

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ŘÍZENÍ DYNAMICKÝCH SYSTÉMŮ V REÁLNÉM ČASE

DIPLOMOVÁ PRÁCE

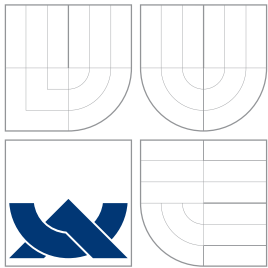
MASTER'S THESIS

AUTOR PRÁCE

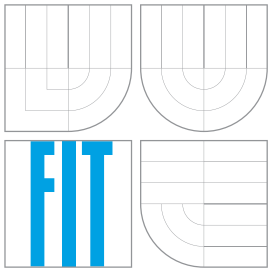
AUTHOR

Bc. PAVEL ADAMÍK

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

ŘÍZENÍ DYNAMICKÝCH SYSTÉMŮ V REÁLNÉM ČASE

REAL TIME DYNAMIC SYSTEM CONTROL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PAVEL ADAMÍK

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. JIŘÍ KUNOVSKÝ, CSc.

BRNO 2009

Abstrakt

Tato práce se zabývá metodikou řízení dynamických systémů v reálném čase. Obsahuje přehled základů teorie řízení a základy stavby regulátorů. Dále následuje přehled matematických základů pro modelování systémů, matematický základ pro simulace systémů s pomocí diferenciálního počtu, metody řešení diferenciálních rovnic. Dále je uveden metodický postup návrhu obecného regulátoru s využitím simulačních metod. Po ověření výsledku v systému Matlab pokračuje problematika modelování zpoždění a kvantování.

Klíčová slova

Taylorova řada, teorie řízení, simulace, model, diferenciální počet, dopravní zpoždění, Padého rozvoj, kvantizace, návrh, regulátor, numerické metody, integrace, derivace, diferenciální rovnice, zpětná vazba, bloková algebra, TKSL, Matlab, Simulink, integrační krok, řád rovnice.

Abstract

This thesis focuses on the methodology of controlling dynamic systems in real time. It contains a review of the control theory basis and the elementary base of regulators construction. Then the list of mathematic formulaes follows as well as the math basis for the system simulations using a differential count and the problem of differential equations solving. Furthermore, there is a systematic approach to the design of general regulator enclosed, using modern simulation techniques. After the results confirmation in the Matlab system, the problematics of transport delay & quantization modelling follow.

Keywords

Taylor power series, control theory, simulation, model, differential count, transport delay, Pade approximation, quantization, design, regulator, numeric methods, integration, derivation, differential equation, closed loop, block algebra, TKSL, Matlab, Simulink, integration step, equation power.

Citace

Pavel Adamík: Řízení dynamických systémů v reálném čase, diplomová práce, Brno, FIT VUT v Brně, 2009

Řízení dynamických systémů v reálném čase

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana doc. Ing. Jiřího Kunovského, CSc. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Adamík
27. ledna 2009

© Pavel Adamík, 2009.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Teorie řízení systémů	4
2.1	Úvod	4
2.2	Problematika návrhu	4
2.3	Rozdělení modelů	5
2.4	Popis jevů v lineárních systémech	5
2.5	Popisy spojitých lineárních systémů	6
2.5.1	Popis vnější	6
2.5.2	Popis vnitřní	6
3	Regulační systémy	10
3.1	Základní pojmy	10
3.2	Regulační obvody	10
3.3	Realizace regulátorů	12
3.3.1	P-regulátor	13
3.3.2	I-regulátor	13
3.3.3	D-regulátor	14
3.4	Složené typy regulátorů	14
3.4.1	PD regulátor	14
3.4.2	PI regulátor	15
3.4.3	PID regulátor	15
3.5	Rozvětvené regulační obvody	15
3.5.1	Obvody s pomocnou regulovanou veličinou	15
3.5.2	Obvod s měřením poruchy	16
3.5.3	Obvod s modelem regulované soustavy	16
4	Matematické nástroje	18
4.1	Diferenciální rovnice	18
4.1.1	Pojem	18
4.1.2	Analytické řešení	18
4.1.3	Numerické řešení	19
4.2	Systematické řešení diferenciálních rovnic	21
4.2.1	Metoda snižování řádu derivace	21
4.2.2	Metoda postupné integrace	22

5	Metodický postup návrhu regulátoru	24
5.1	Myšlenka	24
5.2	Stabilní soustava	24
5.2.1	Vytvoření modelu	24
5.2.2	Vyladění modelu	25
5.2.3	Připojení PI regulátoru	29
5.2.4	Zavedení do reálné soustavy	32
5.2.5	Zavedení korekce	32
5.2.6	Uzavření zpětné vazby přes soustavu	33
5.2.7	Test přičtením poruchy	35
5.3	Nestabilní soustava	35
6	Simulační nástroje	37
6.1	TKSL	37
6.2	Matlab a Simulink	37
6.2.1	Soustava lehce rozladěná s velkou chybou výstupu	38
6.2.2	Soustava silně rozladěná s malým šumem na výstupu	39
6.2.3	Soustava silně rozladěná s velkým šumem na výstupu	40
7	Dopravní zpoždění	41
7.1	Modelování časového zpoždění	41
8	Kvantizace a převodníky	44
8.1	Vzorkování	44
8.2	Aliasing	44
8.3	Kvantování	44
8.4	Kvantizační šum	45
8.5	Modifikace modelu	45
8.5.1	Aplikace kvantizace v modelu	46
8.5.2	Vliv dopravního zpoždění v převodnících	47
9	Závěr	50

Kapitola 1

Úvod

Cílem této práce jsou dvě komplementárně spojené stěžejní věci. Hlavním cílem je předvést metodiku návrhu systému pro řízení teoreticky jakéhokoliv dynamického systému. Podruzným cílem, který plyne z předchozího, je potřeba čtenáře uvést do problematiky teorie řízení stejně jako do základů používaných nástrojů a přístupů. První část textu je věnována potřebné teorii nutné pro snazší vkročení do části druhé, která je věnována metodickému postupu návrhu regulátoru s využitím možností moderních metod simulace dynamických dějů.

Kapitola 2

Teorie řízení systémů

2.1 Úvod

Řízení dynamických systémů, potažmo teorie řízení, dosáhla hlavního rozmachu po roce 1980[2]. Jejím jádrem je v podstatě propojení techniky a matematického přístupu. Každý ze subjektů aplikace přináší nové problémy v druhém subjektu, výsledek v rovině matematické zavádí problém jeho technické realizace v praxi a naopak, reálný systém nás staví před problém, jak tento systém přenést do úrovně matematické teorie. V dnešní době již však obecně nemáme potíž realizovat ani jednu z rovin, a tak je teorie řízení běžným nástrojem používaným v mnoha formách reálných aplikací.

Teorie řízení v zásadě předpokládá, že jisté (sledované) signály - jako chybové vlivy a výstupy akčních členů - nebudou překračovat jisté předem specifikované meze. Matematické modely, které k tomuto účelu využíváme, jsou pak vlastně idealizací reálného systému; i chybové signály jsou ovlivněny citlivostí sensorů, které budou skutečně použity. Vždy tedy bude přítomna jistá úroveň nejistoty, nicméně moderní metody prokázaly, že dokáží věrně a funkčně modelovat systém při udržení sledovaných signálů v požadovaných mezích. [2]. Zanedbávání detailů za jistou zvolenou mez je nezbytný kompromis mezi přesností a složitostí ovládaného systému.

2.2 Problematika návrhu

Proces návrhu řídicího systému obecně zahrnuje mnoho kroků. Typický scénář je následující:

- Studium systému, jež má být řízen, a rozhodnutí, jakých typů sensorů a akčních členů bude použito a jak budou rozmístěny.
- Vytvoření modelu řízeného systému.
- Je-li zapotřebí, zjednodušení modelu pro lepší popisnost.
- Analýza výsledného modelu, určení jeho vlastností.
- Určení skutečných průběhů systému.
- Rozhodnutí o typu použité regulace.
- Návrh regulátoru tak, aby co nejvíce odpovídal specifikacím průběhů; pokud neodpovídá, upravení specifikací nebo použití obecnějšího typu regulátoru.

- Simulace výsledného řízeného systému, a to na počítači nebo na elektronickém obvodu.
- Zpět na krok 1 je-li zapotřebí nový návrh.
- Výběr hardware a software potřebného k implementaci regulátoru.
- Doladění regulátoru za chodu, je-li potřeba.[2]

2.3 Rozdělení modelů

Ještě před tím, než můžeme začít rozebírat postup modelování fyzického systému, je důležité uvést a rozlišit čtyři rozdílné objekty:

- *Reálný fyzický systém*: funkční systém, vyskytující se ve skutečnosti.
- *Ideální fyzický model*: je získán schématickým rozborem reálného fyzického systému do funkčních bloků. Ty jsou tvořeny například rezistory, hmotou, paprsky, výhněmi, izotropickými médii, poli, elektrony atd.
- *Ideální matematický model*: je získán aplikací přírodních zákonů na ideální fyzický model. Skládá se typicky z nelineárních diferenciálních rovnic či z jiných matematických popisů.
- *Redukovaný matematický model*: je získán z ideálního matematického modelu jeho linearizací, zanedbáváním apod. Obvykle má podobu racionálních funkcí.

Ve skutečnosti používáme často stejných slov pro popis součástí reálného i ideálního modelu, například slovo odpor může označovat jak zařízení složené z keramiky a kovu, tak ideální objekt sloužící jako reprezentace Ohmova zákona. V takových případech by se měla správně používat rozlišovací slova *reálný* a *ideální*.

Žádný matematický systém nemůže přesně namodelovat reálný fyzický systém, vždy dosahujeme jisté nejistoty. Nejistota znamená, že nemůžeme přesně předpovědět, jaký bude výstup skutečného systému, i když známe přesně vstupní hodnoty. Nejistota má v podstatě dva zdroje: neznámé či nepředvídatelné vstupy (šum, porucha) a nepředvídatelný dynamický vývoj.

Cílem modelu je poskytnout úrovně předpovědi odezvy výstupu na vstup takovým způsobem, abychom byli schopni navrhnout řídicí systém a byli si jisti, že navržená soustava bude fungovat i ve skutečnosti. Samozřejmě to není možné. Na straně návrhu je vždy zapotřebí jistá míra víry ve výsledek; tento fakt se nedá odstranit, ale lze jej umírnit použitím efektivního modelování, analýzou a použitím správných technik návrhu. [2]

2.4 Popis jevů v lineárních systémech

U lineárních systémů se vyskytují dvě hlavní metody, jak dosáhnout jejich popisu, a to popis *vnitřní* a *vnější*.

Vnější popis je založen na vztahu mezi výstupními a vstupními veličinami. Vnitřní popis je pak založen na naší znalosti funkce systému z hlediska jeho částí; využíváme schopnosti odhadnout vliv vstupních veličin na stavové veličiny a zároveň vliv vstupních a stavových veličin na výstup.

2.5 Popisy spojitých lineárních systémů

2.5.1 Popis vnější

Vztah mezi vstupem a výstupem systému může být vyjádřen různými způsoby. V literatuře jsou tedy použity rozmanité notace; v této práci budu vstup označovat jako $z(t)$ a výstup jako $y(t)$.

Lineární stacionární spojitý systém se vstupem $z(t)$ a výstupem $y(t)$ může být popsán obecnou diferenciální rovnicí ve tvaru:

$$a_n y^n(t) + a_{n-1} y^{(n-1)}(t) + \dots + a_1 y'(t) + a_0 y(t) = b_m z^{(m)}(t) + b_{m-1} z^{(m-1)}(t) + \dots + b_1 z'(t) + b_0 z(t) \quad (2.1)$$

kde $a(i)$, $i = 0, 1, 2, \dots, n$ a $b(j)$, $j = 0, 1, 2, \dots, m$ jsou konstantní koeficienty.

Přenos systému v operátorovém tvaru je za předpokladu nulových počátečních podmínek roven poměru Laplaceova obrazu výstupního signálu k Laplaceově obrazu vstupu. V diferenciálním tvaru je pak přenos systému dán následujícím obecným vztahem:

$$F(p) = \frac{Y(p)}{Z(p)} = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0} \quad (2.2)$$

V teorii systémů platí poučka: aby byl systém realizovatelný, musí být stupeň polynomu v čitateli menší nebo roven stupni polynomu ve jmenovateli přenosu. Pro analýzu systémů je důležitý právě polynom ve jmenovateli přenosu, který lze též vyjádřit jako součin kořenových činitelů:

$$a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0 = a_n (p - p_1)(p - p_2) \dots (p - p_n) \quad (2.3)$$

kde $p_1 \dots p_n$ jsou póly přenosu systému.

Kořeny polynomu v čitateli:

$$B(p) = b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0 = 0 \quad (2.4)$$

označujeme jako nuly přenosu systému a lze psát:

$$b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0 = b_m (p - n_1)(p - n_2) \dots (p - n_m) \quad (2.5)$$

Pokud využijeme takto určených pólů a nul, můžeme přenos vyjádřit v následujícím vztahu:

$$F(p) = \frac{b_m ((p - n_1)(p - n_2) \dots (p - n_m))}{a_n (p - p_1)(p - p_2) \dots (p - p_n)} \quad (2.6)$$

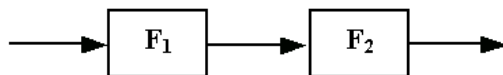
2.5.2 Popis vnitřní

Jde o nejobecnější možný popis systému. Vychází z vnitřních dějů, a proto umožňuje odhalit - a popsat - některé z vlastností systému, které nelze vyčíst při použití popisu vnějšího.

Teorie řízení zavádí jako základ vnitřního popisu systému blokovou algebru. Každý systém se tak skládá z vzájemně propojených podsystémů *bloků*. Prioritou bývá určit, jaké vlastnosti bude mít složený systém, pokud známe chování a projevy všech jeho podsystémů a způsob jejich vzájemného propojení. V zásadě existují tři různé způsoby propojení různých bloků:

- Sériové

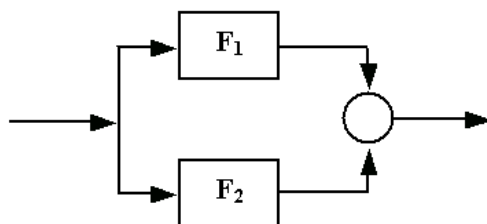
$$F(p) = F_1(p) \cdot F_2(p) \quad (2.7)$$



Obrázek 2.1: Sériové spojení bloků

- Paralelní

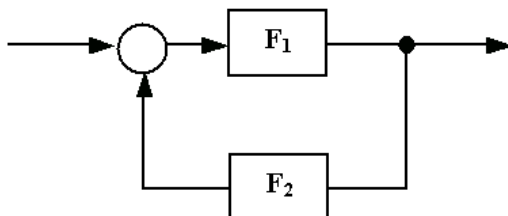
$$F(p) = F_1(p) + F_2(p) \quad (2.8)$$



Obrázek 2.2: Paralelní spojení bloků

- Zpětnovazební

$$F(p) = \frac{F_1(p)}{1 - F_1(p) \cdot F_2(p)} \quad (2.9)$$



Obrázek 2.3: Bloky zapojené do zpětné vazby

Při práci s blokovou algebrou se využívá nástroje zvaného graf signálových toků. Tyto grafy mají v literatuře zavedeny různé notace; zde jsou základní grafové operace:

- Násobení signálu konstantou

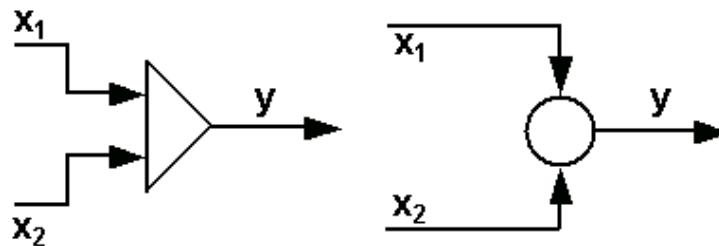
$$y = a \cdot x \quad (2.10)$$



Obrázek 2.4: Násobení signálu konstantou

- Součet dvou (či více) signálů

$$y = x_1 + x_2 \quad (2.11)$$

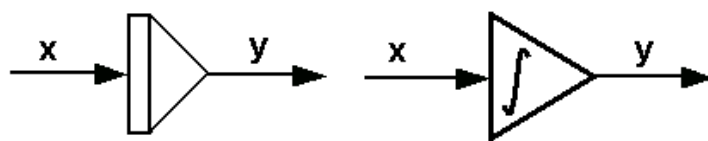


Obrázek 2.5: Sčítání signálů

V této publikaci používám levou notaci, pravá je rozšířenější ve světové literatuře.

- Integrace signálu - integrátor

$$y = \int x \quad (2.12)$$

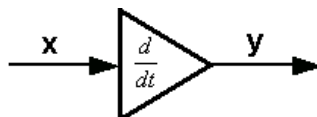


Obrázek 2.6: Integrace signálu

V této publikaci opět používám levou notaci, pravá je rozšířenější ve světové literatuře.

- Derivace signálu - derivátor

$$y = \frac{dx}{dt} \quad (2.13)$$

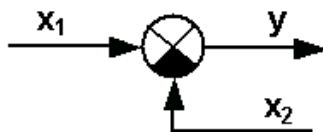


Obrázek 2.7: Integrace signálu

V praxi se derivátorů příliš nepoužívá, protože dosáhnout spolehlivé derivace je poměrně náročné. Potřebujeme-li derivaci signálu, můžeme si vypomoci jinými způsoby (viz níže).

- Rozdílový člen

$$y = x_1 - x_2 \quad (2.14)$$



Obrázek 2.8: Rozdílový člen

Technicky vzato se rozdílový člen dá realizovat pomocí násobení signálu a součtového členu, nicméně pro větší přehlednost se v grafové algebře často používá tento symbol. Zejména v teorii regulačních členů má své specifické místo.

