

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-5384-23218

**VIZUALIZÁCIA DYNAMICKÝCH SYSTÉMOV
V MODERNÝCH INTERNETOVÝCH PREHLIADAČOCH
DIPLOMOVÁ PRÁCA**

Študijný program:	Aplikovaná informatika
Číslo študijného odboru:	2511
Názov študijného odboru:	9.2.9 Aplikovaná informatika
Školiace pracovisko:	Ústav riadenia a priemyselnej informatiky
Školiteľ:	Ing. Pavol Bisták

Bratislava 2012

Bc. Róbert Kurcz



ZADANIE DIPLOMOVEJ PRÁCE

Evidenčné číslo: FEI-5384-23218
ID študenta: 23218
Autor práce: Bc. Robert Kurcz (23218)
Študijný program: Aplikovaná informatika
Študijný odbor: 9.2.9 aplikovaná informatika

Vedúci práce: Ing. Pavol Bisták

Miesto vypracovania: Ústav riadenia a priemyselnej informatiky

Názov témy: **Vizualizácia dynamických systémov v moderných internetových prehliadačoch**

Špecifikácia zadania:

Cieľom práce je preskúmať možnosti JavaScript-u a HTML pre animáciu dynamických systémov a ich implementácia pre zadané systémy.

Úlohy:

1. Naštudujte problematiku tvorby animácií v nových verziách JavaScript a HTML. Zoznámte sa so zadanými dynamickými systémami a ich riadením.
2. Navrhnite animácie s riadením a interaktívne používateľské rozhranie pre ovládanie animácií.
3. Realizujte animácie s navrhnutým používateľským rozhraním a riadením.
4. Porovnajte realizované animácie s animáciami vytvorenými pomocou iných softvérových nástrojov.
5. Vypracujte technickú dokumentáciu.

Dátum zadania diplomovej práce: **19. 09. 2011**

Termín odovzdania diplomovej práce: **22. 05. 2012**

Bc. Robert Kurcz
študent

prof. RNDr. Otokar Grošek, PhD.
vedúci pracoviska

prof. RNDr. Otokar Grošek, PhD.
garant študijného programu

Pod'akovanie

Týmto by som chcel vyjadriť pod'akovanie vedúcemu diplomovej práce, Ing. Pavlovi Bistákovi za trpezlivé objasňovanie problematiky fyzikálnych sústav a praktické pripomienky, ktoré boli existenčné pri tvorbe tejto diplomovej práce.

ANOTÁCIA DIPLOMOVEJ PRÁCE

Slovenská technická univerzita v Bratislave
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný odbor: 9.2.9 Aplikovaná informatika

Študijný program: Aplikovaná informatika

Autor: Bc. Róbert Kurcz

Diplomová práca: Vizualizácia dynamických systémov v moderných internetových prehliadačoch

Vedúci diplomovej práce: Ing. Pavol Bisták

Mesiac, rok odovzdania: Máj, 2012

Kľúčové slová: vizualizácie dynamických systémov, JavaScript, HTML5, canvas, Runge-Kutta

Práca jasne popisuje tvorbu vizualizácií dynamických systémov v internetových prehliadačoch. Pomocou jazyka JavaScript bolo vytvorených 5 vizualizácií, ktoré verne kopírujú správanie sa dynamických systémov v reálnom svete.

Opis správania sa týchto systémov bol dosiahnutý aproximáciou pomocou metódy typu Runge-Kutta pre všetky systémy. Tieto opisy boli následne naprogramované v programovacom jazyku JavaScript, ktorý umožňuje vykresľovanie v rámci značky "canvas" programovacieho jazyka HTML5. Vizualizácie sú umiestnené na lokálnom serveri, avšak je možné ich umiestniť na ľubovoľný Apache server. Postup inštalácie je detailne popísaný v práci.

Vytvorením tejto práce sa podarilo doceliť jednu z prvých aplikácií, ktorá popisuje dynamické systémy v prostredí internetových prehliadačov pomocou programovacích jazykov JavaScript a HTML5, ktoré sú alternatívou voči hardwarovo náročným aplikáciám z prostredia Flash.

