

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ TECHNOLOGIE

Katedra informatizácie a riadenia procesov



DIPLOMOVÁ PRÁCA

Tvorba e-learningovej učebnice: Návrhy regulátorov

Bratislava 2005

Bc. Katarína Rybárová



SLOVENSKÁ TECHNICKÁ UNIVERZITA

Fakulta chemickej a potravinárskej technológie

Radlinského 9, 812 37 Bratislava

Katedra: **informatizácie a riadenia procesov**

Číslo: 8/KIRP/2005

Vec: **Zadanie diplomovej práce**

Meno a priezvisko študentky: **Bc. Katarína Rybárová**

Meno a priezvisko vedúceho diplomovej práce: **Ing. Ľuboš Čirka, PhD.**

Meno a priezvisko konzultanta diplomovej práce:

Názov diplomovej práce:

Tvorba e-learningovej učebnice: Návrhy regulátorov

Termín odovzdania diplomovej práce: **21. mája 2005**

Diplomová práca sa odovzdáva v 3 exemplároch vedúcemu katedry.

Bratislava **1. februára 2005**

Doc. Dr. Ing. Miroslav Fikar
vedúci katedry

Prof. Ing. Dušan Bakoš, DrSc.
dekan

Prehlásenie:

Čestne prehlasujem, že zadanú diplomovú prácu som vypracovala samostatne s použitím literatúry, uvedenej v osobitnom zozname a podľa pokynov diplomového vedúceho. Ďalej prehlasujem, že súhlasím s akýmkoľvek prípadným využitím diplomovej práce katedrou.

V Bratislave, 20. mája 2005

.....

Pod'akovanie

Chcem poďakovať vedúcemu diplomovej práce *Ing. Ľubošovi Čirkovi, PhD.* za všetky cenné rady, pripomienky a odborné vedenie, ktoré mi ochotne poskytol pri vypracovávaní zadania diplomovej práce.

Abstrakt

Táto diplomová práca sa zaoberá vytváraním e-learningovej aplikácie umožňujúcej on-line riešenie vybraných úloh z oblasti návrhu regulátorov. Jednotlivé úlohy vo forme *MWS* prezentácií boli robené tak, aby bolo možné ich použiť spolu s [1] ako podporu výuky, k názornejšiemu riešeniu jednotlivých príkladov. Aplikácie využívajú možnosti *MATLAB Web Servera*, ktorý je schopný prijímať údaje z internetových aplikácií a umožňuje ich spracovanie v *MATLABe*. V práci je vytvorený postup ako s *MATLAB Web Serverom* pracovať a ako vytvárať vlastné prezentácie.

Abstract

This graduation thesis deals with the creating of the e-learning application, which enables on-line solution of choosen problems in area of Controller Design. The each individual case in form of *MWS* presentation was processed like support teaching together with [1], for illustration of problem solving. Application use capabilities of *MATLAB Web Server*, which is able to receive data from Internet applications and enables to work with them in MATLAB. In thesis is explained the way how to work with *MATLAB Web Server* and how to create the own *MWS* presentations.

Obsah

ÚVOD	1
1 MATLAB WEB SERVER	2
1.1 Prevádzka a využívanie <i>MWS</i>	2
1.2 Fungovanie MATLAB Web Servera	3
2 APACHE HTTP SERVER	5
2.1 Inštalácia Apache HTTP Servera	5
2.2 Otestovanie funkčnosti Apache HTTP Servera	6
3 TVORBA <i>MWS</i> PREZENTÁCIE	7
3.1 Vstupný formulár	7
3.1.1 Deklarácia formulára	8
3.1.2 Parametre tagu <code><form></code>	8
3.1.3 Parametre tagu <code><input></code>	8
3.1.4 Kontrola zadávaných údajov pomocou JavaScriptu	9
3.2 Tvorba m-súboru pre MATLAB Web Server	12
3.2.1 Ako funguje táto štruktúra	13
3.2.2 Príkazy MATLAB Web Servera	14
3.2.3 Použitie Simulinku	14
3.2.4 Formát dát	15
3.2.5 Kontrola zadávaných údajov v MATLABe	16
3.3 Výstupný formulár	17
4 NÁVRHY REGULÁTOROV	19
4.1 PID regulátor	20
4.1.1 Zásady výberu regulátora	20
4.1.2 Vplyv parametrov regulátora na rýchlosť regulácie a stabilitu URO	21
4.1.3 Prenosy PID regulátora	21
4.1.4 Proporcionálny regulátor	22
4.1.4.1 Príklady na riadenie pomocou P regulátora	22
4.1.5 Integračný regulátor	23
4.1.5.1 Príklady na riadenie pomocou PI regulátora	24
4.1.6 Derivačný regulátor	25
4.1.6.1 Príklad na riadenie pomocou PID regulátora	26
4.1.7 Saturácia integrátora – Vplyv obmedzení na integračnú zložku regulátora	26

4.1.7.1	Príklad na saturáciu integrátora	27
4.1.8	Nastavovanie regulátora Strejcovou metódou	28
4.1.8.1	Príklad na Strejcovu metódu	28
4.2	Polynomický návrh regulátora	29
4.2.1	Riešenie diofantickej rovnice	29
4.2.2	Polynomický návrh PA (pole assignment) regulátorov	30
4.2.2.1	Príklady na riešenie polynomickeho návrhu PA regulátora	31
4.2.3	Polynomický PA regulátor s integračnou činnosťou	31
4.2.3.1	Príklady na riešenie polynomickeho návrhu PA regulátora s integračnou činnosťou	33
4.3	Zoznam <i>MWS</i> prezentácií	33
5	ŠTRUKTÚRA E – LEARNINGOVÝCH STRÁNOK	35
5.1	Vypĺňanie formulárov	36
6	PROBLÉMY PRI TVORBE <i>MWS</i> PREZENTÁCIÍ	38
6.1	Nesprávna konfigurácia vstupných údajov	38
6.2	Inicializácia polynomickeho toolboxu	38
6.3	Nekompatibilita internetových prehliadačov	39
	ZÁVER	40
	LITERATÚRA	42
	PRÍLOHA	43

Úvod

V súčasnej dynamickej dobe, keď svetom hýbe najmä sila informácií, nie je pochyb o nutnosti neustáleho sa vzdelávania. Nové informačné a komunikačné technológie prinášajú zmeny aj do vzdelávacieho systému. Umožňujú vnímanie poznatkov viacerými zmyslami a tým, v porovnaní s tradičnými formami vzdelávania, umožňujú dosiahnuť vyšší efekt vo vzdelávaní. Jednou z foriem je využitie e-learningu.

E-learning je proces elektronického vzdelávania, ktorý využíva počítačovú sieť na realizáciu, interakciu a podporu výučby. Ideálne podmienky pre jeho vznik vytvoril “boom” na poli informačných technológií a Internetu. Výučba tak môže prebiehať z akéhokoľvek miesta na svete, kde je dostupný internet, pričom štúdium je časovo nezávislé, takže každý môže absolvovať kurz podľa vlastných potrieb a schopností. Študent si sám vyberie oblasť tzv. e-learningový kurz, ktorý ho zaujíma.

Cieľom tejto diplomovej práce je vytvorenie e-learningovej aplikácie, ktorá umožní riešenie vybraných problémov týkajúcich sa návrhu regulátorov využitím Internetu. Aplikácie využívajú možnosti poskytované programom MATLAB, konkrétne toolbox *MATLAB Web Server* poskytujúci možnosti komunikácie MATLABu s užívateľom, prostredníctvom Internetu.

Prvá kapitola je venovaná *MATLAB Web Serveru*, podmienkam pre inštaláciu a jeho fungovanie.

V druhej kapitole je jednoducho popísaná inštalácia *Apache HTTP Servera*, ktorý slúži na vytváranie a správu serverov a bol použitý na prevádzku *MATLAB Web Servera*.

Tretia kapitola sa zaoberá vytváraním jednotlivých častí *MWS* prezentácie, vstupného a výstupného formulára v HTML kóde a m-súboru v MATLABe.

Štvrtá kapitola obsahuje teoretické podklady k návrhom regulátorov a vychádza z [1]. V jej závere sú uvedené príklady, ktoré boli riešené on-line.

Štruktúru HTML stránok vytvorených vo forme e-learningovej aplikácie, ako aj spôsob a možnosti vyplňania vstupných formulárov *MWS* prezentácií, možno nájsť v piatej kapitole.

V šiestej kapitole sú uvedené chyby, ktoré pri vytváraní *MWS* prezentácií boli najčastejšie a tiež možnosti ich odstránenia.

1 MATLAB WEB SERVER

MATLAB Web Server (MWS) toolbox je už štandardne voliteľnou súčasťou inštalácie programu MATLAB, distribuovaného firmou MathWorks [2].

MATLAB Web Server je nástroj, ktorý umožňuje vytváranie aplikácií v MATLABe, nazývaných prezentácie. *MWS* prezentácia je umiestnená na serveri a je možné ovládať ju z Internetu prostredníctvom HTML formulára. Vstupné dáta pre *MWS* prezentáciu sú získavané od užívateľa internetového prehliadača, tým že ich zadá v HTML formuláre a odošle. Dáta sú vstupnými parametrami pre aplikačný program (**m-súbor**), ktorý ich spracuje a vráti užívateľovi výsledky výpočtov vo forme výstupnej HTML stránky.

Výsledkom výpočtu môžu byť:

- číselné údaje vo forme jednoduchých čísel, vektorov, matíc, tabuliek atď.
- obrázky a grafy generované MATLABom a uložené v niektorom z grafických formátov (**.jpeg**, **.bmp**, **.png**, ...)

1.1 Prevádzka a využívanie *MWS*

Prevádzkovateľ takéhoto servera, musí splniť nasledovné podmienky:

Na prevádzku je potrebné pripojenie na sieť (TCP/IP), operačný systém napr. MS Windows NT (98/2000/XP), Linux resp. Unix. Nainštalovaný MATLAB, ktorého súčasťou je *MWS* toolbox. Keďže *MWS* vyžaduje pre svoju činnosť klasický Web (HTTP) Server podporujúci CGI (*Common Gateway Interface*) skripty, je potrebné mať nainštalovaný aj program, ktorý toto umožňuje. Vhodný je napr. *Apache HTTP Server*, ktorý slúži na vytváranie a správu serverov. Je to voľne šíriteľný program a možno ho stiahnuť na stránke <http://www.apache.org/> [3].

Z pozície jednoduchého užívateľa, postačí pripojenie k internetu (TCP/IP) a niektorý z internetových prehliadačov (MS Internet Explorer, Mozilla, Mozilla Firefox, Opera, atď.).