Kapitola 3

Regulační systémy

Ve skutečném provozu se dnes téměř všude využívá přístupů automatizace a tedy je zapotřebí skloubit skutečnost a teorii řízení. Zařízení, které aplikuje principy teorie řízení na reálný fyzický systém se běžně nazývá *regulátor* či *kontroler*. Tato zařízení jsou běžně postavena tak, aby zcela nahradila potřebu lidské činnosti.

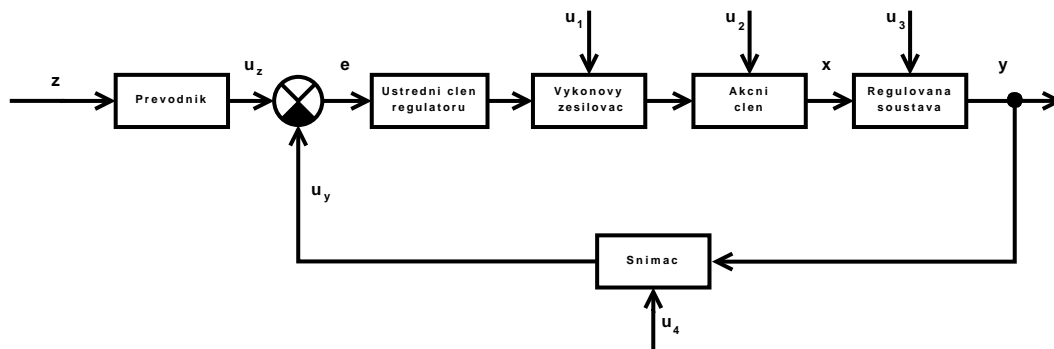
3.1 Základní pojmy

Dříve, než je možné začít rozebírat problematiku, je třeba vytvořit přehled základních pojmů tématu. Mezi tyto pak patří:

- *Mechanizace* - proces, který zaváděním strojních mechanismů osvobozuje člověka od namáhavé, fyzicky náročné práce.
- *Automatizace* - vývojové stádium rozvoje techniky, ve kterém automatizační zařízení vykonává nejen fyzickou, ale i řídicí práci.
- *Řízení* - každé cílevědomé působení na řízený objekt tak, aby bylo dosaženo určitého předepsaného cíle. Podle toho, jak je řízení prováděno, rozlišujeme řízení ruční a automatické.
- *Ovládání* - neboli *řízení bez zpětné vazby* - řízení, které probíhá na základě zadaných pravidel (algoritmů) bez zpětné kontroly výsledků řídicích zásahů měřením.
- *Regulace* - neboli *řízení se zpětnou vazbou* - probíhá tak, že velikost zvolené veličiny je měřena, porovnávána s žádanou hodnotou a řízení probíhá v závislosti na vypočtené odchylce. Tato metoda řízení je daleko dokonalejší než metoda předchozí, a tedy jsou dosahované výsledky daleko kvalitnější.

3.2 Regulační obvody

Automatické řízení se dnes takřka vždy provádí aplikací regulátoru, tedy řízení se zpětnou vazbou. Jde tedy o takové systémy, jejichž výstupní veličiny jsou měřeny a porovnávány s požadovanou hodnotou. Ve výsledku nám vznikne systém, který je blokově znázorněn pod tímto odstavcem. Jsou na něm znázorněny některé typické základní veličiny všech regulačních obvodů.



Obrázek 3.1: Schéma regulátoru se zpětnou vazbou

- *Regulovaná veličina (y)* - výstup regulované soustavy. Udržení úrovně této veličiny na požadované hodnotě je zásadním úkolem regulačního obvodu.
- *Akční veličina (x)* - vstup regulované soustavy. Vytváří ji akční člen, který je napájen výkonovým zesilovačem. *Ústřední člen* regulátoru určuje algoritmus řízení, což znamená, že v něm probíhají požadované matematické operace.
- *Řídicí veličina (z)* - neboli požadovaná hodnota. Je nositelem informace o tom, jaká hodnota regulované veličiny má být nastavena.
- *Regulační odchylka (e)* - je definována jako rozdíl požadované hodnoty a regulované veličiny, $e = z - y$. Regulační odchylku zpracovává ústřední člen regulátoru, pro který je odchylka vstupem. Rozdíl $z - y$ se realizuje v diferenciálním (rozdílovém) členu. K tomu je ovšem zapotřebí, aby obě veličiny měly stejný fyzikální rozměr. Proto je v obvodu umístěn převodník, který převede řídicí veličinu na stejný tvar jako má výstup z měřícího členu. Je možný i opačný přístup, kdy se převodník umístí do větve zpětné vazby; záleží na konkrétní implementaci systému a požadavcích, jež jsou kladeny na jeho chod a regulaci.
- *Poruchové veličiny (u_1, u_2, u_3, u_4)* - obecně mohou působit kdekoli v systému. Nejčastěji se ale uplatňují vlivy poruch přímo v regulované soustavě.

Regulační obvod se zpětnou vazbou má dva hlavní úkoly. Prvním je zabezpečit, aby regulovaná veličina co nejlépe sledovala časový průběh řídicí veličiny. Splnění tohoto požadavku charakterizují vlastnosti obvodu z hlediska řízení. Druhým úkolem je kompenzace působení poruchových signálů tak, aby se jejich vliv projevil na regulované soustavě co nejméně. Oba tyto požadavky lze shrnout do jediného - s využitím pojmu *regulační odchylka*.

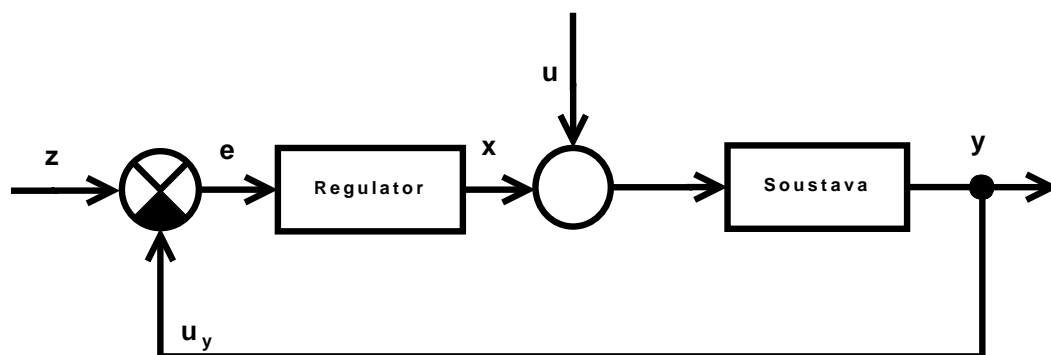
Snahou regulátoru je udržet tuto odchylku za plného provozu co nejmenší, v ideálním případě rovnu nule. Důležitým požadavkem je pak nejen splnění této podmínky v ustáleném stavu, ale i v průběhu přechodného děje.

Podle časového průběhu řídicí veličiny rozpoznáváme několik druhů regulace.

- *Regulace na konstantní hodnotu* - v praxi asi nejběžnější případ, kdy je žádaná hodnota konstantní. U těchto systémů je kladen důraz na funkci kompenzace poruch.

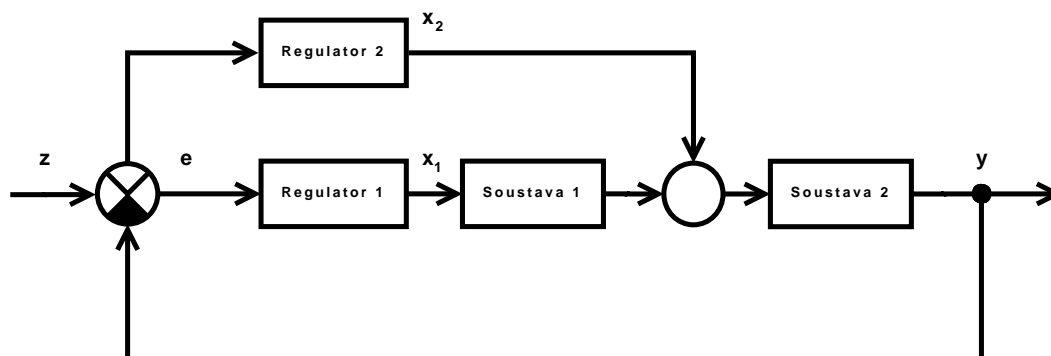
- *Programová regulace* - je charakteristická tím, že řídicí veličina se mění podle předem daného známého průběhu, tedy programu. V průběhu řídicí veličiny mohou být jak konstantní úseky, tak úseky lineárně nebo kvadraticky proměnné s časem, popřípadě skokové změny. V takovém případě jsou požadavky kladené na regulační obvod z hlediska řídicí veličiny daleko přísnější.
- *Vlečná regulace* - v tomto případě je řídicí veličina proměnná a její průběh není dopředu znám. Zvláštním případem je regulace poměrová, kdy regulovaná veličina má být stále v určitém poměru k další zvolené nezávislé proměnné.

Nejjednodušší schéma regulované soustavy:



Obrázek 3.2: Jednoduchá regulovaná soustava

Následuje příklad složitějšího, rozvětveného obvodu, který je charakteristický tím, že obsahuje dopředné i zpětné vazby. Tato soustava se skládá ze dvou podčástí, nazvané Soustava 1 a Soustava 2. Má dvě akční veličiny, x_1 a x_2 , řízení provádějí dva regulátory 1 a 2. Ty zpracovávají stejnou odchylku e .



Obrázek 3.3: Složitá regulovaná soustava

3.3 Realizace regulátorů

Jak bylo již řečeno, regulátor je zařízení, které pomocí akční veličiny působí na soustavu tak, aby odchylka regulované veličiny od požadované hodnoty byla co nejmenší. Všechny

členy zpětnovazebního regulačního obvodu, s výjimkou regulované soustavy, shrnujeme pod název regulátor. Jeho úlohou je měřit velikost regulované veličiny, porovnat ji s velikostí požadované hodnoty, tuto regulační odchylku vhodně zesílit (popřípadě upravit) a po výkonovém zesílení pomocí akčního členu působit na soustavu. Neuvažujeme-li poruchové signály, má regulátor dva vstupy - $y(t)$, $z(t)$ - a jeden výstup $x(t)$.

Blok, dříve označený jako ústřední člen regulátoru, určuje, jakými časovými, frekvenčními či logickými úpravami je regulační odchylka $e(t)$ zpracována, než se změní ve výstupní signál regulátoru (akční veličinu). Tento člen má z hlediska regulace zásadní význam, neboť určuje řídicí algoritmus, jež je spolu s vlastnostmi regulované soustavy nejdůležitější pro výslednou kvalitu regulace. Podle druhu energie, která napájí samotný regulátor, dělíme regulátory na:

- *Přímočinné regulátory* - nemají svůj vlastní zdroj energie a ke své činnosti využívají pouze energii odebíranou z regulované soustavy. Většina jednoduchých průmyslových regulátorů spadá do této kategorie. Tyto regulátory bývají většinou nelineární, akční veličina v nich může nabývat pouze omezený počet hodnot (často jen dvě, zapnuto-vypnuto). Jsou to známé reléové regulátory používané například v lednicích či v regulátorech topení.
- *Regulátory s pomocným zdrojem energie* - oproti předchozím jde o složitější zařízení, jehož jádrem je vždy nějaký druh zesilovače. Dosahovaná kvalita regulace je podstatně vyšší (úměrná nákladům a složitosti). Statické vlastnosti těchto regulátorů považujeme v určitém pracovním rozsahu za lineární.

Rozlišujeme tři základní typy regulátorů:

3.3.1 P-regulátor

P-regulátor, neboli proporcionální, je nejjednodušším typem. Jde o prostý zesilovač, v úplně nejjednodušším případě o přímé spojení výstupu se vstupem. Název plyne od faktu, že akční odchylka je v tomto případě proporcionálně úměrná odchylce regulační.

Rovnice výstupu a přenosu:

$$x(t) = r_0 \cdot e(t) \quad (3.1)$$

$$F_R(p) = \frac{X(p)}{E(p)} \quad (3.2)$$

Nejjednodušším příkladem proporcionálního regulátoru může být prostý odporový dělič napětí.

3.3.2 I-regulátor

Neboli integrační regulátor. Častým požadavkem na regulaci je, aby regulovaná veličina přesně odpovídala veličině řídicí, tj. aby regulační odchylka byla v ustáleném stavu nulová. Akční veličina však nulová není, protože když je obvod statickou soustavou, musí nenulové regulované veličině odpovídat též nenulová akční veličina. Regulátor je pak článek, který i při nulovém vstupním signálu (odchylce) má nenulový výstup.

Rovnice výstupu a přenosu:

$$x(t) = r_i \cdot \int_0^t e(t) dt + x(0) \quad (3.3)$$

$$F_R(p) = \frac{X(p)}{E(p)} = \frac{r_i}{p} = \frac{1}{T_i p} \quad (3.4)$$

Elementárním příkladem I-regulátoru je RC integrátor (rezistor a kondenzátor).

3.3.3 D-regulátor

Neboli derivační regulátor. Jeho výstupní signál je derivací signálu vstupního, tedy je změna akční veličiny úměrná rychlosti změny odchylky. Pro konstantní odchylku je nulová a regulátor se tedy chová jako rozpojený spínač. Proto jej nelze samostatně zařazovat do přímé větve regulačního obvodu. Může být pouze součástí regulátorů PD a PID. Samostatně jej lze použít do tzv. malé zpětnovazební smyčky sloužící ke stabilizaci regulačního obvodu.

K derivačním regulátorům je třeba dodat, že v praxi se jim snažíme při návrhu vyhnout. Fyzikální realizace derivací bývá problematická a mnohdy je nevyhovující. Větší význam mají v případě číslicového provedení, ovšem v analogové rovině je D-regulátor zřídkačným jevem.

Rovnice výstupu a přenosu:

$$x(t) = T_d' \cdot e(t) \quad (3.5)$$

$$F_R(p) = \frac{X(p)}{E(p)} = T_d p \quad (3.6)$$

Typickým elementárním derivátorem, tedy D-regulátorem, je základní CR článek.

3.4 Složené typy regulátorů

Z výše uvedených základních typů se používají tři základní složené typy regulátorů.

3.4.1 PD regulátor

Neboli proporcionálně derivační regulátor. Jeho přenosy jsou dány vztahy:

$$x(t) = r_0 \cdot e(t) = r_d \frac{d \cdot e(t)}{d(t)} \quad (3.7)$$

$$F_R(p) = \frac{X(p)}{E(p)} = r_0 + r_d = K_R(T_d p + 1) \quad (3.8)$$

Platí:

$$K_R = r_0, T_d = \frac{r_d}{r_0} \quad (3.9)$$

3.4.2 PI regulátor

Proporcionálně integrační regulátor. Jeho přenosy jsou dány vztahy:

$$x(t) = r_0 \cdot e(t) + r_i \cdot \int_0^t e(t)dt + x(0) \quad (3.10)$$

$$F_R(p) = \frac{X(p)}{E(p)} = r_0 + \frac{r_i}{p} = K_R \frac{T_R p + 1}{p} = \frac{T_R p + 1}{T_i} \quad (3.11)$$

$$K_R = r_i = \frac{1}{T_i}, T_R = \frac{r_0}{r_i}, r_0 = \frac{T_R}{T_i} \quad (3.12)$$

3.4.3 PID regulátor

Potencionálně nejvyšší a zároveň nejsložitější regulační obvod. Jak název napovídá, je složen ze všech základních regulátorů. Platí pak pro něj rovnice:

$$x(t) = r_0 \cdot e(t) + r_d \frac{d.e(t)}{dt} + r_i \cdot \int_0^t e(t)dt + x(0) \quad (3.13)$$

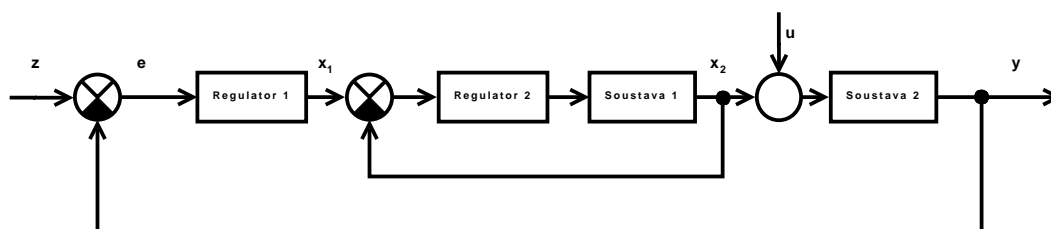
$$F_R(p) = \frac{X(p)}{E(p)} = r_0 + \frac{r_i}{p} + r_d p = K_R \left(1 + D \cdot p + \frac{1}{I \cdot p}\right) = k_r \left(i + T_d p + \frac{1}{T_i p}\right) = \frac{(T_1 p + 1) \cdot (T_2 p + 1)}{T_c p} \quad (3.14)$$

$$r_i = \frac{K_R}{I} = \frac{k_r \cdot i}{T_i}, r_d = K_R D = \frac{T_1 T_2}{T_c} = k_r T_d, r_0 = K_R = \frac{T_1 + T_2}{T_c} = k_i \cdot i \quad (3.15)$$

3.5 Rozvětvené regulační obvody

Nejvyšší možnou kombinací základního rozdělení regulátorů jsou regulátory větvené. Jde o:

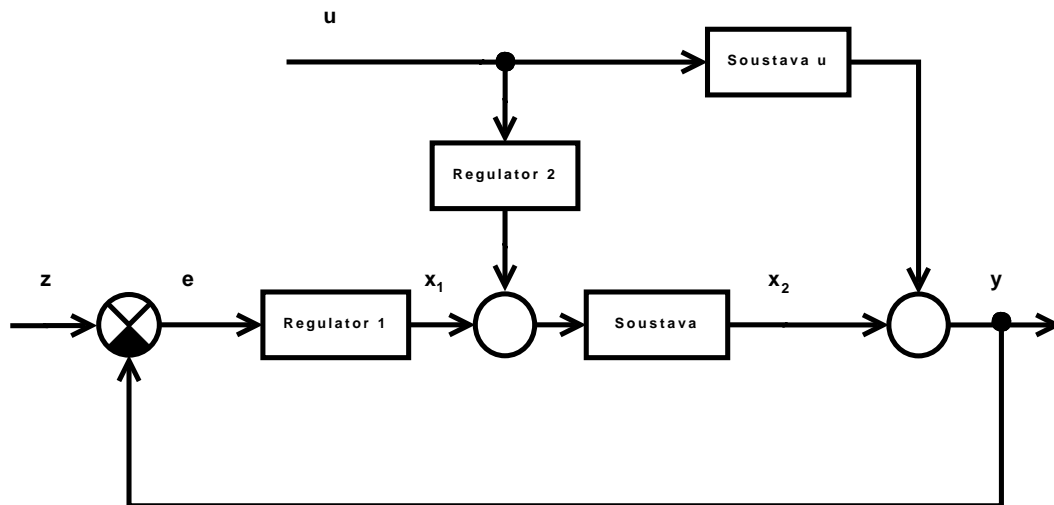
3.5.1 Obvody s pomocnou regulovanou veličinou



Obrázek 3.4: Obvod s pomocnou regulovanou veličinou

Ve výše uvedeném příkladě vidíme, že sériovým spojením soustav 1 a 2 vzniká měřená veličina x_2 , která je regulátorem 2 řízena podle požadované hodnoty x_1 . Jako hlavní regulátor pracuje blok regulátoru 1.

