

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ
TECHNOLÓGIE**

Flash prezentácie pre predmet identifikácia

Bakalárska práca

FCHPT-5415-70048

Študijný program: automatizácia, informatizácia a manažment v chémii a potravinárstve

Číslo študijného odboru: 2621

Názov študijného odboru: 5.2.14 automatizácia, 5.2.52 priemyselné inžinierstvo

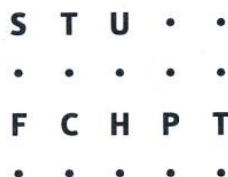
Školiace pracovisko: oddelenie informatizácie a riadenia procesov

Vedúci záverečnej práce/školiťel': Ing. Ľuboš Čirka, PhD.

Konzultant: Ing. Martin Kalúz

Bratislava 2012

Viktor Kukla



ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Viktor Kukla**
ID študenta: 70048
Študijný program: automatizácia, informatizácia a manažment v chémii a potravinárstve
Kombinácia študijných odborov: 5.2.14 automatizácia, 5.2.52 priemyselné inžinierstvo
Vedúci práce: Ing. Ľuboš Čírka, PhD.
Konzultant: Ing. Martin Kalúz

Názov práce: **Flash prezentácie pre predmet Identifikácia**

Špecifikácia zadania:

Cieľom práce je vytvoriť prezentácie v Adobe Flash, ktoré budú použité ako doplnkové výukové materiály v predmete Identifikácia.

Úlohy:

- Naštudovať problematiku experimentálnej identifikácie.
- Naštudovať program Adobe Flash a skriptovací jazyk ActionScript.
- Programovo realizovať prezentácie vo Flash a ActionScript.
- Implementovať vytvorené prezentácie do LMS Moodle a overiť ich funkčnosť.

Rozsah práce: 30

Zoznam odbornej literatúry:

1. MIKLEŠ, J. – FIKAR, M. *Identifikácia systémov*. Bratislava: STU v Bratislave, 1999. 112 s. ISBN 80-227-1177-2.
2. MIKLEŠ, J. – FIKAR, M. *Modelovanie, identifikácia a riadenie procesov I : Modely a dynamické charakteristiky spojitých procesov*. Bratislava: STU v Bratislave, 1999. 192 s. ISBN 80-227-1289-2.
3. KERMAN, P. *ActionScript ve Flashi : Podrobná příručka*. Praha: Computer Press, 2002. 507 s. ISBN 80-7226-615-2.
4. KELEMEN, S. – STUHLÍKOVÁ, Ľ. Adobe Flash ako efektívny nástroj popularizácie vedy a techniky. *Posterius : Internetový časopis august*. s. 11370. ISSN 1338-0087.
5. ADOBE CREATIVE TEAM. *Adobe Flash CS3 : Oficiální výukový kurz*. Brno: Computer Press, 2008. 341 s. ISBN 978-80-251-2109-2.

Riešenie zadania práce od: 13. 02. 2012

Dátum odovzdania práce: 19. 05. 2012

L. S.



Viktor Kukla

študent

prof. Ing. Miroslav Fikar, DrSc.

vedúci pracoviska

prof. Ing. Miroslav Fikar, DrSc.

garant študijného programu

Čestné prehlásenie a poďakovanie

Čestne prehlasujem, že som prácu vypracoval samostatne, podľa pokynov vedúceho práce a s použitím zdrojov uvedených v zozname literatúry. Na tomto mieste si ďalej dovoľujem poďakovať školiteľovi Ing. Ľubošovi Čirkovi, PhD. a Ing. Martinovi Kalúzovi za cenné rady, odborné vedenie a pomoc pri vypracovaní tohto projektu.

V Bratislave, máj 2012

Viktor Kukla.

Abstrakt

Bakalárska práca popisuje prezentácie vytvorené v programe Adobe Flash, ktoré sa použijú ako doplnkové výukové materiály pre študijný predmet Identifikácia. V práci je vo všeobecnosti popísaný program Adobe Flash, jeho špecifiká, vlastnosti, jeho porovnanie s inými alternatívami, pomocou ktorých by bolo podobne možné vytvoriť cielené prezentácie. Najväčšia pozornosť je venovaná experimentálnej identifikácii systémov. Práca popisuje návrh a tvorbu aplikácie a dokumentuje funkcie a jej súčasti.

Kľúčové slová: identifikácia, Flash, prezentácia

Abstract

The thesis describes the presentations created in Adobe Flash, that will be used as supporting teaching material in the course Identification. Adobe Flash is described in general with emphasis on its specifications, properties and comparison with other possible alternatives allowing creation of presentations with similar content. Particular emphasis is laid on experimental identification of systems. Thesis describes design and development of the application and documents its functions and components.

Keywords: identification, Flash, presentation

Obsah

ZOZNAM OBRÁZKOV.....	8
ZOZNAM TABULIEK.....	9
ZOZNAM POUŽITÝCH SKRATIEK.....	10
1 ÚVOD.....	11
1.1 CIEĽ PRÁCE.....	11
2 TEORETICKÝ ÚVOD.....	13
2.1 ADOBE FLASH	13
2.1.1 ActionScript	13
2.1.2 Typy dát	13
2.2 ALTERNATÍVNE TECHNOLOGIE	14
2.2.1 HTML	14
2.2.2 AJAX	15
2.2.3 Java Applet.....	15
2.2.4 Microsoft Silverlight.....	15
2.2.5 Porovnanie spomínaných technológií	15
2.3 IDENTIFIKÁCIA SYSTÉMOV.....	16
2.3.1 Postup pri odhade modelu.....	16
2.3.2 Spracovanie prechodových charakteristík	17
2.3.3 Systém 1. rádu.....	18
2.3.4 Kmitavý systém 2. rádu	20
2.3.5 Systém vyššieho rádu.....	21
3 PRAKTICKÁ ČASŤ	25
3.1 HLAVNÁ FUNKCIONALITA A KRITÉRIÁ	25
3.2 PRACOVNÉ PROSTREDIE ADOBE FLASH	25
3.2.1 Hlavná scéna, objekty, časová os.....	25
3.2.2 Osi x a y hlavnej scény.....	27
3.2.3 Časová os	28
3.3 PREZENTÁCIA VO FLASHI	29
3.3.1 Vykreslenie prechodovej charakteristiky.....	29
3.3.2 Priamka $y(\infty)$	31
3.3.3 Dotyčnica k inflexnému bodu	32
3.3.4 Priamka dopravného oneskorenia	33
3.3.5 Priesečníky priamok $y(0)$ a $y(\infty)$ s vykreslenou dotyčnicou	34
3.3.6 Vypísanie výsledných hodnôt a identifikovaných prenosov	35
4 ZÁVER.....	37
ZOZNAM POUŽITEJ LITERATÚRY	38

Zoznam obrázkov

Obr. 1: PCH systému 1. rádu

Obr. 2: PCH periodického systému 2. rádu

Obr. 3: PCH systému vyššieho rádu

Obr. 4: Hlavná scéna

Obr. 5: Knižnica objektov

Obr. 6: Panel nástrojov

Obr. 7: Osi x a y hlavnej scény v Adobe Flash

Obr. 8: Časová os

Obr. 9: Funkcia pre načítanie XML dokumentu

Obr. 10: Funkcia na vypočítanie minima

Obr. 11: Funkcia na vypočítanie maxima

Obr. 12: Použitie funkcií na nájdenie maxima a minima pre polia hodnôt x a y

Obr. 13: Výpočty dielikov pre stupnice x a y

Obr. 14: Vzorová prechodová charakteristika vyššieho rádu

Obr. 15: Vykreslenie priamky $y(\infty)$

Obr. 16: Funkcia na vykreslenie priamky $y(\infty)$

Obr. 17: Vykreslenie dotyčnice k inflexnému bodu

Obr. 18: Funkcia na nájdenie hodnoty dopravného oneskorenia

Obr. 19: Vykreslenie dopravného oneskorenia

Obr. 20: Vykreslenie čiary priesečníka priamky $y(\infty)$ a dotyčnice k inflexnému bodu

Obr. 21: Vykreslenie čiary priesečníka priamky $y(0)$ a dotyčnice k inflexnému bodu

Obr. 22: Vypísanie výsledných hodnôt

Obr. 23: Vypísanie výsledného prenosu

Zoznam tabuliek

Tabuľka 1: Hodnoty parametrov n , $f(n)$, $g(n)$ pri identifikácii Strejcovou metódou

Tabuľka 2: Hodnoty parametrov n , $f(n)$, $g(n)$ pri identifikácii Broidovou metódou

Zoznam použitých skratiek

- HTML – HyperText Markup Language (hypertextový značkový jazyk)
- XML – eXtensible Markup Language (rozšíriteľný značkový jazyk)
- AJAX – Asynchronous JavaScript and XML (skupina technológií pre vývoj interaktívnych webových aplikácií)
- MS – Microsoft (softvérová firma)
- RIA – Rich Internet Application (plnohodnotné internetové aplikácie)

1 Úvod

Štúdium a vzdelávanie na vysokých školách sa v súčasnosti netýka iba kníh, prednášok alebo skript, ale aj hľadania informácií na internete, elektronických výukových kurzov či video prednášok. Študent má teda širokú škálu možností učiť sa ako v laviciach a aulách v priamom kontakte s učiteľmi, tak aj mimo školy čo napomáha rozširovaniu vedomostí a zručností v odbore, v ktorom študuje.

Naša univerzita takisto pomáha rozvíjať podobné spôsoby štúdia a to najmä vďaka online portálu pre vytváranie a správu vzdelávacích kurzov na internete a web stránkach s názvom Moodle [1]. Moodle je modulárny softvér patriaci do triedy LMS systémov (Learning Management System). Na našej fakulte je poskytovaný takpovediac každému študentovi, pre rôzne predmety je tu možnosť stiahnuť si alebo prezerať samotné skriptá a učebnice ku kurzom, obsahuje pomocné vzdelávacie aplikácie a prezentácie pre lepšie pochopenie učiva a zahŕňa mnohé iné možnosti na zlepšenie kvality štúdia na tejto vysokej škole.