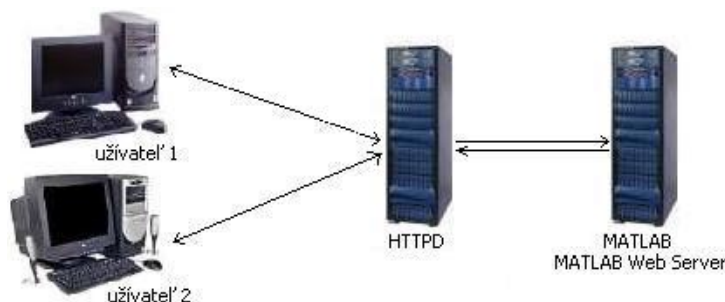
MWS je možné prevádzkovať dvoma spôsobmi:

- HTTP server a *MWS* môžu fungovať na jednom počítači (obr.1)



Obr. 1 HTTP server a *MATLAB Web Server* na jednom počítači

- HTTP server a *MWS* môžu fungovať každý na inom počítači (obr.2)



Obr. 2 HTTP server a *MATLAB Web Server*, každý bežiaci na inom počítači v sieti

1.2 Fungovanie MATLAB Web Servera

Mechanizmus predávania dát medzi internetovou aplikáciou a MATLABom je založený na CGI (*Common Gateway Interface*) rozhraní. Odoslaním sú dáta, ktoré užívateľ zadá do vstupného HTML formulára, predané cez HTTP Server CGI skriptu **matweb.exe**. Klient **matweb.exe** prevezme dáta z HTML formulára a predá ich TCP/IP serveru **matlabserver.exe**, čo je viacvláknový TCP/IP server, ktorý zaistí vykonanie **m-súboru** príslušnej *MWS* prezentácie. S CGI skriptom je spojený jeho konfiguračný súbor **matweb.conf**. Tento je umiestnený v rovnakom adresári ako **matweb.exe** a zapisujú sa v ňom jednotlivé spustiteľné *MWS* prezentácie.

Príklad zápisu *MWS* prezentácií v konfiguračnom súbore **matweb.conf**:

```
[pr1]
mlserver=localhost
mldir=D:\www\priklady\pr_3.4.1\

[pr2]
mlserver=localhost
mldir=D:\www\priklady\pr_3.4.2\
```

Položka *mlserver* určuje URL adresu serveru, na ktorom je spustený *MATLAB Web Server* a položka *mldir* je adresárová cesta danej prezentácie. V hranatých zátvorkách je uvedený názov **m-súboru**, prislúchajúci danej *MWS* prezentácii.

2 APACHE HTTP SERVER

Apache HTTP Server je jeden z najrozšírenejších programov, slúžiaci na vytváranie a správu serverov. Je voľne šíriteľný a možno ho stiahnuť na stránke <http://www.apache.org/> [3]. Možno ho prevádzkovať napr. pod operačným systémom MS Windows alebo Unix/Linux.

2.1 Inštalácia Apache HTTP Servera

1. Spustíme inštalačný program a nainštalujeme *Apache HTTP Server*. Na inštaláciu programu nie je potrebné mať žiadne špeciálne znalosti ohľadom správy alebo vytvárania serverov. Inštalačný program zrozumiteľne ponúka všetky dostupné možnosti inštalácie.
2. Pred samotným spustením *Apache HTTP Servera* musíme urobiť určité zmeny v konfiguračnom súbore **httpd.conf**, ktorý sa nachádza v nainštalovanom adresári

```
C:\Program Files\Apache Group\Apache\conf\httpd.conf
```

- **ServerAdmin** - e-mailová adresa správcu servera
- **ServerName** - IP počítača resp. localhost, podľa toho na akom počítači je server umiestnený
- **DocumentRoot** - cesta k adresáru, v ktorom sú HTML stránky *MWS* prezentácií
- **Directory** - cesta k adresáru, v ktorom sú HTML stránky *MWS* prezentácií

Pre skúsenejších správcov serverov je v konfiguračnom súbore viac parametrov, ktoré možno nastavovať. Vo všeobecnosti však stačí nastavenie vyššie uvedených. Po uložení konfiguračného súboru **httpd.conf**, treba program reštartovať.

V internetovom prehliadači teraz možno zadať **ServerName**, na ktorom je umiestnený *Apache HTTP Server*. Ak server pracuje správne, mal by sa v prehliadači zobrazíť obsah adresára, ktorý bol zadaný v konfiguračnom súbore **httpd.conf** ako **DocumentRoot** a **Directory**.

2.2 Otestovanie funkčnosti Apache HTTP Servera

Pre overenie funkčnosti nainštalovaného *Apache HTTP Servera* stačí dodržať nasledovný postup. Do koreňového adresára, ktorý bol určený pre HTML prezentácie, je potrebné skopírovať z MATLABu adresár **wsdemos**, ktorý sa nachádza

```
C:\Matlab6p5\toolbox\webserver\
```

Ďalej z adresára

```
C:\Matlab6p5\webserver\bin\win32\
```

skopírujeme súbory **matweb.exe** a **matweb.conf** do adresára

```
C:\Program Files\Apache Group\Apache\cgi-bin\
```

V konfiguračnom súbore **matweb.conf** treba zmeniť položku *mlserver* na IP adresu serveru (resp. localhost), na ktorom je umiestnená daná prezentácia a položka *mldir* je adresárová cesta danej prezentácie. Nakoniec len v prehliadači treba zadať **ServerName**, vybrať adresár **wsdemos** a zvoliť niektorú z ponúkaných demo *MWS* prezentácií.

3 TVORBA MWS PREZENTÁCIE

Vytvorenie prezentácie pre *MATLAB Web Server* možno rozdeliť do niekoľkých krokov.

Prvým krokom by mala byť analýza úlohy, ktorá je spracúvaná ako *MWS* prezentácia. Je potrebné definovať vstupné dáta, t.j. aké dáta bude užívateľ zadávať a aké dáta je potrebné mať na výstupe, teda čo presne má byť výsledkom prezentácie (grafy, obrázky, hodnoty veličín, čísla, matice atď.).

Druhým krokom je vytváranie samotnej *MWS* prezentácie, ktorá sa skladá z troch častí. Prvú časť tvorí HTML stránka, ktorá sa nazýva vstupný formulár a slúži na zadávanie dát užívateľom. Druhou časťou je vytvorenie funkcie pre *MATLAB* (**m-súboru**), ktorá vykonáva súbor príkazov (program) v nej zapísaných, pričom celý výpočet sa realizuje prostredníctvom *MATLAB Web Serveru*. Vstupné dáta pre tento **m-súbor** sú prevzaté zo vstupného formulára metódou **post**. Po spracovaní dát je vygenerovaná HTML stránka, ktorá sa nazýva výstupný formulár a tvorí tretiu časť *MWS* prezentácie. Výstupný formulár obsahuje výsledky, ktoré sme si zvolili ako výstupné dáta.

Takže kvôli prehľadnosti, si zhrňme jednotlivé kroky:

- Užívateľ (klient) sa pripojí na HTTP server pomocou internetového prehliadača a vyberie si prezentáciu, zobrazením príslušného vstupného formulára.
- Užívateľ vyplní vstupný formulár a odošle ho.
- Dáta z formulára sa spracujú cez *MWS* v *MATLABe*.
- Užívateľ získa vo svojom internetovom prehliadači výstupný formulár s výsledkami výpočtov.

3.1 Vstupný formulár

Vstupný formulár je klasický HTML dokument a tvorí vstup pre každú prezentáciu. Vo vstupnom formulári sú prvky pomocou, ktorých sa získava z tohto formulára súbor dát použitých následne **m-súborom** pre výpočet.

3.1.1 Deklarácia formulára

Každý formulár musí obsahovať nasledovnú časť HTML kódu

```
<html>
    <form action="/cgi-bin/matweb.exe" method="post">
        <input type="hidden" name="mlmfile" value="názov_m-súboru" />

        <!--časť HTML kódu, obsahujúca prvky pre zadanie vstupných dát-->

        <input type="submit" value="Odoslať údaje" />
    </html>
```

3.1.2 Parametre tagu <form>

<form> ... </form> je párový tag vymedzujúci formulár, ktorým sa získavajú dáta od užívateľa. Jednotlivé formuláre nemožno vnoriť!

Tabuľka 1 Najpoužívanéjšie parametre tagu <form> v MWS prezentácii

ACTION	URL	určuje adresu alebo cestu k programu (skriptu), ktorý sa spustí po odoslaní údajov formulára. Tomuto programu odovzdá prehliadač všetky údaje z formulára.
METHOD	GET	údaje z formulára (premenné a ich hodnoty) pripoja k URL adrese programu, ktorá je zadaná v parametri ACTION , v tvare. O spracovanie týchto údajov sa musí postarať daný program. Používa sa ako všeobecná metóda v prípade, že sa nevyžaduje bezpečnosť. V MWS môže dôjsť k problémom, ak sú odosielané dáta dlhšie než je maximálna povolená dĺžka URL .
	POST	údaje z formulára sprístupní program uvedený v hodnote parametra ACTION načítaním zo štandardného vstupu. Údaje majú rovnaký tvar ako pri metóde GET . Používa sa, ak prenášame údaje, ktoré nechceme zobrazovať ako časť URL . Okrem toho umožňuje posielat' s formulármi aj súbory. Preto budeme používať túto metódu.

3.1.3 Parametre tagu <input>

Vo formuláre musí byť deklarovaný skrytý vstup určujúci, ktorý **m-súbor** bude použitý pre spracovanie dát.

```
<input type="hidden" name="mlmfile" value=" názov_m-súboru " />
```


Nepárový tag `<input>` vloží prvok (položku) formulára daného typu. Každá položka je označená svojim menom. Po odoslaní formulára sa zašlú hodnoty všetkých položiek programu, ktorý je uvedený v parametri **ACTION** príkazu `<form>`.

Tabuľka 2 Najpoužívanéjšie parametre tagu `<input>` v MWS prezentácii

TYPE	<i>hidden</i>	umožní vložiť do formulára skrytú informáciu, ktorá sa bude prenášať pri každom odoslaní formulára. Položka sa nezobrazuje v prehliadači. Hodnotou skrytého poľa je hodnota uvedená v parametri VALUE .
	<i>text</i>	vloží jednoriadkové textové vstupné pole formulára.
	<i>checkbox</i>	vytvorí zaškrŕavacie políčko formulára. Počiatočné "zaškrtnutie" možno dosiahnuť použitím parametra CHECKED .
	<i>radio</i>	vytvorí prepínač, t.j. zo skupiny vstupných polí typu "radio", ktoré majú rovnaké meno (parameter NAME) a z ktorých môže byť aktívny práve jeden. Počiatočné "zaškrtnutie" možno dosiahnuť použitím parametra CHECKED v jednom z prepínačov.
	<i>button</i>	vloží tlačidlo, ktoré možno využiť iba pomocou skriptov na stránke.
	<i>submit</i>	vloží tlačidlo na odoslanie formulára.
	<i>reset</i>	vygeneruje formulár s pôvodnými hodnotami.
NAME	<i>text</i>	určuje meno formulárovej položky pre účely neskoršieho spracovania hodnoty. Funguje ako meno premennej.
VALUE	<i>text</i>	určuje počiatočnú hodnotu formulárovej položky (a zodpovedajúcej premennej). Jej zmysel závisí od typu položky.
SIZE	<i>veľkosť</i>	určuje počiatočnú šírku položky formulára.

3.1.4 Kontrola zadávaných údajov pomocou JavaScriptu

JavaScript je programovací jazyk, ktorého kód sa vkladá do internetových stránok. To znamená, že príkazy tohto jazyka sa vykonávajú na strane klienta, nie servera. Výhodou je preto menšie zaťaženie servera, na ktorom sú stránky umiestnené.

