



SLOVENSKÁ TECHNICKÁ UNIVERZITA

Fakulta chemickej a potravinárskej technológie

Katedra informatizácie a riadenia procesov

Radlinského 9, 812 37 Bratislava

E – learningový internetový modul pre experimentálnu identifikáciu

Školiteľ: Ing. Ľuboš Čírka, PhD.

Vypracoval: Pavel Dovál

Bratislava 2005

Ďakujem vedúcemu bakalárskej práce Ing. Ľubošovi Čirkovi, PhD. za pomoc pri získavaní vedomostí z oblasti programovania a celkovo za vedenie, rady a pripomienky, ktoré mi poskytol pri vypracovaní bakalárskej práce.

1. ÚVOD	- 4 -
2. TEORETICKÁ ČASŤ	- 5 -
2.1. Identifikácia systémov	- 5 -
2.1.1. Systém 1. rádu	- 6 -
2.1.2. Systém 2. rádu (nekmitavý, aperiodický)	- 7 -
2.1.3. Systém 2. rádu (kmitavý, periodický)	- 8 -
2.1.4. Systém vyššieho rádu	- 9 -
2.1.4.1 Strejcova metóda	- 9 -
2.1.4.2 Broidova metóda	- 9 -
2.2. Matlab web server	- 10 -
3. IDENTIFIKÁCIA PROSTREDNÍCTVOM MATLABU	- 11 -
3.1. Prechodová charakteristika systému 1. rádu	- 12 -
3.1.1. Definovanie údajov zistených z prechodovej charakteristiky	- 12 -
3.1.2. Načítanie údajov prechodovej charakteristiky z textového súboru	- 19 -
3.2. Prechodová charakteristika aperiodického systému 2. rádu	- 23 -
3.3. Prechodová charakteristika periodického systému 2. rádu	- 25 -
3.4. Prechodová charakteristika systému – Strejcova metóda	- 27 -
3.5. Prechodová charakteristika systému – Broidova metóda	- 28 -
4. ZÁVER:	- 29 -
5. LITERATÚRA:	- 30 -
6. PRÍLOHA	- 31 -

1. Úvod

Mojou úlohou na tejto práci bolo oboznámiť sa so základmi programovacieho jazyka html, ktorý tvorí statické www stránky a neskôr so základmi programovacieho jazyka php, tvoriaceho dynamické www stránky.

Po zvládnutí týchto úloh som prešiel na samotnú úlohu môjho projektu, a to bola tvorba stránok zameraných na výpočet prechodových funkcií pomocou zadania základných atribútov charakterizujúcich danú prechodovú funkciu, alebo načítaním tejto krivky z textového súboru. Všetky prenosové charakteristiky som spracovával pomocou programu Matlab.

Venoval som sa prechodovým funkciám 1. rádu, 2. rádu (kmitavej aj nekmitavej) vyššieho rádu (Strejcova metóda, Broidova metóda).

Na tomto mieste by bolo vhodné povedať, čo to vlastne html a php je.

Pôvodné www stránky boli vytvárané v jazyku, ktorý popisoval, čo sa vlastne má na stránke zobraziť. Teda išlo o html. Tento jazyk hovorí klientskému počítaču, ako má prijaté dáta zobraziť. Tieto statické stránky možno spojiť s kódom jazyka php. Tým vznikne dynamická www stránka. Postupom času bol do php zabudovaný SQL modul, ktorý mu umožňuje pracovať s databázami.

2. Teoretická časť

2.1. Identifikácia systémov

Základom pri identifikácii systémov sú vstupné a výstupné veličiny. Vstupné veličiny delíme na technickým spôsobom ovládané (tzv. akčné veličiny) a na ostatné vstupy (označované aj ako poruchy).

Medzi najčastejšie používané vstupné signály patria:

1. skoková funkcia
2. pulz
3. suma sínusov

ako vstupný signál je vhodné zvoliť si signál zaručujúci správne odhadnutie (vypočítanie) parametrov systému.

Existuje niekoľko možností, ako identifikovať klasifikačné metódy.

Jednotlivé typy sú:

1. identifikácia z aktívneho alebo pasívneho experimentu
2. spracovanie matematickou metódou: deterministické a stochastické metódy
3. podľa spôsobu spracovania: metódy pre jednorázové spracovanie a metódy priebežné

V ďalšej časti sa už budem venovať výhradne deterministickým metódam. Jedná sa o metódy, pri ktorých neuvažujeme pôsobenie náhodných veličín na systém. Najčastejším z týchto je jednotkový skok. Patria sem však aj harmonický signál, všeobecný deterministický signál. Vo všetkých prípadoch je nevyhnutný aktívny experiment, ten je však jednoducho realizovateľný a slúži na zistenie približných charakteristík skúmaného systému. Pri uskutočnení experimentu (napríklad metódou jednotkového skoku) je nevyhnutné, aby bol skúmaný systém v ustálenom stave. Časový priebeh výstupnej veličiny je tzv. prechodová charakteristika. Je nevyhnutné tento experiment opakovať niekoľkokrát a to so skokmi rôznej veľkosti a znamienka. Po týchto nevyhnutných úkonoch možno pristúpiť k určovaniu prechodovej funkcie. Tá môže byť 1., 2., či n-tého rádu.

2.1.1. Systém 1. rádu

Predpokladáme náhradu modelu systému prenosom prvého rádu:

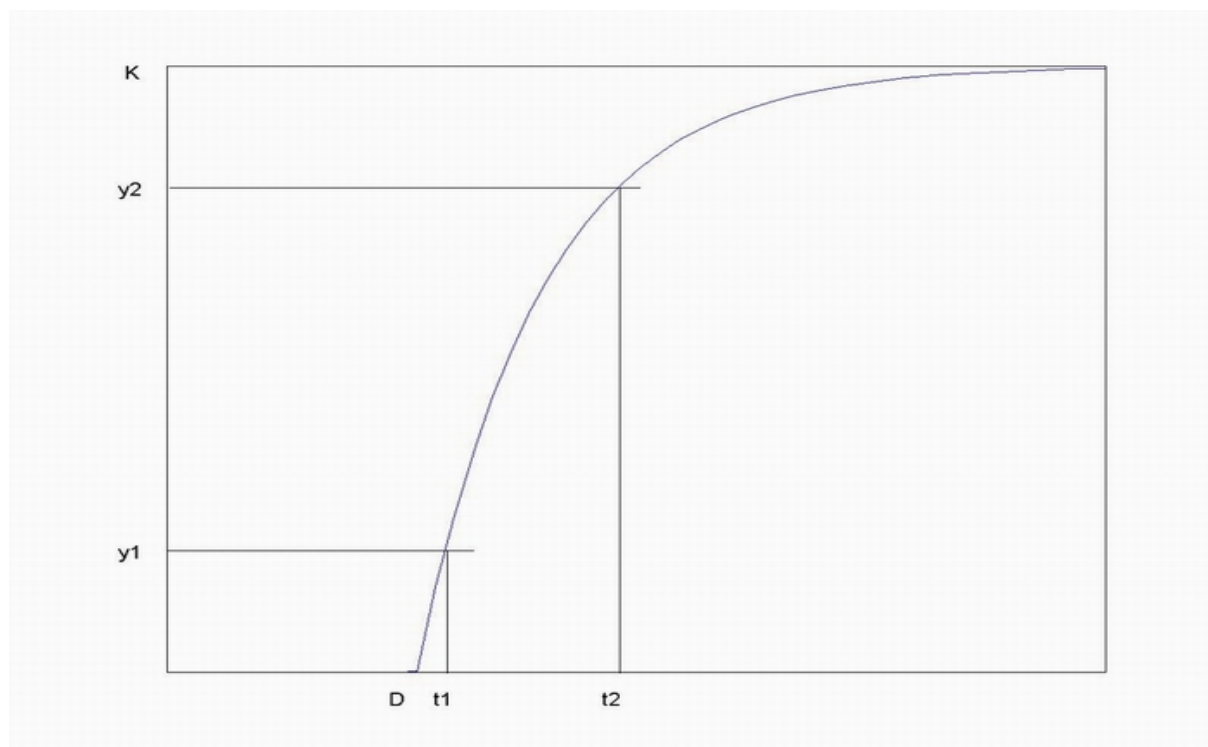
$$F(s) = \frac{K}{Ts + 1} e^{-Ds}$$

Kde K je zosilnenie, T je časová konštanta a D dopravné oneskorenie. Zosilnenie systému je dané rozdielom hodnôt: $K = y(\infty) - y(0)$. Uvažujeme, že poznáme dva body so súradnicami: $[t_1, y_1]$ a $[t_2, y_2]$ na prechodovej charakteristike, názorne sú zobrazené tieto body na obrázku č.1. Podľa týchto dvoch bodov vieme zapísať rovnice:

$$y_1 = K \left(1 - e^{-\frac{t_1 - D}{T}} \right)$$
$$y_2 = K \left(1 - e^{-\frac{t_2 - D}{T}} \right)$$