Aplikácia, ktorú v tejto práci budem opisovať by mala byť teda ďalšou pomôckou pre študentov, ktorí majú vo svojom rozvrhu predmety súvisiace s experimentálnou identifikáciou. Jej kvalitu a hodnotu sa snažím zaručiť používaním kvalitných a overených zdrojov a publikácií.

Existuje veľké množstvo technológií vhodných pre vytváranie takýchto webových aplikácií. V dnešnej dobe najrozšírenejšími sú značkovacie jazyky ako HTML a XML, skriptovacie jazyky ako JavaScript, ASP a PHP, ale aj technológie a platformy pre tvorbu plnohodnotných internetových aplikácií, akými sú Java, Adobe Flash a MS Silverlight. Každý z nich má svoje špecifiká, výhody a nevýhody. V mojej práci sa však zameriavam iba na platformu Adobe Flash a jej súčasť ActionScript.

1.1 Cieľ práce

Cieľom práce je vytvoriť prezentácie v Adobe Flash, ktoré budú použité ako doplnkový výukový materiál v predmete Identifikácia. Oboznámiť sa so základnými postupmi pri experimentálnej identifikácii systémov, naštudovať základy práce v programe Adobe Flash CS5 Professional a základy programovania v skriptovacom jazyku ActionScript. Prezentácia je zverejnená v e-learningovom portáli Moodle pre predmet Identifikácia. Je zameraná na identifikáciu systémov prvého a vyšších rádov z dát aperiodických

prechodových charakteristík načítaných z XML dokumentov. Názorne prezentuje známe spôsoby experimentálnej identifikácie a všetky kroky potrebné na vytvorenie prenosových funkcií opisujúcich dynamiku systémov.

2 Teoretický úvod

2.1 Adobe Flash [2]

Adobe Flash (predtým Macromedia Flash) je multimediálna platforma, ktorá slúži na vytváranie animácií, videí a interaktívnych prvkov webových stránok. Flash sa často používa pre reklamy, hry a Flash animácie pre online vysielanie. Jeho obsah môže byť zobrazený na rôznych počítačových systémoch a zariadeniach s použitím prehrávača Adobe Flash Player , ktorý je zdarma k dispozícii pre bežné webové prehliadače, niektoré mobilné telefóny a niekoľko ďalších elektronických zariadení.

2.1.1 ActionScript [3]

Flash obsahuje objektovo orientovaný jazyk s názvom ActionScript. ActionScript sa používa predovšetkým na vývoj webových aplikácií a softvéru zameraného na Adobe Flash Player platformu, ktorý sa používa na webových stránkach vo forme vložených súborov SWF.

Verzie ActionScriptu:

ActionScript 1.0 - prvá a najjednoduchšia forma, používaná aj dnes v niektorých verziách prehrávača Flash Lite Player

ActionScript 2.0 - výrazne obsiahlejší ako verzia 1.0, vhodný pre použitie vo výpočtovo nenáročných aplikáciách a pre tvorbu vzhľadovo orientovaného obsahu

ActionScript 3.0 - oproti verziám 1.0 a 2.0 je plne vyhovujúci špecifikáciám ECMAScript, rozšírený o množstvo funkcií, hlavne na lepšiu prácu s XML, obrazovými elementmi a modelom udalostí

2.1.2 Typy dát [2]

ActionScript sa prevažne skladá zo "základných" alebo "jednoduchých" dátových typov, ktoré sa používajú na vytvorenie iných dátových typov.

Niektoré dátové typy na najvyššej úrovni používané v ActionScript 2.0

- **String** – súbor znakov zoradených do reťazca
- **Number** – akákoľvek číselná hodnota
- **Boolean** – jednoduché binárne ukladanie, ktoré môže nadobúdať iba hodnoty "true" alebo "false"
- **Object** – objekt je dátový typ, z ktorého vychádzajú všetky komplexné dátové typy

Vybrané komplexné dátové typy používané v ActionScript 2.0

K dispozícii sú ďalšie "komplexné" dátové typy. Tieto sú viac náročné na procesor a pamäť a skladajú sa z mnohých "jednoduchých" dátových typov.

- **MovieClip** – umožňuje jednoduché použitie viditeľných objektov
- **TextField** – jednoduché statické, dynamické alebo vstupné textové pole
- **Button** – jednoduché tlačidlo, použiteľné na 4 stavy: Up („hore“, bez akejkoľvek interakcie s myšou), Over („cez“, myš nad tlačidlom), Down („dole“, stlačené myšou) a Hit („úder“, definuje oblasť interakcie tlačidla s myšou)
- **Date** – umožňuje prístup k informáciám o konkrétnom okamihu
- **Array** – pole, umožňuje lineárne ukladanie dát
- **XML** – objekt XML
- **XMLNode** – uzol XML
- **LoadVars** - umožňuje ukladanie a odosielanie HTTP POST a HTTP GET

Aplikácia ktorú som vytvoril bola napísaná programovacím jazykom ActionScript vo verzii 2.0.

2.2 Alternatívne technológie [4]

Na vytvorenie podobnej aplikácie existujú aj rôzne iné technológie a programovacie jazyky ako je ActionScript. Na porovnanie som si však vybral niektoré z webových technológií a technológií RIA (plnohodnotné webové aplikácie).

2.2.1 HTML

HTML (HyperText Markup Language) je hypertextový značkovací jazyk, ktorý sa používa na vytváranie statického obsahu webových stránok. Jazyk HTML umožňuje vytvárať dokumenty obsahujúce text, hypertextové odkazy, multimediálny a iný obsah, formuláre, skripty a metainformácie prehliadateľné v tzv. webovom prehliadači.

Jazyk HTML je textový, umožňuje čítanie a upravovanie priamo v textovom editore. Existujú aj špecializované programy na grafické editovanie stránok HTML, napríklad Macromedia Dreamweaver alebo Microsoft FrontPage.

2.2.2 AJAX

AJAX (Asynchronous JavaScript and XML) je súhrnné označenie pre technológie vývoja interaktívnych webových aplikácií, ktoré umožňujú meniť obsah stránok bez potreby ich kompletného opätovného načítania zo servera. V porovnaní s klasickými webovými aplikáciami môžu AJAX-ové aplikácie pri vhodnom návrhu poskytovať používateľsky komfortnejšie prostredie, vyžadujú však použitie moderných webových prehliadačov.

2.2.3 Java Applet

Java Applet je program vytvorený v jazyku Java a vložený na internetovú stránku. Využíva všetky vlastnosti samotného programovacieho jazyka Java, ako napr. vysoká operačná rýchlosť, široká podpora komunikačných technológií, grafické knižnice a pod.

Java Applety majú široké využitie v oblasti on-line prostriedkov pre podporu vzdelávania a často sa používajú aj ako grafické užívateľské rozhrania u technológií prepojených so serverom.

2.2.4 Microsoft Silverlight

Microsoft Silverlight je aplikačná platforma pre tvorbu plnohodnotných internetových aplikácií, podobná platforme Adobe Flash. MS Silverlight ponúka možnosti tvorby interaktívnych aplikácií s podporou multimediálneho obsahu, tvorby animácií, práce s dátami a pod.

MS Silverlight sa dá zaradiť medzi tri najpoužívanejšie technológie RIA. Na rozdiel od technológií Java a Adobe Flash, MS Silverlight nie je multi-platformová technológia, keďže nemá podporu pre operačné systémy unixového typu.

2.2.5 Porovnanie spomínaných technológií

HTML je veľmi vhodná technológia na použitie práve v spojení s AJAX. Nevyžaduje inštaláciu doplnkového softvéru, funguje priamo v internetovom prehliadači. Nevýhodou HTML je slabá podpora multimediálneho obsahu (to platí pre verziu HTML 4 a staršie).

AJAX sa v porovnaní s ostatnými ukázala ako najbezpečnejšia a univerzálna technológia. Všetky poskytované technológie AJAX sú voľne dostupné a nepotrebujú inštaláciu doplnkového softvéru. Technológia však nemá veľkú flexibilitu kvôli naviazaniu na internetový prehliadač.

Technológia Java poskytuje teda dobré možnosti pre tvorbu RIA hlavne z pohľadu bezpečnosti a vývojárskych aspektov. Vo výkone mierne zaostáva a neposkytuje, z hľadiska univerzálnosti, také možnosti ako napr. Adobe Flash, AJAX a HTML.

Adobe Flash teda vyniká hlavne z pohľadu tvorby aplikácií vysokou podporou multimediálneho obsahu, poskytuje plnú podporu 64-bitovej architektúry a u mobilných zariadení je podpora platformy Flash značne širšia. Teda aj kvôli týmto dôvodom je podľa mňa lepším výberom spomedzi možných alternatív.

2.3 Identifikácia systémov [6]

Pre opis procesov chemickej technológie sa používajú rôzne typy dynamických modelov. Obvykle sa vytvárajú na základe materiálových a entalpických bilancií v tvare nelineárnych diferenciálnych rovníc. Aby riadenie malo význam najčastejšie sa však využívajú modely opísané lineárnymi diferenciálnymi či diferenčnými rovnicami. Parametre modelov berieme za známe a konštantné a získavame ich experimentálnym pozorovaním správania sa procesov a spracovaním odpovede na vhodne zvolené vstupné signály. V tomto prípade sa dá hovoriť o odhade parametrov, resp. o identifikácii systémov.

Identifikácia opisuje hľadaný proces reláciami medzi vstupmi a výstupmi procesu pričom pomocou nej sa zisťuje jeho matematický model.

2.3.1 Postup pri odhade modelu

Všeobecný opis odhadu modelu procesu zahŕňa viacero krokov:

Určenie štruktúry modelu

Pri odhade štruktúry sa často využívajú empirické skúsenosti o procese. Je potrebné vybrať typ modelu (napr. nelineárny a lineárny) a takisto aj jeho zložitosť. Kritériá, ktorými sa dá pri výbere riadiť, je viac: kvalita, flexibilita, prípadne cena modelu.