JavaScript sa väčšinou používa na úpravu stránok. Programy vytvorené v JavaScripte môžu na stránke robiť rôzne veci napr. zobrazit' dátum, zmenit' polohu okna prehliadača, otvoriť nové okno, urobiť kontrolu zadaných údajov, atď. Pri tvorbe MWS prezentácií bol JavaScript použitý práve na kontrolu zadávaných údajov.

Kód skriptu možno vložiť ako časť stránky, alebo ako samostatný súbor s príponou **.js**. Skripty, ktoré majú byť volané len raz, t.j. pri načítaní stránky, je vhodné vložiť do

hlavičky HTML súboru medzi tagy `<HEAD>...</HEAD>`. Tam je vhodné tiež vkladať funkcie, ktoré budú volané z viac miest skriptu.

Pre vkladanie JavaScriptu do stránok sa používa tag

```
<script type="text/javascript" language="javascript">... </script>
```

Funkcia je blok príkazov a používa sa, ak sa niektoré bloky príkazov budú opakovať. Pretože funkcia je často volaná z viacerých skriptov, jej umiestnenie je najvhodnejšie v hlavičke stránky medzi tagy `<HEAD>...</HEAD>`, čím je zabezpečené to, že funkciu bude možné zavolať, z ktoréhokoľvek skriptu na stránke.

Pre kontrolu údajov v *MWS* prezentácií bolo použitých niekoľko funkcií vytvorených v JavaScripte.

- Funkcia `isblank`

```
function isblank(s)
{
    for(var i = 0; i < s.length; i++) {
        var c = s.charAt(i);
        if ((c != ' ') && (c != '\n') && (c != '\t')) return false;
    }
    return true;
}
```

kontroluje v určených položkách, či daná položka bola vyplnená, resp. či neobsahuje tzv. prázdne znaky.

- Funkcia `IsNumeric`

```
function IsNumeric(PossibleNumber)
{
    var PNum = new String(PossibleNumber);
    var regex = /^[+\-]?[0-9]+(\.[0-9]+)?$/;
    var tmp = regex.test(PNum);
    return tmp;
}
```

bola vytvorená pomocou pravidiel, ktoré platia pre regulárne výrazy a overuje, či údaj zadaný do formulára je číslo. Ak je zadávané desatinné číslo je akceptované len ak je desatinným znakom bodka, čo je výhodné pre MATLAB.

- Funkcia `IsVector`

```
function IsVector(PossibleNumber)
{
    var PNum = new String(PossibleNumber);
    var regex = /^\[([\\+\\-]?[0-9]+(\\. [0-9]+)?\\s*?)+\\]$ /;
    var tmp = regex.test(PNum);
    return tmp;
}
```

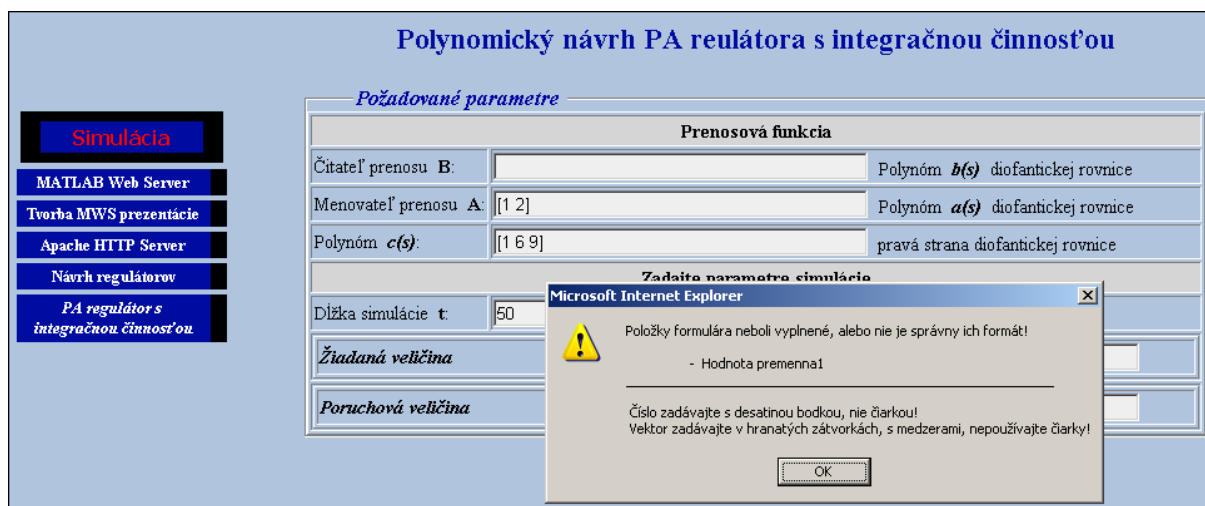
je podobná ako funkcia `IsNumeric`. Pri tejto funkcii však musí byť položka formulára vyplnená ako vektor, t.j. musí byť v hranatých zátvorkách a jednotlivé čísla vektora je treba oddelovať medzerou, nie čiarkou ako je to v MATLABe možné.

Samotná kontrola jednotlivých určených položiek by potom mohla vyzerat' nasledovne

```
<!--
function kontrola(formular)
{
    missinginfo = "";
    if((formular.premenna1.value=="") || isblank(formular.premenna1.value)
    || !IsNumeric(formular.premenna1.value)) {
        missinginfo += "\n - Hodnota premenna1";
    }
    if((formular.premenna2.value=="") || isblank(formular.premenna2.value)
    || !IsVector(formular.premenna2.value)) {
        missinginfo += "\n - Hodnota premenna2";
    }
    if (missinginfo != "") {
        missinginfo = "Položky formulára neboli vyplnené, alebo nie je správny
        ich formát!\n" +
        missinginfo+
        "\n_____ " + "\n\n
        Číslo zadávajúte s desatinou bodkou, nie čiarkou!\n Vektor zadávajúte v
        hranatých zátvorkách, s medzerami, nepoužívajte čiarky!";
        alert(missinginfo);
        return false;
    }
    else return true;
}
-->
```

Vo formuláre potom treba určiť, akým spôsobom a kedy sa má spustiť kontrola údajov JavaScriptom. Možné je na to použiť niekoľko tzv. udalostí. Vo formuláre však slúži na to udalosť `onsubmit` zapísaná v tagu `<form>`

```
<form onsubmit="return kontrola(this);" action="/cgi-bin/matweb.exe"
method="post">
```



Obr. 3 Hlásenie o chybe vrátené JavaScriptom pri nevyplnení položky vo formuláre

3.2 Tvorba m-súboru pre MATLAB Web Server

Súbor nazývaný **m-file** resp. **m-súbor** spracúva dáta na strane servera a je to vlastne postupnosť príkazov pre MATLAB. **M-súbor** vytvorený pre *MWS* prezentáciu musí byť schopný prevziať údaje zo vstupného formulára, tie následne spracovať a odovzdať výsledné hodnoty výstupnému formuláru. Na vstupnej stránke sa pomocou skrytého vstupu s názvom *mlfile* a parametrom **VALUE**, ktorý obsahuje názov volaného súboru, odkazujeme na **m-súbor**, ktorý zabezpečí spracovanie vstupných dát. Názov **m-súboru** musí byť rovnaký ako je názov *MWS* prezentácie a spolu s *URL* adresou, ktorá udáva kde sa daný **m-súbor** nachádza, je zapísaný v skripte konfiguračného súboru **matweb.conf**.

Všeobecný skript **m-súboru** so základnými časťami kódu vyzerá nasledovne:

```
function HTMLout = názov_m-súboru(in,out)

imdir = 'D:/www/images'; % ==>> umiestnenie generovaných obrázkov
cd (in.mldir);           % ==>> nastavenie cesty pracovného adresára
mlid = in.mlid;          % ==>> získanie jedinečného reťazca
wscleanup('*.jpeg',0.2,imdir); % ==>> vymazanie grafov
% ==>> Načítanie dát z HTML formulára
premenna1 = in.premenna1;
premenna2 = in.premenna2;
% ==>> Prevod údajov z formulára na dáta spracovateľné Matlabom
premenna1 = str2num(premenna1);
premenna2 = str2num(premenna2);
% ==>>

    %% Nasledujú všeobecné príkazy pre výpočet, v tvare v akom sa
        udávajú bežne v Matlabe
% ==>>
% ==>> Výstupné dáta pre výstupný HTML formulár
out.premenna1 = premenna1;
out.premenna2 = premenna2;
    % Načítanie výstupnej šablóny
templatefile = which('názov_výstupného formulára.html');
% ==>> Vygenerovanie výstupného formulára
HTMLout = htmlrep(out, templatefile);
```

3.2.1 Ako funguje štruktúra m-súboru

Dáta sú po odoslaní užívateľom prevzaté CGI skriptom **matweb.exe**. Tento skript vytvorí štruktúrovanú premennú, ktorej jednotlivé prvky sú tvorené hodnotami položiek **NAME** zo vstupného formulára. V jednotlivých prvkoch tejto štruktúry sú uložené hodnoty položiek **VALUE** pre jednotlivé prvky. Okrem toho sa do štruktúrovanej premennej pridávajú položky *mldir* a *mlid*. Premenná *mldir* určuje názov pracovného adresára *MWS* prezentácie, ktorý sa získa zo súboru **matweb.conf** a premenná *mlid* obsahuje jedinečný reťazec, ktorý sa používa pri tvorbe obrázkov a grafov, za pomoci ktorého je možné rozlíšiť, ktorému užívateľovi vygenerované údaje patria, resp. ktoré dáta boli generované ako posledné. Takto vytvorená štruktúrovaná premenná je predaná príslušnému **m-súboru** ako vstupný parameter nazvaný v tomto prípade *in*. Po spracovaní

sa výsledky výpočtov, názvy vygenerovaných obrázkov atď. uložia do výstupnej štruktúrovanej premennej nazvanej napr. *out*. Príkaz *htmlrep* vygeneruje výstupný formulár obsahujúci potrebné výstupné údaje.

3.2.2 Príkazy MATLAB Web Servera

Základnými príkazmi *MATLAB Web Server* toolboxu privytváraní *MWS* preztnácií sú:

- `htmlrep` – Nahradí názvy premenných v šablóne výstupného formulára ich hodnotami získanými z *MATLAB Web Servera*.
- `wscleanup` – Vymaže súbory v určenom adresár, ktoré sú staršie než je zadaný čas.
- `wsprintjpeg` – Uloží graf do súbore go formáte **.jpeg**.
- `wssetfield` – Nastaví alebo modifikuje jednotlivé položky štruktúrovanej premennej.

3.2.3 Použitie Simulinku

Pre zložitejšie výpočty a vytváranie grafov je nutné využitie Simulinku. Toto je možné len použitím príkazov MATLABu zadaných v **m-súbore**.