Po matematických úpravách týchto rovníc dostávame vzťahy:

$$T = \frac{t_2 - t_1}{\ln \frac{K - y_1}{K - y_2}} \quad x = \frac{\ln \frac{K - y_1}{K}}{\ln \frac{K - y_2}{K}}$$
$$D = \frac{t_2 x - t_1}{x - 1} \text{ kde}$$



Obr. č.1: Prechodová charakteristika systému prvého rádu

2.1.2. Systém 2. rádu (nekmitavý, aperiodický)

V tomto prípade platí vzťah:

$$F(s) = \frac{K}{(T_1 s + 1)(T_2 s + 1)}$$

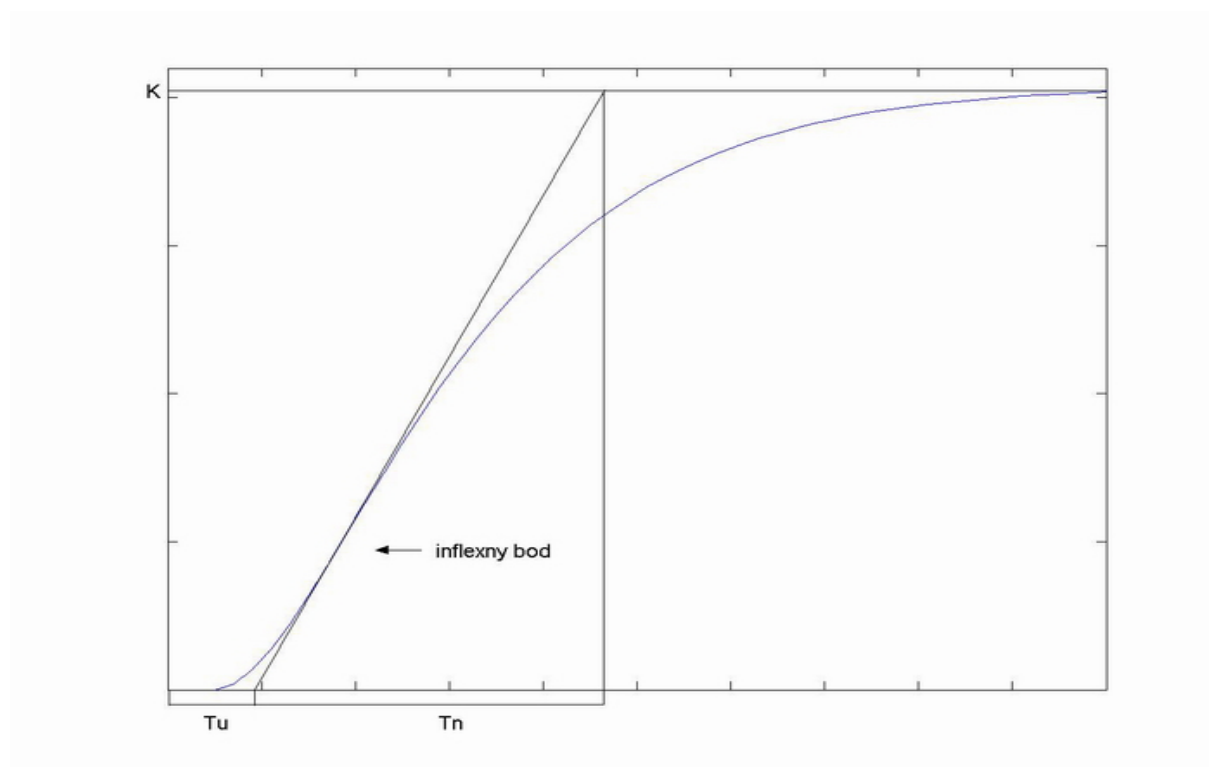
Zosilnenie systému je definované rovnakým spôsobom ako v predchádzajúcom prípade.

Uvažujeme, že sme zistili čas prietahu T_u a čas nábehu T_n , ktoré sú znázornené na obrázku

č.2. Na základe týchto údajov je potrebné určiť hodnoty časových konštánt. Postup je nasledovný:

1. $f_1(k) = T_n / T_u$
2. určenie hodnoty K z prechodovej charakteristiky
3. $T_1 = T_n / f_2(k)$
4. $T_2 = k * T_1$

Kde $f_2(k)$ a k sú hodnoty získané z grafickej závislosti od $f_1(k)$.



Obr. č.2: Prechodová charakteristika aperiodického systému 2. rádu.

2.1.3. Systém 2. rádu (kmitavý, periodický)

Prenos je v tvare:

$$F(s) = \frac{K}{T^2 s^2 + 2T\xi s + 1} = \frac{K\omega_0^2}{s^2 + 2\omega_0\xi s + \omega_0^2}$$

Úlohou je určiť hodnotu zosilnenia, vlastnej frekvencie ω_0 , časovej konštanty $T = 1 / \omega_0$ a tlmenia ξ . Hodnota derivácie v lokálnych extrémoch je nulová. Tieto lokálne extrémny možno vidieť na obrázku č.3, majú súradnice: $[t_1, y_1]$ a $[t_2, y_2]$.

Postup získavania neznámych parametrov je potom nasledovný:

1. $K = y(\infty) - y(0)$

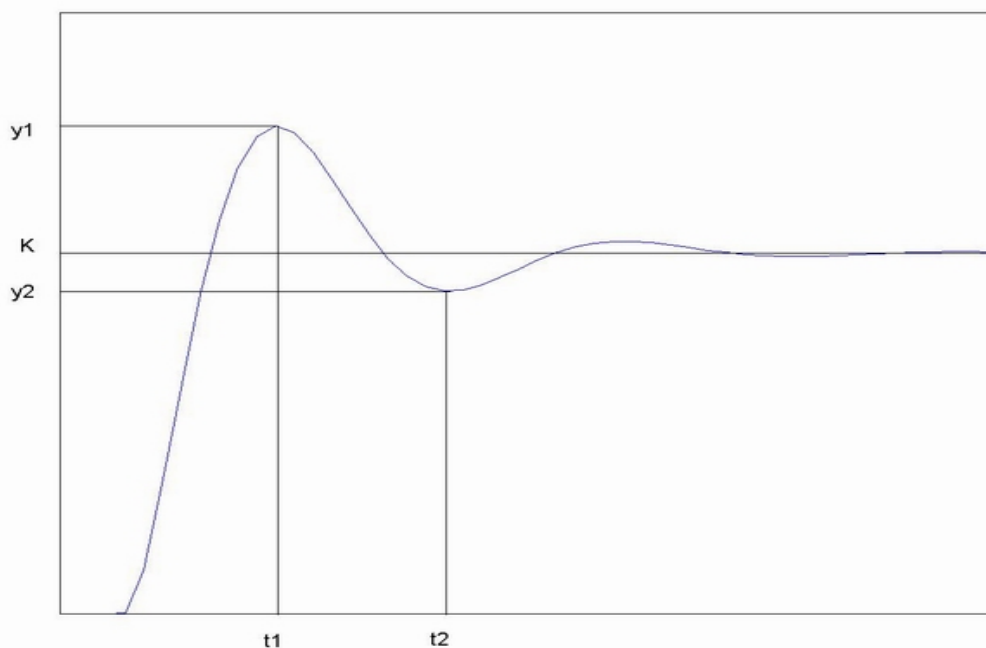
2. $M = \frac{y_1 - y_2}{y_1}$

3. $\xi = \left| \frac{\ln M}{\sqrt{\pi^2 + \ln^2 M}} \right|$

4. $P = \sqrt{1 - \xi^2}$

5. $\omega_0 = \frac{\pi}{(t_2 - t_1)P}$

6. $T = \frac{1}{\omega_0}$



Obr. č.3: Prechodová charakteristika periodického systému 2. rádu.