Odhad parametrov

Pre odhad parametrov modelu bolo vypracovaných veľa postupov. Záleží obvykle na type a charakteristike vstupného signálu ako aj na type modelu, ktorý chceme získať.

Verifikácia modelu

Vhodný model by mal súhlasiť s nameranými údajmi, mal by dostatočne presne opisovať proces a vyhovovať účelom, pre ktoré bol získaný. Ďalej môžeme skúmať, či odhadnuté parametre sú v hraniciach daných fyzikálnou podstatou. Je možné tiež skúsiť model redukovať a porovnať, či pôvodný model nie je príliš komplikovaný.

Klasifikácia identifikačných metód

Existuje niekoľko možností ako klasifikovať identifikačné metódy. Dajú sa uviesť jednotlivé triedy.

- A. Identifikácia z aktívneho alebo pasívneho experimentu. Použitie jednej z týchto metód je spravidla dané možnosťami danej technológie a náročnosťou experimentu, ktorý je možné technologickým zariadením vykonať. To znamená, že je možné buď generovať na vstupoch špeciálne signály, alebo identifikovať len v bežnej prevádzke.
- B. Z hľadiska matematického aparátu, ktorý je potrebný pri spracovaní nameraných údajov, je možné rozlíšiť nasledovné metódy:
 - deterministické,
 - stochastické.

Pod *deterministickými* metódami sa myslia metódy, pri ktorých presne poznáme vstupy aj výstupy z procesu a neuvažujeme pri výpočtoch žiadne náhodné vplyvy.

Pod *stochastickým* prístupom sa dá rozumieť najčastejšie metóda najmenších štvorcov (a jej modifikácie), metóda maximálnej vierohodnosti a Bayesov prístup. Každá zo štatistických metód predpokladá určité vlastnosti náhodných porúch a každá vyžaduje inú mieru znalostí o ich vlastnostiach (najmenej náročná je z tohto hľadiska MN©).

- C. Podľa spôsobu spracovania nameraných údajov možno metódy deliť na:

- metódy pre jednorazové (dávkové) spracovanie,
- metódy priebežné (rekurzívne).

Metódy pre jednorazové spracovanie ďalej delíme na manuálne a počítačové. Metódy pre manuálne spracovanie nie sú vhodné pre počítačový výpočet, alebo sú ťažko algoritmizovateľné, pretože závisia od subjektívneho hodnotenia nejakého ukazovateľa (napr. výška inflexného bodu na nameranej prechodovej charakteristike). Počítačové jednorazové metódy spracúvajú súbor nameraných údajov nie najlepšou metódou pre manuálny výpočet. Niekedy sa tieto metódy nazývajú off-line.

Priebežné metódy spracúvajú namerané údaje postupne, identifikované parametre sa spresňujú na základe každého ďalšieho merania. Sú používané rekurentné vzťahy - preto rekurzívne metódy. Použitie týchto metód je vhodné pri on-line nasadení riadiacich počítačov, sú základom pre adaptívne riadenie.

2.3.2 Spracovanie prechodových charakteristík

Spracovanie prechodových charakteristík radíme medzi deterministické metódy, pretože vstupný signál je deterministický a ani u procesu neuvažujeme náhodné vplyvy. Tieto

metódy slúžia na prvotné odhady procesu a poskytujú informácie o približnej hodnote zosilnenia, dominantnej časovej konštanty a oneskorenia.

Používaným vstupným signálom je skoková zmena jednej zo vstupných veličín pri zachovaní ostatných vstupných veličín konštantných. Pred uskutočnením skokovej zmeny je dôležité zabezpečiť, aby bol skúmaný systém v ustálenom stave. Časový priebeh výstupnej veličiny je reálna prechodová charakteristika. Pre ďalšie spracovanie je ju obvykle vhodné normalizovať tak, aby bol jej počiatok v nule.

Keďže skúmaný systém môže vo všeobecnosti byť nelineárny, je potrebné vykonať niekoľko skokových zmien rozličných veľkostí a znamienok. Pre stanovenie výslednej prechodovej charakteristiky sa používa vyhodnocovací vzorec určený z podmienky minima kvadratických chýb v tvare

$$\hat{y}_i = \sum_{k=1}^N \Delta u_k y_{ik} / \sum_{k=1}^N (\Delta u_k)^2 \quad (1)$$

kde i - i -ty bod prechodovej charakteristiky
 k - k -te meranie, $k=1, \dots, N$,
 Δu_k - skoková zmena vstupu pri k -tom meraní,
 y_{ik} - hodnota výstupu pri k -tom meraní v i -tom intervale,
 $\hat{y}_i$ - výsledná hodnota PCH v čase i .

2.3.3 Systém 1. rádu

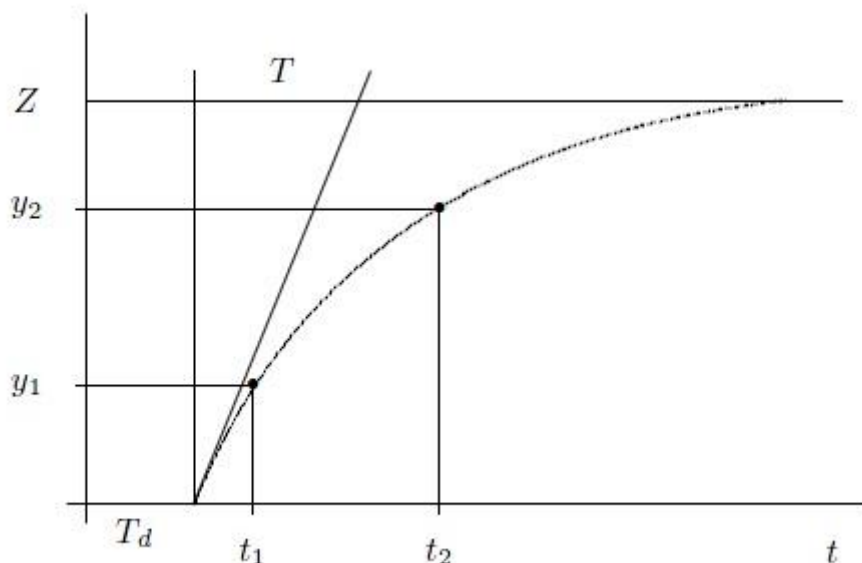
Uvažujme náhradu modelu systému prenosom prvého rádu

$$G(s) = \frac{Z}{Ts+1} e^{-T_d s} \quad (2)$$

kde Z - zosilnenie,
 T - časová konštanta,
 T_d - dopravné oneskorenie systému, ktoré potrebujeme určiť.

Prechodová funkcia pre tento prenos sa dá získať pomocou spätnej Laplaceovej transformácie výstupu a je daná vzťahom

$$y(t) = \begin{cases} 0, & t < T_d \\ Z \left(1 - e^{-\frac{t-T_d}{T}} \right), & t \geq T_d \end{cases} \quad (3)$$



Obr.1: PCH systému 1. rádu

Predpokladajme normalizovanú prechodovú charakteristiku. Zosilnenie systému je dané ako hodnota prechodovej charakteristiky v nekonečne $Z = y(\infty)$, ak skok na vstupe má jednotkovú veľkosť (Obr. 1). Ak poznáme dva body t_1, y_1 a t_2, y_2 na prechodovej charakteristike z rovnice (3) potom vyplýva

$$y_1 = Z \left(1 - e^{-\frac{t_1 - T_d}{T}} \right) \quad (4)$$

$$y_2 = Z \left(1 - e^{-\frac{t_2 - T_d}{T}} \right) \quad (5)$$

Logaritmovaním týchto rovníc a ich následnou úpravou dostaneme výsledné vzťahy

$$T = \frac{t_2 - t_1}{\ln \frac{Z - y_1}{Z - y_2}} \quad (6)$$

$$T_d = \frac{t_2 x - t_1}{x - 1}, x = \frac{\ln \frac{Z - y_1}{Z}}{\ln \frac{Z - y_2}{Z}} \quad (7)$$

Hodnota T sa dá tiež odčítať z prechodovej charakteristiky, ako je to naznačené na obrázku 1, prípadne ako čas, za ktorý dosiahne výstupná veličina 63% svojej ustálenej hodnoty.

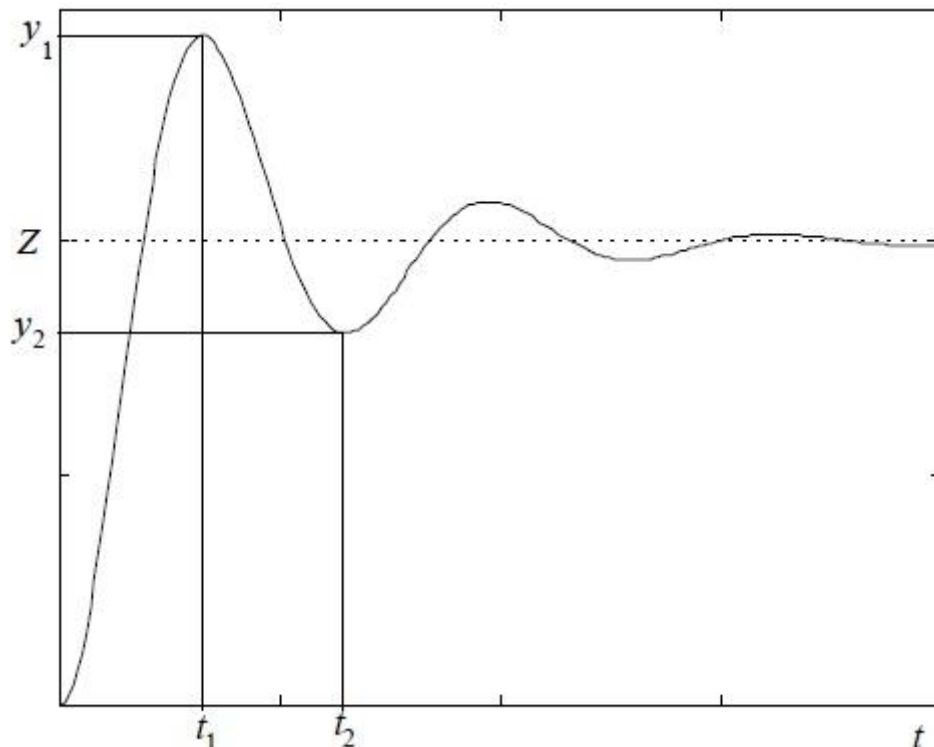
2.3.4 Kmitavý systém 2. rádu

Ak prechodová charakteristika skúmaného procesu má kmitavý priebeh (Obr. 2), je možné ho aproximovať vlastným systémom druhého rádu s prenosom v tvare

$$G(s) = \frac{Z}{T^2 s^2 + 2T\xi s + 1} = \frac{Z\omega_0^2}{s^2 + 2\omega_0\xi s + \omega_0^2} \quad (8)$$