MASTER THESIS ABSTRACT

Slovak University of Technology in Bratislava
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION
TECHNOLOGY

Branch of Study: 9.2.9 Applied informatics

Study Programme: Applied informatics

Author: Bc. Robert Kurcz

Master Thesis: Visualization of dynamic systems in modern web browsers

Supervisor: Ing. Pavol Bistak

Year, Month: 2012, May

Keywords: visualization of dynamic systems, JavaScript, HTML5, canvas, Runge-Kutta

The work clearly describes the creation of visualizations of dynamical systems in web browsers. Using the JavaScript was created 5 visualizations that faithfully replicate the behavior of dynamic systems in the real world.

Description of the behavior of these systems was achieved by an approximation using method of the Runge-Kutta type for all systems. These descriptions were then programmed in JavaScript programming language that allows the rendering within the "canvas" tag of HTML5 programming language. Visualizations are located on a local server, but can be placed on any Apache server. The installation procedure is described in detail in the work.

The creation of this work has managed to achieve one of the first application that describes the dynamic systems in an internet browser with JavaScript programming language and HTML 5, which is an alternative to hardware-intensive applications from within Flash.

Čestné prehlásenie

Prehlasujem, že som diplomovú prácu vypracoval samostatne s využitím poznatkov získaných od vedúceho diplomovej práce a s použitím uvedenej literatúry.

Bratislava, máj 2012

Róbert Kurcz

Obsah

Zoznam použitých skratiek	9
Úvod	10
1 Použité programovacie jazyky a postupy	12
1.1 HTML5.....	12
1.2 JavaScript	13
1.3 PHP.....	15
1.4 CSS.....	16
1.5 knižnica JQuery.....	16
1.6 šablónovací systém Smarty	16
1.7 Runge-Kutta 4	17
2 Analýza použitia programovacích jazykov a opisných metód	18
2.1 Využitie HTML.....	18
2.2 Využitie JavaScriptu	19
2.2.1 Využitie knižnice HighCharts	20
2.2.2 Využitie knižnice JQuery	21
2.3 Využitie PHP a Smarty	21
3 Dynamické sústavy a ich modely	23
3.1 Rovnomerne zrýchlený pohyb	23
3.2 Hydraulický systém jednohladinový.....	24
3.2.1 Schéma a opis sústavy.....	24
3.2.2 Realizácia vizualizácie	24
3.3 Hydraulický systém dvojhladinový.....	27
3.3.1 Schéma a opis sústavy.....	27
3.3.2 Realizácia vizualizácie	28
3.4 Magnetická levitácia	32
3.4.1 Schéma a opis sústavy.....	32
3.4.2 Realizácia vizualizácie	34
3.5 Inverzné kyvadlo	36

3.5.1	Schéma a opis sústavy.....	36
3.5.2	Realizácia vizualizácie	38
4	Kompatibilita aplikácie	41
5	Inštalácia aplikácie	43
	Záver	44
	Literatúra	45

Zoznam použitých skratiek

RK4 – metóda typu Runge-Kutta štvrtého rádu

HTML - HyperText Markup Language

PHP - Hypertext Preprocessor

CSS - Cascading Style Sheets

XML – eXtensible Markup Language

XHTML – Extensible Hypertext Markup Language

SVG – Scalable Vector Graphics

Úvod

Vizualizácie dynamických systémov v internetových prehliadačoch boli do terajšej doby praktizované výhradne pomocou nástroja Flash resp. programovacieho jazyka ActionScript. Aplikácie vytvorené v programovacom jazyku ActionScript sú považované za hardwarovo náročné a aj preto sa vytvorila alternatíva v podobe párových značiek canvas, audio a video jazyka HTML5, ktoré umožňujú prehrávanie videa, vkladania obrázkov a v poslednom rade tvorbu animácií pomocou jazyka JavaScript.