2.1.4. Systém vyššieho rádu

2.1.4.1 Strejcova metóda

Model systému nahrádzame prenosom v tvare:

$$F(s) = \frac{K}{(Ts + 1)^n} e^{-Ds}$$

Veličiny T, K, D majú rovnaký význam ako v predchádzajúcich prípadoch, n možno označiť pomenovaním rád systému. Na základe hodnôt n možno zostrojiť tabuľku:

N	1	2	3	4	5	6
f(n)	0,000	0,104	0,218	0,319	0,410	0,493
g(n)	1,000	0,368	0,271	0,224	0,195	0,161

Prvým bodom pri postupe identifikácie systému je zistenie hodnôt K, T_u, T_n. Z týchto údajov

vychádzame pri ďalších výpočtoch. Pomer $f = \frac{T_u}{T_s}$ je hodnotou, z ktorej vychádzajú všetky

ďalšie výpočty. Z vyššie uvedenej tabuľky vyberieme prvé menšie f(n) ($f(n) \leq f < f(n+1)$)

a jemu priradíme prislúchajúce g(n) a n. Dopravné oneskorenie a časovú konštantu vypočítame:

$$D = [f(n) - f]T_n$$
$$T = T_n g(n)$$

Prechodová charakteristika je podobná prechodovej charakteristike aperiodického systému 2. rádu, ktorý je znázornený obrázkom č.2.

2.1.4.2. Broidova metóda

Je metóda analogická k Strejcovej, rozdiel je v tabuľke udávajúcej hodnoty n, f(n), g(n).

$$F(s) = \frac{K}{\prod_{k=1}^n \left(\frac{T}{k} s + 1 \right)} e^{-Ds}$$

N	1	2	3	4	5	6
f(n)	0,000	0,096	0,192	0,268	0,331	0,385
g(n)	1,000	0,500	0,440	0,420	0,410	0,400

Prechodová charakteristika je podobná prechodovej charakteristike aperiodického systému 2. rádu, ktorý je znázornený obrázkom č.2.

2.2. Matlab web server

Matlab web server (ďalej skrátene MWS) toolbox je nástroj umožňujúci vytvárať aplikácie v prostredí Matlab. Týmto aplikáciám budeme ďalej hovoriť prezentácie. Táto je umiestnená na MWS servere a je ovládaná prostredníctvom internetu. Na to, aby sme takúto prezentáciu mohli vytvoriť je potrebné zvládnuť jazyk html a programovanie v Matlabe.

Na činnosť MWS je potrebné mať počítač, na ktorom je nainštalovaný Matlab a Matlab web server toolbox. MWS ale potrebuje na svoju činnosť okrem už spomínaných prezentácií aj web server podporujúci CGI (Common Gateway Interface) skripty (napr. Apache). Matlab a web server môžu, ale nemusia byť umiestnené na tom istom počítači. Takýto počítač teda možno nazvať serverovým počítačom.

Každá MWS prezentácia sa skladá z troch častí. Prvou je html súbor nazývaný vstupným formulárom, druhou časťou je súbor príkazov pre Matlab (m-súbor), treťou a poslednou časťou je výstupný html súbor. Obsahovať môže jednotlivé čísla, vektory, matice, ale i grafy. Teda koncový používateľ nepotrebuje mať na svojom počítači nainštalovaný Matlab, jediné čo potrebuje je internetové pripojenie a prehliadač.

V súboroch typu m-súbor je potrebné riadky ukončovať bodko - čiarkou, aby sa predišlo výpisom premenných. Výpis je totiž presmerovaný na klientský www prehliadač (štruktúra html dokumentu by potom bola pozmenená na výpis zdrojového kódu html dokumentu).

3. Identifikácia prostredníctvom Matlabu

Celú stránku som vytvoril pomocou kódu html. Rozdelil som ju na pravú a ľavú časť. V ľavej časti sa nachádza výberové menu, kde si možno vybrať buď to teóriu alebo spracovanie jednotlivých typov prechodových charakteristík. V pravej sa zobrazuje teória alebo prebieha spracovanie prechodovej charakteristiky.

Dôvod, prečo som volil takýto typ www stránky je prozaický. Prezentácia MWS totiž zmení aktuálnu cestu s html súborom na cestu k adresáru, kde je umiestnený, čím linky napísané v skrátenej forme strácajú platnosť. Zdrojový kód úvodnej stránky je:

```
<html lang="style">
<head>
    <title>Bakalársky projekt</title>
    <meta http-equiv="content-type" content="text/html charset=windows-1250">
    <link rel="stylesheet" type="text/css" href="vzor.css">
</head>

<frameset cols="20%,*">
<frame src="Teoria/menu.html" name="menu">
<frame src="Teoria/stranka.html" name="stranka">
</frameset>
</html>
```

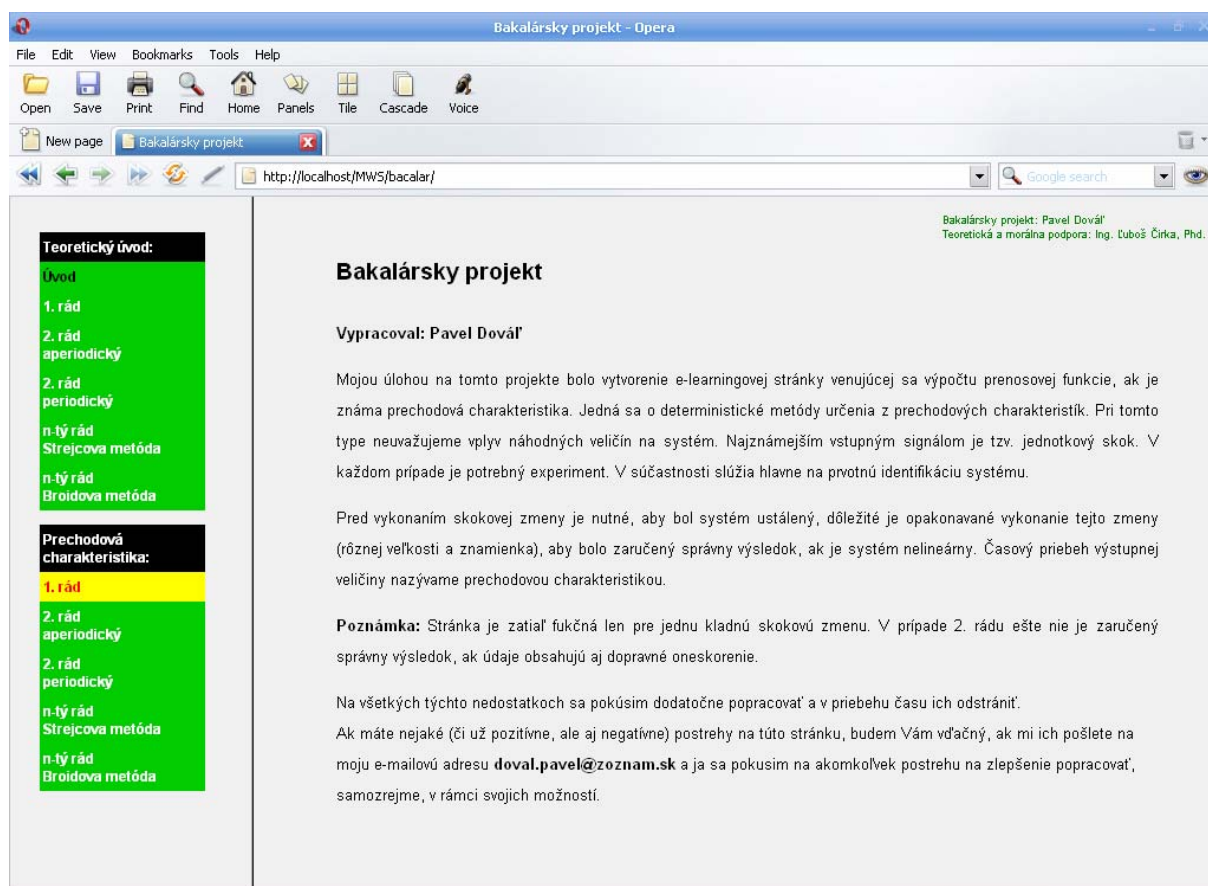
Zdrojový kód č.1

Celkový výsledok možno vidieť na obrázku č.4.

Teoretická časť je kompletne napísaná v kóde html. Pre jej jednoduchosť nebudem na tomto mieste uvádzať ani jej ukážku, ani zdrojové kódy.

Spracovanie prechodových charakteristík bolo zložitejšie a tvorilo podstatnú časť mojej práce, z tohto dôvodu sa budem venovať predovšetkým tejto časti. Tu som použil aj prvky php a samozrejme m-súbory potrebné na MWS prezentácie.