Úlohou je pre dané body $[t_1; y_1]$, $[t_2; y_2]$ a ustálenú hodnotu $y(\infty)$ určiť hodnoty zosilnenia Z , frekvencie ω_0 , časovej konštanty $T = 1/\omega_0$ a tlmenia ξ . Pri odvodení využijeme fakt, že derivácia prechodovej charakteristiky v bodoch t_n (lokálne extrémny) je nulová. Zosilnenie systému je dané podobne ako v predchádzajúcich prípadoch ako hodnota prechodovej charakteristiky v nekonečne $Z = y(\infty)$. Prechodová charakteristika je v tvare

$$y(t) = Z \left[1 - \frac{1}{P} e^{-\xi\omega_0 t} \sin(\omega_0 P t + \varphi) \right], \varphi = \arccos \xi, P = \sqrt{1 - \xi^2} \quad (9)$$



Obr. 2: PCH periodického systému 2. rádu

Derivovaním $y(t)$ podľa času dostaneme

$$\dot{y}(t) = Z \frac{\omega_0}{P} e^{-\xi \omega_0 t} [\xi \sin(\omega_0 P t + \varphi) - P \cos(\omega_0 P t + \varphi)] = Z \frac{\omega_0}{P} e^{-\xi \omega_0 t} \sin(\omega_0 P t) \quad (10)$$

kde bol využitý súčtový vzorec $\sin(a - \varphi)$. V bodoch lokálnych extrémov musí platiť

$$\dot{y}(t_n) = 0 \Leftrightarrow \sin(\omega_0 P t_n) = 0 \Rightarrow t_n = \frac{n\pi}{\omega_0 P} \quad (11)$$

Dosadením t_n do rovnice výstupu (9) dostaneme

$$\begin{aligned} y(t_n) &= Z \left[1 - \frac{1}{P} e^{-\frac{1}{P} n\pi \xi} \sin(n\pi + \varphi) \right] \\ &= Z \left[1 - (-1)^n \left(e^{-\frac{1}{P} \pi \xi} \right)^n \right] \\ &= Z(1 - (-1)^n M^n), M = e^{-\frac{1}{P} \pi \xi} \end{aligned} \quad (12)$$

Postup získavania parametrov je potom nasledovný:

1. $Z = y(\infty)$,
2. $y_1 = Z(1 + M)$, $y_2 = Z(1 - M^2) \Rightarrow M = \frac{y_1 - y_2}{y_1}$,
3. $M = e^{-\frac{1}{P} \pi \xi} \Rightarrow \xi = \left| \frac{\ln M}{\sqrt{\pi^2 + \ln^2 M}} \right|$,
4. $t_1 = \frac{\pi}{\omega_0 P}$, $t_2 = \frac{2\pi}{\omega_0 P} \Rightarrow \omega_0 = \frac{\pi}{(t_2 - t_1)\sqrt{1 - \xi^2}}$, $T = 1/\omega_0$.

2.3.5 Systém vyššieho rádu

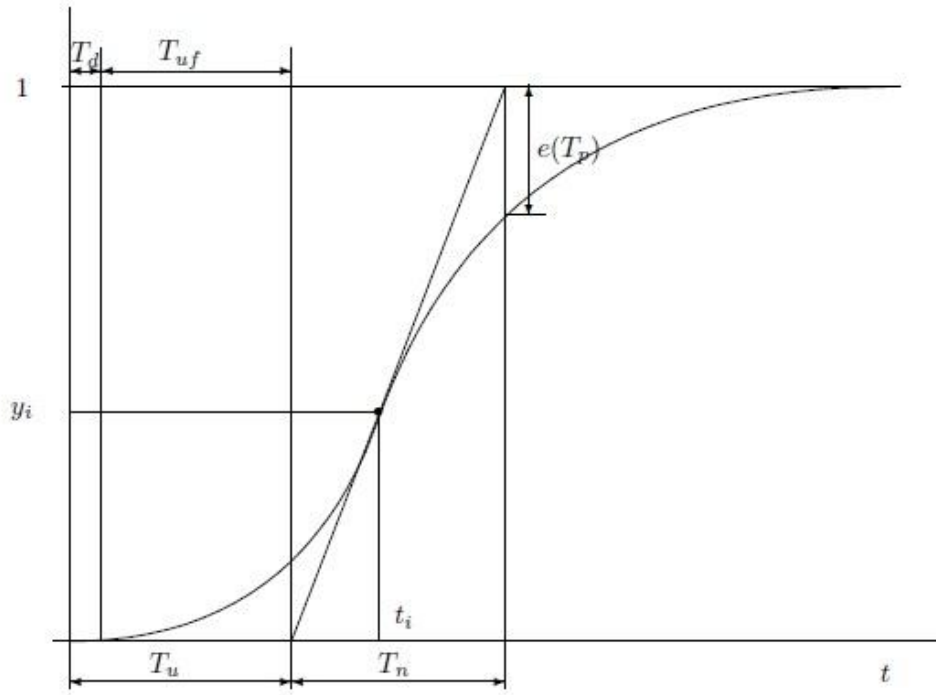
Strejcova metóda

Uvažujme náhradu modelu systému prenosom n -tého rádu

$$G(s) = \frac{Z}{(Ts+1)^n} e^{-T_d s} \quad (13)$$

- kde
- | | | |
|-------|---|---------------------------------------|
| Z | - | zosilnenie, |
| T | - | časová konštanta, |
| T_d | - | dopravné oneskorenie systému, |
| n | - | řád systému, ktoré potrebujeme určiť. |

Z prechodovej charakteristiky (Obr. 3) odčítame čas nábehu T_n a čas priet'ahu T_u na základe preloženia dotýčnice cez inflexný bod prechodovej charakteristiky.



Obr. 3: PCH systému vyššieho rádu

Skúmame vlastnosti normovanej PCH pre prípad $T_d = 0$ v tvare

$$G(s) = \frac{z}{(Ts+1)^n} \quad (14)$$

Pri odvodení využijeme nasledovné fakty:

1. Dotýčnica prechodovej charakteristiky v inflexnom bode $[t_i, y_i]$ je daná rovnicou priamky $p : y = a + bt$,
2. Priamka p prechádza bodmi $[T_u, 0]$, $[T_u + T_n, 1]$, $[t_i, y_i]$ a platí $b = 1/T_n$ a $a = -T_u b$,
3. Smernica priamky je daná ako $b = \dot{y}(t_i)$,
4. $\ddot{y}(t_i) = 0$ (inflexný bod).

Normovaná odozva $y(t)$ na jednotkový skok na vstupe sa dá získať pomocou spätnej Laplaceovej transformácie a je daná vzťahom

$$y(t) = 1 - e^{-\frac{t}{T}} \sum_{k=0}^{n-1} \frac{1}{k!} \left(\frac{t}{T}\right)^k \quad (15)$$

Vypočítame prvú a druhú deriváciu y

$$\dot{y}(t) = \frac{1}{T^n(n-1)!} t^{n-1} e^{-\frac{t}{T}} \quad (16)$$

$$\ddot{y}(t) = \frac{1}{T^n} \left[\frac{t^{n-2}}{(n-2)!} - \frac{1}{T} \frac{t^{n-1}}{(n-1)!} \right] e^{-\frac{t}{T}} \quad (17)$$

V inflexnom bode $\ddot{y}(t_i) = 0$

$$t_i = T(n-1) \quad (18)$$

Prvá derivácia v čase t_i je

$$\dot{y}(t_i) = \frac{(n-1)^{n-1}}{T(n-1)!} e^{-(n-1)} \quad (19)$$

Z obrázku je zrejmé, že platí $\dot{y}(t_i) = 1/T_n$ a teda

$$\frac{T}{T_n} = \frac{(n-1)^{n-1}}{(n-1)!} e^{-(n-1)} = g(n) \quad (20)$$

Vidíme, že táto funkcia je len funkciou n .

Dá sa ukázať, že vzťah medzi T_u a T_n je daný ako

$$\frac{T_u}{T_n} = e^{-(n-1)} \left[\frac{(n-1)^n}{(n-1)!} + \sum_{k=0}^{n-1} \frac{1}{k!} (n-1)^k \right] - 1 = f(n) \quad (21)$$

Táto funkcia je opäť iba funkciou n .

Na základe rozličných hodnôt n môžeme zostrojiť tabuľku (Tab. 1):

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

Tabuľka 1: Hodnoty parametrov n , $f(n)$, $g(n)$ pri identifikácii Strejcovou metódou

Postup pri identifikácii je nasledovný:

1. Z nameranej PCH určíme hodnoty $Z = y(\infty)$, T_{us} , T_n ,
2. Vypočítame podiel $f_s = T_{us}/T_n$,
3. V tabuľke vyberieme rád systému n_0 tak, aby platilo

$$f(n_0) \leq f_s < f(n_0 + 1),$$
4. Dopravné oneskorenie T_d určíme ako rozdiel medzi skutočným a fiktívnym časom nábehu T_u

$$T_d = [f_s - f(n_0)]T_n$$
 pretože platí $T_u = T_n f(n)$,
5. Časová konštanta T sa určí pomocou hodnôt z riadku funkcie $g(n)$ pre príslušné n_0 .
 Odčíta sa $g(n_0)$ a T sa určí ako

$$T = T_n g(n_0).$$

Broidova metóda

Predpoklad zhody všetkých časových konštánt nemusí byť vždy splnený. Broida zaviedol prenos

$$G(s) = \frac{Z}{\prod_{k=1}^n \left(\frac{T}{k^s + 1}\right)} e^{-T_d s} \quad (22)$$

Analogicky ako pri strejcovej metóde boli odvodené $f(n)$, $g(n)$, ktorých hodnoty udáva tabuľka 2.