Hlavné príkazy sú:

- `open_system('názov_modelu');`
=> otvorenie simulinkovej schémy
- `set_param('názov_modelu/názov_bloku', 'parameter1', 'hodnota1', ...);`
=> nastavenie parametrov jednotlivých blokov schémy, keďže nieje možné použiť načítavanie paramatrov a premenných zadaných priamo vo Workspace
- `sim('názov_modelu', 'simulačné_parametre');`
=> spustenie simulácie, podľa zadaných simulačných parametrov

3.2.4 Formát dát

Pri práci s jednotlivými údajmi treba dávať pozor na to, aký formát dát kde použijeme. Nemožno spracovávať textový reťazec rovnako ako číslo a nemožno číslo odosielať ako textový reťazec. Preto je potrebné si uvedomiť, že zo vstupnej stránky sa všetky údaje načítavajú ako textové reťazce, pritom nezáleží na tom, že do formulára sú zadávané užívateľom čísla. Keďže na výpočet v MATLABe sú potrebné číselné údaje, musíme jednotlivé textové reťazce na ne previesť.

Slúžia na to nasledovné funkcie MATLABu:

```
%%% Prevod textového reťazca z formulára na číslo
premennal = str2num(premennal);
% resp.
premennal = eval(premennal);
```

Keď sa dáta odosielajú pre výstupný formulár je treba, aby boli znovu vo forme textového reťazca. Použiť možno tieto funkcie

```
%%% Prevod čísla na textový reťazec
premennal = num2str(premennal);
% resp.
premennal = feval(premennal);
```

Rovnako pre Simulink sa jednotlivé parametre blokov udávajú ako textové reťazce a je dôležité použiť správny formát týchto dát.

```
%%% Prevod a vloženie čísla ako textového reťazca
set_param('model/blok', 'parameter', num2str(premennal));

%%% Vloženie vektora ako textového reťazca, v zátvorkách
set_param('model/blok', 'parameter', sprintf('[%s]', num2str(vektor)));
% príkaz sprintf vloží na miesto premennej %s textový reťazec
num2str(vektor)

% resp.
set_param('model/blok', 'parameter', strcat('[', num2str(vektor), ']'));
% príkaz strcat spája niekoľko textových reťazcov do jedného
```

3.2.5 Kontrola zadávaných údajov v MATLABe

Podobne ako JavaScript je na kontrolu dát možno použiť aj priamo MATLAB. Na kontrolu zadaných parametrov, t.j. či boli vyplnené potrebné položky formulára môžeme do príslušného **m-súboru** na začiatok, pred načítanie dát z formulára zaradiť nasledovný kód

```
function HTMLout = názov_m-súboru(in,out)
imdir = 'D:/www/images';
cd (in.mldir);
mlid = in.mlid;
wscleanup('*.jpeg',0.2,imdir);
errorcode=0;
%%% Kontrola zadaných parametrov
if ( isempty(in.premenna1) | isempty(in.premenna2)),
    out.errormsg='Vyplňte vo formuláre žiadané položky!';
    errorcode=1;
end
```

a na koniec súboru treba pridať informácie o tom, ktorý výstupný formulár má byť použitý v prípade chyby vo formuláre.

```
%%% Predanie údajov o chybe programu matweb.exe, ktorý odovzdá
výstupnú šablónu 'error_pr1.html' www serveru
templatefile = which('error_pr1.html');
if (nargin == 1)
    HTMLout = htmlrep(out, templatefile);
end
```

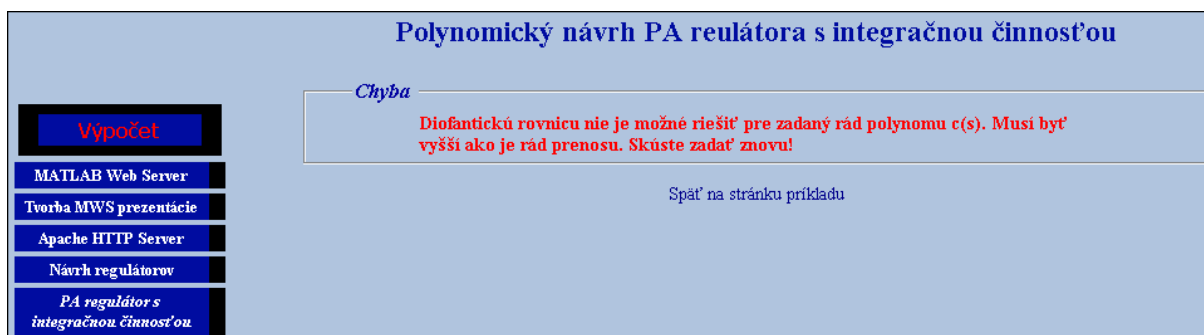
To znamená, že ak nie je vyplnená niektorá z položiek formulára, táto funkcia zachytí chybu a na obrazovke monitora sa zobrazí užívateľovi výstupný formulár, ktorý obsahuje hlásenie o chybe (obr.4).

Okrem funkcií MATLABu môžeme na výpis chyby vytvoriť aj vlastné algoritmy, podľa toho aká úloha je riešená. Väčšinu týchto algoritmov možno vytvoriť na základe jednoduchých vetvení a cyklov (*if, for* atď.). Napríklad kontrola dĺžky polynómov, ktorá bola vytvorená pre MWS prezentáciu, ktorá rieši návrh polynomickeho PA regulátora s integračnou činnosťou.


```

%%% Kontrola dĺžky polynómov
m = length(in.men);
n = length(in.cs);
if ( n <= m),
out.errormsg='Diofantickú rovnicu nie je možné riešiť pre zadaný rád
    polynómu c(s). Musí byť vyšší ako je rád prenosu. Skúste zadať
    znovu!';
    errorcode=1;
end

```



Obr. 4 Výstupná šablóna o chybe

3.3 Výstupný formulár

Súčasťou každej *MWS* prezentácie je šablóna výstupného formulára. Je to rovnako, ako vstupný formulár klasický HTML súbor, ktorý sa zobrazí užívateľovi v internetovom prehliadači. Tento formulár obsahuje výsledky výpočtov z MATLABu, t.j. čísla, vektory, matice, grafy atď. Výstupný formulár je generovaný po ukončení výpočtov v MATLABe pomocou funkcie *htmlrep*, ako výsledok *MWS* prezentácie. Výstupnými parametrami *htmlrep* sú štruktúrovaná premenná *out* a šablóna výstupného formulára.

Funkcia *htmlrep* pracuje nasledovne:

Nech napr. v **m-súbore** pre danú výstupnú šablónu sú výstupné údaje

```

out.premenna1 = "15";
out.premenna2 = "10";

```

potom, ak v príslušnej šablóne *MWS* prezentácie chceme zobrazit' dané hodnoty, časť jej kódu by mala vyzerat' približne takto:

```
<html>
  <body>
    Hodnota prvej premennej je: $premenna1$ <br />
    Hodnota druhej premennej je: $premenna2$
  </body>
</html>
```

Vykonaním príkazu *htmlrep*, reťazce \$premenna1\$ a \$premenna2\$ budú nahradené príslušnými hodnotami premenných získaných z MATLABu. Zo zápisu HTML kódu je vidieť, že pre zobrazenie premenných vo výstupnej šablóne je potrebné, aby bol názov každej výstupnej premennej vložený medzi znaky \$...\$.

4 NÁVRHY REGULÁTOROV

Pri spracovávaní teoretických podkladov k jednotlivým *MWS* prezentáciám bola použitá literatúra [1], pričom jednotlivé príklady *MWS* prezentácií sú on-line riešením úloh uvedených v kapitolách 3.4 PID regulátor a 4.6 Analýza stavového spätnoväzbového riadenia s pozorovaním stavu a polynomicke umiestnenie pólov v podkapitolách 4.6.3, 4.6.4 a 4.6.5.

Regulátor je aktívny člen regulačného obvodu, ktorého úlohou je generovať akčnú veličinu spracovaním regulačnej odchýlky. Pri návrhu regulátorov sa väčšinou vykonáva určitá postupnosť krokov, ktorou je možné dosiahnuť uspokojivé správanie sa riadeného systému.

Všeobecný postup pre všetky možné prípady riešenia regulátorov je nasledovný

1. Skúmanie vlastností riadeného systému a získanie počiatočnej informácie o účele riadenia.
2. Modelovanie procesu.
3. Určenie vlastností modelu procesu.
4. Výber riadiacich a riadených veličín.
5. Výber štruktúry regulačného obvodu.
6. Výber typu regulátora.
7. Určenie kritérií kvality regulácie založených na účeloch riadenia.
8. Návrh regulátora.
9. Analýza vlastností regulátora, kontrola splnenia kritéria kvality regulácie, prípadne zmena regulátora.
10. Simulácia uzavretého regulačného obvodu, v prípade nevyhovujúceho riešenia treba postup opakovať od bodu 2.
11. Výber hardvéru a softvéru pre implementáciu regulátora.
12. Testovanie regulátora a jeho on-line nastavenie.

4.1 PID regulátor

Najčastejšie používaným typom regulátora je proporcionálno-integračno-derivačný (PID) regulátor. Jeho použiteľnosť sa zvyšuje najmä jeho výhodami ako jednoduchosť, robustnosť a zároveň, že je realizovateľný v rozličných prevedeniach. V súčasnosti je používaný hlavne v digitálnej podobe. Princíp PID regulátora je v spracovaní vstupnej veličiny, ktorou je regulačná odchýlka.

4.1.1 Zásady výberu regulátora

- Najlepší regulátor býva často ten najjednoduchší - P regulátor. Použiť ho možno, ak je povolená určitá nenulová regulačná odchýlka v ustálenom stave, alebo keď riadený systém obsahuje integrátor. Takto možno riadiť napr. tlak v plynových zásobníkoch alebo aj výšku hladiny v zásobníkoch kvapaliny.
- Ak na riadenie nie je vhodný P regulátor a systém sa dá opísať jednoduchou dynamikou použijeme PI regulátor. Výhodou je odstránenie trvalej regulačnej odchýlky, nevýhodou zase väčšie preregulovanie a menej tlmené kmity. A tiež zvyšovanie zosilnenia môže viesť k nestabilite.
- PD regulátor, keďže neobsahuje I zložku zanecháva trvalú regulačnú odchýlku, rovnakú ako P regulátor, ale vďaka D zložke sa rýchlo ustáľuje, pretože nevzniká veľké preregulovanie. Používa sa pre procesy, ktoré majú integračné správanie alebo veľké časové konštanty.
- PID regulátor je robustný, predpovedá budúce správanie sa procesu a všeobecne je jeho zosilnenie nižšie ako u zodpovedajúceho PI regulátora. Nevýhodou je citlivosť D zložky na šum merania.
- Problematické procesy z hľadiska nasadenia PID regulátora zahŕňajú systémy s dopravným oneskorením, s časovo premennými parametrami, prípadne procesy vykazujúce vlastné kmity.