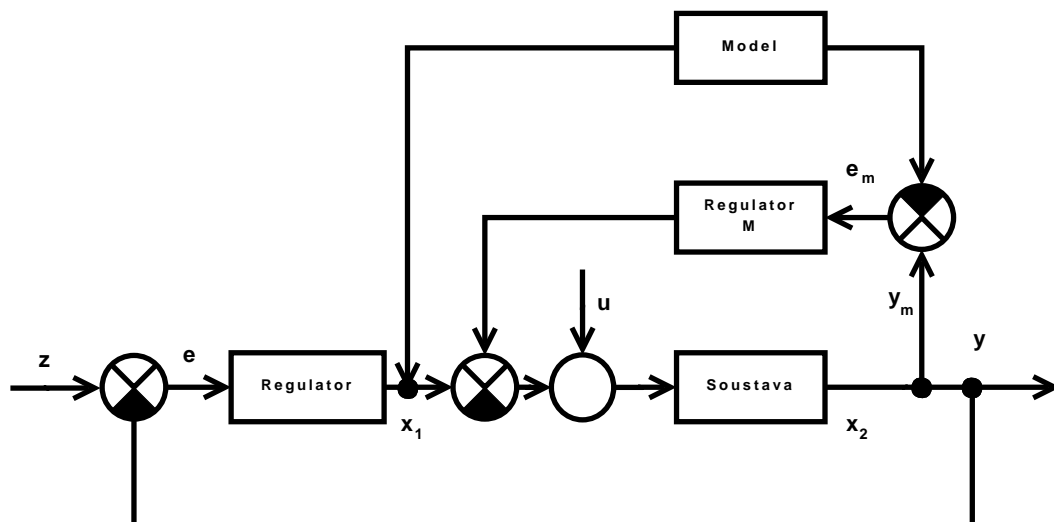
3.5.2 Obvod s měřením poruchy



Obrázek 3.5: Obvod s měřením poruchy

V regulačním obvodu z obrázku 3.5 porucha $u(t)$ prochází článkem s přenosem $S_u(p)$ a přičítá se k výstupu regulované soustavy. Současně tuto poruchu měříme a přes regulátor 2 přičítáme k akční veličině $x_1(t)$. Přenos řízení zůstává touto přídavnou vazbou nezměněn. Přenos poruchy je možno regulátorem 2 výrazně měnit. Teoreticky lze dosáhnout úplné kompenzace poruchového signálu.

3.5.3 Obvod s modelem regulované soustavy



Obrázek 3.6: Obvod s modelem soustavy

Systémy s modelem regulované soustavy se používají hlavně v adaptivních obvodech, mohou však zlepšit i kvalitu regulace v jednoduchém regulačním obvodu. Blokové schéma

takového systému je na obrázku 3.6. Akční veličina $x_1(p)$ působí jak na regulovanou soustavu, tak na model. Rozdíl výstupů $y(t) - y_m(t)$ tvoří pomocnou odchylku e_m , kterou zpracovává regulátor modelu M.

Kapitola 4

Matematické nástroje

Základem pro výpočet, modelování a návrh regulačních obvodů je využití diferenciálního počtu a metod jeho řešení.

4.1 Diferenciální rovnice

4.1.1 Pojem

Diferenciální rovnice je název pro rovnici, v níž se jako neznámá vyskytuje funkce a její derivace.[3]. Při výpočtech, v nichž se vyskytují derivace funkcí, zjišťujeme, že mezi funkcemi a jejich derivacemi platí řada vztahů. Např. pro funkci $y(t) = A \cdot \sin t$ platí $y'(t) = A \cdot \cos t$, $y''(t) = -A \cdot \sin t$. Potom

$$y''(t) + y(t) = 0 \quad (4.1)$$

Vezmeme-li jinou funkci $y(t)$, např. $y(t) = B \cdot e^t$, je $y'(t) = B \cdot e^t$, pak můžeme psát

$$y'(t) - y(t) = 0 \quad (4.2)$$

Je vidět, že řešení dané diferenciální rovnice není jediné, ale může jich být i nekonečně mnoho. Řešením diferenciální rovnice se rozumí nalezení takové funkce, jejímž dosazením do diferenciální rovnice dostaneme identitu. Uvažujme kupříkladu rovnoměrný přímočarý pohyb tělesa. Z fyziky víme, že velikost rychlosti je derivací dráhy podle času, tj. $s'(t) = v(t)$. Při rovnoměrném přímočarém pohybu je $v(t) = v = konst.$ Dostaneme pak

$$s'(t) = v \quad (4.3)$$

, což je vlastně diferenciální rovnice pro dráhu $s(t)$. Integrací dostaneme

$$s(t) = vt + C \quad (4.4)$$

, kde C je libovolná integrační konstanta. Tu určíme z počátečních podmínek; necht' v čase $t = t_0$ urazilo těleso již dráhu s_0 , tj. $s(t_0) = s_0$. Dosadíme-li do rovnice 4.4 za $t = t_0$, dostaneme $C = s_0 - vt_0$. Pak můžeme psát $s(t) = s_0 + v(t - t_0)$.

4.1.2 Analytické řešení

Výše naznačený princip se nazývá analytickým principem řešení. Jeho výsledkem je tedy funkce času. Konkrétní hodnotu v určitém čase získáme dosazením tohoto času do výsledné

funkce. Lze určit hodnotu v libovolném bodě, v němž je funkce definována. Analytické metody bývají obvykle složité a časově náročné, na druhou stranu jsou ale velmi přesné. Jedním z hlavních znaků tohoto přístupu je ale fakt, že lidská mysl s intuitivním přístupem k problému touto metodou vládne daleko lépe než napodobení tohoto přístupu ve strojích, a proto není téměř možné v rámci informačních technologií v systémech reálného času využívat tento přístup. Teorie obyčejných diferenciálních rovnic tedy vybírá určité modely jistých skupin diferenciálních rovnic, pro které je nalezeno obecné schéma řešení.

4.1.3 Numerické řešení

Numerickým řešením diferenciální rovnice je posloupnost hodnot v určitých předem zvolených časových bodech. Hodnoty funkce mezi zvolenými body lze určit buď interpolací z okolních vypočtených bodů, nebo opětovnou aplikací metody s menším rozestupem (krokem) časových bodů. Numerické metody bývají obvykle jednodušší a rychlejší pro strojový výpočet než analytické. Při špatné volbě kroku však může výpočet dosahovat příliš velké chyby.

Eulerova metoda

Jde o nejjednodušší metodu, nejrychlejší a zároveň nejméně přesnou. Vychází přímo z definice fyzikálních veličin. Je to jednokroková metoda prvního řádu, využívá první stupeň Taylorova rozvoje - extrapolace přímkou. Rovnice jsou při přepsání do geometrického tvaru rovnicí přímky jdoucí bodem na normále funkce; neboli, vždy se pohybujeme po tečně k přesnému řešení.

$$y_{n+1} = y_n + h y'_n \quad (4.5)$$

$$y(x_{n+1}) = y(x_n) + h \cdot y'(x_n) = y(x_n) + h \cdot f(x_n, y(x_n)) \quad (4.6)$$

$$y_{n+1} = y_n + h \cdot f(x_n, y_n) \quad (4.7)$$

Při použití této metody jsou stabilní systémy spojité, ovšem obecně nestabilní jsou systémy diskrétní. Děje se tomu kupříkladu z důvodu příliš velké volby kroku výpočtu. Je možná modifikace metody s automatickou volbou kroku.

Runge-Kuttova metoda

Jedním z problémů plynoucích z používání Eulerovy metody je fakt, že je nesymetrická. K výsledku se dobírá na intervalu h , ovšem k derivaci používá pouze informaci z počátku intervalu. To znamená (a lze to ověřit rozepsáním na mocninnou řadu), že chyba kroku je jen o jednu mocninu h menší než jeho oprava.

Mezi další důvody, proč se Eulerova metoda v praxi příliš nedoporučuje, je její výpočetní přesnost. Nejpoužívanější náhradou je metoda Runge-Kutta. Její základní filozofie spočívá v tom, že se pokouší najít bod uprostřed momentálního intervalu a ten použít pro výpočet hodnoty. Existují varianty různého řádu, kdy se pro výpočet používá v praxi až čtyř různých mezibodů v intervalu (Runge-Kuttova metoda čtvrtého řádu, nejběžnější varianta).

$$\begin{aligned}
y_{n+1} &= y_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \\
k_1 &= f(x_n, y_n) \\
k_2 &= f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) \\
k_3 &= f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2\right) \\
k_4 &= f(x_n + h, y_n + hk_3)
\end{aligned} \tag{4.8}$$

Taylorova metoda

Tato metoda vychází z tzv. Taylorovy řady:

$$p(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^{(n)}}{n!}(x-a)^n \tag{4.9}$$

Sice se jeví, že možnosti využití Taylorovy řady pro řešení diferenciálních rovnic jsou již zcela vyčerpány (vycházejí z ní mnohé numerické metody), přesto ale nachází v současnosti opět své uplatnění (rychlý rozvoj počítačové techniky). Její využití předpokládá využití většího počtu členů rozvoje, a to v řádech desítek. Vzhledem k délce integračního kroku h poskytuje daleko přesnější výsledky než metody jako je Eulerova nebo Runge-Kuttova.

Základ metody leží v matematických operacích nad řadou. Pokud položíme jako počáteční podmínku $z = 0$ a položíme $h = x_1 - z_1$, pak rovnice přejde do tvaru:

$$f(x) = f(0) + h \cdot f'(0) + \frac{h^2}{2!}f''(0) + \frac{h^3}{3!}f'''(0) + \dots \tag{4.10}$$

Nyní položme $z_2 = x_1$ za předpokladu $h = x_2 - z_2 = x_1 - z_1$.

$$f(x_2) = f(x_1) + h \cdot f'(x_1) + \frac{h^2}{2!}f''(x_1) + \frac{h^3}{3!}f'''(x_1) + \dots \tag{4.11}$$

Hodnoty funkce $f(x)$ v bodech x_1, x_2 lze vypočítat postupně za využití Taylorovy řady. Výsledek jednoho kroku je nutný pro výpočet kroků dalších, následujících. Parametr h označuje integrační krok, který nemusí být konstantní a lze jej pro jednotlivé kroky výpočtu měnit. Závisí na něm rychlost i přesnost výpočtu, čím menší krok, tím vyšší přesnost a nižší rychlost. Proto musíme před začátkem výpočtu vždy určit, jakou přesnost vyžadujeme. Během samotného procesu pak sčítáme dílčí výsledky a pokud je rozdíl dvou po sobě jdoucích výsledných čísel menší, než je požadovaná přesnost, výpočet končí.

K dílčím výpočtům potřebujeme znát vyšší derivace funkce. Výpočet vyšších derivací je časově náročný a prakticky zbytečný, vyšší derivace lze totiž odvodit z předchozích výpočtů. Toto lze ukázat na následující soustavě diferenciálních rovnic.

$$\begin{aligned}
y' &= A \cdot y + B \cdot z \\
z' &= C \cdot y + d \cdot z
\end{aligned} \tag{4.12}$$

při počátečních podmínkách $y(0) = y_0$, $z(0) = z_0$. Řešení klasickým způsobem:

$$y_1 = y_0 + h \cdot y'(0) + \frac{h^2}{2!} \cdot y''(0) + \frac{h^3}{3!} \cdot y'''(0) + \dots$$

$$z_1 = z_0 + h \cdot z'(0) + \frac{h^2}{2!} \cdot z''(0) + \frac{h^3}{3!} \cdot z'''(0) + \dots \quad (4.13)$$

Mezi hlavní klady této metody patří její rychlost a s tím spojené malé nároky na výpočetní techniku. Rovněž je výraznou výhodou možnost paralelního zpracování dílčích výpočtů, které se uplatňuje při numerických výpočtech soustav diferenciálních rovnic. Mezi úskalí ovšem patří potřeba správného odhadu integračního kroku, při jeho špatném odhadu hrozí zavedení nestability. Původcem je fakt, že při chybném určení jednoho kroku se tato chyba přenáší i do následných výpočtů a celková chyba se tak může kumulovat.

4.2 Systematické řešení diferenciálních rovnic

V reálném světě se vyskytuje celá řada jevů či soustav, které umíme do nějaké míry popsat matematickými rovnicemi aplikovanými na oblasti fyziky, chemie či jiné oblasti. Každé takové vyjádření lze převést na diferenciální tvar a následnými úpravami pak dále převést do tvaru, kdy získáme jedinou, obecně rozsáhlou diferenciální rovnici vysokého řádu. Obecný příklad takové rovnice byl uveden například již v rovnici 2.1.

Pokud takové rovnice dosáhneme a tato rovnice splňuje poučku řešitelnosti, existují postupy, jak takový problém poměrně snadno vyřešit za použití numerických výpočtů. Mezi základní metody patří *metoda snižování řádu derivace* a *metoda postupné integrace*. Tyto metody jsou navzájem ekvivalentní; cílem obou je z diferenciální rovnice libovolného řádu vytvořit programové schéma výpočtu a to pak numericky vyřešit. V této práci využívám metody snižování řádu derivace, protože je pro potřeby simulací vhodnější.

4.2.1 Metoda snižování řádu derivace

Mějme rovnici třetího řádu.

$$p'''y + a_2y'' + a_1y' + ay = b_3z''' + b_2z'' + b_1z' + b_0z \quad (4.14)$$

při nulových počátečních podmínkách. Naším hlavním problémem je zbavit se derivací vnučovací (vstupní) funkce na pravé straně. Prvním krokem je převést tuto rovnici na operátorový tvar a vyjádřit z ní hledanou proměnnou.

$$p^3y + a_2p^2y + a_1py + a_0y = b_3p^3z + b_2p^2z + b_1pz + b_0z$$

$$y(p^3 + a_2p^2 + a_1p + a_0) = z(b_3p^3 + b_2p^2 + b_1p + b_0)$$

$$y = \frac{b_3p^3 + b_2p^2 + b_1p + b_0}{p^3 + a_2p^2 + a_1p + a_0} \quad (4.15)$$

Těmito kroky jsme vlastně vyjádřili přenos systému a tedy je vidět korespondence filozofie výpočtu a reálného systému. Nyní si vypomůžeme zavedením pomocné proměnné.

$$w = \frac{1}{p^3 + a_2p^2 + a_1p + a_0} \cdot z$$

$$y = b_3p^3w + b_2p^2w + b_1pw + b_0w \quad (4.16)$$

Tento výsledek ovšem dosadíme zpět do naší *pomocné rovnice* s neznámou w .

$$p^3w + a_2p^2w + a_1pw + a_0w = z \quad (4.17)$$

Tato rovnice již neobsahuje derivace vynuocovací funkce (na pravé straně není derivace). Můžeme na ni již praktikovat vlastní metodu snižování řádu derivace triviálním způsobem - vyjádříme z rovnice nejvyšší derivaci a derivace nižší pak přepíšeme jako integrace těch vyšších.

$$p^3w = z - a_2p^2w - a_1pw - a_0w$$

$$p^2w = \frac{1}{p}p^3w$$

$$pw = \frac{1}{p}p^2w$$

$$w = \frac{1}{p}pw$$

Takovéto vyjádření již není problém poskládat z elementárních bloků blokové algebry. Mé simulace jsou založeny na rozkladu diferenciální rovnice touto metodou.

4.2.2 Metoda postupné integrace

Filozofie této metody je v zásadě podobná, pouze přístup k substitucím je jiný. Touto metodou sice dosáhneme výsledku stejně jako u předchozí, nicméně přístup k mezivýsledkům v programu je náročnější vzhledem k povaze substitucí.