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

Tabuľka 2: Hodnoty parametrov n , $f(n)$, $g(n)$ pri identifikácii Broidovou metódou

Postup pri identifikácii je rovnaký ako pri Strejcovej metóde.

V mojej práci sa zameriavam na identifikáciu systémov 1., 2. a vyššieho rádu s aperiodickými priebehmi prechodových charakteristík a s predpokladom, že sa v ich priebehu nevyskytuje šum. Pri vyššom ráde určujem prenosy pomocou Strejcovej a Broidovej metódy.

3 PRAKTICKÁ ČASŤ

3.1 Hlavná funkcionálnosť a kritériá

Hlavná funkcionálnosť vytvorenej aplikácie je jednoducho a prehľadne prezentovať postup pri experimentálnej identifikácii systémov. Pred napísaním samotného kódu a práce v programovacom jazyku ActionScript sa muselo určiť pre vytvorenú aplikáciu viacero kritérií:

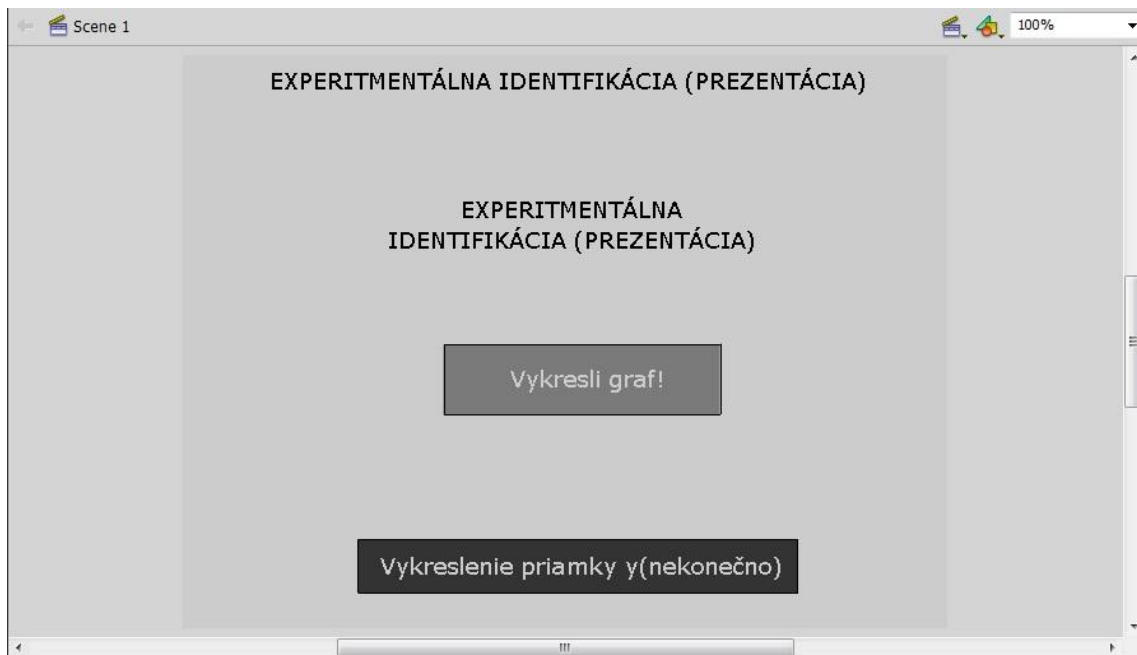
- celá aplikácia by mala mať rovnaký vzhľad, teda upraviť všetky tlačidlá, nadpisy, pozadia na podobný farebný štýl, rovnako je upravená na jednotné rozmery (600 x 450px),
- prezentácia musí byť jednoznačná, jasná, prehľadná a mala by správne vyčísliť všetky parametre potrebné na identifikáciu systémov,
- aplikácia má jednoznačne súhlasiť obsahovo so známymi a použitými metódami pre experimentálnu identifikáciu,
- jej funkcia je čiastočne obmedzená len na aperiodické priebehy prechodových charakteristík bez šumu,
- samotný kód má byť písaný všeobecne, prehľadne a zrozumiteľne nielen pre autora,
- je obmedzená načítavať dáta len z XML súborov, z ktorých neskôr vykresľuje prechodové charakteristiky,
- aplikácia je stavaná tak, aby sa dala otvoriť v každom prehliadači, ktorý obsahuje flash prehrávač.

3.2 Pracovné prostredie Adobe Flash

Na vytváranie mojej aplikácie som používal pracovné programové prostredie Adobe Flash CS5 Professional. V ňom som používal viacero možností, ktorými som mohol modifikovať a meniť rôzne súčasti, pozadia, objekty či písma a zároveň napísať samotný kód celej aplikácie.

3.2.1 Hlavná scéna, objekty, časová os

Hlavná scéna (Obr. 4) slúži na pracovanie s vytvorenými objektmi, či už dynamicky (kódom) alebo staticky (priamym vykreslením grafickými nástrojmi). Je to vlastne tá časť aplikácie, ktorá sa po jej spustení zobrazí užívateľovi.



Obr. 4: Hlavná scéna

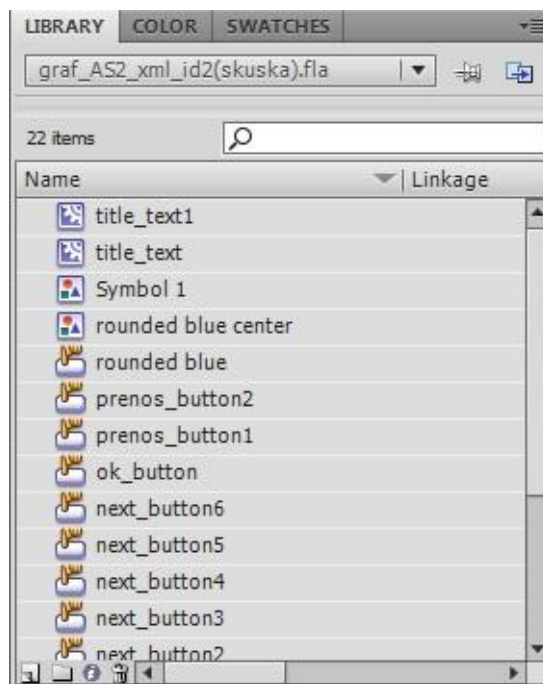
Veľkou výhodou tejto scény je konvertovanie objektov na tzv. symboly, ktorým môžeme priradiť rôzne názvy použiteľné v kóde a ukladať ich do vlastnej knižnice (Obr. 5).

Zaujímavou funkciou, ktorá spája hlavnú scénu (objektovú časť) s kódom je napr. *visible* („viditeľnosť“), ktorá nadobúda hodnoty *true* alebo *false* a teda umožňuje skrytie nejakého objektu na scéne

```
instance_name._visible = false;
```

či následné zobrazenie toho istého objektu v ďalšej snímke

```
instance_name._visible = true.
```



Obr. 5: Knižnica objektov

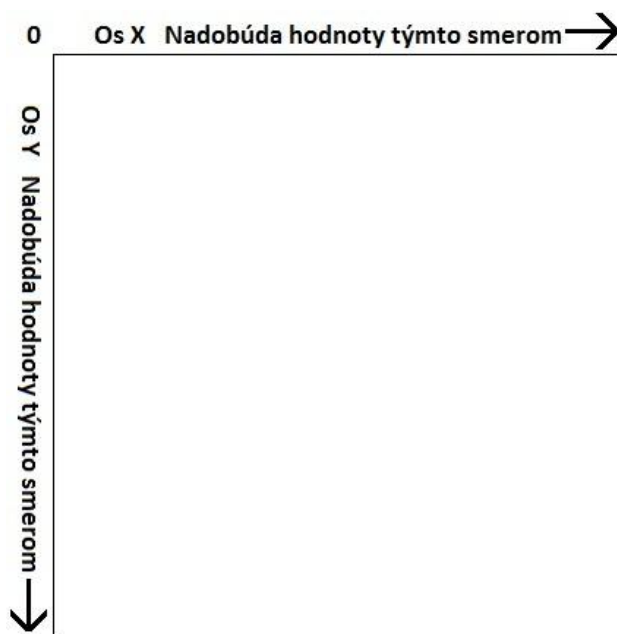
Na scéne som okrem dynamického generovania rôznych objektov či textových alebo číselných polí, niektoré vytváral aj manuálne, teda som ručne kreslil všetky tlačidlá, prirad'oval im farbu, vpisoval do nich text, menil ich veľkosť alebo pozíciu na scéne, či priradil som pozadiu farbu. Na všetky tieto funkcie som mal k dispozícii panel nástrojov (Obr. 6).



Obr. 6: Panel nástrojov

3.2.2 Osi x a y hlavnej scény

Poloha všetkých objektov je na hlavnej scéne vždy definovaná súradnicami x a y . Avšak na rozdiel od klasickej predstavy dvoj rozmerového priestoru, v programe Adobe Flash os y smeruje smerom nadol, čiže nadobúda hodnoty zhora dole. Pre lepšie pochopenie je priložený obrázok 7.