Ako názornú ukážku použijem systém 1. rádu, na ktorom by som chcel ukázať základné činnosti mnou napísanej www stránky, ale i spôsob práce s ňou. Neskôr iba uvediem odlišnosti v spôsobe výpočtu v prípade vyšších rádov.



Obr. č.4: Úvodná stránka.

3.1. Prechodová charakteristika systému 1. rádu

Po kliknutí na daný odkaz v ľavom okne (na obr. č.4 je to žltá vysvietená časť) sa zobrazí html stránka s možnosťou výberu spracovania prechodovej charakteristiky načítaním textového súboru, ktorý obsahuje údaje o čase, skokovej zmene a výstupnej veličine systému, pričom príklad je uvedený na obrázku č.6. Druhým typom je spracovanie jednotlivých charakteristík krivky z vopred vyhodnotenej prechodovej charakteristiky systému, táto časť je znázornená obrázkom č. 5.

3.1.1. Definovanie údajov zistených z prechodovej charakteristiky

Toto je druhý typ spracovania prechodovej charakteristiky, je jednoduchší, ale aj názornejší, preto sa budem venovať najskôr jemu. Po kliknutí na daný odkaz sa zobrazí

stránka, ktorá je napísaná prevažne kódom html, ale obsahuje aj prvky php. Jej zdrojový kód je nasledovný:

```
<html lang="style">
<head>
    <title>Napisanie</title>
    <meta http-equiv="content-type" content="text/html charset=windows-1250">
    <link rel="stylesheet" type="text/css" href="../vyzor.css">
</head>
<body>

<div id="obsah">
<h2>Náhrada modelu systému prenosom 1. rádu<br>
definovaním údajov zistených z PCH[1/2]</h2>
<h3>Zadajte nasledujúce údaje z PCH:</h3>
<?php
$r = mktime;
$u0 = $r; $y0 = $r; $t1 = $r; $t2 = $r;
$ux = $r; $yx = $r; $y1 = $r; $y2 = $r;
?>
<form action="kontrola_1rad.php" method="post">

<table>
<tr><td align="right">u(0):</td><td><INPUT TYPE="text" NAME="u0" MAXLENGTH="10"></td></tr>
<tr><td align="right">u(nekonecno):</td><td><INPUT TYPE="text" NAME="ux" MAXLENGTH="10"></td></tr>
<tr><td align="right">y(0):</td><td><INPUT TYPE="text" NAME="y0" MAXLENGTH="10"></td></tr>
<tr><td align="right">y(nekonecno):</td><td><INPUT TYPE="text" NAME="yx" MAXLENGTH="10"></td></tr>
<tr><td align="right">t1:</td><td><INPUT TYPE="text" NAME="t1" MAXLENGTH="10"></td></tr>
<tr><td align="right">t2:</td><td><INPUT TYPE="text" NAME="t2" MAXLENGTH="10"></td></tr>
<tr><td align="right">y(t1):</td><td><INPUT TYPE="text" NAME="y1" MAXLENGTH="10"></td></tr>
<tr><td align="right">y(t2):</td><td><INPUT TYPE="text" NAME="y2" MAXLENGTH="10"></td></tr>
<tr><td></td><td><INPUT TYPE="submit" VALUE="Ďalej &raquo;"></td></tr>
</table>
</form>
<p><strong>Poznámka:</strong> Na oddelenie desatinného čísla použite bodlu (nie čiarku).</p>
</div>

<hr>
<p id="footer"> Bakalársky projekt: Pavel Dovál<br>
    Teoretická a morálna podpora: Ing. Ľuboš Čírka, Phd.<br></p>

</body>
</html>
```

Zdrojový kód č.2

Bakalársky projekt - Opera

File Edit View Bookmarks Tools Help

Open Save Print Find Home Panels Tile Cascade Voice

New page Bakalársky projekt

http://localhost/MWS/bacalar/ Google search

Bakalársky projekt: Pavel Dovál
Teoretická a morálna podpora: Ing. Ľuboš Čírka, Phd.

Teoretický úvod:

Úvod

1. rád

2. rád aperiodický

2. rád periodický

n-tý rád

Strejcová metóda

n-tý rád

Broidova metóda

Prechodová charakteristika:

1. rád

2. rád aperiodický

2. rád periodický

n-tý rád

Strejcová metóda

n-tý rád

Broidova metóda

Náhrada modelu systému prenosom 1. rádu definovaním údajov zistených z PCH[1/2]

Zadajte nasledujúce údaje z PCH:

u(0):

u(nekonecno):

y(0):

y(nekonecno):

t1:

t2:

y(t1):

y(t2):

Ďalej >

Poznámka: Na oddelenie desatinného čísla použite bodku (nie čiarku).

Obr. č.5: Náhrada systému definovaním údajov zistených z prechodovej charakteristiky.

Výsledný vzhľad je zobrazený na obrázku č.5. Po kliknutí na tlačidlo „Ďalej“ môžu nastať dve situácie. O oboch rozhoduje krátka php stránka, ktorej činnosť nie je príliš viditeľná. Jej úlohou je zistiť, či užívateľ zadal všetky parametre potrebné na výpočet. Ak sa tak nestalo, spustí php stránku, ktorá bude vyžadovať doplnenie potrebných údajov, alebo spustí stránku, ktorá len požiada o potvrdenie užívateľovej požiadavky. Zdrojový kód je nasledovný:

```
<body>
<?php
if ($u0 == $r)
    { $u0 = ''; require 'doplnenie_1rad.php'; }
elseif ($ux == $r)
    { $ux = ''; require 'doplnenie_1rad.php'; }
elseif ($y0 == $r)
    { $y0 = ''; require 'doplnenie_1rad.php'; }
elseif ($yx == $r)
    { $yx = ''; require 'doplnenie_1rad.php'; }
elseif ($t1 == $r)
```

```

        { $t1 = ''; require 'doplnenie_1rad.php'; }
elseif ($t2 == $r)
        { $t2 = ''; require 'doplnenie_1rad.php'; }
elseif ($y1 == $r)
        { $y1 = ''; require 'doplnenie_1rad.php'; }
elseif ($y2 == $r)
        {$y2 = ''; require 'doplnenie_1rad.php';}
else require 'odoslanie_1rad.php';
?>
</body>

```

Zdrojový kód č.3

Pokračovať budem len prípadom správneho zadania (pretože v prípade nesprávneho zadania je zdrojový kód veľmi podobný mnou už uvádzanému kódu č.1). V tomto prípade je uvedenie zdrojového kódu dosť dôležité. Vysvetľuje totiž, ako vlastne dôjde k samotnému odoslaniu údajov MWS. V podstate je to spôsob ekvivalentný kódu č.1, opäť s príkazom `<form action="">`, podstata sa však skrýva pred zrakom užívateľa:

```

<div id="obsah">
<h2>Náhrada modelu systému prenosom 1. rádu<br>
definovaním údajov zistených z PCH[2/2]</h2>
<h3>Vami zadane údaje z PCH sú nasledovné:</h3>

<form action="/cgi-bin/matweb.exe" method="post">
<input type="Hidden" name="mlmfile" value="MWS_1rad_udaje">

        <INPUT TYPE="hidden" NAME="u0" value="<?php echo $u0;?>">
        <INPUT TYPE="hidden" NAME="ux" value="<?php echo $ux;?>">
        <INPUT TYPE="hidden" NAME="y0" value="<?php echo $y0;?>">
        <INPUT TYPE="hidden" NAME="yx" value="<?php echo $yx;?>">
        <INPUT TYPE="hidden" NAME="t1" value="<?php echo $t1;?>">
        <INPUT TYPE="hidden" NAME="t2" value="<?php echo $t2;?>">
        <INPUT TYPE="hidden" NAME="y1" value="<?php echo $y1;?>">
        <INPUT TYPE="hidden" NAME="y2" value="<?php echo $y2;?>">

        <table>
        <tr> <td align="right">u(0):</td> <td>&nbsp;&nbsp;&nbsp;&nbsp;<?php echo $u0;?></td> </tr>
        <tr> <td align="right">u(nekonecno):</td> <td>&nbsp;&nbsp;&nbsp;&nbsp;<?php echo $ux;?></td> </tr>
        <tr> <td align="right">y(0):</td> <td>&nbsp;&nbsp;&nbsp;&nbsp;<?php echo $y0;?></td></tr>
        <tr> <td align="right">y(nekonecno):</td> <td>&nbsp;&nbsp;&nbsp;&nbsp;<?php echo $yx;?></td> </tr>
        <tr> <td align="right">t1:</td> <td>&nbsp;&nbsp;&nbsp;&nbsp;<?php echo $t1;?></td> </tr>
        <tr> <td align="right">t2:</td> <td>&nbsp;&nbsp;&nbsp;&nbsp;<?php echo $t2;?></td> </tr>

```

```

<tr> <td align="right">y(t1):</td> <td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<?php echo $y1;?></td> </tr>
<tr> <td align="right">y(t2):</td> <td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<?php echo $y2;?></td> </tr>
<tr> <td></td> <td><INPUT TYPE="submit" VALUE="Ďalej &raquo;"></td> </tr>
</table>

```

</div>

Zdrojový kód č.4

Ako vidieť tesne pod spomínaným príkazom, ktorý hovorí, že sa má spustiť MWS sa nachádza riadok udávajúci, aký m-súbor má MWS otvoriť. Tým, že je udaný parameter hidden, užívateľ nemôže ani schválne a ani nedopatrením tento názov m-súboru zmeniť.