Ďalšou nevýhodou prehrávania multimediálneho obsahu pomocou technológie Flash je potreba inštalácie prídavného modulu, ktorá úplne odpadá pri použití jazyka HTML5. Inštalácia samotného modulu však nie je najmarkantnejšou nevýhodou. Sú to predovšetkým výrazné zahlcovanie zdrojov CPU a RAM, sú náchylné k pádom a predstavujú nezanedbateľné bezpečnostné riziko. Nedá sa však hovoriť o nahradení Flash komponentu, ktorý funguje už viac ako 13 rokov a za túto dobu sa stal štandardom pre vydavateľský priemysel. HTML5 je veľkým prínosom pre jednoduché prehrávanie videa avšak ak by sme chceli pracovať na streamovaní alebo zachytávaní videa boli by sme nútení použiť Flash, nakoľko HTML5 tieto prvky zatiaľ neovláda. Aj preto sa zatiaľ HTML5 považuje za vysokú pridanú hodnotu a nie za kompletné pokrytie tejto problematiky v oblasti multimediálneho obsahu internetových prehliadačov, avšak má potenciál prevýšiť a vytlačiť technológie ako VML, SVG, Flash či Silverlight.

V súčasnej dobe je možné renderovať len 2D scény avšak výrobcovia internetových prehliadačov pracujú na štandarde WebGL, ktorý umožní funkcii canvas renderovať komplexné 3D scény.

V tejto práci sme výhradne používali párovú značku canvas, ktorá definuje priestor na webových stránkach, kde engine prehliadača renderuje grafiku v reálnom čase. To napríklad umožňuje integrovať do internetovej stránky kresliaci program alebo priamo hru bez špeciálneho doplnku.

Praktická realizácia týchto vizualizácií pomocou technológií JavaScript a HTML5 ešte nebola realizovaná aj vzhľadom na relatívne krátke obdobie od ich implementácie v internetových prehliadačoch. Počas vypracovávania zadanie sme sa stretli s istými realizáciami modelov reálneho sveta avšak ani v jednom prípade sa nejednalo o vizualizáciu dynamického systému.

1 Použité programovacie jazyky a postupy

Na vypracovanie zadania diplomovej práce sme použili nasledovné programovacie jazyky a knižnice:

- HTML5
- JavaScript
- PHP
- CSS
- Knižnica jazyka JavaScript – JQuery
- Šablónovací systém jazyka PHP – Smarty

Na aproximáciu opisu dynamických sústav bola použitá metóda typu Runge-Kutta, konkrétne metóda Runge-Kutta štvrtého rádu označovaná aj ako RK4.

1.1 HTML5

HTML = Hypertext Markup Language Language je hlavným značkovacím jazykom pre tvorbu web stránok. Elementy jazyka HTML sú základnými stavebnými blokmi jazyka HTML. HTML je písané formou HTML elementov skladajúcich sa zo značiek, uzatvorených v zátvorkách "<" a ">". Väčšina bežne používaných značiek je párových, hoci niektoré, ktoré sú známe ako prázdne elementy sú nepárové (napr.:). Prvá značka sa nazýva otváracia, tá druhá je označovaná ako ukončovacia. Medzi tieto značky sa môže pridávať text, značky alebo komentáre.

Princípom internetových prehliadačov je prečítať html dokumenty a z nich vytvoriť vizuálne web stránky. Prehliadač nezobrazuje samotné značky ale interpretuje ich ako obsah stránky.

Elementy HTML formujú stavebné bloky všetkých webových stránok. HTML povoľuje pripojiť obrázky a objekty, čo umožňuje vytvárať interaktívny obsah. Poskytuje spôsoby na vytváranie štruktúrovaných dokumentov takzvanou štruktúrovanou sémantikou ako sú hlavičky, paragrafy, zoznamy, odkazy, citácie a iné. Dokáže pripojiť skripty jazyka JavaScript, ktoré ovplyvňujú správanie sa elementov jazyka.

Autorom jazyka HTML je fyzik Tim Berners-Lee toho času (1980) pôsobiaci v Cerné, navrhol ENQUIRE, systém slúžiaci pre výskumníkov z Cernu na používanie a zdieľanie dokumentov. HTML špecifikoval, napísal software serveru a prehliadača v roku 1990.