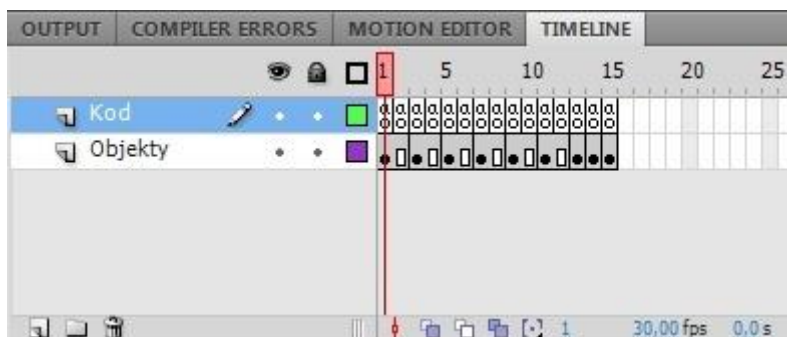


Obr. 7: Osi x a y hlavnej scény v Adobe Flash

Preto polohu každého dynamicky generovaného objektu bolo potrebné prerátať do spomínaného súradnicového systému.

3.2.3 Časová os

Časová os (Obr. 8) sa skladá z dvoch základných častí a to *layer* (vrstvy) a *frame* (snímky). V prvej vrstve som si ukladal časť s kódom a v druhej časť s objektmi, s ktorými som v kóde pracoval. Pre každú snímku vo vrstve „kód“ prináleží snímka vo vrstve „objekty“. V snímkach po spustení aplikácie prebiehajú tzv. frame-cykly, v ktorých sa spustí istá časť kódu pričom na scéne sa objaví žiadaný objekt, či text alebo sa na pozadí programu prerátajú isté parametre z hodnôt dátového súboru.



Obr. 8: Časová os

3.3 Prezentácia vo Flashi

3.3.1 Vykreslenie prechodovej charakteristiky

Pre experimentálnu identifikáciu systémov je základom poznať priebeh prechodovej charakteristiky. Predovšetkým sa však musí načítať XML dokument, z ktorého si dokáže odseparovať parametre t (čas) a y (výstup) potrebných na vykreslenie prechodovej charakteristiky. To robí funkcia na obrázku 9.

```
xmlGraph.onLoad = function(ok:Boolean)
{
    if (ok)
    {
        var numOfNodes:Number = this.firstChild.childNodes.length;
        for(var i=0;i<numOfNodes;i++)
        {
            _root.xArray[i]=Number(this.firstChild.childNodes[i]
                                   .childNodes[0].firstChild.nodeValue);
            _root.yArray[i]=Number(this.firstChild.childNodes[i]
                                   .childNodes[1].firstChild.nodeValue);
        }
    }
    else
    {
        trace("XML did not load");
    }
};
```

Obr. 9: Funkcia pre načítanie XML dokumentu

Obsah XML dokumentu musí byť ale napísaný v určitom tvare, ktorý som si vopred zvolil. Potom vie táto funkcia odseparovať dáta do čísel a vkladať ich do dvoch polí. Samozrejme je ošetrená aj pre prípad ak by sa súbor nenachádzal vo zvolenom umiestnení, v tom prípade len vypíše správu, že sa XML dokument nenačítal. Ak sa načíta správne potom hodnoty pre čas t používam ako súradnice na osi x a hodnoty pre výstup y ako súradnice na osi y .

Samotné vykreslenie prechodovej charakteristiky je založené na dynamickom generovaní krátkych čiar ako MovieClipov do grafu, ako ďalšieho generovaného MovieClipu. Na to bolo zase potrebné vypočítanie jedného dielika pre obe osi. Tu sa museli vytvoriť vlastné definované funkcie pre výpočet maxima a minima hodnôt na oboch stupniciach (Obr. 10 a Obr. 11).

```

minValue = function (array:Array)
{
    var mn:Number = array[0];
    for (i=0; i<array.length; i++)
    {
        if (array[i]<mn && (!isNaN(array[i])))
        {
            mn = array[i];
        }
    }
    return mn;
};

```

Obr. 10: Funkcia na vypočítanie minima

```

maxValue = function (array:Array)
{
    var mxm:Number = array[0];
    for (i=0; i<array.length; i++)
    {
        if (array[i]>mxm && (!isNaN(array[i])))
        {
            mxm = array[i];
        }
    }
    return mxm;
};

```

Obr. 11: Funkcia na vypočítanie maxima

Tieto funkcie sa dali teda následne použiť na nájdenie maxima a minima v hodnotách z polí pre x a y súradnice (Obr. 12).

```

var minX:Number = minValue(xArray);
var maxX:Number = maxValue(xArray);
var minY:Number = minValue(yArray);
var maxY:Number = maxValue(yArray);

```

Obr. 12: Použitie funkcií na nájdenie maxima a minima pre polia hodnôt x a y

Nakoniec už s nájdenými minimami a maximami a zadanou šírkou a výškou grafu sa dali vypočítať dieliky pre x-ovú a y-ovú stupnicu (Obr. 13).

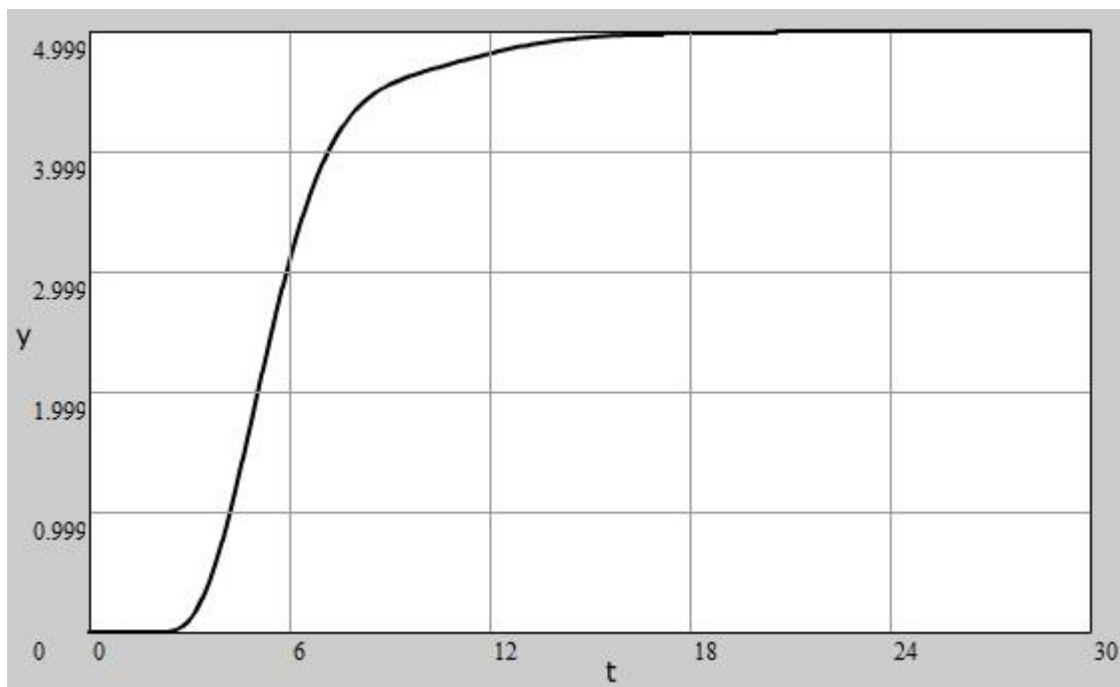
```

var Xscale:Number = graphWidth/(maxX-minX);
var Yscale:Number = graphHeight/(maxY-minY);

```

Obr. 13: Výpočty dielikov pre stupnice x a y

Pri vykresľovaní krivky prechodovej charakteristiky bol použitý cyklus *for* a pomocou vypočítaných dielikov sa postupne generujú a ukladajú sa na správne pozície čiary (resp. body) krivky. Pre príklad uvádzam prechodovú charakteristiku vyššieho rádu (Obr. 14).

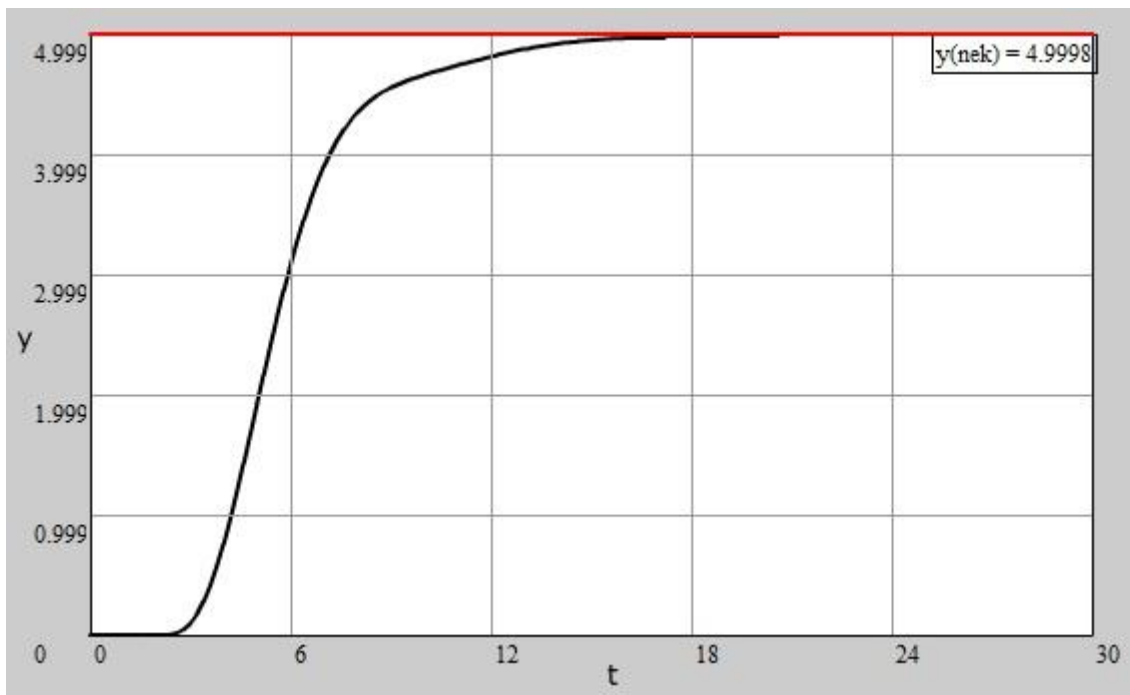


Obr. 14: Vzorová prechodová charakteristika vyššieho rádu

Samotný graf je ohraničený jednoduchým vykresľovaním čiar okolo neho pomocou funkcie *.lineTo* a pomocou ďalšej vlastnej definovanej funkcie sa vygeneruje mriežka v jeho vnútri. Do tejto funkcie som zakomponoval aj dynamické vypísanie hodnôt t a y pozdĺž osi x a y na začiatkoch čiar mriežky, ako aj začiatkové a konečné hodnoty na vrcholoch grafu.