Po kliknutí na posledné tlačidlo „Ďalej“ dôjde k spracovaniu údajov MWS a vyhodnoteniu. Tejto časti sa budem venovať najpodrobnejšie. A to z dvoch dôvodov. Jednak je to najzložitejšia a najdôležitejšia časť môjho projektu a jednak jeho podstatou:

číslo riadku	samotný kód
1	function retstr = MWS_1rad_udaje(instruct)
2	retstr = char("");
3	conf2
4	%instruct
5	
6	yx = str2num(instruct.yx);
7	y0 = str2num(instruct.y0);
8	ux = str2num(instruct.ux);
9	u0 = str2num(instruct.u0);
10	y1 = str2num(instruct.y1);
11	y2 = str2num(instruct.y2);
12	t1 = str2num(instruct.t1);
13	t2 = str2num(instruct.t2);
14	
15	K = (yx - y0) / (ux - u0);
16	T = (t2 - t1) / log((K - y1) / (K - y2));
17	x = (log((K - y1) / K) / log((K - y2) / K));
18	D = (t2 * x - t1) / (x - 1);
19	if D < 0
20	D = 0;
21	end;
22	
23	open_system('schema2');
24	set_param('schema2/Funkcia','Numerator',num2str(K));
25	set_param('schema2/Skok','Before',instruct.u0);
26	set_param('schema2/Skok','After',instruct.ux);
27	set_param('schema2/Transport','DelayTime',num2str(D));

28	set_param('schema2/Funkcia','Denominator',sprintf('[%d 1]',T));
29	StopTime = t2/0.7 + 3 + D;
30	sim('schema2',StopTime);
31	
32	wscleanup('ml*.jpg',1);
33	
34	fout = figure('visible','off');
35	plot(tout,yout,'r');
36	xlabel('t (s)');
37	ylabel('y');
38	jpgname = sprintf('%s.jpg',instruct.mlid);
39	drawnow;
40	wsprintjpeg(fout,jpgname);
41	close('all');
42	
43	outstruct.K = K;
44	outstruct.T = T;
45	outstruct.D = D;
46	outstruct.suma_s = 'Is not possible to define.';
47	outstruct.graf = jpgname;
48	
49	%outstruct
50	templatefile = which('zaver_1rad.html');
51	retstr = htmlrep(outstruct, templatefile);

Zdrojový kód č.5

- Riadok 3: Udáva cestu, kam má byť vygenerovaný graf uložený. Načítava sa z ďalšieho (pomocného) m-súboru.
- Riadky 6-13: Načítanie jednotlivých charakteristík spracovanej prechodovej charakteristiky do mnou navolených premenných.
- Riadky 15-21: Samotný výpočet zosilnenia, časovej konštanty a dopravného oneskorenia.
- Riadok 23: Otvorenie schémy potrebnej na simuláciu (znázornená je na obrázku č.4)
- Riadky 24-30: Vloženie K, T, a D do schémy a spustenie simulácie (riadok 30)
- Riadok 32: Vymazanie starých grafov.
- Riadky 34-41: Vygenerovanie grafu a zatvorenie simulačnej schémy.
- Riadky 43-47: Vytvorenie výstupných veličín.
- Riadky 50,51: Určenie záverečnej stránky, ktorá zobrazí výsledky výpočtov.

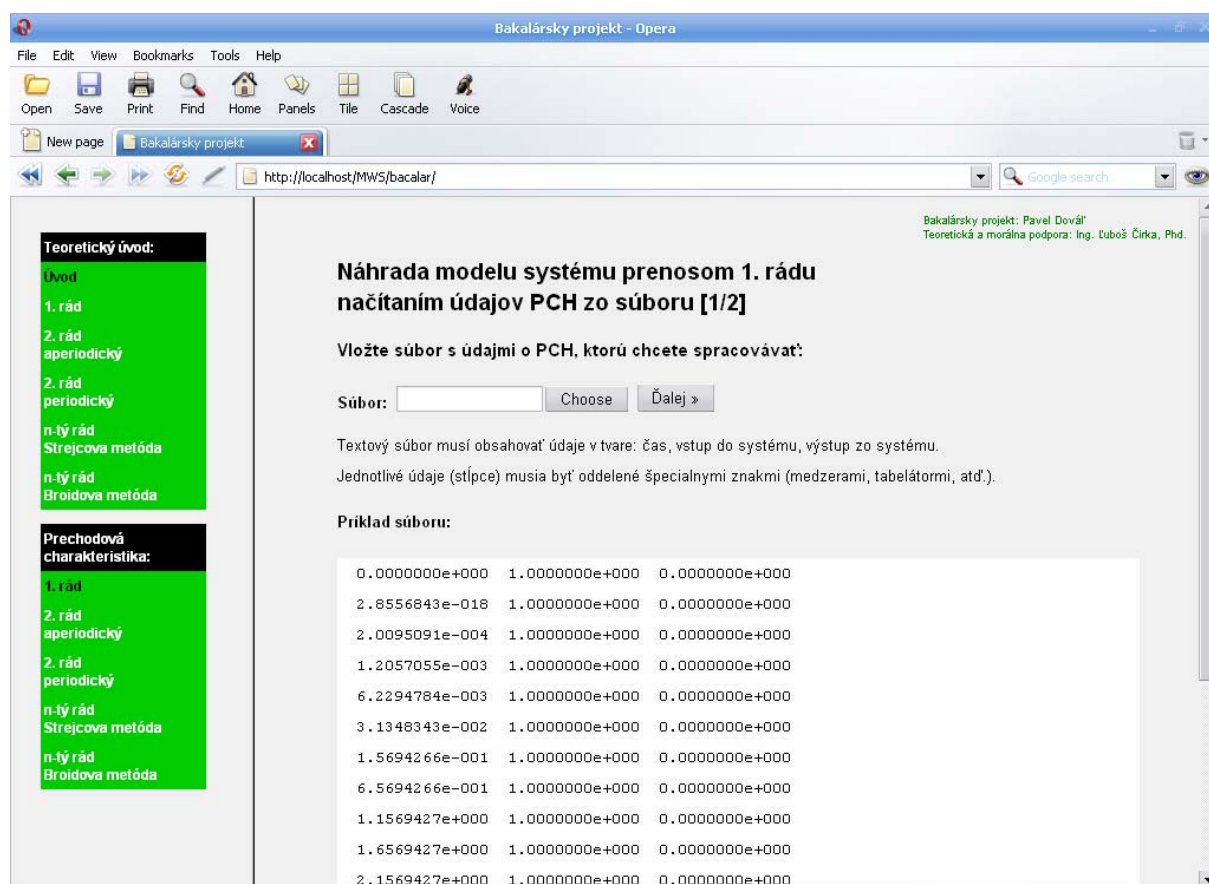
MWS je schopný pritom vložiť výsledné veličiny (určené riadkymi 43 až 47) medzi dva rovnaké znaky: \$. Tým je vlastne ukončený celý proces. Zdrojový kód výslednej stránky je:

```
<body>
<h2>Náhrada modelu systému prenosom 1. rádu</h2>
<p>
Vypočítaná prenosová funkcia má tvar:
<div class="eq"> F(s) = ( <b>$K$</b> / ( <b>$T$</b>*s + 1)) * e<sup>( <b>$D$</b>*s )</sup> </div>
</p>

<p>
Suma štvorcov odchýliek PCH identifikovaného a neznámeho systému: $suma_s$
</p>
```

Zdrojový kód č.6

Jej vzhľad po procese vidieť na obrázku č.9



Obr. č.6: Náhrada modelu systému načítaním údajov zo súboru, ktorého ukážka je v pravej spodnej časti obrázku.