Rovnici 4.14 si pro zpracování touto metodou vyjádřením stupňů derivace převedeme do následujícího tvaru:

$$p^3y = b_3p^3z + p^2(b_2z - a_2y) + p(b_1z - a_1y) + (b_0z - a_0y) \quad \setminus \cdot \frac{1}{p} \quad (4.18)$$

$$p^2y = b_3p^2z + p(b_2z - a_2y) + (b_1z - a_1y) + \underbrace{\frac{1}{p}(b_0z - a_0y)}_{P1} \quad \setminus \cdot \frac{1}{p} \quad (4.19)$$

$$py = b_3pz + (b_2z - a_2y) + \underbrace{\frac{1}{p}(b_1z - a_1y + P1)}_{P2} \quad \setminus \cdot \frac{1}{p} \quad (4.20)$$

$$y = b_3 + \underbrace{(b_2z - a_2y + P2)}_{P3} \quad \setminus \cdot \frac{1}{p} \quad (4.21)$$

Blokovou algebrou pak modelujeme děje:

$$y = b_3z + P_3$$

$$P_3 = \frac{1}{p}(b_2z - a_2y + P_2)$$

$$P_2 = \frac{1}{p}(b_1z - a_1y + P_1)$$

$$P_1 = \frac{1}{p}(b_0z - a_0y) \tag{4.22}$$

Kapitola 5

Metodický postup návrhu regulátoru

5.1 Myšlenka

V této kapitole je na konkrétním příkladu ukázán nejen ideový, ale i metodický postup návrhu elektronického regulátoru. Cílem je vytvořit regulátor s modelem soustavy (viz. 3.5.3), návrh bude probíhat pomocí simulátoru.

Při návrhu je uvažován matematický model s diferenciální rovnicí 4. řádu. Jak ale bude názorně ukázáno, postup není řádem derivace nijak omezován a je volně přenesitelný. Je založen na znalosti požadovaného chování a na předpokladu, že dokážeme reálné veličiny převádět pomocí převodníků z analogových na elektronické a nazpět; veškerá regulace v tomto návrhu probíhá na číslicové, tedy výpočetní, úrovni.

5.2 Stabilní soustava

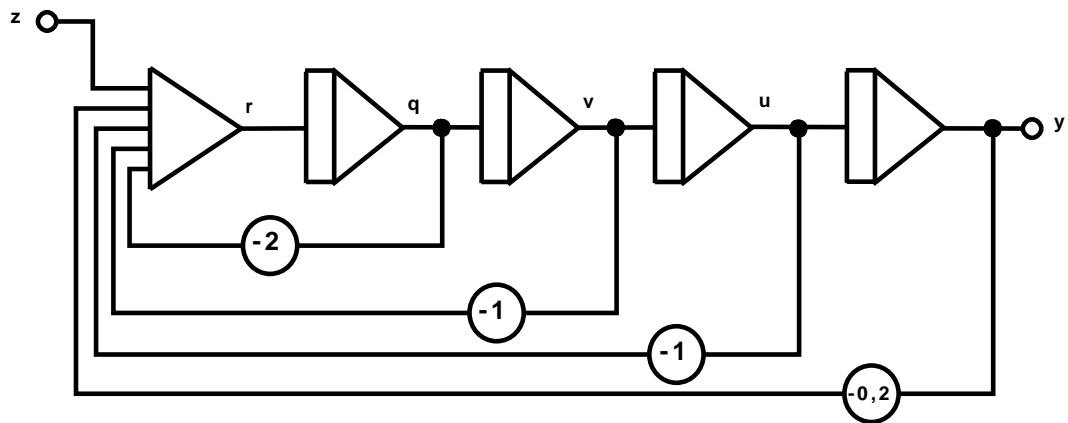
Jak bylo naznačeno v předchozí kapitole, jakýkoli systém popsatelný diferenciálními rovnicemi lze převést na programový výpočetní model s využitím blokové algebry. Pro tento proces tedy není podstatné, co vlastně daný model znázorňuje.

5.2.1 Vytvoření modelu

Prvním krokem nabízené metody je tedy vytvoření modelu. Jako příklad stabilní soustavy použijeme tuto:

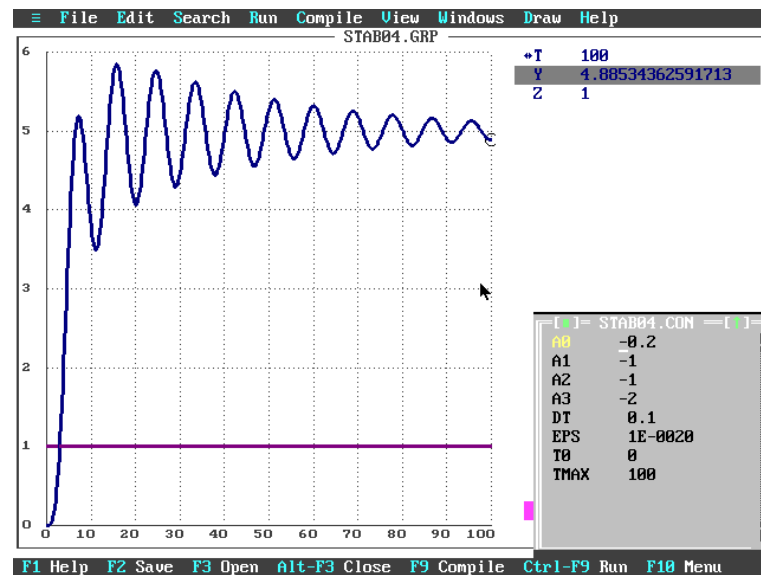
$$r = z + a_3y''' + a_2y'' + a_1y' + a_0y \quad (5.1)$$

$$r = z - 2y''' - y'' - y' - 0.2y \quad (5.2)$$



Obrázek 5.1: Stabilní soustava

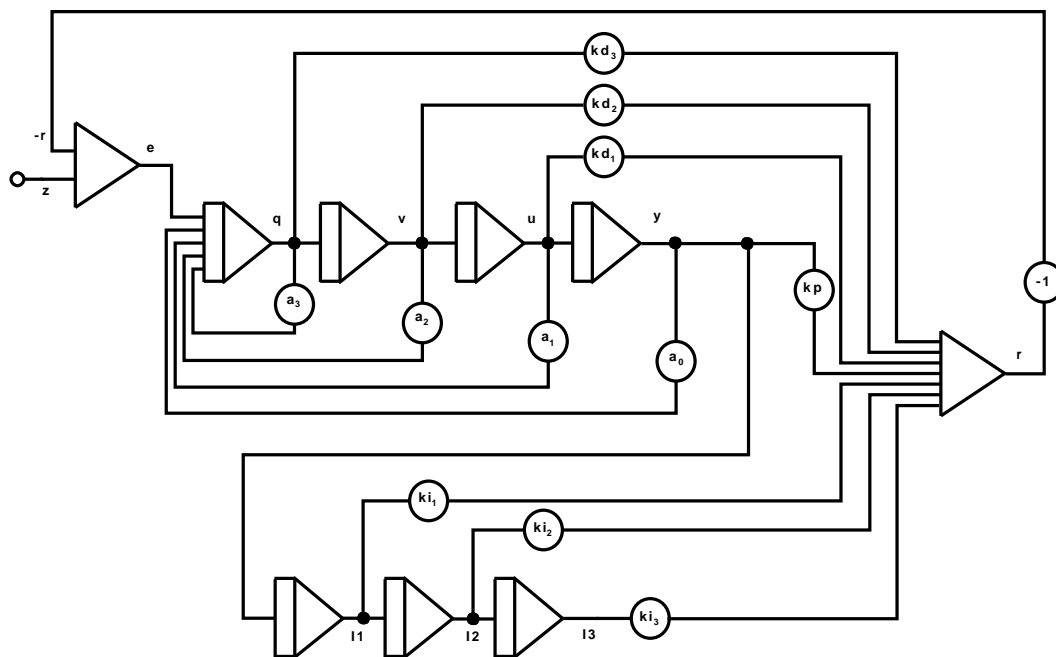
Že je tato soustava dlouhodobě stabilní nám dokáže simulace obvodu v programu TKSL.



Obrázek 5.2: Simulace stabilní soustavy

5.2.2 Vyladění modelu

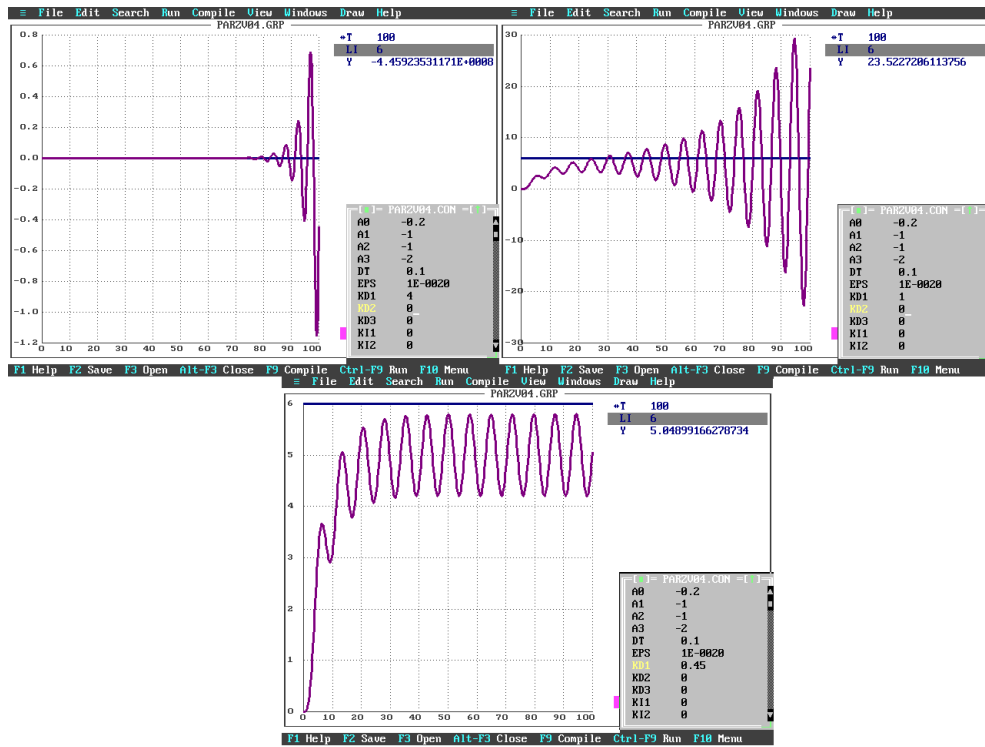
Dalším krokem je úprava modelu tak, abychom vyladili jeho chování. Tento krok je obecně velmi vhodné udělat, protože v modelu máme jedinečnou příležitost přistoupit k derivacím a integracím nosné proměnné. Prakticky přidáme do programového schématu nové prvky tak, abychom získali jednotlivé derivace a integrace sledovaného výstupního signálu y , a to vše za cílem experimentálního zjištění koeficientů. Programové schéma nyní vypadá následovně:



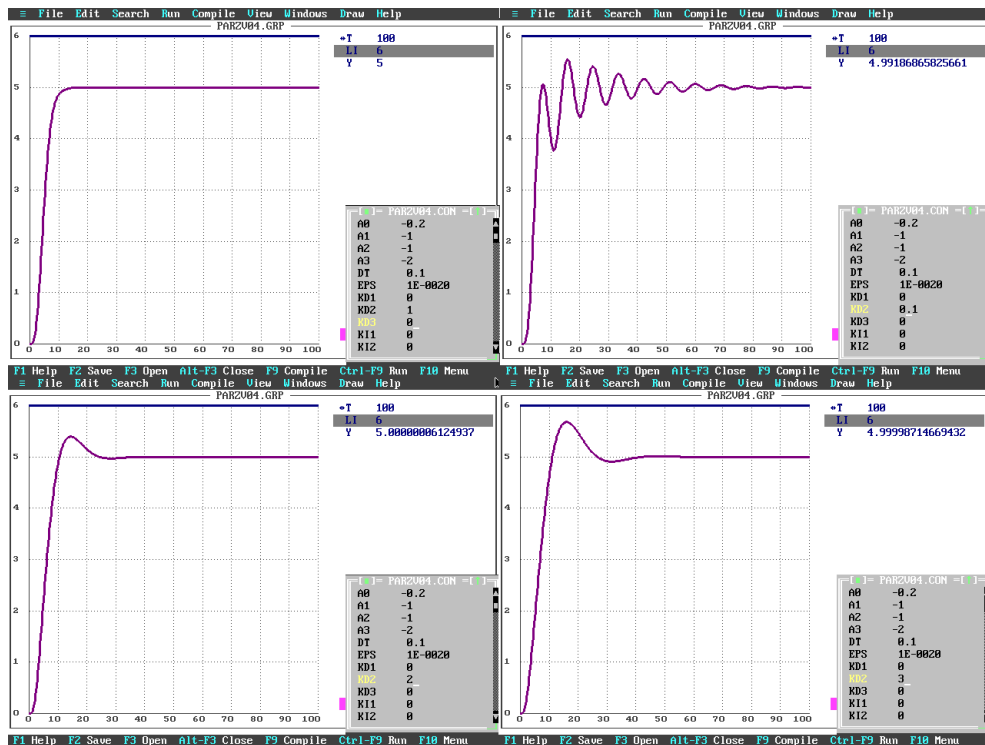
Obrázek 5.3: Rozšířená soustava

Povšimněme si zajímavého prvku této metodiky, a sice filozofii získání derivace signálu. Ačkoli je spolehlivý derivátor obtížně realizovatelný prvek, ve výpočtu jej můžeme nahradit stavem signálu před integrátorem. Tímto způsobem namodelované derivace poskytují v praxi velice použitelné výsledky. Nyní zbývá nastavit v simulátoru koeficienty tak, aby výsledný průběh odpovídal našim představám.

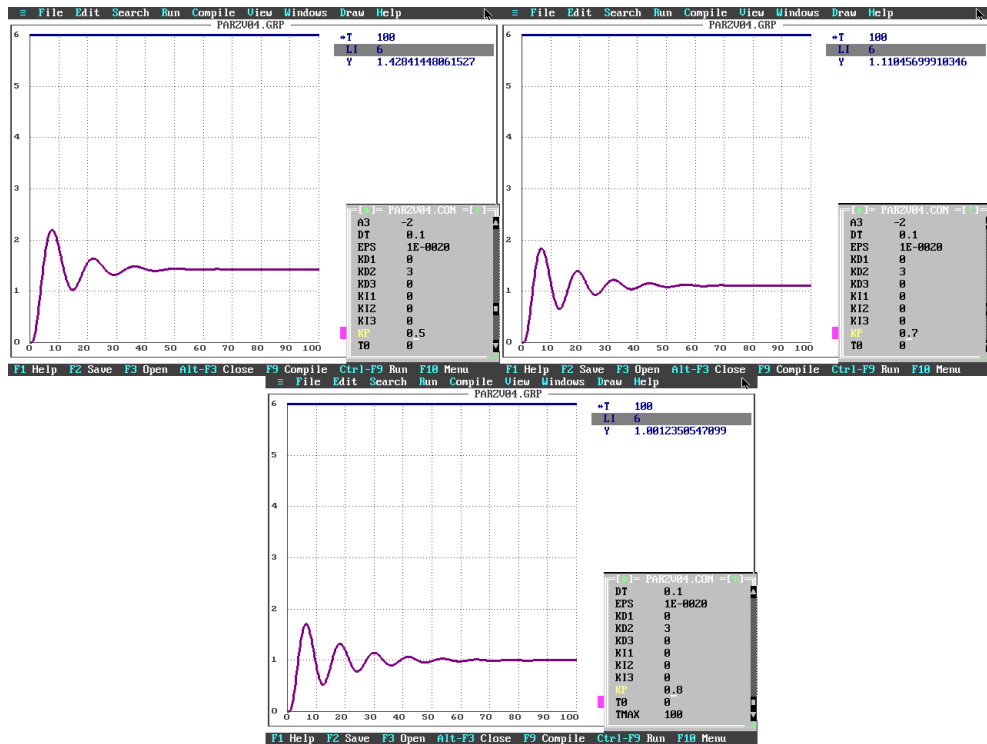
Některé z průběhů simulací, kdy bylo mj. zjištěno, že integrační konstanty nemají žádný pozitivní vliv na stabilitu obvodu:



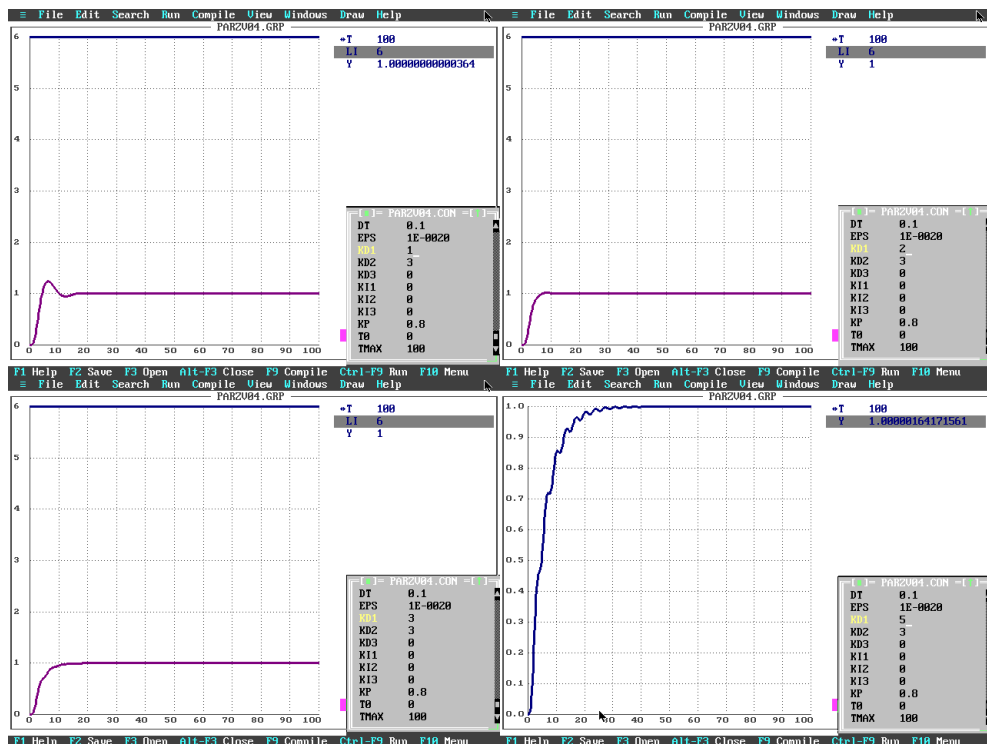
Obrázek 5.4: První experimenty s Kd_1



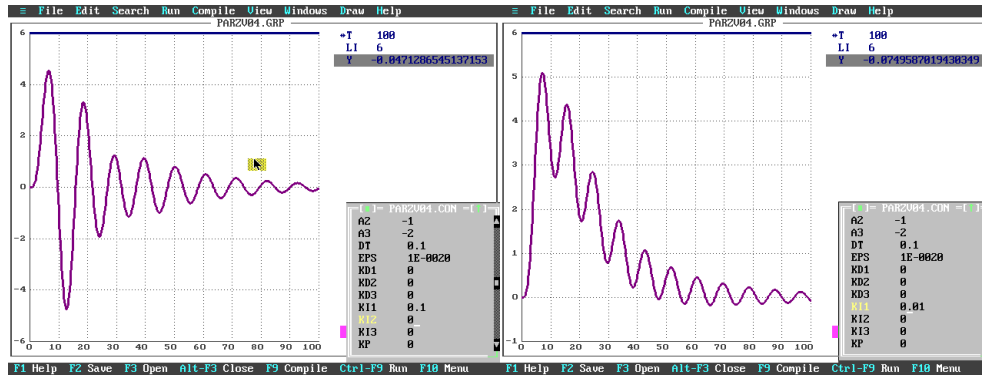
Obrázek 5.5: Vliv Kd_2



Obrázek 5.6: Vliv proporcionálního koeficientu



Obrázek 5.7: Další experimenty s K_{d1}

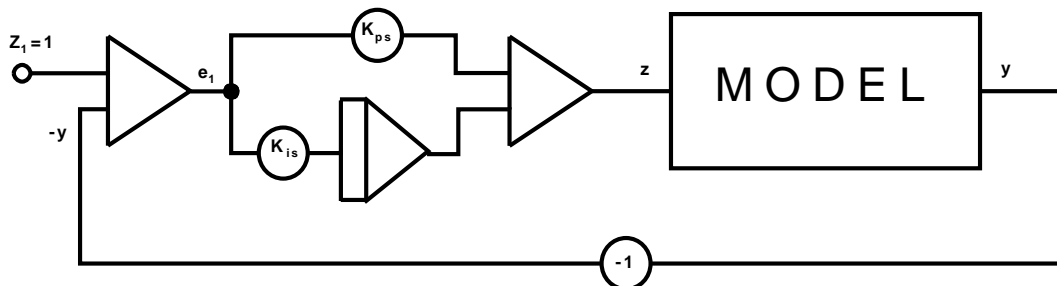


Obrázek 5.8: Negativní vliv integrací

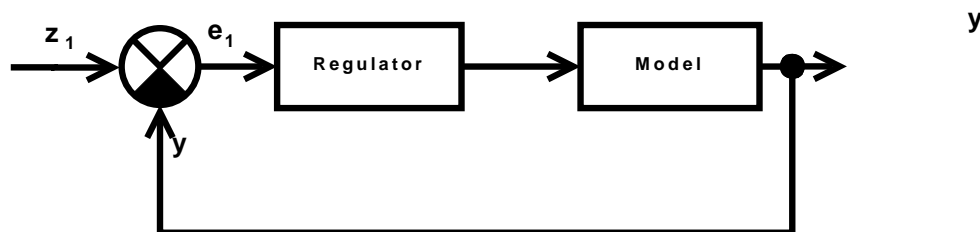
Tento krok je krokem obecným; jeho průběh se bude lišit vždy podle požadavků. Smyslem je pouze získat koeficienty pro danou aplikaci.

5.2.3 Připojení PI regulátoru

Dalším krokem je připojení PI regulátoru (D složku nevyužijeme z výše uvedených důvodů). Tentokrát je tedy cílem stanovit tímtéž způsobem koeficienty regulátoru samotného, při hodnotách ustáleného modelu získaného v minulém kroku. Bloková reprezentace tohoto kroku:

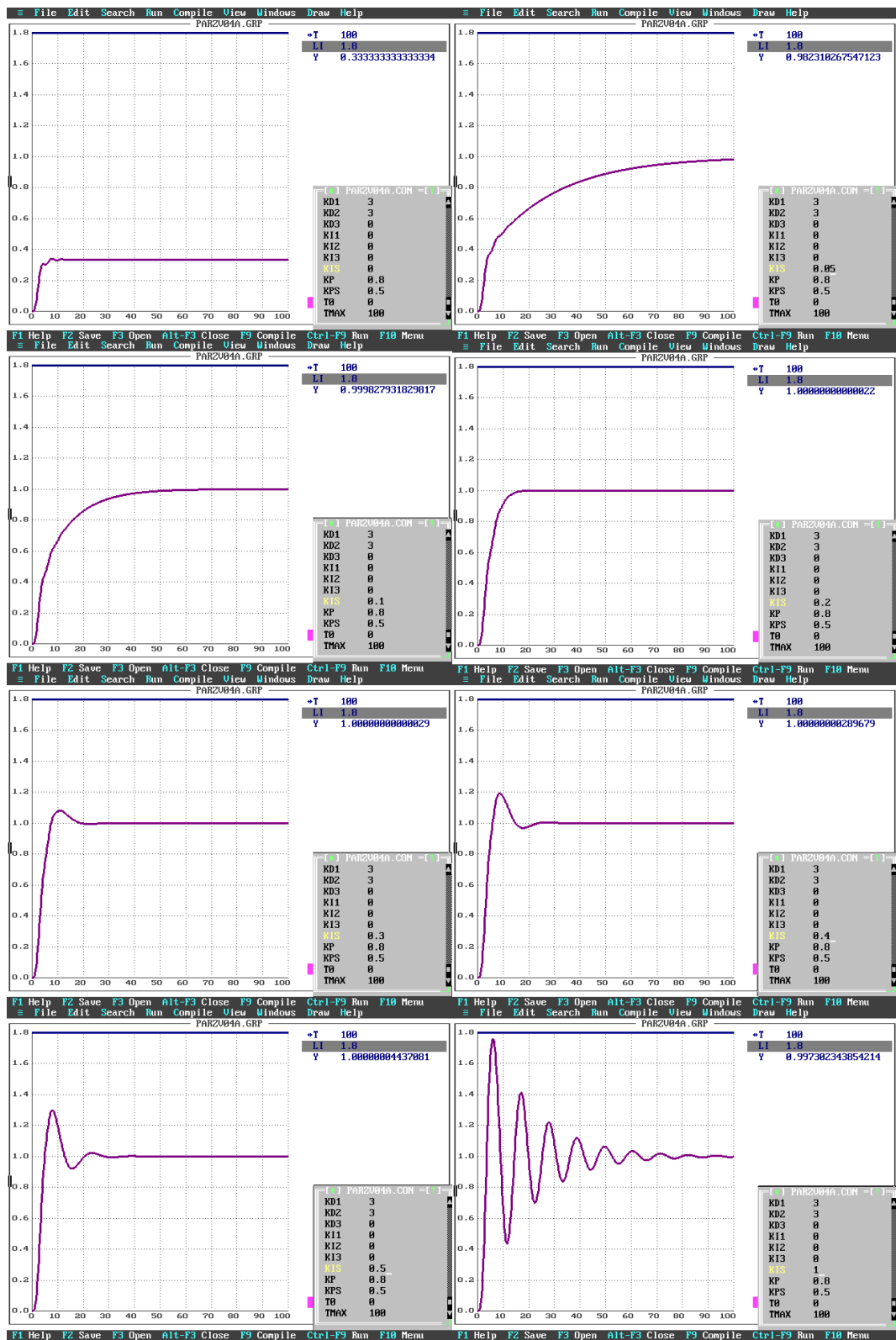


Obrázek 5.9: Idea připojení PI regulátoru

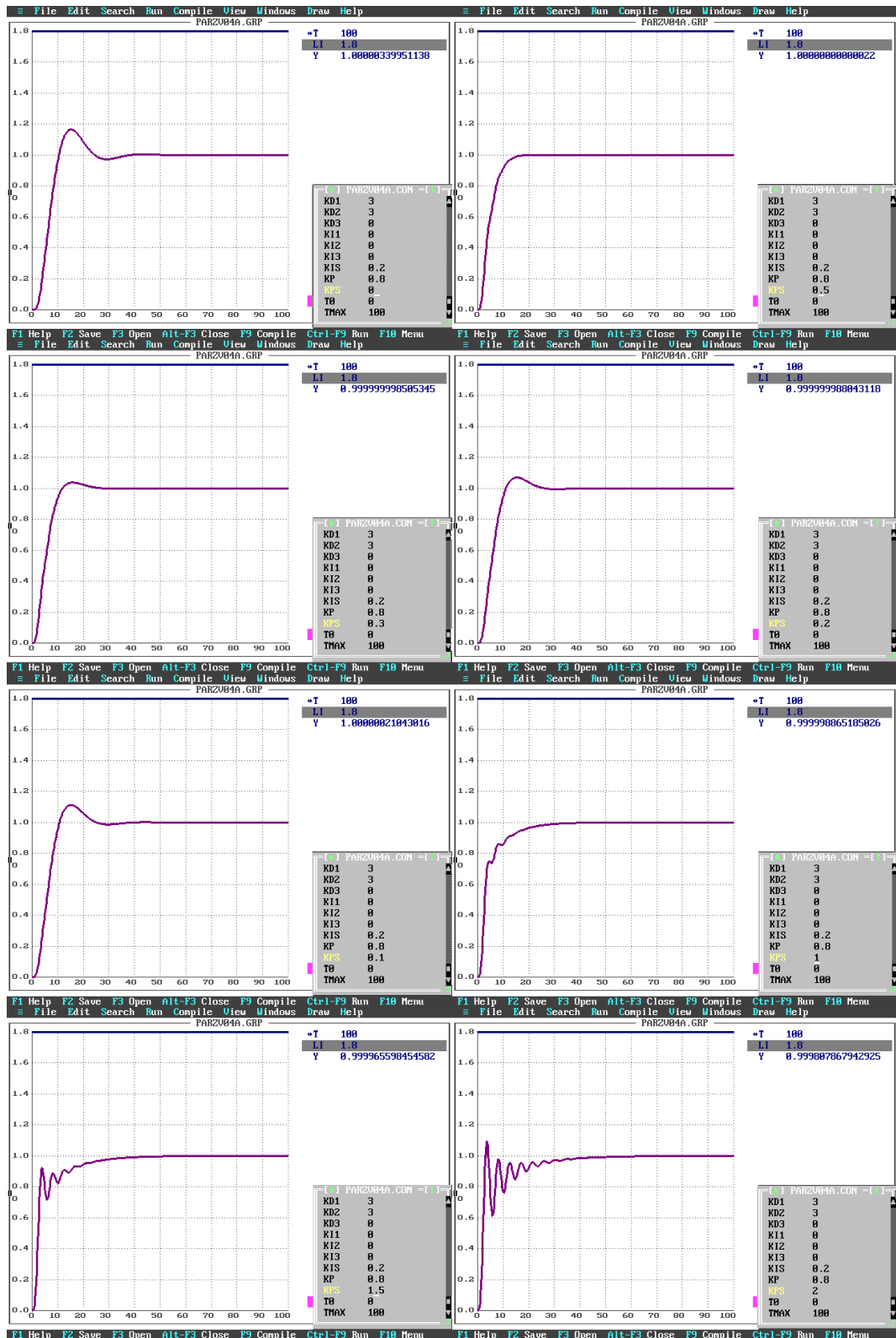


Obrázek 5.10: Idea v blokové algebře

Nyní je na řadě další řada simulací. Smyslem tohoto kroku v podstatě je přiblížit se co nejvíce výsledkem kroku předešlému.



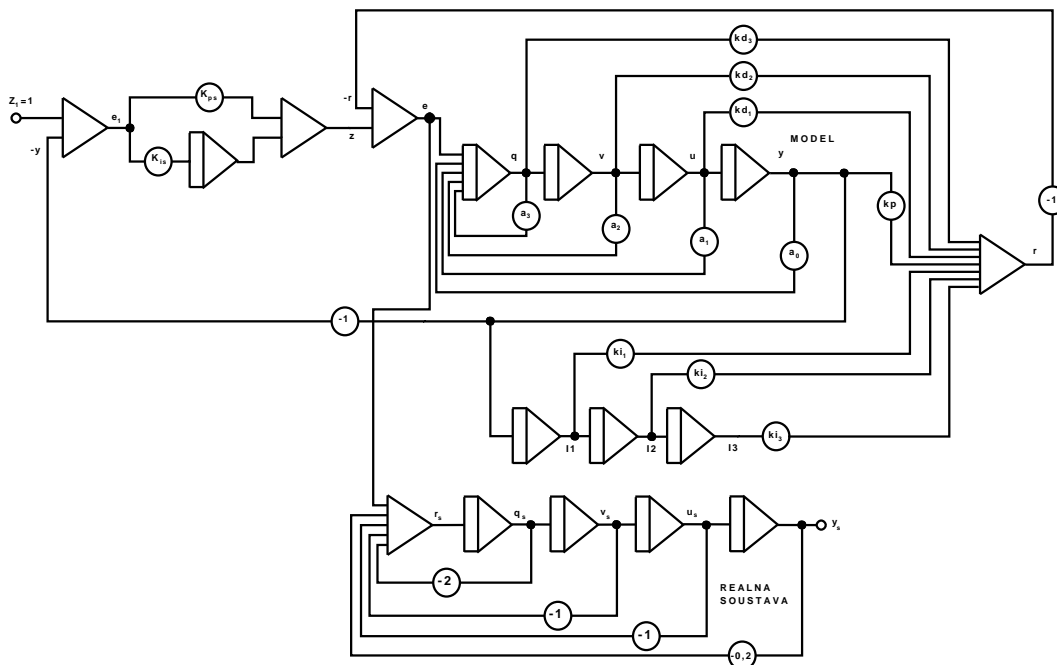
Obrázek 5.11: Vliv parametru $K_{i,s}$



Obrázek 5.12: Vliv parametru K_{ps}

5.2.4 Zavedení do reálné soustavy

Dalším ideovým krokem je zavedení získaného řídicího signálu e do reálné fyzické soustavy. Vznikne nám tak zpětná vazba uzavřená z výstupu modelu. Jak se ale při simulacích rychle ukáže, tento postup sám o sobě nestačí; již malá odchylka parametrů skutečné soustavy způsobí selhání řízení.

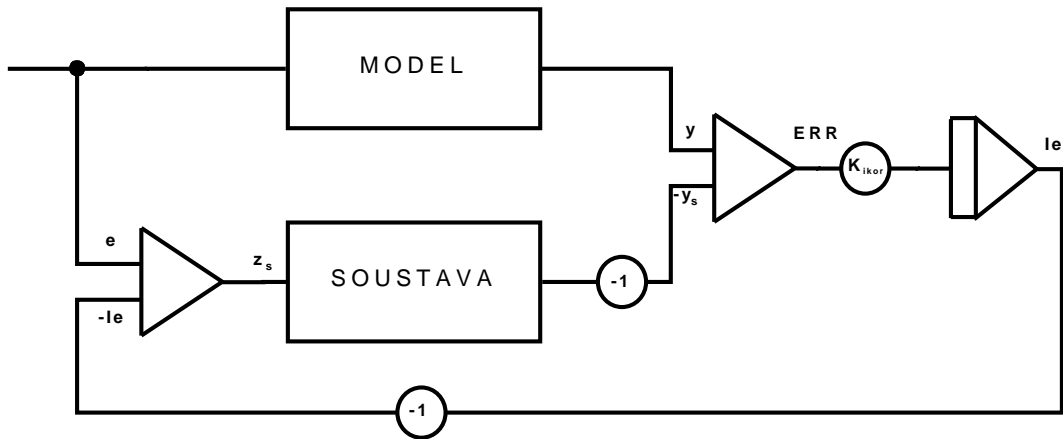


Obrázek 5.13: Přímé propojení modelu a soustavy

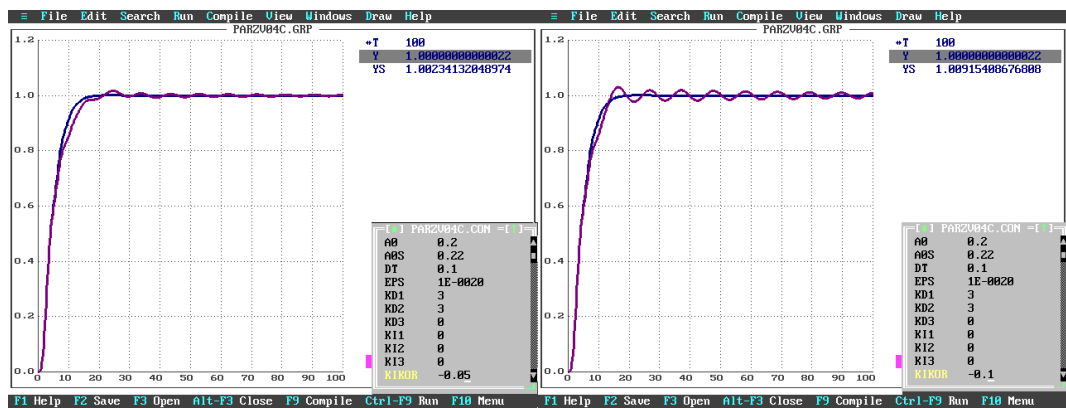
Tento přístup má ten důsledek, že pomocí analýzy odchylky můžeme v modelu zpětně identifikovat, který parametr skutečné soustavy se změnil a na jakou hodnotu. Tato skutečnost může být v praxi užitečná v mnoha směrech.

5.2.5 Zavedení korekce

Aby předchozí přístup fungoval, musíme ještě zavést opravu regulace podle výstupu reálné soustavy - v předchozím případě by totiž nešlo ani tolik o regulátor jako spíš o poněkud nefunkční ovládání. Výsledku dosáhneme zavedením dodatečného I-regulátoru s odpovídajícím koeficientem K_{ikor} . Ten zjistíme, jako v předchozích případech, experimentálně.