4.1.2 Vplyv parametrov regulátora na rýchlosť regulácie a stabilitu URO (uzavretý regulačný obvod)

- zväčšovanie P zložky zvyšuje rýchlosť a znižuje stabilitu
- zväčšovanie I zložky znižuje rýchlosť a zvyšuje stabilitu
- zväčšovanie D zložky zvyšuje rýchlosť a zvyšuje stabilitu

4.1.3 Prenosy PID regulátora

Prenos PID regulátora, ktorý je zložený z 3 zložiek: proporčnej, integračnej a derivačnej, môže mať rôznu štruktúru. Najčastejšie využívané štruktúry PID regulátora sú nasledovné:

- štruktúra bez interakcie

$$R(s) = Z_R \left(1 + \frac{1}{T_I s} + T_D s \right) \quad (1)$$

- štruktúra s interakciou

$$R(s) = Z_R \left(1 + \frac{1}{T_I s} \right) (1 + T_D s) \quad (2)$$

- paralelná štruktúra

$$R(s) = Z_R + \frac{1}{T_I s} + T_D s \quad (3)$$

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

Dôležité je tiež vedieť, že tvar PID regulátora používaný v MATLABe je

$$G_R(s) = P + \frac{I}{s} + D s \quad (4)$$

kde pre štruktúru bez interakcie platia nasledovné vzťahy

$$P = Z_R, \quad I = \frac{Z_R}{T_I}, \quad D = Z_R T_D \quad (5)$$

4.1.4 Proporcionálny regulátor

Najjednoduchšou formou spätnoväzbového regulátora je dvojpohový regulátor. Možno ho opísať nasledovnou podmienkou

$$u(t) = \begin{cases} u_{\max} & \text{ak } e(t) > 0 \\ u_{\min} & \text{ak } e(t) < 0 \end{cases} \quad (6)$$

Riadiaca veličina takto môže nadobúdať len dve hodnoty, podľa znamienka regulačnej odchýlky. Nevýhodou takéhoto regulátora je, že aj malá hodnota regulačnej odchýlky, môže spôsobiť veľkú zmenu riadiacej veličiny. Pre takéto prípady sa zavádza regulátor, ktorý generuje riadiacu veličinu, ktorá je priamoúmerná regulačnej odchýlke. Jeho správanie opisuje rovnica

$$u(t) = Z_R e(t) \quad (7)$$

Takýto regulátor sa nazýva proporcionálny regulátor (P regulátor).

Ak je regulačná odchýlka veľká (v absolútnej hodnote), P regulátor sa chová ako dvojpohový, t.j. u je ohraničené svojimi hraničnými hodnotami. Ak riadený systém neobsahuje integrátor, obvod bude vykazovať trvalú regulačnú odchýlku.

4.1.4.1 Príklady na riadenie pomocou P regulátora

4.1.4.1.1 Riadenie plášťového výmenníka tepla s P regulátorom

Pre tento príklad uvažujeme spätnoväzbové riadenie plášťového výmenníka tepla s P regulátorom, ktorého prenos je

$$R(s) = Z_R \quad (8)$$

Výmenník možno opísať diferenciálnou rovnicou

$$T_1 y' + y = Z_1 u + \frac{Z_2}{Z_1} d \quad (9)$$

kde u je riadiaca a d poruchová veličina.

Predpokladajme napr. riadenie pri $W(s) = 0$ a $D(s) = A/s$. Prenos URO vzhľadom na poruchovú veličinu možno napísať

$$G_{yd} = \frac{G G_d}{1 + G R} = \frac{\frac{Z_1}{T_1 s + 1} \frac{Z_2}{Z_1}}{1 + Z_R \frac{Z_1}{T_1 s + 1}} \quad (10)$$

Pre výstupnú veličinu y potom platí

$$Y(s) = \frac{\frac{Z_1}{T_1 s + 1} \frac{Z_2}{Z_1}}{1 + Z_R \frac{Z_1}{T_1 s + 1}} \frac{A}{s} = \frac{AZ_2}{1 + Z_R Z_1} \frac{1}{1 + Z_R \frac{Z_1}{T_1 s + 1}} \frac{A}{s} \quad (11)$$

$$\frac{y(t)}{AZ_2} = \frac{1}{1 + Z_R Z_1} \left(1 - e^{-\frac{t}{T_1/(1+Z_R Z_1)}} \right) \quad (12)$$

Výstup z výmenníka je stabilný pre ľubovoľné Z_R a ustáli sa na hodnote

$$y(\infty) = \frac{AZ_2}{1 + Z_R Z_1} \quad (13)$$

pričom trvalá regulačná odchýlka (TRO) je

$$e(\infty) = w(\infty) - y(\infty) = -\frac{AZ_2}{1 + Z_R Z_1} \quad (14)$$

t.j. so zvyšovaním zosilnenia sa znižuje trvalá regulačná odchýlka. So zvyšujúcim sa zosilnením narastá aj sklon prechodovej charakteristiky.

4.1.4.1.2 Systém vyššieho rádu riadený P regulátorom

Uvažujme spätnoväzbové riadenie systému vyššieho rádu opísaného prenosom

$$G(s) = \frac{Z}{(Ts + 1)^n} \quad (15)$$

s P regulátorom, ktorého prenos je $R(s) = Z_R$.

Platia tu tie isté črty aplikovania P regulátora, ako pre výmenník tepla. Trvalá regulačná odchýlka sa znižuje so zvyšujúcim zosilnením regulátora. Môže to však byť na úkor kvality riadenia, pretože pre dostatočne vysoké Z_R sa môže stať obvod nestabilným.

4.1.5 Integračný regulátor

Pre riešenie problému trvalej regulačnej odchýlky je najjednoduchšie zaviesť do URO integrátor. Akčný zásah sa tak bude meniť dovtedy, kým nebude regulačná odchýlka nulová. Integrátor totiž pracuje ako pamäťový člen, t.j. aj pri nulovom vstupnom signále dokáže generovať výstup rôzny od nuly. Táto vlastnosť zaručuje, že TRO môže byť odstránená, ak regulátor zahŕňa v sebe I zložku.

Matematicky to možno zapísať nasledovne

$$\frac{de(t)}{dt} = \frac{1}{T_I} e(t) \quad (16)$$

konštanta T_I sa nazýva integračná časová konštanta a určuje rýchlosť zmeny akčného zásahu v prípade jednotkovej regulačnej odchýlky. Čím bude veľkosť T_I menšia, tým väčšie zmeny riadenia budú generované.

4.1.5.1 Príklady na riadenie pomocou PI regulátora

4.1.5.1.1 Riadenie plášťového výmenníka tepla s PI regulátorom

Opäť uvažujeme spätnoväzbové riadenie pre daný výmenník tepla s PI regulátorom, ktorého prenos je

$$R(s) = Z_R \left(1 + \frac{1}{T_I s} \right) \quad (17)$$

Pre odozvu systému na skokovú zmenu poruchy môžeme v Laplaceovej transformácii napísať

$$Y(s) = \frac{\frac{Z_2}{Z_1} \frac{Z_1}{T_I s + 1}}{1 + Z_R \frac{T_I s + 1}{T_I s} \frac{Z_1}{T_I s + 1}} \frac{A}{s} = \frac{AZ_2 T_I}{T_I T_1 s^2 + (T_I + Z_R T_I Z_1)s + Z_R Z_1} \quad (18)$$

takáto prechodová charakteristika bude stabilná a ustáľovať sa bude v nule, pričom jej priebeh bude závisieť od vlastností výmenníka a konštánt regulátora.

Odozva systému na skokovú zmenu žiadanej veličiny bude

$$Y(s) = \frac{Z_1 Z_R (T_I s + 1)}{T_I T_1 s^2 + (T_I + Z_R T_I Z_1)s + Z_R Z_1} \frac{A}{s} \quad (19)$$

Tento regulátor odstraňuje trvalú regulačnú odchýlku, preto jej hodnota je daná nasledovne

$$e(\infty) = w(\infty) - y(\infty) = A - \lim_{s \rightarrow 0} s Y(s) = A - A = 0 \quad (20)$$

Pri (19) sa v čitateli prechodovej charakteristiky nachádza stabilná nula $s = -1/T_I$, ktorej vplyv sa pri znižovaní hodnoty T_I zosilňuje, pretože sa blíži k hranici nestability. Z praktického hľadiska sa to prejaví na zväčšovaní maximálneho preregulovania.

4.1.5.1.2 Systém vyššieho rádu riadený PI regulátorom

Uvažujme opäť spätnoväzbové riadenie systému vyššieho rádu tentokrát s PI regulátorom, opísaného rovnakým prenosom ako pri P regulátore

$$G(s) = \frac{Z}{(Ts + 1)^n} \quad (21)$$

Prenos s PI regulátora uvažujme rovnaký ako pri riadení výmenníka tepla s PI regulátorom (17). Zároveň tu platia rovnaké zásady aplikovania PI regulátora ako v predchádzajúcom príklade. Znamená to, že TRO je nulová, len pre vysoké hodnoty T_I sa výstup bude ušľachťovať pomaly. A zase pre dostatočne nízke hodnoty T_I sa môže stať systém nestabilným.

4.1.6 Derivačný regulátor

Derivačná zložka zlepšuje stabilitu URO. Jej zavedenie vyplýva z toho, že účinok ako P tak aj I zložky sa prejaví až po určitom čase, čiže nie je okamžitý.

Derivačný regulátor, ktorý možno opísať nasledovnou rovnicou

$$u(t) = T_D \frac{de(t)}{dt} \quad (22)$$

T_D je derivačná časová konštanta a predpokladá vývoj regulačnej odchýlky. Ukázať sa to dá na rozvoji $e(t+T_D)$ do Taylorovho radu, kde platí

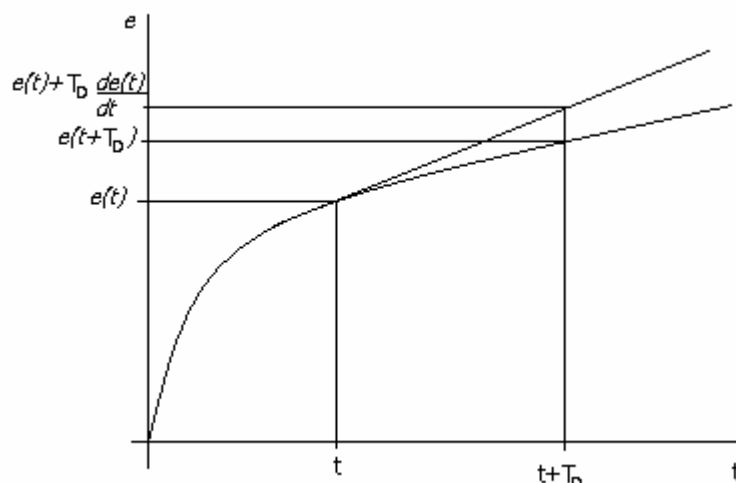
$$e(t+T_D) \approx e(t) + T_D \frac{de(t)}{dt} \quad (23)$$

Grafické znázornenie rovnice je na obr.5.

Ideálny D regulátor je citlivý na šum v riadenej veličine, pretože derivácia nepresného signálu môže viesť k veľkým a častým zmenám amplitúdy riadenia. Ideálny derivačný člen je nerealizovateľný, pretože je nekauzálny. Preto sa zavádza filtrovaný D regulátor s prenosom

$$R(s) = \frac{T_D s}{1 + \frac{T_D}{N} s} \quad (24)$$

Ide vlastne o ideálny D regulátor zapojený v sérii so systémom prvého rádu s časovou konštantou $T_D = N$, pričom hodnoty N sú v intervale medzi 5 a 20.