3.1.2. Načítanie údajov prechodovej charakteristiky z textového súboru

Počiatok spracovania je rovnaký. Opäť vidieť prvý odkaz na obrázku č.4. Po kliknutí naň a vybratí načítania údajov z textového súboru sa zobrazí stránka , ktorá je na obrázku č.6. Po vložení textového súboru a jeho odoslání dôjde k jeho spracovaniu www stránkou, obsahujúcou php kód. Je potrebná na uloženie daného textového súboru a vymazanie starších súborov zo servera:

```
<?php
//kopirovanie suboru
$PaperDir = "../Cleandata/";
$name = time();
$file = $_FILES['subor'];
if (@copy($file['tmp_name'], $PaperDir.$name))
    $error_msg = "<p>Súbor sa podarilo úspešne skopírovať na server.</p>";
else
    $error_msg = "<p>Súbor sa nepodarilo skopírovať na server.</p>";
echo $error_msg;

//vymazanie suboru
chdir("../Cleandata");
$Directory = opendir(".");
$filemx = time() - 3600;
while($File = readdir($Directory))
{
    if ($File!="." && $File!="..")
    {
        //echo "&nbsp;&nbsp;&nbsp;$FileList[$f]<br>";
        if((int)$File < $filemx) @unlink($File);
    }
}

$filemx = time() - 3600;
$f < $filemx;
@unlink($f);
chdir("../1_rad/");
?>

<form action="/cgi-bin/matweb.exe" method="post">
<input type="Hidden" name="mlmfile" value="MWS_1rad_nacitanie">
<p>
Ak chcete tento súbor spracovať <b>M</b>atLab <b>W</b>eb <b>S</b>erver-om,
```

kliknite na tlačítko Ďalej.

</p>

<input type="Hidden" name="subor" value="<?php opendir(\$PaperDir); echo \$name; ?>">

<input type="Submit" value="Ďalej »">

</div>

Zdrojový kód č.7

MWS-u sa už len posielajú údaje o jeho umiestnení na serveri. Spôsob je to rovnaký ako v kapitole 3.1.1., presnejšie povedané v zdrojovom kóde č.3. Tento súbor spracúva ďalej MWS:

číslo riadku	samotný kód
1	function retstr = MWS_1rad_nacitanie(instruct)
2	retstr = char("");
3	conf1
4	
5	eval(sprintf('load %s;',instruct.subor));
6	eval(sprintf('mat = X%s;',instruct.subor));
7	
8	t = mat(:,1);
9	u = mat(:,2);
10	y = mat(:,3);
11	ux = max(u);
12	u0 = min(u);
13	if u0 == ux
14	u0 = 0;
15	end;
16	yx = max(y);
17	y0 = min(y);
18	
19	y1h = 0.2 * yx;
20	y2h = 0.8 * yx;
21	%20percenten
22	for yh = 1:length(y),
23	if y(yh) < y1h
24	y1 = y(yh);
25	t1 = t(yh);
26	end;
27	end;
28	%80percent
29	for yhh = 1:length(y),
30	if y(yhh) < y2h
31	y2 = y(yhh);
32	t2 = t(yhh);
33	end;
34	end;
35	
36	K = (yx - y0) / (ux - u0);
37	T = (t2 - t1) / log((K - y1) / (K - y2));
38	x = (log((K - y1)/K) / log((K - y2)/K));

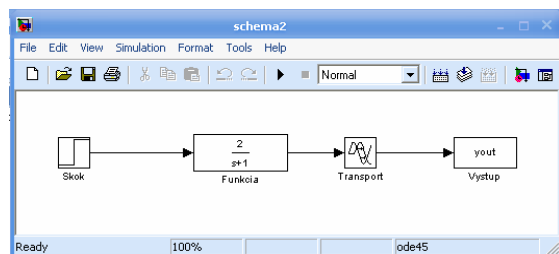
39	D = (t2 * x - t1) / (x - 1);
40	
41	if D < 0
42	D = 0;
43	end;
44	conf2
45	StopTime = max(t);
46	x = K / T + D;
47	mat = mat';
48	save(sprintf('data_%d.mat',x),'mat');
49	
50	open_system('schema');
51	set_param('schema/FromFile','FileName',sprintf('data_%d.mat',x));
52	set_param('schema/Funkcia','Numerator',num2str(K));
53	set_param('schema/Transport','DelayTime',num2str(D));
54	set_param('schema/Funkcia','Denominator',sprintf('1/s',1/T));
55	sim('schema',StopTime);
56	rozdiel = yout - ynac;
57	suma_stvorcov = rozdiel*rozdiel;
58	
59	wscleanup('ml*.jpg',1);
60	wscleanup('data_*.mat',1);
61	
62	fout = figure('visible','off');
63	plot(tout,yout,'r',tout,ynac,'b');
64	legend('PCH identifikovaného systému','PCH neznámeho systému');
65	xlabel('t (s)');
66	ylabel('y');
67	jpgname = sprintf('%s.jpg',instruct.mlid);
68	drawnow;
69	wsprintfjpeg(fout,jpgname);
70	close('all');
71	
72	outstruct.K = K;
73	outstruct.T = T;
74	outstruct.D = D;
75	outstruct.graf = jpgname;
76	outstruct.suma_s = suma_stvorcov;
77	
78	templatefile = which('zaver_1rad.html');
79	retstr = htmlrep(outstruct, templatefile);

Zdrojový kód č.8

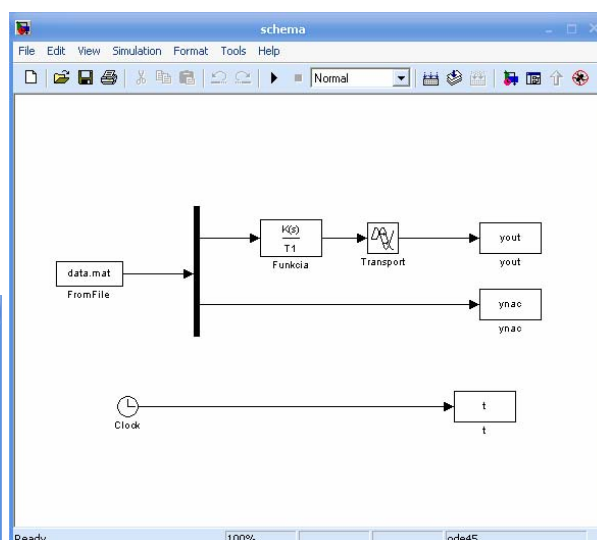
- Riadky 5,6: Uloženie textového súboru do matice nazvanej mat.
- Riadky 8-10: Vybranie jednotlivých stĺpcov a priradenie ich premenným.
- Riadky 11-17: Nájdenie minimálnych a maximálnych hodnôt výstupnej veličiny a skokovej zmeny.
- Riadky 19-34: Nájdenie hodnôt výstupnej veličiny v približne 20 a 80 % maximálnej hodnoty a im prislúchajúce časy.

- Riadky 36-39: Výpočty rovnako ako v kóde 4 riadky 24-30.
- Riadky 47,48: Vytvorenie transponovanej matice a jej následné uloženie s príponou .mat (matica programu Matlab).
- Riadky 50-57: Otvorenie simulačnej schémy, zmena premenných a výpočet sumy štvorcov odchýliek.
- Riadky 59,60: Vymazanie starších súborov s príponami .jpg, .mat.
- Riadky 62-79: Vytvorenie grafu, uloženie a otvorenie záverečnej html stránky.

Ako vidieť, je to skutočne miene zložitejšie. Najväčším problémom ale bolo zaručiť rovnaký krok simulácie pre nami simulovaný proces a načítané údaje. To bolo možné jediný spôsobom. Uložením načítanej matice vo formáte .mat a urobiť z nej vstup do simulačnej schémy. Tým sa zaručil rovnaký krok simulácie (výstupné hodnoty majú vždy rovnaké časové údaje) a tým pádom bolo možné vypočítať sumu štvorcov odchýliek. Celá schéma je znázornená obrázkom č.8.