Obrázek 5.14: Zapojení s opravou chyby



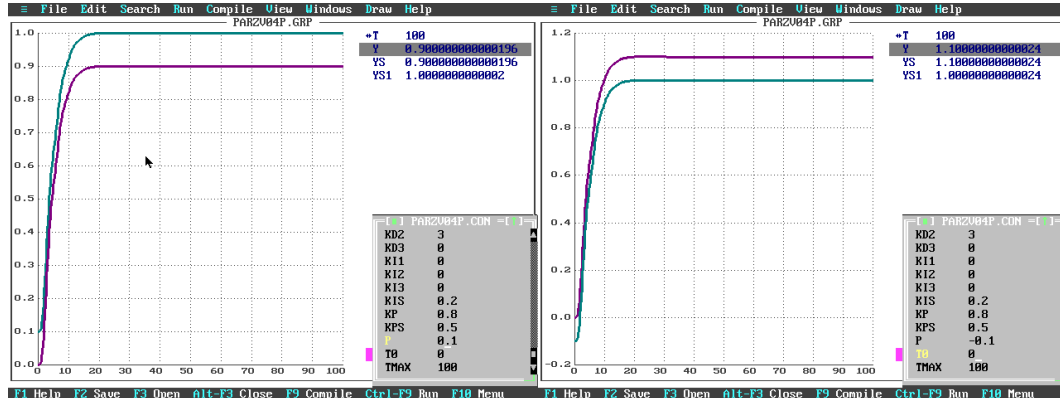
Obrázek 5.15: Simulace opravy chyby

5.2.6 Uzavření zpětné vazby přes soustavu

Dalším možným krokem je uzavření zpětné vazby přes reálnou soustavu. V zásadě jde o obměnu zavedeného principu, ovšem simulačně bylo prokázáno, že tato metoda nese stabilní výsledky. Technicky vzato je hlavní změnou ve výpočtu, že nyní $e_1 = z_1 - y_s$.

5.2.7 Test přičtením poruchy

Zbývá jen ověřit funkčnost systému při situaci, kdy se k výstupu přičte nějaká porucha - například konstantní výchylka v rozumné toleranci. Z reálné soustavy tak bude vycházet signál $y_{s1} = y_s + p$.

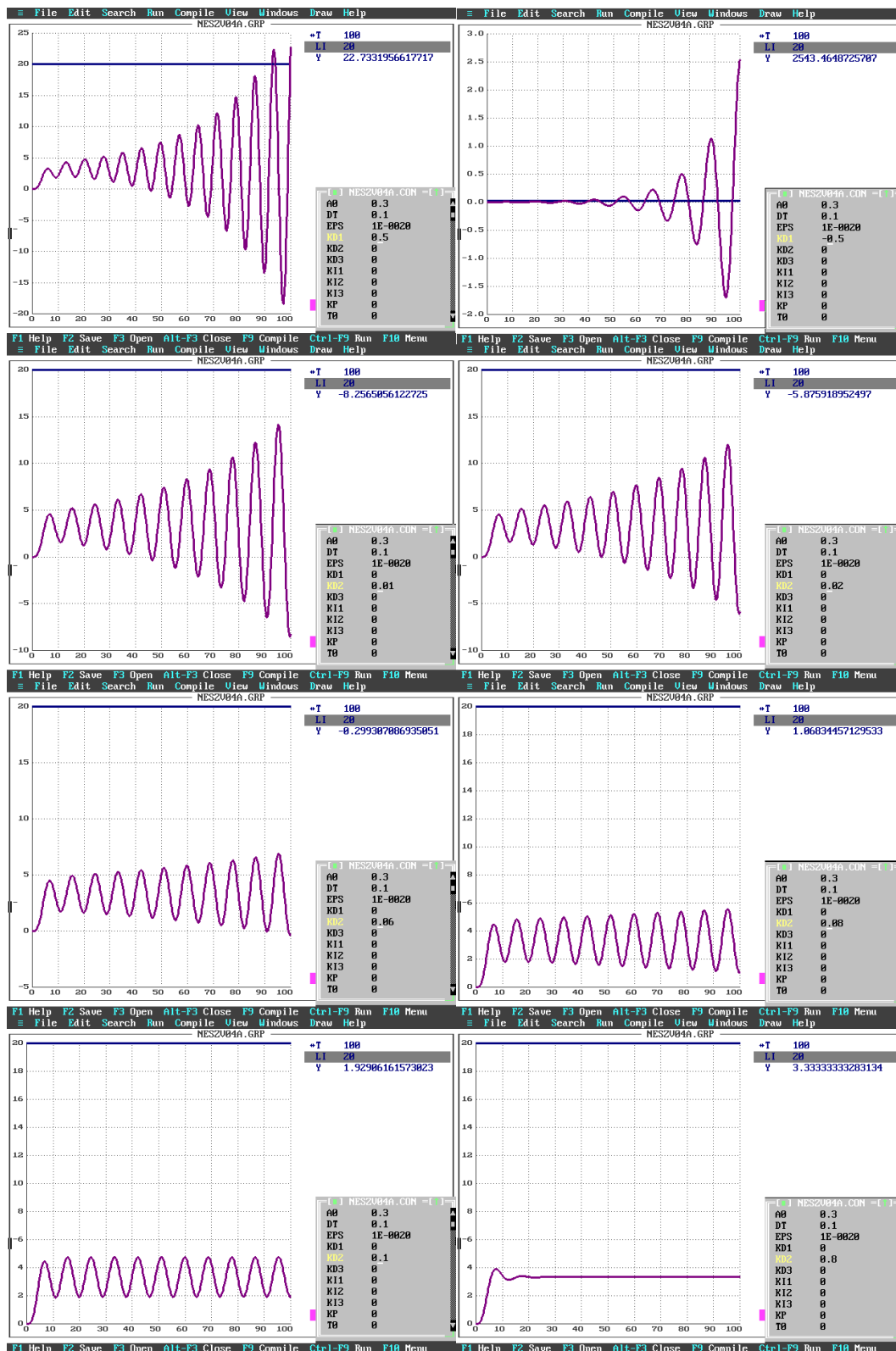


Obrázek 5.18: Výsledky pokusů s poruchou

Metodika se v simulacích osvědčila jako funkční, principiálně jednoduchá a efektivní metoda návrhu.

5.3 Nestabilní soustava

V případě dosažení nestabilní soustavy je celý postup prakticky totožný. I když soustava sama není stabilní, po nalezení správných regulačních koeficientů jsme schopni její průběh ustabilizovat a řídit stejně tak, jako by byla stabilní. Proto zde uvádím jen několik výsledků znázorňujících začátek postupu hledání hodnot.



Obrázek 5.19: Demontrace stabilizace nestabilního systému

Kapitola 6

Simulační nástroje

Protože již byly prezentovány některé výsledky simulací, je nejvyšší čas zmínit použítá prostředí.

6.1 TKSL

Předchozí kapitoly používaly pro simulace výhradně nástroje TKSL. Jde o simulační nástroj založený na numerickém výpočtu s využitím Taylorovy řady. Ačkoli je současná verze již staršího data a vývoj novější implementace stále probíhá, daří se TKSL dosahovat přesných výsledků v krátkém čase. Data vstupují v podobě zdrojového kódu, který je založen právě na blokových grafech programových schémat, metodách jejich řešení a implementaci.

6.2 Matlab a Simulink

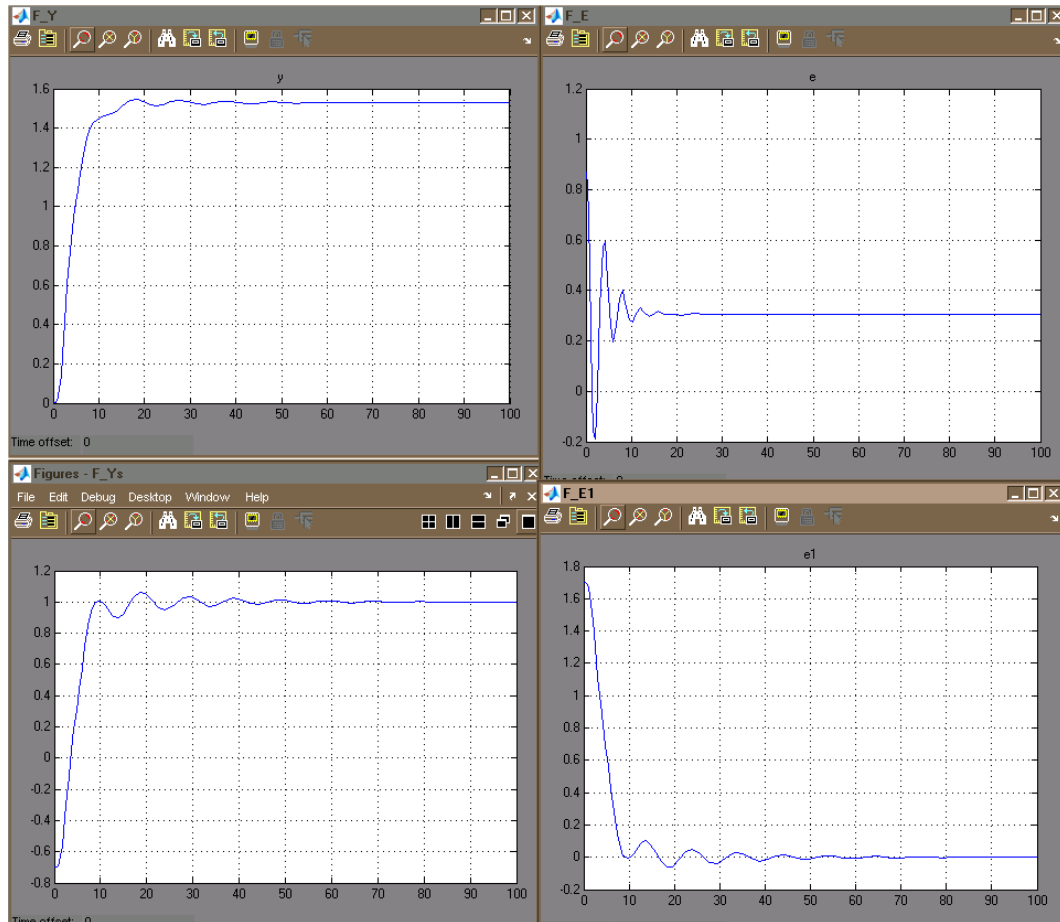
Na výsledky dosažené v TKSL by mohl někdo namítnout, že nejsou důkazy o jeho přesnosti; proto je vhodné využít i jiného, rozšířeného a všeobecně uznávaného matematického nástroje. Systém Matlab má pověst jednoho z nejrobustnějších programů pro řešení matematických problémů. Obsahuje mnoho dílčích modulů, a mezi nimi simulační nástroj Simulink.

Pro ověření jsem v tomto prostředí namodeloval kompletní výslednou soustavu. Výsledky byly shodné, proto pro další rozvoj metodiky uvedu namodelované situace, kdy byl do modelu reálného systému zaveden nějaký stupeň chyb.

6.2.1 Soustava lehce rozladěná s velkou chybou výstupu

V tomto modelu bylo použito mírně posunutých koeficientů reálné stabilní soustavy. Pro připomenutí, koeficienty modelu (a reálné nerozladěné soustavy) jsou $a_0 = -0.2$, $a_1 = -1$, $a_2 = -1$, $a_3 = -2$.

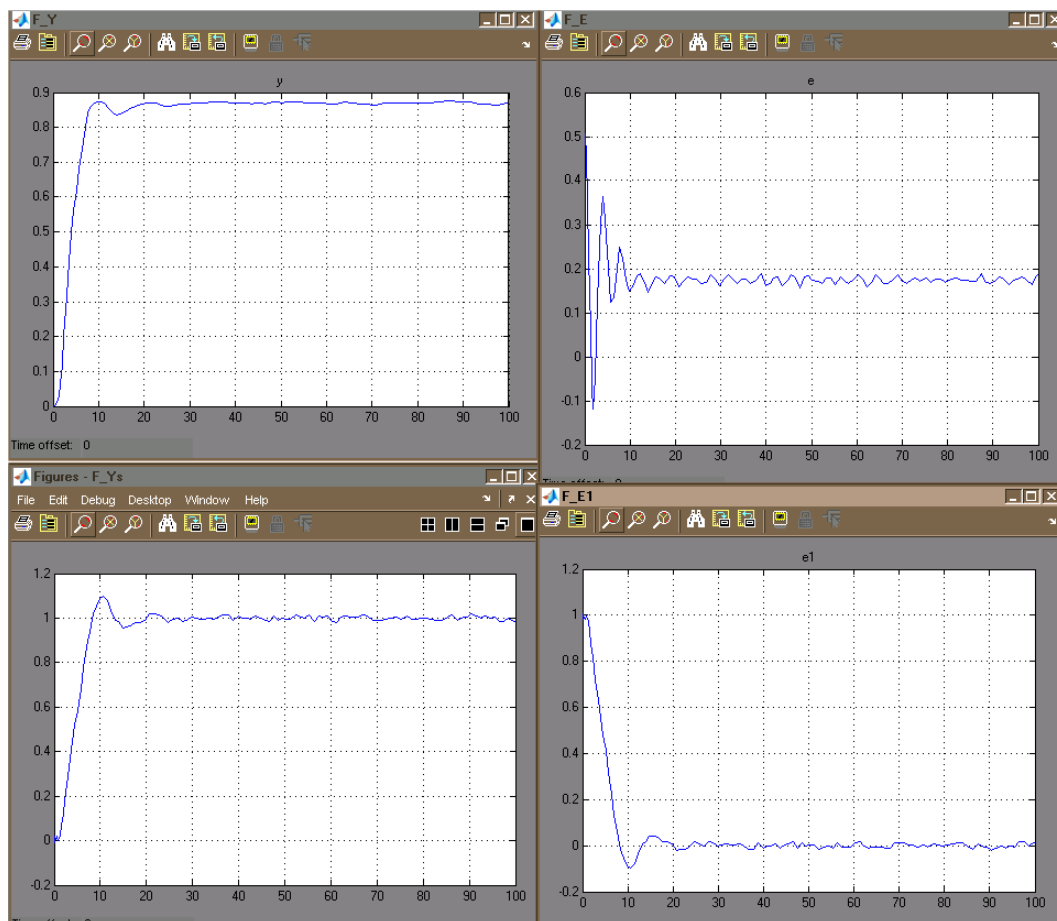
V tomto experimentu byla reálná soustava rozladěna následujícím způsobem: $a_{0s} = -0,18$, $a_{1s} = -0,9$, $a_{2s} = -1,1$, $a_{3s} = -2,1$. K výstupu y_s pak byla připojena velká parazitní chyba $p = -0,7$ (regulátor má snahu udržet hodnotu 1). Na snímcích jsou zleva doprava po řádcích hodnoty y , e , y_s , e_1 .



Obrázek 6.1: Systém s velkou parazitní chybou

6.2.2 Soustava silně rozladěná s malým šumem na výstupu

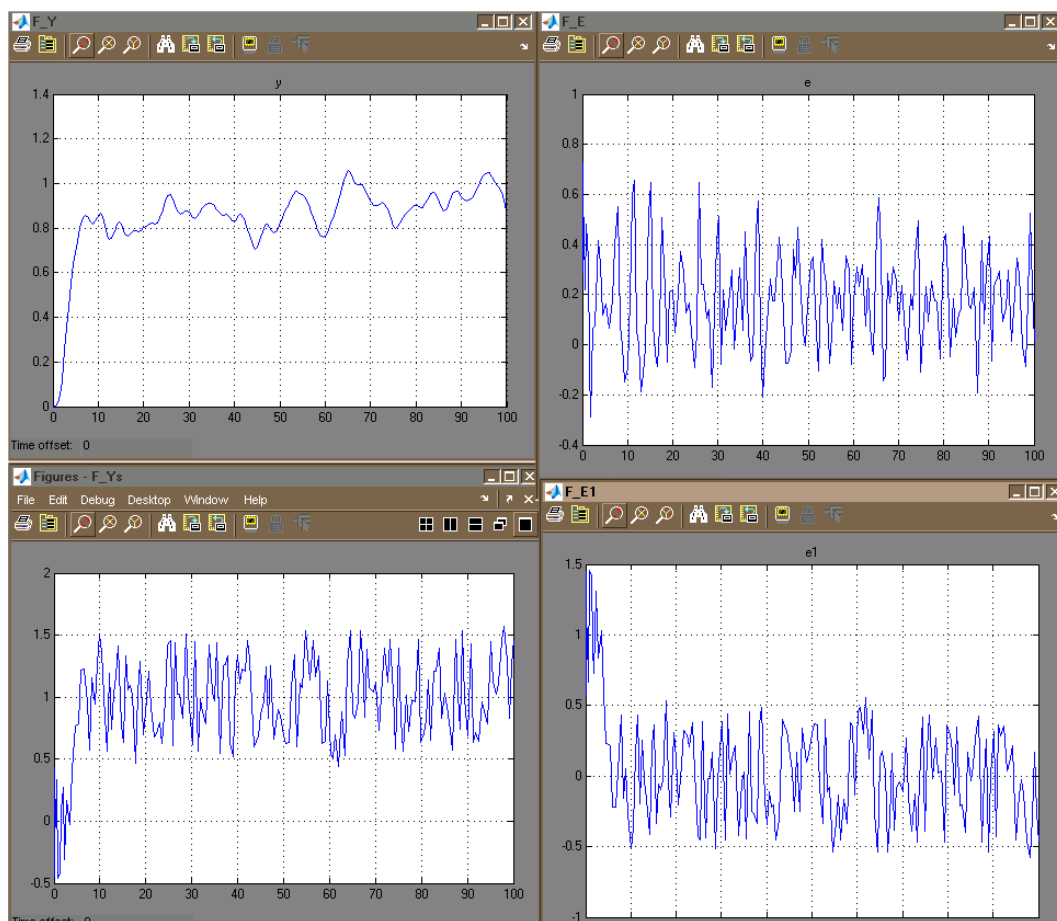
Následující experiment testoval chování v situaci s opačným typem chyby, a sice s relativně hodně rozladěným reálným systémem a takřka zanedbatelným šumem zkreslujícím výstupní signál. V tomto případě měla daná soustava parametry $a_{0s} = -0,175$, $a_{1s} = -0,91$, $a_{2s} = -1,4$, $a_{3s} = -2,1$ a zkreslení výstupu p nabývalo hodnot od $-0,1$ do $0,02$.