Obr. 5 Účinok D zložky

4.1.6.1 Príklad na riadenie pomocou PID regulátora

4.1.6.1.1 Systém vyššieho rádu riadený PID regulátorom

Aj pre nasledujúci príklad uvažujme spätnoväzbové riadenie systému vyššieho rádu

$$G(s) = \frac{Z}{(Ts + 1)^n} \quad (25)$$

tentokrát s PID regulátorom.

Aplikácia T_D zložky spôsobí, že regulácia je rýchlejšia a tiež, že vzniká menšie preregulovanie. Teoreticky sa teda môže zdať, že T_D má za následok neustále zlepšovanie kvality riadenia. Táto úvaha je chybná, pretože T_D zložka má predikčný charakter čo znamená, že prítomnosť šumu znemožňuje zvyšovať ľubovoľne T_D .

4.1.7 Saturácia integrátora – Vplyv obmedzení na integračnú zložku regulátora

Akčná veličina je v praxi obmedzená v intervale (u_{min}, u_{max}) . Ak regulátor obsahuje I zložku a akčný zásah je na obmedzení, integrátor pokračuje v integrovaní regulačnej odchýlky. Dôsledkom toho, ani opačné znamienko regulačnej odchýlky nespôsobí okamžitú zmenu akčného zásahu, pretože I zložka dominuje regulátoru. Klasickým príznakom saturácie integrátora je vysoké preregulovanie spôsobené oneskorenou aktivitou regulátora.

4.1.7.1 Príklad na saturáciu integrátora

Opäť uvažujme spätnoväzbové riadenie systému vyššieho rádu, ktorého prenos je

$$G(s) = \frac{Z}{(Ts + 1)^n} \quad (26)$$

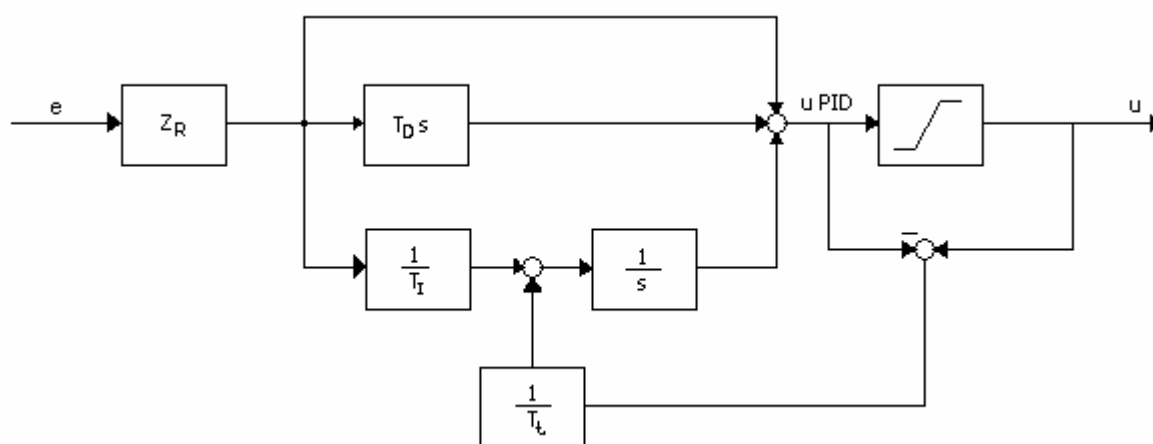
a je riadený PI regulátorom v tvare

$$R(s) = Z_R + \frac{1}{T_I s} \quad (27)$$

Odstrániť problém saturácie integračnej zložky možno niekoľkými spôsobmi. Najčastejšie používaným je metóda spätného výpočtu (obr.6). Princíp spätného výpočtu je v prepočítavaní integračnej zložky tak, aby sa akčný zásah nachádzal v danom obmedzení. V praxi to je možné robiť len dynamicky s časovou konštantou T_t , ktorá je daná empirickým vzťahom

$$T_t = \sqrt{T_I T_D} \quad (28)$$

a nie v každom časovom bode výpočtu.



Obr. 6 Korekcia saturácie integrátora spätným výpočtom

Na vstupe do prídavného spätnoväzbového obvodu je rozdiel medzi vypočítaným a skutočným akčným zásahom. Ak je vypočítaný zásah v dovolenom intervale, je rozdiel nulový a regulátor sa správa ako pôvodný. Ak sú signály rozdielne, signál integračnej zložky sa mení, kým nedosiahne požadovanú hodnotu.

4.1.8 Nastavovanie regulátora Strejcovou metódou

4.1.8.1 Príklad na Strejcovu metódu

Pre nastavenie parametrov PID regulátora v tvare

$$R(s) = Z_R \left(1 + \frac{1}{T_I s} + T_D s \right) \quad (29)$$

uvažujeme nasledovný prenos riadeného systému

$$G(s) = \frac{Z}{(T s + 1)^n} e^{-T_d s} \quad (30)$$

pričom Z je zosilnenie regulátora, T je časová konštanta, T_d je dopravné oneskorenie systému a n je rád riadeného systému. Parametre regulátora podľa Strejca sú určované tabuľkou 3.

Vyberieme regulátor a z príslušného riadku tabuľky vypočítame jeho parametre. Ak volíme štruktúru PID regulátora, potom musí byť riadený prenos minimálne tretieho rádu, pre P a PI regulátor je nutný prenos minimálne druhého rádu. Vyplýva to zo vzťahu pre výpočet Z_R v tabuľke 3.

Tabuľka 3 Nastavovanie parametrov regulátora Strejcovou metódou

Regulátor	Z_R	T_I	T_D
P	$\frac{1}{Z} \frac{1}{n-1}$		
PI	$\frac{1}{Z} \frac{n+2}{4(n-1)}$	$T \frac{n+2}{3}$	
PID	$\frac{1}{Z} \frac{7n+16}{16(n-2)}$	$T \frac{7n+16}{15}$	$T \frac{(n+1)(n+3)}{7n+16}$

4.2 Polynomický návrh regulátora

4.2.1 Riešenie diofantickej rovnice

Lineárna skalárna polynomická rovnica

$$a(s)x(s) + b(s)y(s) = c(s) \quad (31)$$

sa nazýva diofantická. Pre jednotlivé polynómy platí, že $a(s)$, $b(s)$ a $c(s)$ sú dané a polynómy $x(s)$ a $y(s)$ sú neznáme. Riešenie diofantickej rovnice charakterizuje nasledujúca veta.

Veta o riešiteľnosti diofantickej rovnice:

Nech $a(s)$, $b(s)$ a $c(s)$ sú polynómy s reálnymi koeficientmi. Potom diofantická rovnica má riešenie vtedy a len vtedy, ak najväčší spoločný deliteľ $a(s)$ a $b(s)$ delí $c(s)$.

Na základe vety o riešiteľnosti diofantickej rovnice platí, že ak $a(s)$ a $b(s)$ sú nesúdeliteľné, potom diofantická rovnica je riešiteľná pre ľubovoľný polynóm $c(s)$ vrátane prípadu, ak $c(s) = 1$. Keď je diofantická rovnica riešiteľná, potom má nekonečne veľa riešení.

Všeobecné riešenie diofantickej rovnice je teda

$$x(s) = x'(s) + \bar{b}(s)t(s) \quad (32)$$

$$y(s) = y'(s) - \bar{a}(s)t(s) \quad (33)$$

pričom $x(s)$ a $y(s)$ sú partikulárnym riešením diofantickej rovnice, $t(s)$ je ľubovoľný polynóm a $a(s)$, $b(s)$ sú nesúdeliteľné polynómy, pre ktoré platí vzťah

$$\frac{\bar{b}(s)}{\bar{a}(s)} = \frac{b(s)}{a(s)} \quad (34)$$

Z predchádzajúceho vyplýva, že zo všetkých riešení diofantickej rovnice existuje jediné riešenie, dvojica polynómov $x(s)$ a $y(s)$ pre ktorú platí, že

$$\deg x(s) < \deg \bar{b}(s) \quad (35)$$

$$\deg y(s) < \deg \bar{a}(s) \quad (36)$$

Tieto dve riešenia sa zhodujú len vtedy, ak platí

$$\deg a(s) + \deg b(s) \geq \deg c(s) \quad (37)$$

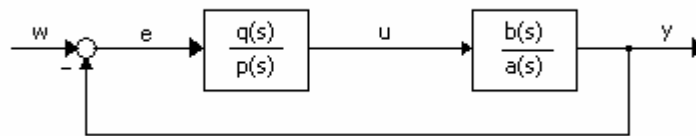
Ak platí predchádzajúci vzťah, potom riešenie diofantickej rovnice je minimálneho stupňa z čoho vyplýva, že

$$\deg x(s) = \deg y(s) \quad (38)$$

ak vzťah neplatí, potom riešenie diofantickej rovnice je možné hľadať aj pre minimálny stupeň polynómu $x(s)$ alebo pre minimálny stupeň polynómu $y(s)$.

4.2.2 Polynomický návrh PA (pole assignment) regulátorov

Majme nasledovný spätnoväzbový systém s jedným stupňom voľnosti



Obr. 7 Zapojenie polynomickeho PA regulátora

Prenos riadeného systému je

$$G(s) = \frac{b(s)}{a(s)} \quad (39)$$

a jeho polynómy sú

$$a(s) = a_n s^n + a_{n-1} s^{n-1} + \dots + a_0 \quad (40)$$

$$b(s) = b_{n-1} s^{n-1} + b_{n-2} s^{n-2} + \dots + b_0 \quad (41)$$

a sú nesúdeliteľné. Systém je jednorozmerný, striktné rýdzi.

Prenos regulátora má tvar

$$R(s) = \frac{q(s)}{p(s)} \quad (42)$$

polynómy regulátora sú tieto

$$p(s) = p_{n_p} s^{n_p} + p_{n_p-1} s^{n_p-1} + \dots + p_0 \quad (43)$$

$$q(s) = q_{n_p} s^{n_q} + p_{n_q-1} s^{n_q-1} + \dots + q_0 \quad (44)$$

Žiadaný stabilný charakteristický polynóm uzavretého obvodu je

$$c(s) = c_{n_p} s^{n_c} + c_{n_c-1} s^{n_c-1} + \dots + c_0 \quad (45)$$

Voľba stupňov polynómov regulátora a polynómu $c(s)$ sa riadi určitými pravidlami. Aby bol regulátor čo najjednoduchší, z diofantickej rovnice vyplýva, že stupeň polynómu $q(s)$ musí byť menší ako stupeň polynómu $a(s)$, čiže $n_q = n - 1$. Aby bola splnená fyzikálna realizovateľnosť regulátora musí platiť, že stupeň polynómu menovateľa $p(s)$ musí byť minimálne rovný alebo väčší ako stupeň $q(s)$. Stupeň polynómu $c(s)$ je potom možné určiť na základe stupňa výsledného polynómu na ľavej strane diofantickej rovnice.

4.2.2.1 Príklady na riešenie polynomického návrhu PA regulátora

Prípad 1:

Nech $c(s)$ je ľubovoľný polynóm stupňa $n_c = 2n - 1$, potom existuje polynóm $p(s)$ stupňa $n_p = n - 1$ a polynóm $q(s)$ stupňa $n_q = n - 1$, takže platí

$$a(s)p(s) + b(s)q(s) = c(s) \quad (46)$$

Takto navrhnutý regulátor $R(s)$ je rýdži.