Tab. 1: Verzie jazyka HTML

Dátum	Verzia
1991 október	neformálny dokument CERNu obsahujúci 12 HTML značiek
1992 jún	prvý neformálny návrh HTML DTD (document type definition)
1992 november	HTML DTD 1.1
1993 jún	HTML 2.0
1993 november	HTML + (zaniklo v máji 1994)
1995 apríl	HTML 3.0
1998 december	HTML 4.0
1999 december	HTML 4.01 štandard ISO/IEC 15445:2000
2008 január	HTML 5, prvý návrh

1.2 JavaScript

JavaScript - skriptovací programovací jazyk, ktorý v súčasnosti podporuje objektovo-orientované programovacie štýly. V jeho počiatkoch bol označovaný ako Mocha, neskôr pod menom LiveScript. Jeho autorom je Brendan Eich a meno JavaScript dostal kvôli

vtedajšej popularite jazyka Java, nakoľko syntax JavaScriptu je veľmi podobná tomuto jazyku.

JavaScript je client-side (vykonáva sa na strane klienta) programovací jazyk, implementovaný ako časť internetových prehliadačov za účelom vytvoriť užívateľské prostredie s pridanou hodnotou a teda dynamický obsah internetovej stránky.

Tento programovací jazyk veľmi rýchlo nadobudol úspech ako client-side skriptovací jazyk pre internetové stránky. Ako dôsledok, Microsoft pomenoval jeho implementáciu ako JScript aby sa vyhol problémom s ochranou autorských práv. JScript priniesol nové dátumové metódy ako riešenia Y2K problematiky v JavaScripte, ktoré vychádzali z java triedy java.util.Date. JScript bol súčasťou prehliadača Internet Explorer 3.0, ktorý bol vydaný v auguste 1996.

Po počiatočnej veľkej obľube sa začalo od tohto jazyka upúšťať, avšak príchod AJAXu vrátil JavaScript na výslnie a priniesol viac profesionálneho programovacieho záujmu o tento jazyk. Výsledkom bola profilácia obsiahlych frameworkov a knižníc (JSON alebo JQuery).

JSON alebo JavaScript Object Notation je univerzálny formát na výmenu dát, ktorý je definovaný ako podmnožina syntaxe JavaScriptu.

JQuery je populárna JavaScriptová knižnica navrhnutá na zjednodušenie DOM orientovaného písania client-side skriptov.

Pri veľkých netriviálnych programoch je potreba prístupu k debuggeru nakoľko tu môžu vznikať implementačné rozdiely hlavne v rámci DOM(Document Object Model) a je teda veľmi príhodné mať k dispozícii debugger pre každý internetový prehliadač. Väčšina moderných internetových prehliadačov má v sebe debugger zabudovaný (Internet Explorer, Firefox, Google Chrome, Opera, Safari).

1.3 PHP

PHP – je populárny open source skriptovací programovací jazyk, ktorý sa používa najmä na programovanie klient-server aplikácií (na strane servera) a pre vývoj dynamických webových stránok.

Je jedným z prvých vyvinutých server-side skriptovacích jazykov, ktorý je možné vložiť do HTML zdrojového kódu namiesto volania externého súboru na výpočet dát. Všeobecne, kód je interpretovaný Web serverom s PHP modulom.

Tento programovací jazyk má slabú až žiadnu typovú kontrolu. Jednotlivé typy premenných môžu byť dynamicky zmenené.

PHP bolo inšpirované a postavené na základe programovacích jazykov C alebo Perl a teda má z nich prebraných množstvo prvkov. V neskorších verziách sa objavila možnosť používania objektov.

PHP dokáže spolupracovať s relačnými databázami, ako napríklad MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL a SQLite a zachováva si veľmi priamočiaru syntax. Je možné ho prevádzkovať na rôznych operačných systémoch avšak architektúra známa pod skratkou LAMP(Linux, Apache, MySQL, PHP) sa teší veľkej popularite v internetovom odvetví.

Tab. 2: Verzie jazyka PHP

Dátum	Verzia
1995	PHP 1: Oficiálne nazvaná Personal Home Page Tools
1997	PHP 2
1998	PHP 3
2000	PHP 4
2004	PHP 5

1.4 CSS

Jazyk CSS sa využíva pre štylovanie prezentačnej sémantiky (vzhľad a formátovanie) dokumentu napísaného v značkovacom jazyku. Jeho najčastejšie použitie je na štylovanie internetových stránok napísaných v HTML a XHTML, ale jazyk môže byť takisto aplikovaný na akýkoľvek typ dokumentu XML, SVG a XUL.