3.3.2 Priamka $y(\infty)$

Na ďalšom snímku sa vykresľuje priamka pre výslednú hodnotu $y(\infty)$ (Obr. 15). Definovaná funkcia pre túto priamku je ukázaná na obrázku 16.



Obr. 15: Vykreslenie priamky $y(\infty)$

```
function drawYInfLine (MC:MovieClip,graphWidth:Number)
{
    var X:Number=MC._x;
    var Y:Number=MC._y;
    _root.createEmptyMovieClip("vertical_line",
        _root.getNextHighestDepth());
    _root.vertical_line.lineStyle(2,0xFF0000);
    _root.vertical_line.moveTo(X,Y);
    _root.vertical_line.lineTo(X+graphWidth,Y);
};
```

Obr. 16: Funkcia na vykreslenie priamky $y(\infty)$

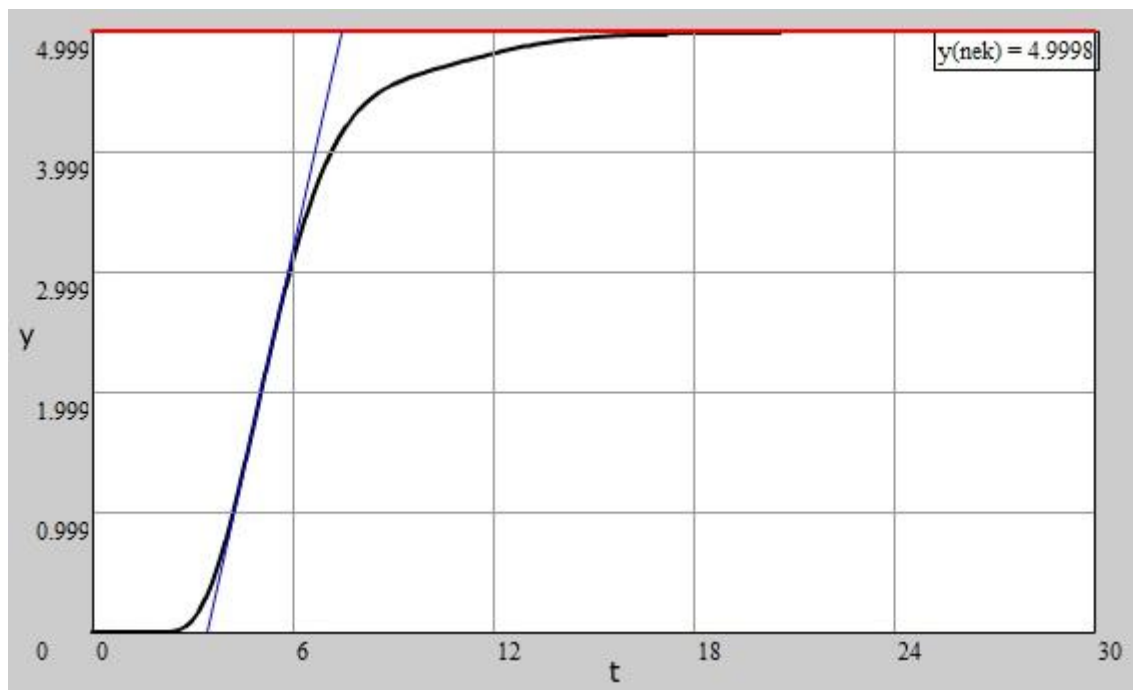
3.3.3 Dotýčnica k inflexnému bodu

V treťom snímku sa zavolá definovaná funkcia na vykreslenie dotýčnice k inflexnému bodu. Táto funkcia najprv nájde samotný inflexný bod. Keďže ActionScript neráta derivácie, zvolil som postup výpočtu pomocou diferenciálu. Funkcia teda počíta rozdiel medzi dvomi po sebe nasledujúcimi y hodnotami. Najväčší rozdiel predstavuje inflexný bod a nájde sa už spomínanou funkciou na hľadanie maxima (Obr. 11). Keďže dotýčnica je len priamka, pomocou súradníc bodov z najväčšieho diferenciálu (medzi ktorými je inflexný bod) sa vypočítajú parametre pre rovnicu priamky z dvoch rovníc o dvoch neznámych.

$$y_1 = a + bx_1 \quad (23)$$

$$y_2 = a + bx_2 \quad (24)$$

Dotyčnica je nakoniec vykreslená ako nový MovieClip jednoduchou funkciou `.lineTo` a presunutá na správnu pozíciu na scéne funkciou `.moveTo` (Obr. 17)



Obr. 17: Vykreslenie dotyčnice k inflexnému bodu

3.3.4 Priamka dopravného oneskorenia

V prechodových charakteristikách musíme rátať aj s možným výskytom dopravného oneskorenia systému. Preto aj táto aplikácia hľadá možné dopravné oneskorenie vlastnou funkciou (Obr. 18). Táto funkcia nájde jeho hodnotu a po jej zavolaní ju vráti ako *x*-ovú súradnicu.

```
function findDelay(xArray:Array,yArray:Array)
{
    var delayPos:Number = 0;
    for(var i=0;i<yArray.length;i++)
    {
        if(yArray[i]==0) delayPos=i;
    }
    return xArray[delayPos];
};
```

Obr. 18: Funkcia na nájdenie hodnoty dopravného oneskorenia

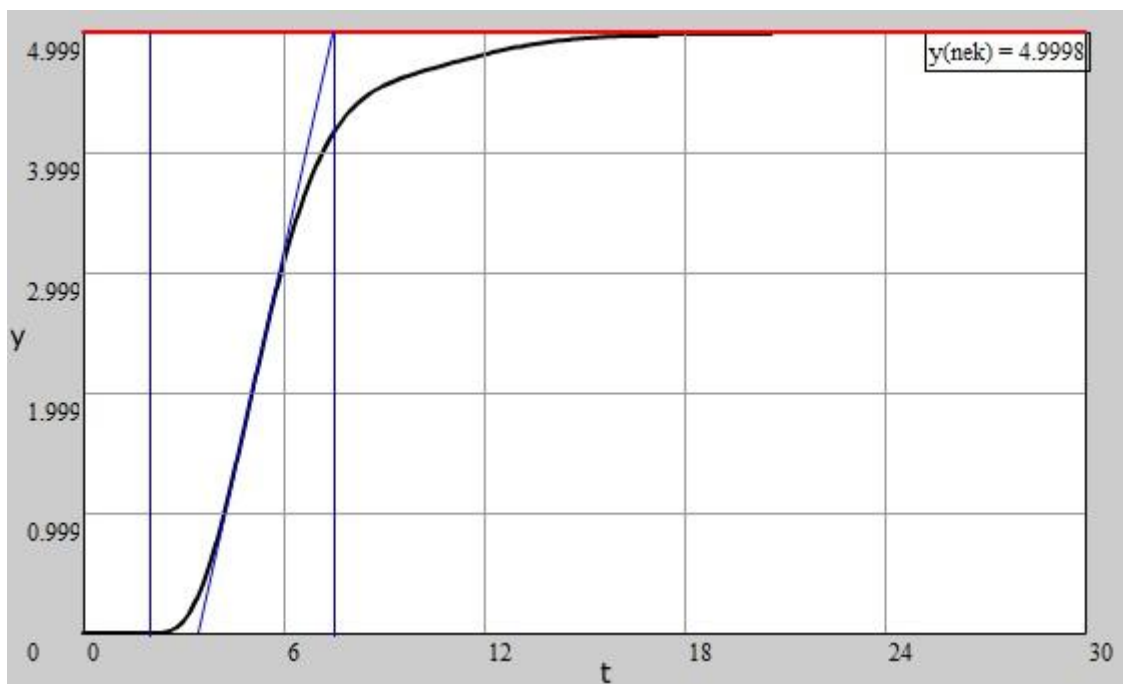
Na transparentné poukázanie, že sa v identifikovanom systéme nachádza dopravné oneskorenie sa dynamicky vykresľuje priamka prechádzajúca vypočítanou x -ovou súradnicou pozdĺž celej výšky grafu (Obr. 19).