Prípad 2:

Nech $c(s)$ je ľubovoľný polynóm stupňa $n_c = 2n$, potom existuje polynóm $p(s)$ stupňa $n_p = n$ a polynóm $q(s)$ stupňa $n_q = n - 1$, takže platí rovnica (46)

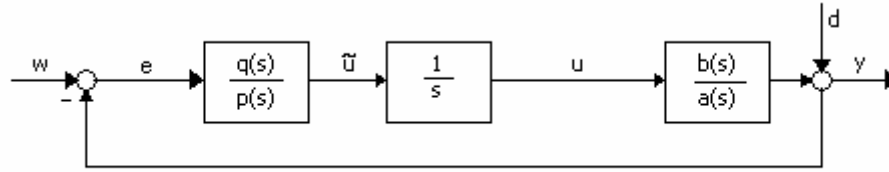
Takto navrhnutý regulátor $R(s)$ je striktné rýdži.

4.2.3 Polynomický PA regulátor s integračnou činnosťou

Klasický uzavretý regulačný obvod (obr.7), ako sme ho uviedli v predchádzajúcej kapitole, s regulátorom a riadeným systémom nezabezpečí pri existencii porúch odstránenie trvalej regulačnej odchýlky. Toto je však možné dosiahnuť zaradením integračného člena do uzavretého obvodu (obr.8).

PA regulátor je tvorený blokom $q(s)/p(s)$ a blokom integrátora $1/s$. Takýto PA regulátor zabezpečí pre skokové zmeny žiadanej veličiny w a skokové zmeny poruchy d ,

že regulačná odchýlka e a veličina \tilde{u} v našom uzavretom obvode sa asymptoticky blížia k nule,



Obr. 8 Zapojenie polynomickeho PA regulatora s integračnou zložkou

ak $q(s)$ a $p(s)$ spĺňajú rovnicu

$$a(s)sp(s) + b(s)q(s) = c(s) \quad (47)$$

a polynómu $c(s)$ je stabilný.

Postup pri návrhu PA regulatora s integračnou činnosťou je nasledovný.

Riadený systém formálne rozšírime o integrátor a polynómy $p(s)$ a $q(s)$ určíme z rovnice

$$\tilde{a}(s)p(s) + b(s)q(s) = c(s) \quad (48)$$

pričom $\tilde{a}(s)$

$$\tilde{a}(s) = sa(s) \quad (49)$$

a jeho stupeň je $n_I = n + 1$.

Po nájdení polynómov $p(s)$ a $q(s)$, ktoré určíme z predchádzajúcej rovnice, integrátor presunieme do regulatora. Skutočný regulátor bude mať teda prenosovú funkciu

$$R(s) = \frac{q(s)}{\tilde{p}(s)} \quad (50)$$

pričom

$$\tilde{p}(s) = sp(s) \quad (51)$$

Aj v tomto prípade voľba stupňov polynómov regulatora a pravej strany nie je náhodná. Vyplýva z požiadavky minimálneho stupňa polynómu $q(s)$, ktorý je vo všetkých prípadoch daný z diofantickej rovnice (47), vzťahom $n_q = n$. Jednotlivé prípady regulátorov vychádzajú z regulatora s minimálnou realizáciou tak, aby menovateľ $sp(s)$ bol minimálne rovnakého stupňa ako čitateľ. Keď sú už teda známe stupne polynómov regulatora, je stupeň polynómu $c(s)$ možné určiť zo stupňa výsledného polynómu na ľavej strane diofantickej rovnice (47). Stupeň polynómu $c(s)$ môže byť aj väčší ako $2n_I$. Takýto

regulátor by však bol už príliš zložitý. Najčastejšie sa pri voľbe stupňov regulátora používajú prípady 1 a 2.

4.2.3.1 Príklady na riešenie polynomickeho návrhu PA regulátora s integračnou činnosťou

Prípady 1:

Nech $c(s)$ je ľubovoľný polynóm stupňa $n_c = 2n$, potom existuje polynóm $p(s)$ stupňa $n_p = n - 1$ a polynóm $q(s)$ stupňa $n_q = n$, takže platí rovnica (49).

V tomto prípade prenos regulátora $q(s)/p(s)$ nespĺňa podmienku rýdzosti, ale prenos skutočne realizovaného regulátora je rýdzi.

Prípady 2:

Nech $c(s)$ je ľubovoľný polynóm stupňa $n_c = 2n_I - 1 = 2n + 1$, potom existuje polynóm $p(s)$ stupňa $n_p = n_I - 1 = n$ a polynóm $q(s)$ stupňa $n_q = n_I - 1 = n$, takže platí rovnica (49). Prenos $q(s)/p(s)$ je v tomto prípade rýdzi. Skutočný regulátor je striktne rýdzi a zodpovedá mu prenos

$$R(s) = \frac{q(s)}{\tilde{p}(s)} \quad (52)$$

Prípady 3:

Nech $c(s)$ je ľubovoľný polynóm stupňa $n_c = 2n_I = 2(n + 1)$, potom existuje polynóm $p(s)$ stupňa $n_p = n_I = n + 1$ a polynóm $q(s)$ stupňa $n_q = n_I - 1 = n$, takže platí rovnica (49). Prenos (42) je v tomto prípade striktne rýdzi.

4.3 Zoznam MWS prezentácií

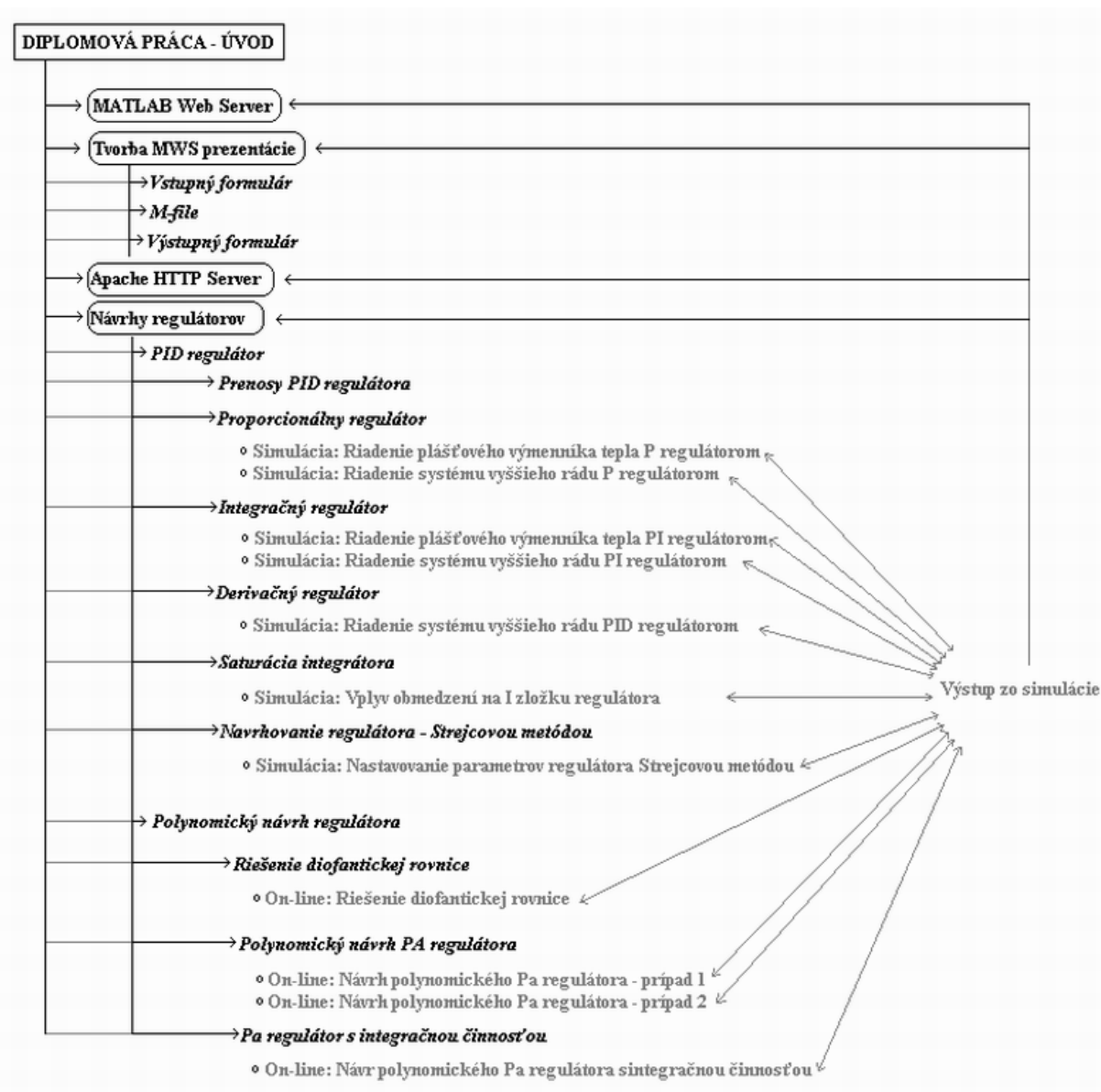
K jednotlivým príkladom teoreticky popísaným v predchádzajúcich dvoch kapitolách boli vytvorené MWS prezentácie. Sú však spracované tak, aby boli čo najvšeobecnejšie a teda bolo možné na základe zadaných parametrov riešiť, čo najviac variant jednotlivých úloh, pri čo najmenších obmedzujúcich podmienkach. Zároveň, aby však zostala zachovaná súvislosť s [1].

Celkovo bolo vytvorených 11 *MWS* prezentácií uvedených nižšie

- [Simulácia riadenia plášťového výmenníka tepla s P regulátorom](#)
- [Simulácia riadenia pre systém vyššieho rádu riadený P regulátorom](#)
- [Simulácia riadenia plášťového výmenníka tepla s PI regulátorom](#)
- [Simulácia riadenia systém vyššieho rádu riadený s PI regulátorom](#)
- [Simulácia pre systém vyššieho rádu riadený s PID regulátorom](#)
- [Simulácia: Vplyv obmedzení na I zložku regulátora](#)
- [Strejcova metóda nastavovania parametrov regulátora](#)
- [On-line riešenie diofantickej rovnice](#)
- [On-line: Prípad 1 návrhu polynomickeho PA regulátora](#)
- [On-line: Prípad 2 návrhu polynomickeho PA regulátora](#)
- [On-line: Návrh polynomickeho PA regulátora s integračnou činnosťou](#)

5 ŠTRUKTÚRA E – LEARNINGOVÝCH STRÁNOK

Všetky WWW stránky boli spracované ako e-learningový dokument k téme návrhu regulátorov. Vytvorené boli tak, aby orientácia v nich bola čo najjednoduchšia. K jednoduchšej orientácii slúži prehľadné navigačné menu nachádzajúce sa vždy v ľavej časti každej stránky, poskytujúce rýchle prepínanie sa medzi jednotlivými tematickými okruhmi a stránkami e-learningovej prezentácie (obr.9).