Hlavný význam jazyka spočíva v oddelení obsahu dokumentu (najčastejšie HTML) od prezentačnej vrstvy vrátane prvkov ako rozloženia, farby a fontov.

1.5 knižnica JQuery

JQuery je univerzálna JavaScript knižnica navrhnutá pre zjednodušenie skriptovania na strane klienta HTML. Bola vydaná v januári 2006 Johnom Resigom. Používa ju viac ako 55% z 10000 najnavštevovanejších internetových stránok sveta a je jednoznačne najpopulárnejšia JavaScript-ová knižnica používaná v súčasnosti.

JQuery je open-source knižnica, jej syntax je navrhnutá pre jednoduchšiu navigáciu dokumentu, prístupovanie k DOM elementom, vytváranie animácií, nadštandardných efektov a upravovateľných “widgetov”.

1.6 šablónovací systém Smarty

Smarty je webový šablónovací systém napísaný v PHP. Tento systém primárne slúži na oddelenie prezentačnej vrstvy od “back-endu”. V ideálnych prípadoch uľahčuje úsilie a znižuje čas spojený s údržbou softvéru.

Smarty vytvára webový obsah umiestnený v špeciálnych smarty značkách v dokumente. Tieto značky sú spracované a nahradené iným kódom, sú vkladané do dokumentu za pomoci Smarty oddelovačov. Tieto značky môžu predstavovať premenné, funkcie alebo logické výroky. Smarty umožňuje PHP programátorom definovať vlastné funkcie, ku ktorým je možno prístupovať pomocou Smarty značiek.

1.7 Runge-Kutta 4

Metódy typu Runge Kutta sú veľmi užitočné najmä v technickej praxi. Tieto metódy sú postavené na Taylorovom rozvoji funkcie, avšak nepriamo, čiže neurčujeme hodnoty derivácie funkcie ale ich aproximujeme výpočtom samotnej funkcie vo vhodne zvolených strategických bodoch.

Metóda typu Runge Kutta štvrtého rádu:

h - krokovanie

$$y' = F(x, y(x))$$

derivácia funkcie

(1.1)

$$y(x + h) = y(x) + \frac{k_1 + 2 * k_2 + 2 * k_3 + k_4}{6}$$

hodnota funkcie v $x + h$ kroku

(1.2)

$$\begin{aligned} k_1 &= h * F(x, y(x)) \\ k_2 &= h * F\left(x + \frac{h}{2}, y(x) + k_1/2\right) \\ k_3 &= h * F\left(x + \frac{h}{2}, y(x) + k_2/2\right) \\ k_4 &= h * F(x, y(x) + k_3) \end{aligned}$$
(1.3)

2 Analýza použitia programovacích jazykov a opisných metód

V nasledujúcej časti sme popísali význam použitia jednotlivých programovacích jazykov a postupov, uviedli konkrétne príklady použitia, ktoré dopomohli k dosiahnutiu celkového cieľu tejto práce.

2.1 Využitie HTML

Využitie jazyka HTML pri tvorbe tejto práce bolo existenčné, nakoľko na jeho súčasťou je značka canvas, ktorá zabezpečuje plochu pre renderovanie jednotlivých objektov pomocou jazyka JavaScript.

Uvádzame príklad použitia z práce:

```
<div style="float:left;">
  <canvas id="rotCanvas" width="40" height="40"></canvas>
  <canvas id="myCanvas" width="350" height="350"></canvas>
</div>

<div id="container"></div>
<div style="clear:both;"></div><br />

  <input type="image" onclick="init();" src="images/start.png" />
  <input type="image" onclick="window.location.reload();"
    src="images/reset.png" />
<div style="clear:both;"></div><br />

Ventil č.:1 uzavrieť: <input type="checkbox" id="ventil1" /><br />
Ventil č.:2 uzavrieť: <input type="checkbox" id="ventil2" /> <br />
<br />
<span>Výška1: </span><span id="vyska1"></span> m <br />
<span>Výška2: </span><span id="vyska2"></span> m
```

2.2 Využitie JavaScriptu

Využitie tohto programovacieho skriptovacieho jazyka malo v našej práci najväčšie využitie, pretože sa pomocou neho nedeje len samotné renderovanie objektov ale aj výpočet opisu správania sa jednotlivých dynamických sústav.