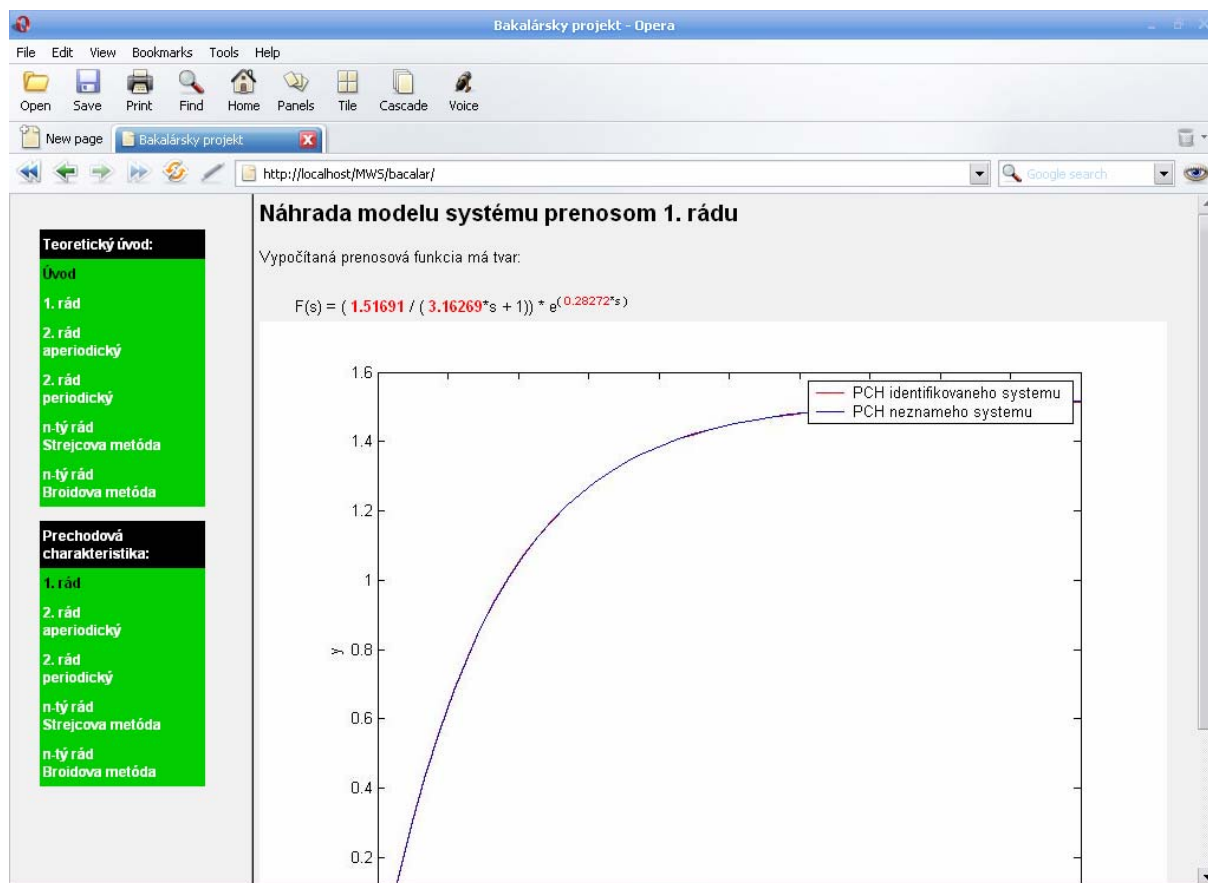


Obr. č.7



Obr. č.8

Tento spôsob výpočtu mal zo všetkých metód najmenšiu sumu štvorcov odchýliek, pretože jediné určenie, ktoré z prechodovej charakteristiky treba urobiť je zistenie minima a maxima prechodovej charakteristiky. K tomu treba určiť približne hodnoty v 20 a 80 % maxima. Tieto dva body som získal z pôvodného, teda klientom uloženého súboru na server.



Obr. č.9: Stránka zobrazujúca výsledky výpočtov vykonaných programom Matlab.

Pri druhom ráde, ani pri vyššom sa už viac nebudem venovať ani predchádzajúcim html stránkam, ani záverečnej html stránke. Je to len ekvivalencia k spomínaným stránkam. O to viac sa budem venovať MWS.

3.2. Prechodová charakteristika aperiodického systému 2. rádu

Ako vyplýva z teórie spomínanej v kapitole 2.1.2., základom celého výpočtu je vlastne inflexný bod. Inflexným bodom prechádza priamka, ktorá určuje čas prietahu a čas nábehu. Čo sa týka zadávania jednotlivých charakteristík krivky z vopred vyhodnotenej prechodovej charakteristiky, je to jednoduchý výpočet. Stačí dosadiť do vzťahov zo spomínanej kapitoly. Väčším problémom je zistenie inflexného bodu, ak prechodovú charakteristiku načítavame z textového súboru.

Tomuto prípadu sa budem venovať podrobnejšie. Uvediem už ale iba časť zo zdrojového kódu:

číslo riadku	samotný kód
1	t = mat(:,1);
2	u = mat(:,2);
3	y = mat(:,3);
4	ux = max(u);
5	u0 = min(u);
6	if u0 == ux
7	u0 = 0;
8	end;
9	y0 = min(y);
10	yx = max(y);
11	K = (yx - y0) / (ux - u0);
12	
13	d1y = y(2:end) - y(1:end-1);
14	d1t = t(2:end) - t(1:end-1);
15	delmat = d1y./d1t;
16	max_dm = max(delmat);
17	indx = find(max_dm==delmat);
18	
19	bod_t1 = t(indx+1);
20	bod_y1 = y(indx+1);
21	bod_t2 = t(indx);
22	bod_y2 = y(indx);
23	
24	a = (bod_y2 - bod_y1) / (bod_t2 - bod_t1);
25	b = bod_y2 - a*bod_t2;
26	Tu = (y0 - b) / a;
27	Tn = (yx - b) / a - Tu;
28	f1kv = Tn / Tu;
29	
30	k = [0.99 0.95 0.9 0.8 0.4 0.2 0.1 0.05]';
31	f2k = [2.705 2.650 2.581 2.441 1.842 1.495 1.292 1.171]';
32	f1k = [9.649 9.653 9.665 9.72 10.91 13.974 20.088 31.737]';
33	indx = find(f1kv<f1k);
34	f2k = f2k(indx(1));
35	k = k(indx(1));
36	
37	T1 = Tn / f2k;
38	T2 = k * T1;
39	T12n = T1 * T2;
40	T12s = T1 + T2;
41	D = 0;
42	
43	conf2
44	
45	StopTime = max(t);
46	
47	x = T12n / T12s + b * a;
48	mat = mat';
49	save(sprintf('data_%d.mat',x),'mat');

Zdrojový kód č.9

- Riadky 1-10: Tak ako v predchádzajúcich prípadoch, načítanie do premenných a určenie minimálnej a maximálnej hodnoty (v ďalších systémoch tento úsek budem vynechávať).
- Riadok 11: Výpočet zosilnenia.
- Riadky 13,14: Výpočet prvej derivácie výstupnej veličiny a času.
- Riadok 15,16: Podiel týchto dvoch matíc a určenie maximálnej hodnoty tohto delenia.
- Riadok 17: Zistenie, na ktorom poradovom čísle sa nachádza toto maximum.
- Riadky 19-22: Zistenie inflexného bodu (lepšie povedané súradnice dvoch bodov priamky prechádzajúcej inflexným bodom – dotyčnice).
- Riadky 24,25: Výpočet parametrov a , b rovnice priamky.
- Riadky 26,27: Výpočet času prietahu a času nábehu.
- Riadok 28: Určenie vzájomného pomeru týchto časov.
- Riadky 30-32: Matematické vyjadrenie grafickej závislosti spomínanej v kapitole 2.1.2
- Riadky 33-35: Určenie ostatných koeficientov z riadkov 30 až 32.
- Riadky 37-40: Výpočty zosilnenia a časovej konštanty.

Dotyčnica k prechodovej charakteristike prechádza inflexným bodom. Rovnicu tejto dotyčnice som zistil pomocou súradníc dvoch bodov, ktoré som určil postupom znázorneným riadkami 19-22 zdrojového kódu č.9. Jedná sa teda o zistenie maxima podielu druhých derivácií výstupnej veličiny a času. Určenie týchto dvoch bodov možno aj iným spôsobom. Je ním druhá derivácia výstupnej veličiny a určenie jej minima. Tým ale zanedbávame variabilný časový krok a určenie inflexného bodu bude nepresné. Ak by sme používali len konštantný časový krok, určenie by bolo možné aj týmto spôsobom. V takom prípade by bol výpočet jednoduchší.