Obr. 9 Mapa e-learningových stránok

5.1 Vypĺňanie formulárov

Vstupné formuláre jednotlivých *MWS* prezentácií sa nachádzajú na stránke zaoberajúcej sa návrhom regulátorov. Táto je rozdelená na dve časti (obr.9). Prvá sa venuje návrhu PID regulátora a druhá polynomickeému návrhu regulátorov. Každá časť obsahuje on-line simulácie jednotlivých príkladov.

Po kliknutí na príslušnú linku príkladu, sa v internetovom prehliadači užívateľovi zobrazí vstupný formulár prislúchajúci danej *MWS* prezentácii (obr.10). Vo formuláre treba len vyplniť požadované položky a kliknúť na tlačidlo “*Odoslať údaje*”. Ak chce užívateľ vo formuláre pôvodné hodnoty, ktoré sú v ňom predvolené nastavené pri otvorení stránky, môže tak urobiť kliknutím na tlačidlo “*Pôvodné údaje*”.

Jednotlivé formuláre sú ošetrené JavaScriptom. Ak teda užívateľ niektorú z položiek zabudne vyplniť alebo ju zadá nesprávne, obdrží hlásenie o chybe (obr.3), t.j. ktoré položky boli vyplnené nesprávne alebo neboli vyplnené vôbec a ako je možné danú chybu odstrániť. Napr. ako desatinný znak je akceptovaná bodka, nie čiarka s ohľadom na to, že dáta sú spracovávané MATLABom.

Obr. 10 Vstupný formulár

Ak sú všetky vstupné údaje zadné správne, po odoslaní formulára a spracovaní dát MATLABom, tak ako bolo uvedené v predchádzajúcich kapitolách, v prehliadači sa zobrazí výstupná šablóna formulára a v nej výsledky výpočtov a simulácie (obr.11).

Z výstupného formulára sa možno vrátiť späť na stránku vstupného formulára pomocou vytvorenej linky “Späť na stránku príkladu“. A rovnako zo vstupného ako aj výstupného formulára sa možno prejsť na ktorúkoľvek stránku e-learningového dokumentu, kliknutím na linku v navigačnom menu v ľavej hornej časti stránky.

Polynomický návrh PA regulátora s integračnou činnosťou

Výpočet

MATLAB Web Server

Tvorba MWS prezentácie

Apache HTTP Server

Návrh regulátorov

PA regulátor s integračnou činnosťou

Vaše zadanie parametre

Prenosová funkcia		
Čitateľ prenosu B =	[1]	
Menovateľ prenosu A =	[1 2]	
Polynóm c(s) =	[1 6 9]	

Parametre simulácie		
Dĺžka simulácie t =	10	
Žiadaná veličina	Initial value =	5
	Final value =	10
Poruchová veličina	Step time =	5
	Final value =	0.5

Prenos systému

$$G(s) = \frac{1}{s+2}$$

Výsledok výpočtu

Riešenie diofantickej rovnice je minimálneho stupňa

Polynóm $p(s) = 1$

Polynóm $q(s) = 4s+9$

Prenos PA regulátora s integračnou činnosťou

$$R(s) = \frac{4s+9}{s}$$

Grafické znázornenie

Priebeh riadiacej veličiny pri pôsobení poruchy podľa zadanych parametrov

Priebeh riadenia meranej veličiny pri pôsobení poruchy podľa zadanych parametrov

Obr. 11 Výstupný formulár

6 PROBLÉMY PRI TVORBE *MWS* PREZENTÁCIÍ

Pri vytváraní a prevádzke *MWS* prezentácií možno naraziť na niekoľko problémov. Uvedené sú tie, ktoré sa vyskytli najčastejšie pri riešení zadania tejto práce.

6.1 Nesprávna konfigurácia vstupných údajov

Jedným z častých problémov je, že po vytvorení *MWS* prezentácie, správnom vyplnení vstupného formulára internetový prehliadač zobrazí nasledujúcu chybu

```
Error: Invalid configuration entry on server: „názov m-súboru“.
```

Ide o chybové hlásenie *MATLAB Web Serveru*, že nenašiel požadovaný **m-súbor**, potrebný pre výpočet, simuláciu a zobrazenie výstupnej šablóny. V takom prípade je vhodné skontrolovať konfiguračný súbor **matweb.conf**, či je v ňom zapísaná daná *MWS* prezentácia a či správne zadaná adresárová cesta. Ak je konfiguračný súbor v poriadku, môže byť chyba v HTML skripte vstupného formulára. Je potrebné skontrolovať v skrytom vstupe, či je správne zadaný parameter **VALUE**, obsahujúci názov prezentácie, viď. [kap. 3.1.3](#).

6.2 Inicializácia polynomickeho toolboxu

Polynomickeý toolbox je produktom spoločnosti Polyx [4]. Je to balíček programov operujúci so širokým spektrom polynomickeých metód. Obsahuje viac ako 222 skriptov vytvorených v *MATLABe*, vďaka čomu je jednoducho použiteľný.

Pri práci v *MATLABe* sa využíva súbor **pinit.m**, ktorým sa inicializuje polynomickeý toolbox. Pri spúšťaní polynomickeho toolboxu príkazom **pinit** cez *MATLAB Web Server* dochádza k chybovému hláseniu typu **500 Internal Server Error** (obr.12). V polynomickeom toolboxe nie je k dispozícii funkcia, ktorá by spúšťala toolbox pri práci s *MATLAB Web Serverom*, preto treba pôvodnú funkciu jednoduchým spôsobom upraviť.

Zo súboru **pinit.m**, musia byť odstránené nasledujúce príkazy

```
disp(' ');  
disp('Polynomial Toolbox initialized. To get started, type one of');  
disp('these: helpwin or poldesk. For product information, visit');  
disp('www.polyx.com or www.polyx.cz.');
```

Takto upravený **m-súbor** je vhodné uložiť pod novým názvom, v našom prípade sme ho pomenovali **pinit_mws.m** a polynomickeý toolbox inicializovať pomocou príkazu **pinit_mws**.



Obr. 12 Chybové hlásenie pri inicializácii polynomickeého toolboxu príkazom **pinit**

6.3 Nekompatibilita internetových prehliadačov

Pri programovaní WWW stránok sa v jazyku HTML sa vyskytlo niekoľko problémov, ktoré súviseli s vzájomnou nekompatibilitou jednotlivých internetových prehliadačov, resp. to bolo spôsobené rôznou implmentáciou *W3C* štandardov. Išlo hlavne o dizajnové rozdiely WWW stránok, ktoré však nijako neovplyvnili ich funkčnosť vytvorených aplikácií.

Záver

Pri tvorbe tejto e-learningovej učebnice bolo využité spojenie samostatných častí, t.j. WWW stránok a *MATLAB Web Serveru*, čím bola vytvorená reálne fungujúca internetová aplikácia, ktorá umožňuje on-line riešenie vybraných úloh. Celkom bolo vytvorených 11 *MWS* prezentácií, zaoberajúcich sa návrhom regulátorov. Vytvorené boli príklady simulácií pre P, PI a PID regulátor, on-line môže byť sledovaná aj saturácia integrátora, návrh regulátora Srejcovou metódou a riešené týmto spôsobom boli aj úlohy polynomickeho návrhu regulátora.

Keďže išlo o vytvorenie e-learningovej aplikácie, okrem jednotlivých častí *MWS* prezentácií, boli vytvorené aj súvisiace WWW stránky, ktoré slúžia na zorientovanie sa v danej problematike. Tieto poskytujú užívateľovi, teoretické podklady k jednotlivým úlohám, ktoré je možné riešiť on-line *MATLAB Web Serverom* a tiež obsahujú všetky potrebné informácie o fungovaní a tvorbe jednotlivých častí *MWS* prezentácií.

Pri tvorbe WWW stránok boli využité kaskádové štýly, ktoré umožňujú jednoduchú úpravu stránok a zjednodušenie jednotlivých HTML kódov. Snahou bolo, aby aplikácia vyzerala rovnako vo všetkých prehliadačoch. Nie vždy to však bolo možné dosiahnuť. Napr. pri nastavovaní šírky tabuľky na WWW stránke dochádza, v prehliadači Internet Explorer 6, k nesprávnej interpretácii vlastnosti `width`. Do tej sa navyše pripočítava aj vnútorný okraj tabuľky (`padding`) a orámovanie (`border`), čo sa prejaví na tom, že tabuľka je širšia ako sú okraje stránky prehliadača. Táto chyba však napríklad nebola zistená v Internet Exploreri 5.5 ani v prehliadačoch Mozilla, Mozilla Firefox a Netscape Navigator. Tieto problémy však boli len dizajnového charakteru a neovplyvnili žiadnym spôsobom fungovanie jednotlivých *MWS* prezentácií.

Na kontrolu zadávaných dát je použitý JavaScript. Pre odladžovanie funkcií JavaScriptu je vhodné použiť napr. Mozillu, ktorej nástroj JavaScript Console, poskytuje možnosť zistiť miesto vzniknutej chyby, aj to, ako ku chybe prišlo. V Internet Exploreri je vytvorený podobný nástroj, kontrolujúci chyby v JavaScripte. Ak sa na stránke vyskytne chyba v stavovom riadku sa objaví upozornenie. Zapnutý JavaScript v prehliadači však nie je nutnou podmienkou fungovania vytvorených aplikácií, pretože okrem JavaScriptu je použité aj preverovanie dát v *MATLABe*. Kontrola údajov pred odoslaním formulára rieši

základné chyby, ktoré sa pri vyplňaní formulárov môžu vyskytnúť: nevyplnenie položky, nesprávne zadanie čísla alebo vektora a tiež či nie sú v položkách formulára zadane namiesto požadovaných hodnôt nezmyselné znaky.

Jednotlivé *MWS* prezentácie, vytvorené v tejto e-learningovej aplikácii boli vytvárané v úzkej súvislosti s literatúrou [1], preto je možné jej využitie pri interaktívnej výuke návrhu regulátorov alebo k názornejšiemu vysvetleniu jednotlivých úloh.

Literatúra

- [1] Mikleš, J., Fikar, M.: *Modelovanie, identifikácia a riadenie procesov II. Identifikácia a optimálne riadenie*. STU Press, Bratislava, 260pp, 2004
- [2] The MathWorks, Inc. *MATLAB Web Server*.
<http://www.mathworks.com/products/webserver>.
- [3] The Apache Software Foundation. *Apache Web Server*. 2001.
<http://www.apache.org>.
- [4] Polyx. *The Polynomial Toolbox for MATLAB*. PolyX, Ltd., Prague, 1999.
<http://www.polyx.com>.
- [5] Vixova príručka webelopera. [on-line]
<http://deja-vix.sk/htmlhelp>.

Príloha

Obsah priloženého CD

Súčasťou tejto diplomovej práce je CD-ROM s nasledujúcim obsahom:

- *Dokumentácia* – je na ňom uložený tento textový dokument vo formátoch DOC a PDF.
- *Zdrojové kódy e-learningového dokumentu* – jednotlivé zdrojové kódy WWW stránok vytvorenej e-learningovej aplikácie.
- *Zdrojové kódy MWS prezentácií* – jednotlivé zdrojové kódy vstupných a výstupných formulárov, m-súborov a simulinkové schémy prezentácie.