Príklad renderovania objektov z našej práce:

```
//gulicka
context.beginPath();
context.arc(centerX, centerY-posun, 47.5, 0, 2 * Math.PI,
false);
context.fillStyle = 'black';
context.fill();
context.stroke();

//cievka
context.beginPath();
context.arc(310, 67, 5, 1.5*Math.PI, (1/2)*Math.PI, false);
context.lineTo(190,79);
context.lineWidth = 2;
context.strokeStyle = "black";
context.stroke();
```

Príklad výpočtu opisu správania sa sústavy:

```
function rungel() {
    regulacia();
    if(ypred != 0){
        k1 = 1/s1*q1 - betal*Math.sqrt(ypred);
        k2 = 1/s1*q1 - betal*Math.sqrt(ypred) + h/2*k1;
        k3 = 1/s1*q1 - betal*Math.sqrt(ypred) + h/2*k2;
        k4 = 1/s1*q1 - betal*Math.sqrt(ypred) + h*k3;
    }else{
        k1 = 1/s1*q1;
        k2 = 1/s1*q1 + h/2*k1;
        k3 = 1/s1*q1 + h/2*k2;
        k4 = 1/s1*q1 + h*k3;
    }
}
```

2.2.1 Využitie knižnice HighCharts

Tieto však neboli jediné prípady kedy vznikla potreba využitia tohto jazyka. JavaScript slúži prevažne na tvorby dynamického obsahu stránok a teda v našom prípade bol vysoko prínosný pri obsluhu jeho knižnice HighCharts, ktorá nám poslúžila na vynášanie údajov do grafu a jeho samotné vykresľovanie.

Príklad inicializácie grafu:

```
var chart;
chart = new Highcharts.Chart({
  chart: {
    renderTo: 'container',
    type: 'spline',
    marginRight: 10,
    events: {
      load: function() {

        var series = this.series[0];
        var time;
        var c =2;
        setInterval(function() {

          var x = c++;

          if(y > 0)  {
            series.addPoint([x, ypred]);
          }else alert('koniec');
          }, 1000);
        }
      },
    title: {
      text: 'Priebeh Výšky Hladiny'
    },
    xAxis: {
      type: 'linear',
      tickPixelInterval: 150,
      maxZoom: 15
    }
  }
  ...
});
```

2.2.2 Využitie knižnice JQuery

Táto knižnica bola prevažne využívaná na “real-time” získavanie údajov zo sústav, ktoré boli následne prezentované na stránke.

Príklad použitia:

```
if( $('#regulacia').attr('checked') ){  
    if( y < $('#reg_hlad').val() ){  
        q1 = 0.000008;  
    }  
}
```

2.3 Využitie PHP a Smarty

PHP programovací jazyk sme využili pri vytvorení indexovej stránky systému, kde obsluhujeme pridelovanie javascriptových súborov do smarty premenných a takisto pridelovanie htm súborov do smarty premenných.

Tieto dva typy súborov sú jediné potrebné na vytvorenie ďalších vizualizácií, čím sme uľahčili pridávanie ďalších vizualizácií v prípade, že by vznikla potreba ich vytvorenia.