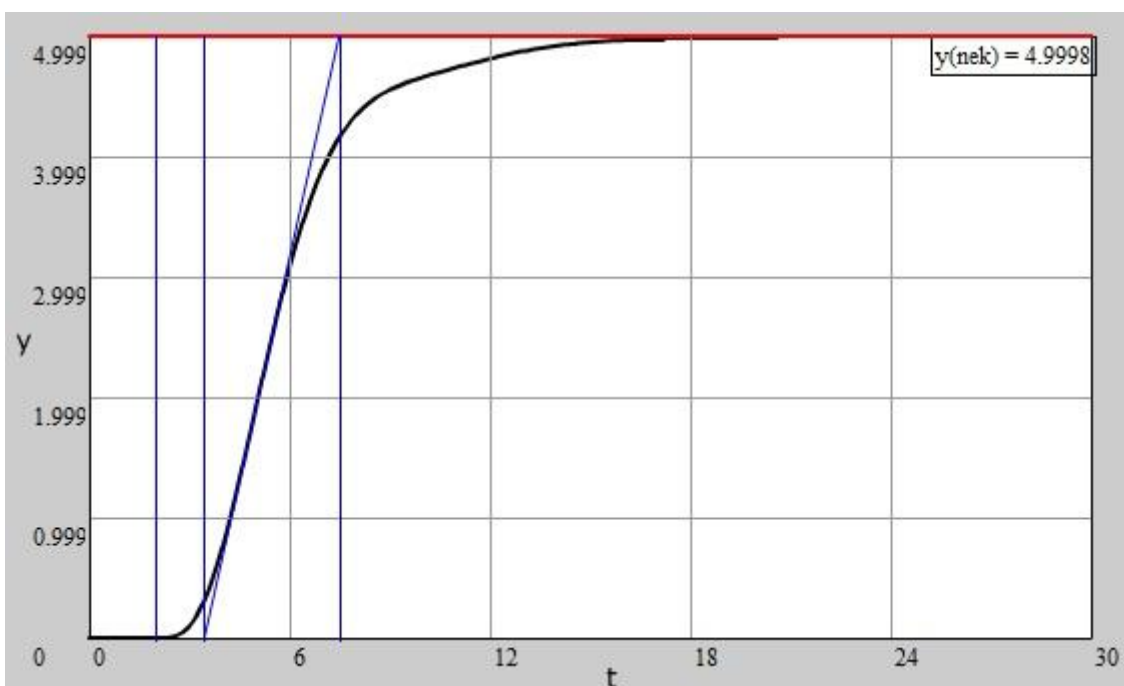
Obr. 19: Vykreslenie dopravného oneskorenia

3.3.5 Priesečníky priamok $y(0)$ a $y(\infty)$ s vykreslenou dotyčnicou

K posledným dôležitým parametrom na určenie výsledného prenosu podľa známych metód patria *čas nábehu* (T_n) a *čas prietahu* (T_u). Ich hodnoty sa dajú jednoducho určiť nájdením priesečníkov priamok $y(0)$ a $y(\infty)$ s dotyčnicou prechádzajúcou inflexným bodom (Obr. 20 a 21).



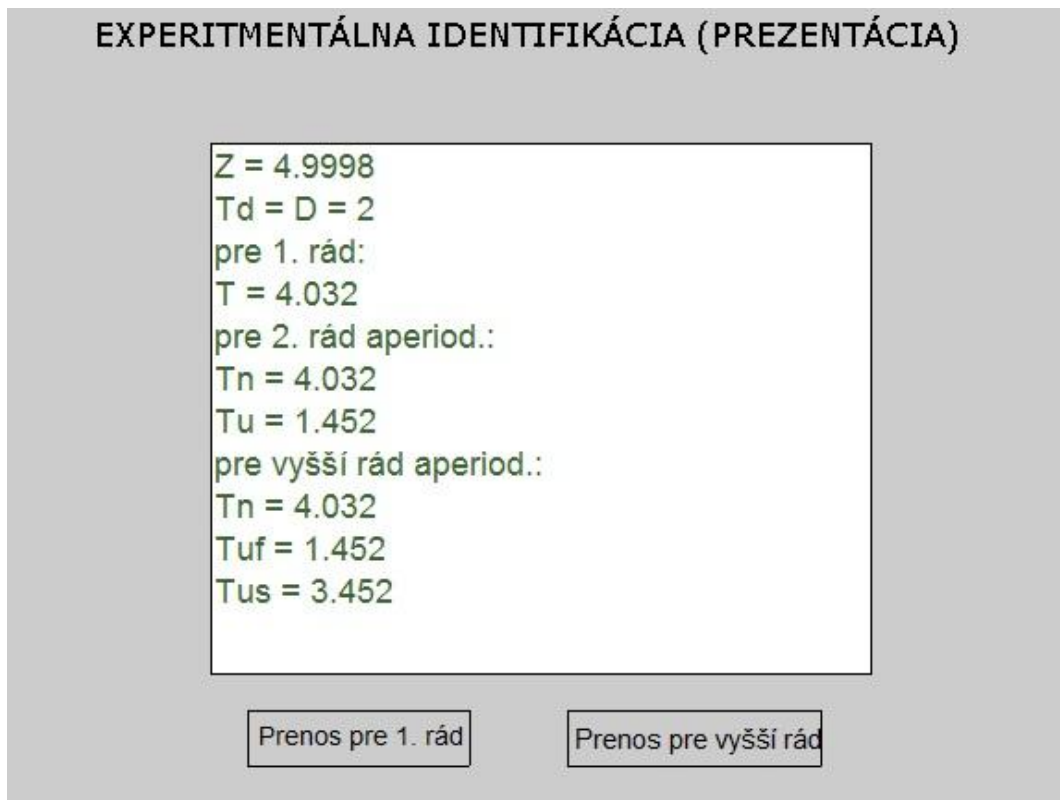
Obr. 20: Vykreslenie čiary priesečníka priamky $y(\infty)$ a dotýčnice k inflexnému bodu



Obr. 21: Vykreslenie čiary priesečníka priamky $y(0)$ a dotýčnice k inflexnému bodu

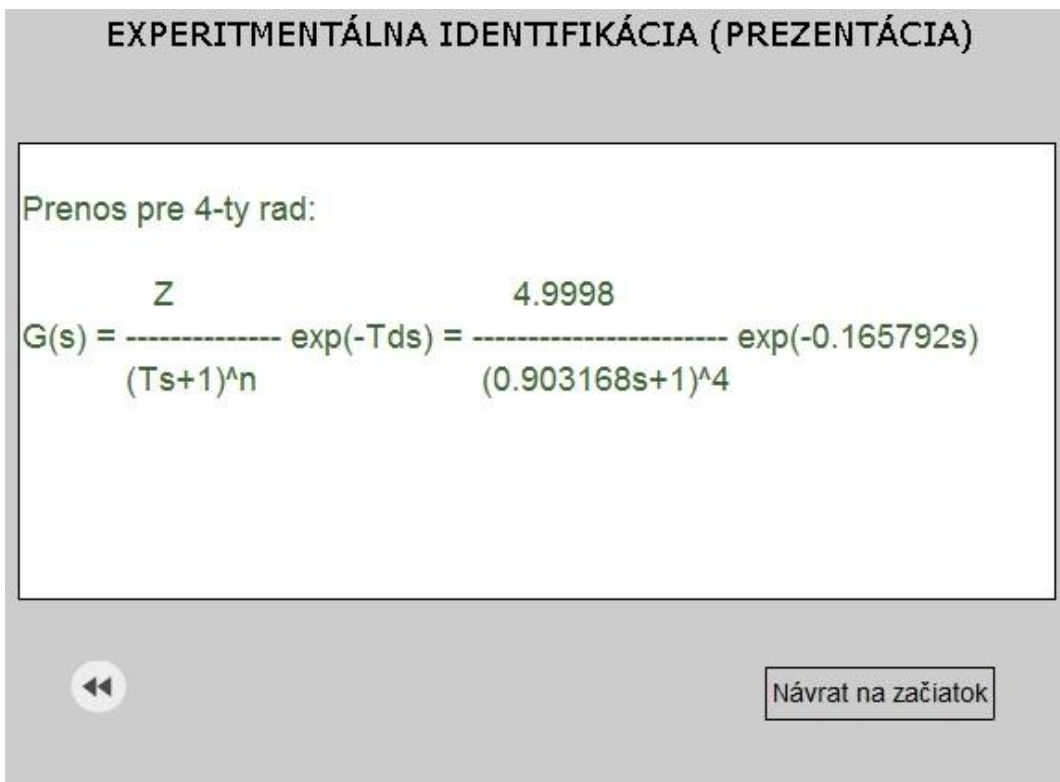
3.3.6 Vypísanie výsledných hodnôt a identifikovaných prenosov

Predposledný snímok nakoniec vypíše zoznam všetkých vypočítaných parametrov (Obr. 22), ktoré potrebujeme na identifikáciu systémov a navrhne možnosť vypísať prenos ako pre systém prvého alebo vyššieho rádu.



Obr. 22: Vypísanie výsledných hodnôt

Po kliknutí na jedno z tlačidiel sa zobrazí požadovaný prenos (Obr. 23). Ak by si užívateľ chcel znova ešte pozrieť výsledné hodnoty je tu možnosť vrátiť sa naspäť na predošlý snímok.



Obr. 23: Vypísanie výsledného prenosu

4 Záver

V tejto práci som sa zaoberal tvorbou prezentácií pre predmet Identifikácia v programe Adobe Flash. Aplikácia, ktorú som vytvoril spĺňa hlavný cieľ celej práce a vytvára jednoduché prezentácie s popisom ich názorného postupu. V teoretickej časti som sa zameriaval na popis programu Adobe Flash, aké má súčasti a vlastnosti a uviedol jeho výhody v porovnaní s inými vhodnými a veľmi podobnými programovacími jazykmi a webovými technológiami ako napríklad HTML, Java Applet, AJAX. Ďalšou časťou bola teória k samotnej experimentálnej identifikácii rôznych systémov, kde sa uvádzajú všeobecné pravidlá, postupy a známe metódy identifikovania. V praktickej časti som názorne opisoval aplikáciu a jej jednotlivé kroky pri identifikovaní systému z prechodovej charakteristiky, ktorá sa vykresľovala z načítaného dátového súboru. Výsledkom bol výpis vypočítaných parametrov a žiadaného identifikovaného prenosu.

Zoznam použitej literatúry

- [1] Čo je Moodle, [online], [cit. 2012-05-05]. Dostupné na:
<http://www.moodle.sk/mod/resource/view.php?id=2>
- [2] Adobe Flash, [online], [cit. 2012-05-05]. Dostupné na:
http://www.en.wikipedia.org/wiki/Adobe_Flash
- [3] KALÚZ, M.: Virtuálne laboratórium, Diplomová práca, FCHPT STU v Bratislave, 2010
- [4] KALÚZ, M.: Virtuálne a vzdialené laboratóriá, Písomná práca k dizertačnej skúške, FCHPT STU v Bratislave, 2012
- [5] MIKLEŠ, J., FIKAR, M.: *Modelovanie, identifikácia a riadenie procesov II., Identifikácia a optimálne riadenie*, STU Bratislava, 2004