Obrázek 6.2: Rozladěná soustava

6.2.3 Soustava silně rozladěná s velkým šumem na výstupu

Závěrečný experiment etapy ověřoval chování soustavy v extrémním případě, kdy se reálný systém nechová tak, jak by měl, a jeho výstup je zatížen náhodnou chybou dosahující zkreslení až 50% požadovaného výstupu. Pro pořádek, reálná soustava měla parametry $a_{0s} = -0,175$, $a_{1s} = -0,91$, $a_{2s} = -1,4$, $a_{3s} = -2,1$, $p = -0,5 \dots 0,5$.



Obrázek 6.3: Silně poruchová soustava

Kapitola 7

Dopravní zpoždění

Dopravní nebo časové zpoždění, někdy též transportní zpoždění, se vyskytuje u řešení celé řady úloh. Časové zpoždění může být konstantní nebo proměnné. Proměnné časové zpoždění se vyskytuje v úlohách z termodynamiky, chemie apod. Konstantní časové zpoždění se vyskytuje v úlohách z automatického řízení, v biologii, v ekonomice, v dynamice úloh s lidským faktorem apod.

Časové zpoždění znamená, že daná funkce $f(t)$ je zpožděna o určitý časový úsek T , čili z funkce $f(t)$ se vytvoří funkce $f(t - T)$.

7.1 Modelování časového zpoždění

Funkci $g(t)$ časově zpožděnou o T vzhledem k funkci $f(t)$ můžeme zapsat výrazem

$$g(t) = f(t - T) \quad (7.1)$$

kde $f(t) = 0$ pro každou hodnotu $t < 0$.

Vyjádríme-li rovnici 7.1 Laplaceovou transformací, obdržíme

$$G(s) = e^{-sT} F(s) \quad (7.2)$$

Z rovnice 7.2 pak obdržíme operátorový přenos

$$H(s) = \frac{G(s)}{F(s)} = e^{-sT} \quad (7.3)$$

Tento výraz však nejde přímo modelovat. Pro tuto potřebu je velmi vhodné užít aproximace funkce e^{-sT} v racionálně lomenou funkcí pomocí tzv. Padého rozvoje

$$e^{-sT} \approx \frac{Pn(s)}{Qn(s)} \quad (7.4)$$

, kde

$$Pn(s) = 1 - \frac{sT}{2} + \frac{n(n-1)s^2T^2}{2n(2n-1) \cdot 2!} - \dots + \frac{(-1)^n n! s^n T^n}{(2n)!}$$
$$Qn(s) = 1 + \frac{sT}{2} + \frac{n(n-1)s^2T^2}{2n(2n-1) \cdot 2!} + \dots + \frac{n! s^n T^n}{(2n)!} \quad (7.5)$$

Řád	$P_n(p)$
$n = 0$	1
$n = 1$	$1 - \frac{Tp}{2}$
$n = 2$	$1 - \frac{Tp}{2} + \frac{(Tp)^2}{12}$
$n = 3$	$1 - \frac{Tp}{2} + \frac{(Tp)^2}{10} - \frac{(Tp)^3}{120}$
$n = 4$	$1 - \frac{Tp}{2} + \frac{3(Tp)^2}{28} - \frac{(Tp)^3}{84} + \frac{(Tp)^4}{1680}$
$n = 5$	$1 - \frac{Tp}{2} + \frac{(Tp)^2}{9} - \frac{(Tp)^3}{72} + \frac{(Tp)^4}{1680} - \frac{(Tp)^5}{7560}$
$n = 6$	$1 - \frac{Tp}{2} + \frac{5(Tp)^2}{44} - \frac{(Tp)^3}{132} + \frac{(Tp)^4}{1584} - \frac{(Tp)^5}{5080} + \frac{(Tp)^6}{665280}$

Tabulka 7.1: Tabulka Padého rozvoju

V tabulce 7.1 jsou uvedeny mnohočleny $P_n(s)$ používané při aproximaci přenosové funkce do šestého řádu při Padého rozvoji. Za předpokladu nulových počátečních podmínek můžeme používat diferenciální operátor p .

Vytvoříme model zpožďovací linky pomocí Padého rozvoje druhého řádu. Podle hodnot v tabulce sestavíme diferenciální rovnici

$$\frac{u_0(t)}{u_1(t)} = \frac{1 - \frac{Tp}{2} + \frac{(Tp)^2}{12}}{1 + \frac{Tp}{2} + \frac{(Tp)^2}{12}} \quad (7.6)$$

Rovnici upravíme vynásobením 12-ti a obdržíme

$$\frac{u_0(t)}{u_1(t)} = \frac{T^2p^2 - 6Tp + 12}{T^2p^2 + 6Tp + 12} \quad (7.7)$$

Rovnici 7.7 upravíme

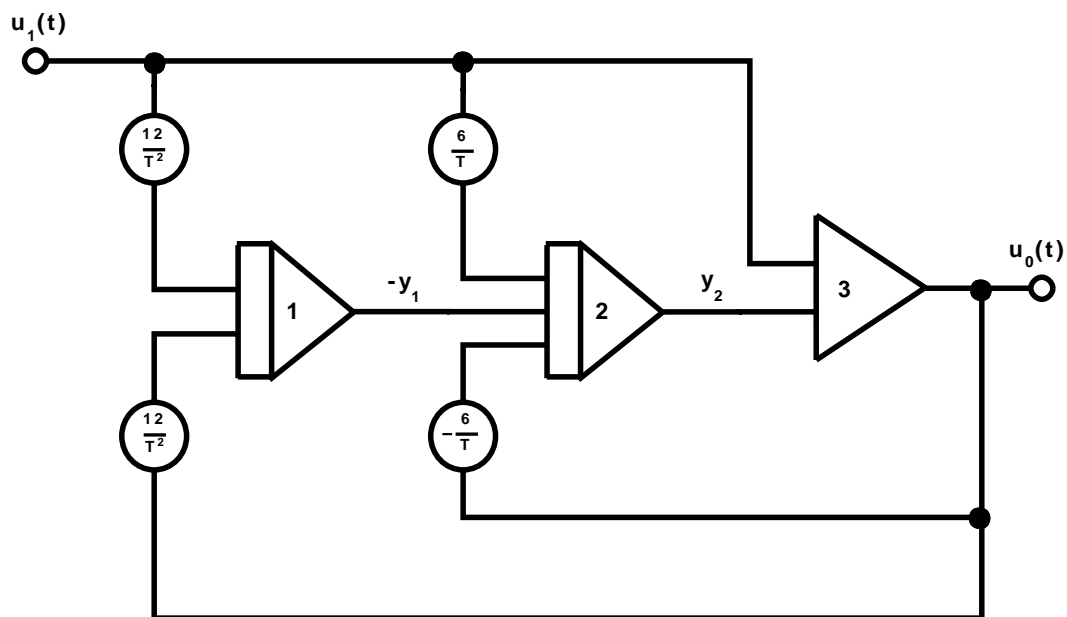
$$T^2p^2u_0 + 6Tpu_0 + 12u_0 = T^2p^2u_1 - 6Tpu_1 + 12u_1$$

$$p^2u_0 + \frac{6}{T}pu_0 + \frac{12}{T^2}u_0 = p^2u_1 - \frac{6}{T}pu_1 + \frac{12}{T^2}u_1 \quad (7.8)$$

$$\textcircled{1} \quad u_0 = -(-u_1 + y_2)$$

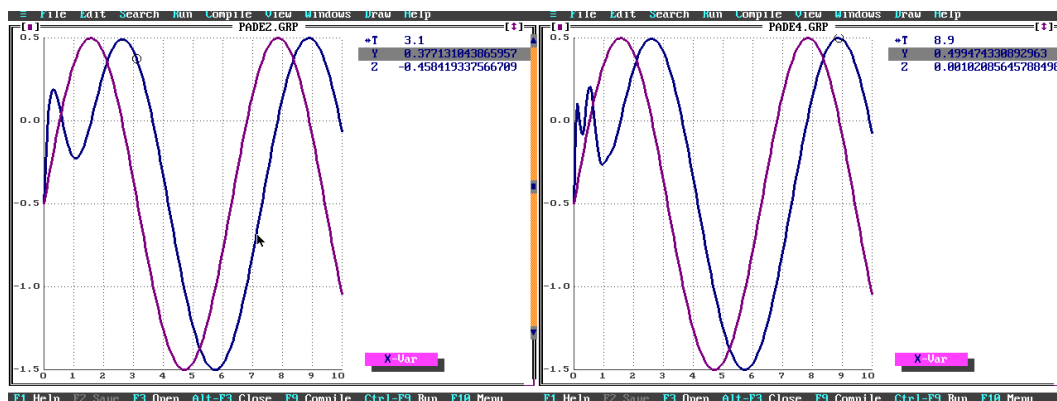
$$\textcircled{2} \quad y_2 = -\frac{1}{p}\left(-\frac{6}{T}u_0 - \frac{6}{T}u_1 - y_1\right)$$

$$\textcircled{3} \quad -y_1 = -\frac{1}{p}\left(\frac{12}{T^2}u_p - \frac{12}{T^2}u_1\right) \quad (7.9)$$



Obrázek 7.1: Programové schéma pro Padého rozvoj 2.řádu

Pro názornost předkládám výsledek simulace Padého rozvoje 2. a 4. řádu aplikovaného na funkci $z = \sin(t) - 0,5$.



Obrázek 7.2: Výsledek namodelovaného rozvoje 2. a 4. řádu

Kapitola 8

Kvantizace a převodníky

Pro převod mezi spojitým a diskrétním signálem, obecně mezi dvěma veličinami různého fyzikálního rozměru, je třeba použít převodníky. Pro naše potřeby jsou podstatné jen A/D a D/A převodníky. S jejich použitím jsou spojeny 2 problémy: dopravní zpoždění a kvantizace.

8.1 Vzorkování

Převod spojitého signálu na diskrétní se provádí pomocí vzorkování. Úsek spojitého signálu se sice dá donekonečna zvětšovat a pozorovat tak jeho nekonečně malé detaily, ale protože počítače mají pouze konečnou kapacitu paměti a ani nejsou nekonečně rychlé, musíme se u reálného vzorkování při A/D převodu omezit pouze na nezbytně nutné množství vzorků, které budeme dále zpracovávat.

Obecně se vzorkování signálu provádí tím způsobem, že rozdělíme vodorovnou osu signálu (ve většině případů čas) na rovnoměrné úseky a z každého úseku odebereme jeden vzorek. Je přitom zřejmé, že tak z původního signálu ztratíme mnoho detailů, protože namísto spojitě čáry, kterou lze donekonečna zvětšovat, dostáváme pouze množinu diskrétních bodů s intervalem odpovídajícím použité vzorkovací frekvenci.

8.2 Aliasing

Chyba vzorkování může ovšem být ještě daleko horší. Pokud se totiž v původním spojitém signálu vyskytuje frekvence vyšší, než je polovina frekvence vzorkovací (nazývaná též Nyquistova frekvence), dojde, jak praví Shannonův teorém, k úplnému a nenávratnému zkreslení signálu díky jevu nazývanému se aliasing. Aliasingu se dá zabránit jedině anti-aliasing filtrem, což je dolní propust' zařazená před převodníkem. Ta nedovolí frekvencím vyšším než je Nyquistova frekvence vstoupit do převodníku.

Pro naše potřeby není aliasing podstatný, přesto však jde o nedílnou součást problematiky.

8.3 Kvantování

Vzhledem k tomu, že počítače a další zařízení dále zpracovávající digitální signál umí vyjádřit čísla pouze s omezenou přesností, je potřeba navzorkované hodnoty upravit i na svislé ose. Protože se hodnota vzorku dá vyjádřit pouze po určitých kvantech, nazýváme tuto fázi A/D převodu kvantování.

Aby bylo možné určit, které hodnoty má po kvantování nabývat určitý vzorek, je třeba rozdělit prostor kolem jednotlivých hodnot na toleranční pásy.

Kterémukoliv vzorku, který padne do daného tolerančního pásu, je při kvantování přiřazena daná hodnota. Je zřejmé, že se kvantované hodnoty ve většině případů liší od skutečných navzorkovaných hodnot. Velikost kvantizační chyby je vzdálenost mezi kvantovanými a původními navzorkovanými body. Velikost této chyby se pohybuje v intervalu $\pm \frac{1}{2}$ kvantizační úrovně. Počet kvantizačních úrovní bývá zpravidla rovný mocnině čísla 2.

8.4 Kvantizační šum

Pokud bychom vynesli velikosti chyb od jednotlivých vzorků do grafu, získali bychom náhodný signál, kterému se říká kvantizační šum. Velikost šumu je zvykem vyjadřovat jako poměrné číslo v decibelech, a sice jako poměr užitečného signálu ku šumu. Protože číslo ve jmenovateli zlomku - kvantizační chyba - je u všech lineárních převodníků stejná (interval $\pm \frac{1}{2}$ kvantizační úrovně), závisí velikost kvantizačního šumu jen na čitateli zlomku, tedy na velikosti užitečného signálu, což je maximální počet kvantizačních úrovní daného převodníku.

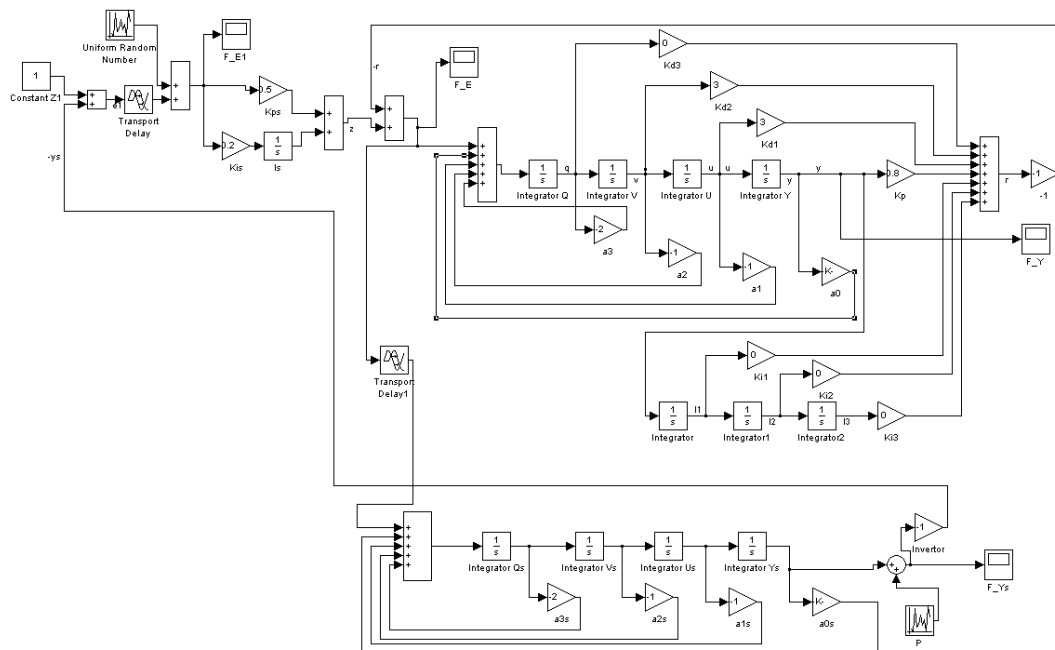
$$S N R_{A/D} = 20 \cdot \log 2^N \approx 6,02 \cdot N [dB] \quad (8.1)$$

Díky diskretizaci původního spojitého signálu ve dvou osách nemůže být ve většině případů signál zpětně převedený z digitální podoby do analogové přesně odpovídat původnímu signálu.

8.5 Modifikace modelu

Nyní, když již máme zavedeny pojmy převodníků, je na čase zavést je do naší soustavy. Ačkoli se oba negativní jevy - dopravní zpoždění a chyba kvantizace - projevují zároveň, pro potřeby modelování je nejen možné, ale i vhodné jejich vliv prozkoumat odděleně.