3.3. Prechodová charakteristika periodického systému 2. rádu

Výpočet periodického systému 2. rádu sa mierne podobá systému 1. rádu uvedeného v kapitole 3.1. Ako vidieť na obrázku č.3, aj tu je potrebné určiť súradnice dvoch bodov. Tie sa vzťahujú na maximum prvého kmitu a následné minimum druhého kmitu. Po týchto určeníach je následný výpočet jednoduchý:

číslo riadku	samotný kód
1	d1y = y(2:end) - y(1:end-1);
2	indx = find(d1y<0);
3	t1 = t(indx(1));
4	y1 = y(indx(1));
5	
6	kon = max(indx);
7	zac = indx;
8	submat = mat(zac:kon,1:3);
9	ysm = submat(:,3);
10	tsm = submat(:,1);
11	d1ysm = ysm(2:end) - ysm(1:end-1);
12	indx = find(d1ysm>0);
13	y2 = ysm(indx(1));
14	t2 = tsm(indx(1));
15	
16	M = (y1 - y2) / y1;
17	m = sqrt(3.14^2 + (log(M))^2);
18	r = abs(log(M) / m);
19	P = sqrt(1 - r^2);
20	om = 3.14 / ((t2 - t1) * P);
21	T = 1 / om;
22	K = y1 / (1 + M);
23	D = 0;

Zdrojový kód č.10

Poznámka: Riadky pred riadkom 1 sú podobné ako v predchádzajúcich prípadoch, napr. riadky 1 až 15 zdrojového kódu č.7. Riadky pod riadkom 23 sú ekvivalentom riadkov 44 až 79 rovnakého kódu.

- Riadok 1: Výpočet prvej derivácie prechodovej charakteristiky.
- Riadky 2-4: Nájdenie maxima a priradenie hodnôt času a výstupnej veličiny systému.
- Riadky 6-8: Vytvorenie submatice matice mat, pričom prvá hodnota tejto matice je nami určená hodnota maxima, poslednou hodnotou je posledná hodnota pôvodnej matice .
- Riadky 9,10: Výber potrebných riadkov submatice.
- Riadok 11: Výpočet prvej derivácie prechodovej charakteristiky submatice.
- Riadky 12-14: Určenie minima a priradenie hodnôt času a výstupnej veličiny systému.
- Riadky 16-23: Výpočet časovej konštanty a zosilnenia (sqrt – druhá odmocnina, abs – absolútna hodnota).

Presnosť celého výpočtu závisí od presnosti určenia bodov 1 a 2, teda maxima a minima. Presnosť určenia týchto dvoch údajov je ovplyvnená hlavne veľkosťou časového kroku. Čím je menší, tým presnejšie údaje možno z prechodovej charakteristiky získať. U ostatných rádov to platí rovnako, pri periodickom systéme je tento vplyv najviac viditeľný.

3.4. Prechodová charakteristika systému – Strejcova metóda

Všeobecné grafické znázornenie je na grafe č.2. Teda rovnako ako v prípade 2. aperiodického systému je potrebné určiť inflexný bod.

číslo riadku	samotný kód
1	d1y = y(2:end) - y(1:end-1);
2	d1t = t(2:end) - t(1:end-1);
3	delmat = d1y./d1t;
4	max_dm = max(delmat);
5	indx = find(max_dm==delmat);
6	
7	bod_t1 = t(indx+1);
8	bod_y1 = y(indx+1);
9	bod_t2 = t(indx);
10	bod_y2 = y(indx);
11	
12	a = (bod_y2 - bod_y1) / (bod_t2 - bod_t1);
13	b = bod_y2 - a*bod_t2;
14	Tu = (y0 - b) / a;
15	Tn = (yx - b) / a - Tu;
16	
17	n = [6 5 4 3 2 1]';
18	fn = [0.493 0.410 0.319 0.218 0.104 0.000]';
19	gn = [0.161 0.195 0.224 0.271 0.368 1.000]';
20	%stupen = [n fn gn];
21	
22	f = round(1000 * Tu / Tn)/1000;
23	K = (yx - y0) / (ux - u0);
24	index_f = find(fn<f);
25	fn = fn(index_f(1));
26	gn = gn(index_f(1));
27	n = n(index_f(1));
28	D = (f - fn) * Tn;
29	T = Tn * gn;

Zdrojový kód č.11

- Poznámka: Riadky pred riadkom 1 sú podobné ako v predchádzajúcich prípadoch, napr. riadky 1 až 15 zdrojového kódu č.7. Riadky pod riadkom 29 sú ekvivalentom riadkov 44 až 79 rovnakého kódu.
- Riadky 1-15: Určenie inflexného bodu (podrobne rozpísané v zdrojovom kóde 8, riadky 13 až 27).
- Riadky 17-19: Vytvorenie vektorov n , f_n , g_n . Sú potrebné na ďalšie výpočty.
- Riadok 22: Výpočet f (pomer času nábehu a prietahu), jeho zaokrúhlenie na tri desatinné miesta.
- Riadok 23: Výpočet zosilnenia systému.
- Riadky 24-26: Určenie f_n , g_n a n prislúchajúce k hodnote f .
- Riadky 28,29: Výpočet času časovej konštanty a času nábehu.

Vzhľadom na to, že som použil rovnaký spôsob výpočtu inflexného bodu, kvalita výpočtu je porovnateľná s aperiodickým systémom 2. rádu. Výhodou Strejcovej metódy oproti spomínanej metóde je, že Strejcova metóda počíta prenosové charakteristiky systémov, ktoré sú vyšších rádov než je druhý rád.

3.5. Prechodová charakteristika systému – Broidova metóda

Pri tomto spôsobe už nemá význam kvôli podobnosti so Strejcovou metódou uvádzať zdrojový kód. Hlavný rozdiel Strejcovej a Broidovej metódy je v načítavanej tabuľke. Obe sú uvádzané v kapitole 2.1.4.

Ako hlavný rozdiel by som uviedol sumu štvorcov odchýliek oboch metód. Mierne nižšie boli v prípade Broidovej metódy (aj keď len na rozdiel stotín, v niektorých prípadoch dokonca tisícín), preto ju možno považovať za presnejšiu, i keď mierne komplikovanejšiu.

4. Záver:

Celá mnou napísaná www stránka spracúva prechodové charakteristiky piatimi metódami. Každú metódu dvomi spôsobmi.

Pri načítavaní prechodovej charakteristiky z textového súboru počíta tzv. sumu štvorcov odchýliek (SSO) medzi údajmi, ktoré sú načítavané a údajmi, ktoré získame z prechodovej charakteristiky simuláciou údajov vypočítanej prenosovej funkcie. Táto SSO bola jednoznačne najnižšia pri systéme 1. rádu. Niekedy mali výsledky získané touto metódou nižšiu hodnotu SSO ako systémy vyšších rádov, aj keď prechodové charakteristiky boli získané simuláciou vyšších aperiodických systémov.

Ako najmenej presný výpočet sa ukázal periodický systém 2. rádu, aj keď pri konštantnom a nižšom časovom kroku dosahoval rovnako presné výsledky ako ostatné systémy. Veľkosť tohto kroku možno považovať za určujúcu veličinu vplývajúcu na kvalitu výpočtu prenosovej funkcie.

Táto www stránka je zatiaľ funkčná len na jednorázové a kladné skoky vstupnej veličiny. V priebehu diplomovej práce by som ju chcel doplniť o viacnásobné skokové zmeny vstupnej veličiny a umožniť aj výpočty záporných skokových zmien. Pri výpočtoch systémov 2. rádu mnou použité metódy nepočítajú s dopravným oneskorením, čo je vlastne ďalšou vecou, o ktorú by som mal záujem obohatiť svoju www stránku.

Zo svojho uhla pohľadu považujem za najzložitejšie systémy, ktoré sú založené na určení časov nábehu a prietahu. Určenie inflexného bodu a spomínaných dvoch časov bolo asi najzložitejšou časťou. Najzložitejšie pritom bolo si predstaviť, ako vlastne vypočítať tieto časy. Matematická realizácia nebola až taká zložitá. Najzložitejšie na naprogramovanie je vygenerovanie grafu a jeho uloženie, ako aj uloženie správnej cesty k nemu, aby tento graf bola schopná výsledná html stránka a zobrazíť.

5. Literatúra:

- [1] Matlab Web Server. dostupné na www.mathworks.com
- [2] M. Fikar a kol. Metódy počítačového spracovania údajov. Vydavateľstvo STU v Bratislave, 1999.
- [3] D.Reschke. Matlab Web Server ve výuce. Diplomová práce. ČVUT Praha, 2002.

6. Príloha

CD médium, ktoré obsahuje:

- pdf verziu tejto práce
- adresár obsahujúci všetky zdrojové kódy
- adresár obsahujúci štyri príklady súborov určených na načítanie údajov z textového poľa