V modelu byly převodníky doplněny v podobě zpožďovacích členů v místě jejich zapojení. V případě A/D převodníku se projeví ještě kvantizace signálu; operační chybu D/A převodníku považuji za zanedbatelnou. Při všech experimentech jsem uvažoval systém s koeficienty přesně odpovídajícími modelu s tím, že na výstup byl zaveden minimální šum nabývající odchylek v intervalu $-0,01 \dots 0,01$



Obrázek 8.1: Kompletní model systému v prostředí Simulink

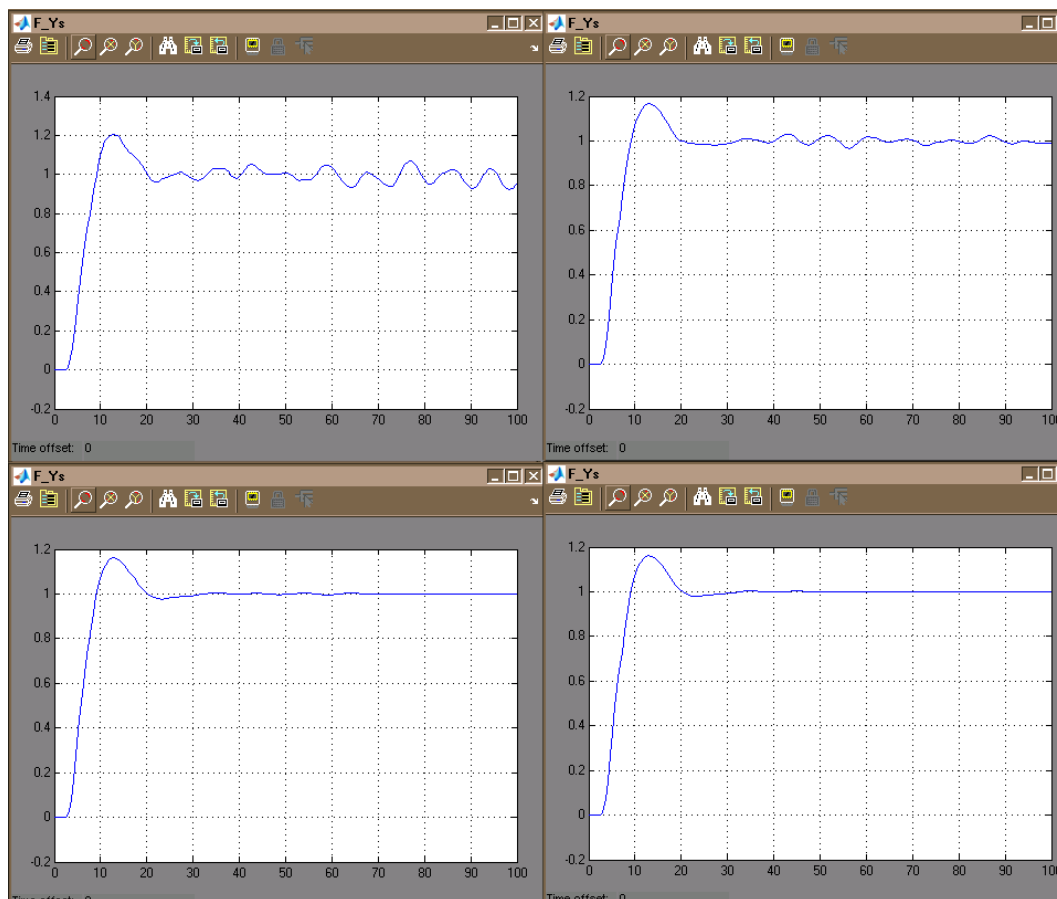
8.5.1 Aplikace kvantizace v modelu

Při pomyšlení na simulaci kvantizace v modelu jistě kdekoho napadne, že jednoduše zkvantujeme signál vstupující do modelu, ovšem to není postup vedoucí k cíli. Musíme si uvědomit, že ačkoli jsou regulátor a soustava odděleny A/D a D/A převodníkem, ve skutečnosti oba zpracovávají stejný signál - jen různými metodami. Nakvantováním signálu by model zpracovával signál odlišný, kvantovaný hned dvakrát.

Pro modelování vlivu kvantovací úrovně jsem tedy využil úvahy, čím se vlastně kvantování v modelu projeví - není to samotnými hodnotami, nýbrž chybou, která je způsobena zaokrouhlováním signálu na diskrétní hodnoty. Tento jev můžeme bez větších obtíží namodelovat generátorem náhodného čísla v rozsahu této chyby daném množstvím použitých bitů. Tento generátor bude nabývat v každém kroku výpočtu náhodné hodnoty v rozsahu

$$k = \pm \frac{1}{2^{n+1}} \quad (8.2)$$

, kde n je počet bitů, do kterých je signál kódován. Jeho výstup se pak sečte se signálem vstupujícím v místě A/D převodníku do systému. Tím je zachován praktický vliv kvantování z hlediska modelu i reálného systému.



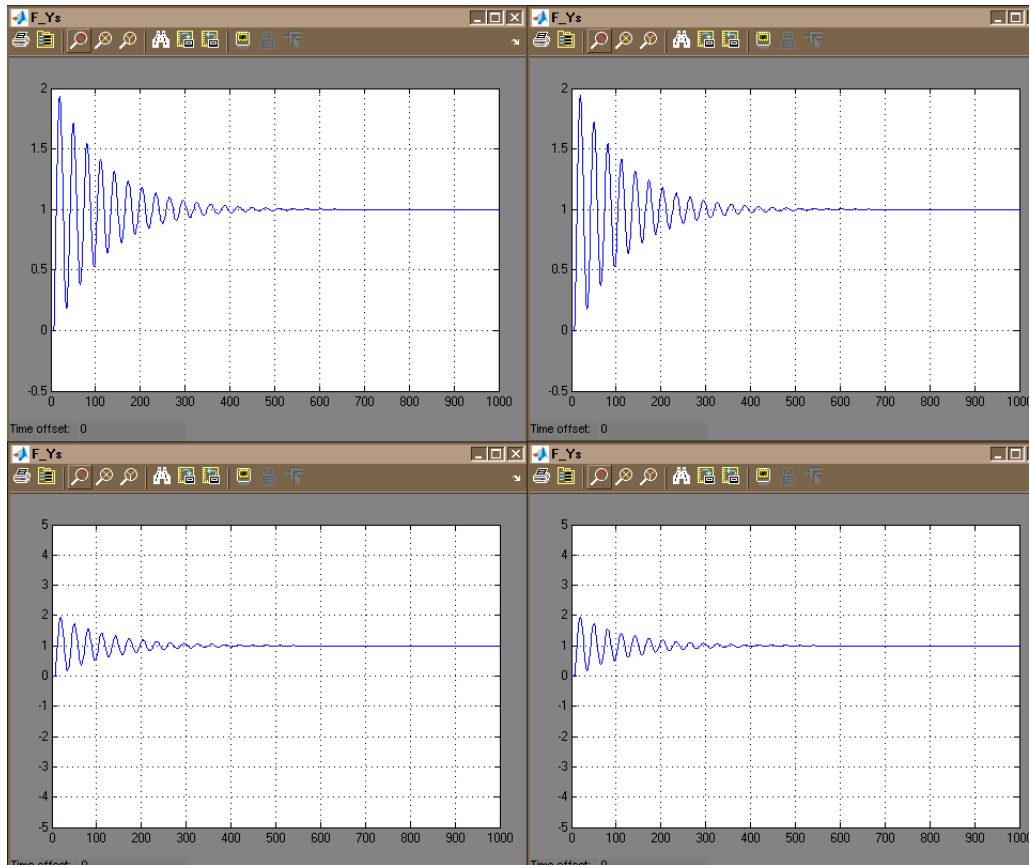
Obrázek 8.2: Vliv různých kvantovacích úrovní

Experiment ukázal výsledky (zleva doprava po řádcích) pro nejběžnější používané úrovně: 3 bity, 4 bity, 8 bitů a 16 bitů.

8.5.2 Vliv dopravního zpoždění v převodnicích

Nyní je na řadě poslední krok, a sice podívat se na vliv časového zkreslení signálů díky nutným mezikomponentům - A/D a D/A převodníku. Jejich rychlost je, jak se ukázalo, kritická pro funkci systému. Jejich namodelování se provádělo v numerickém prostředí, tedy jsou veškeré získané hodnoty v násobcích základního výpočetního kroku použité metody.

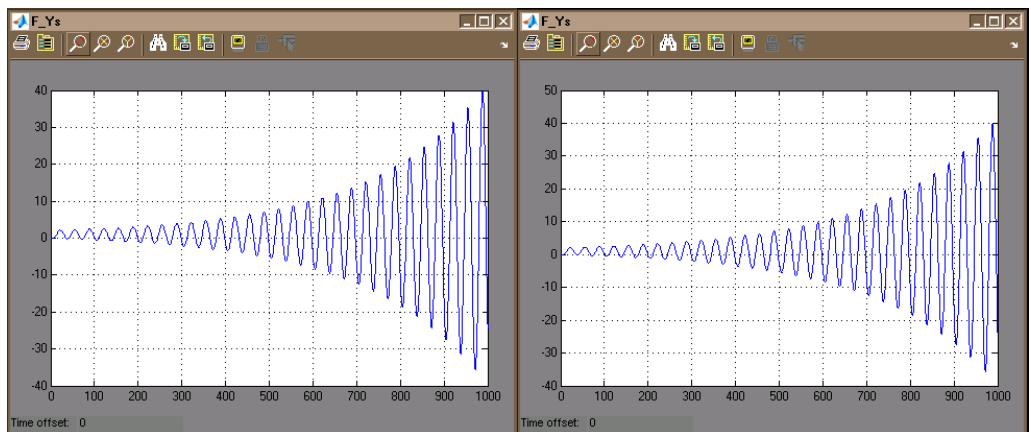
Nejdříve několik příkladů výstupu, který se sice po zavedení zpoždění rozkmital, nicméně soustava a hlavně regulátor se ještě dokázaly ustabilizovat na požadované úrovni.



Obrázek 8.3: Přijatelné hodnoty zpoždění

Zleva doprava po řádcích jde o výstupy na následující dvojice zpoždění ve tvaru A/D - D/A: 2-4, 4-2, 0-6, 6-0. Na dosažených výsledcích vidíme, že ať se zpoždění projeví v kterémkoliv převodníku, ve výsledném poměru si téměř odpovídají. Z toho se dá usuzovat, že vliv zpoždění každého z převodníků je víceméně stejný.

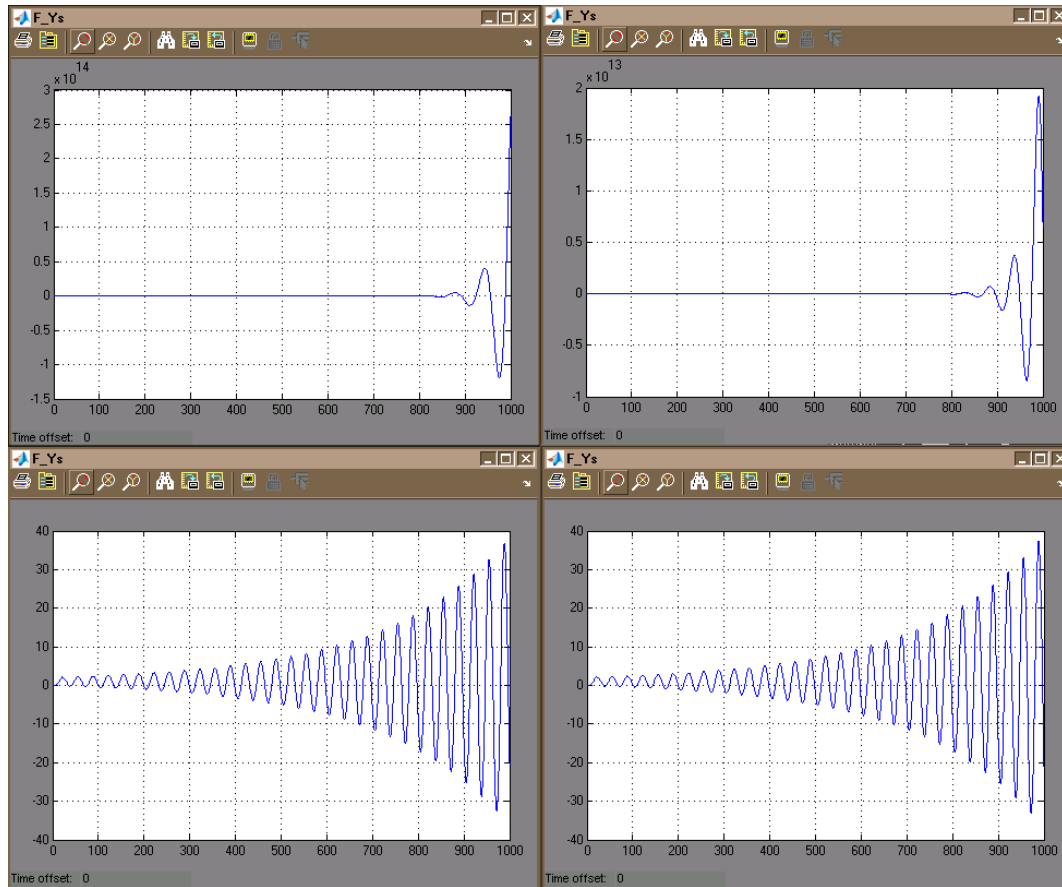
Následující výsledky ukazují další fakt a sice mez zpoždění, od které již soustava neplnila svou regulační funkci.



Obrázek 8.4: První nalezené meze zpoždění

Tato mez byla nalezena na hodnotách 0-7 a 7-0.

Zbytek experimentů odhalil zásadní věc. Uvádím zde některé další nestabilní výsledky pro ilustraci:



Obrázek 8.5: Příliš velké zpoždění

Tyto páry převodníků byly namodelovány, aby představovaly zpoždění 15-3, 7-7, 2-5 a 6-1. Ze získaných dat plyne nejen to, že zpoždění obou převodníků se navzájem nijak neeliminuje, ale zároveň se ukázal být platný fakt: soustava byla vždy nestabilní, pokud byl součet zpoždění obou převodníků větší nebo roven sedmi, a to bez ohledu vzájemné rozložení.

Pro praktický návrh z toho plyne, že stabilita regulátoru může být zajištěna jedině při využití převodníků rychlých přinejmenším tak, jako trvají tři cykly efektivního výpočtu v elektronice regulátoru. Tato hodnota je samozřejmě relativní a může se lišit dle jednotlivých aplikací, nicméně nám ukázala zajímavé kritérium použitelnosti regulátoru navrženého touto metodou.

Kapitola 9

Závěr

V této diplomové práci byl přehledně znázorněn základ teorie řízení jakožto podklad pro navrženou metodu konstrukce elektronického regulátoru. Takto vytvořený regulátor bude dle platných principů obecně schopný řídit jakýkoli matematicky definovatelný, reálný dynamický systém. Popsaný postup nevytváří regulátor obecný, nýbrž slouží k návrhu regulátoru elektronického, jež bude pracovat na principu numerického řešení diferenciálních rovnic. Vzhledem k současným cenám a dostupnosti potřebné elektroniky, jakož i dostupnosti simulačních nástrojů potřebných pro jeho vývoj, je předvedená metoda použitelná v praxi a kromě zmíněné nutnosti matematické definice je omezen pouze potřebou rychlých převodníků mezi elektronickou a reálnou reprezentací dynamického obvodu.

Literatura

- [1] WWW stránky. Závazné pokyny pro vypracování diplomové práce, [online]. URL <https://www.fit.vutbr.cz/info/szz/pokyny.dp.html>
- [2] DOYLE, J., FRANCIS, B., TANNENBAUM, A.: *Feedback Control Theory*, 1990, [online]. URL <http://www.control.utoronto.ca/people/profs/francis/dft.pdf>
- [3] JAREŠOVÁ, M., VYBÍRAL, B.: *Diferenciální rovnice*, [online]. URL <http://fo.cuni.cz/texty/matematika/difro.pdf>
- [4] Prof. Ing. BOBEK, M., CSc.; Doc. Ing. HAŠKA, J., CSc.; Doc. Ing. SERBA, I., CSc.; Doc. Ing. LUKEŠ, M., CSc.: *Analogové počítače*, SNTL Praha 1982
- [5] *Numerical recipes in C: The Art Of Scientific Computing*, ISBN Cambridge University Press 1988-1992, [online]. URL www.ee.nthu.edu.tw/bschen/files/c16-1.pdf
- [6] REKTORYS, K.: *Přehled užití matematiky*, SNTL Praha, 1968
- [7] RALSTON, A.: *Základy numerické matematiky*, Academia Praha, 1973
- [8] HOWE, R.: *Dynamics of Real-Time Simulation*, Applied Dynamics International, Michigan, 1998
- [9] KRAUS, M., KUNOVSKÝ, J., PINDRYC, M., ŠÁTEK, V.: *Taylor Series in Control Theory*, In: Proceedings UKSim 10th International Conference EUROSIM/UKSim2008, Cambridge, GB, IEEE CS, 2008.

PŘÍLOHY

- A - Zdrojový kód TKSL pro Padého rozvoj 4. řádu
- B - Zdrojový kód TKSL pro popis výsledného systému
- C - Úplné schéma systému v prostředí Simulink

A - Zdrojový kód TKSL pro Padého rozvoj 4. řádu

```
var y,z,v,p1v,p2v,p3v,p4v,p5v,p6v;

const eps =1e-20,dt=0.05, tmax=10,
b0=1,a0=1,b1=-1/2,a1=1/2,b2=5/44,a2=5/44,b3=-1/132,a3=1/132,b4=1/1584,a4=1/1584,
b5=-1/5080,a5=1/5080,b6=1/665280,a6=1/665280;

system

p4v=1584*z-1584*a3*p3v-
      1584*a2*p2v-1584*a1*p1v-1584*a0*v;

p3v'=p4v &0;
p2v'=p3v &0;
p1v'=p2v &0;
v'=p1v &0;
y=b0*v+b1*p1v+b2*p2v+b3*p3v+b4*p4v;
z=1;

sysend.
```

B - Zdrojový kód TKSL pro popis výsledného systému

```
var u,v,y,q,z,e,r,I1,I2,I3,e1,Is,z1,li,ys,us,vs,qs,ys1;
const a0=0.2,a0s=0.2,kp=0.8,kd1=3,kd2=3,kd3=0,kI1=0,kI2=0,kI3=0,
      kps=0.5,kis=0.2,dt=0.1, eps=1e-20, tmax=100,p=0.1;
system
y'=u &0;
u'=v &0;
v'=q &0;
q'=e-2*q-v-u-a0*y &0;

ys'=us &0;
us'=vs &0;
vs'=qs &0;
qs'=e-2*qs-vs-us-a0s*ys &0;
ys1=ys+p;
e=z-r;
r=kp*y+kd1*u+kd2*v+kd3*q+kI1*I1+kI2*I2+kI3*I3;
I1'=y &0;
I2'=I1 &0;
I3'=I2 &0;

e1=z1-ys1;
z=kps*e1+Is;
Is'=kis*e1 &0;
z1=12;
li=1.8;
sysend.
```

C - Úplné schéma systému v prostředí Simulink