Vytvorenie nového objektu triedy Smarty:

```
$smarty = new Smarty();
```

Nastavenie ciest k adresárom so šablónami a ku konfiguračným adresárom:

```
$smarty->setTemplateDir('smarty_dirs/templates/');  
$smarty->setCompileDir('smarty_dirs/templates_c/');  
$smarty->setConfigDir('smarty_dirs/configs/');  
$smarty->setCacheDir('smarty_dirs/cache/');
```

Ukážka inicializácie aj priradenia do smarty premenných pomocou metódy assign:

```
if($_GET['page'] == 'rovnomernezrych') {  
    $smarty->assign('jscript','<script type="text/javascript"  
    src="js/rovnomernezrych.js"></script>');  
    $smarty->assign('includeFile','htm/rovnomernezrych.htm');  
}elseif($_GET['page'] == 'nadoba'){
```

```

$smarty->assign('jscript','<script type="text/javascript"
src="js/nadoba.js"></script>');
$smarty->assign('includeFile','htm/nadoba.htm');
}elseif($_GET['page'] == 'sustavanadob'){
    $smarty->assign('jscript','<script type="text/javascript"
src="js/sustavanadob.js"></script>');
    $smarty->assign('includeFile','htm/sustavanadob.htm');
}elseif($_GET['page'] == 'kyvadlo'){
    $smarty->assign('jscript','<script type="text/javascript"
src="js/kyvadlo.js"></script>');
    $smarty->assign('includeFile','htm/kyvadlo.htm');
}elseif($_GET['page'] == 'levitacia'){
    $smarty->assign('jscript','<script type="text/javascript"
src="js/levitacia.js"></script>');
    $smarty->assign('includeFile','htm/levitacia.htm');
}

```

3 Dynamické sústavy a ich modely

3.1 Rovnomerne zrýchlený pohyb

Prvou dynamickou sústavou, ktorú sme namodelovali bol rovnomerne zrýchlený pohyb. V párovej HTML značke canvas širokej 800px simulujeme rovnomerne zrýchlený pohyb na modeli auta, kde na nižšie položenej mierke môžeme sledovať priebeh tohto pohybu.

Jedná sa o najjednoduchšiu vizualizáciu, nakoľko rovnomerne zrýchlený pohyb má lineárny priebeh a teda aj matematický opis správania sa tohto systému nie je komplikovaný:

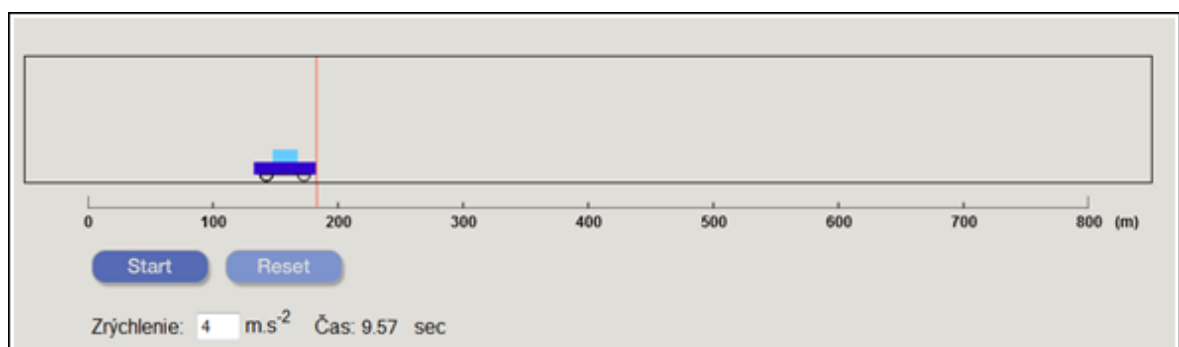
$$s = \frac{1}{2} * a * t^2 \quad (3.1)$$

Následne jednotlivé koeficienty RK4 metódy majú základný tvar:

```
k1 = a; // hodnoty jednotlivych krokov su konstantne
k2 = a;
k3 = a;
k4 = a;
```

Výpočet posunu vozidla:

```
v = v + (h/6) * ( k1 + 2*k2 + 2*k3 + k4 );
t += h;
s = (1/2) * a * (t*t);
```



Obr. 1. Model: Rovnomerne zrýchlený pohyb

3.2 Hydraulický systém jednohladinový

Komplikovanejšou dynamickou sústavou, pre ktorú sme vytvorili vizualizáciu je hydraulický systém jednohladinový. Okrem možnosti ovládania prítoku do systému sme naprogramovali takisto možnosť regulácie tohto systému. Priebeh výšky hladiny je zaznamenávaný pomocou grafu knižnice HighCharts.

3.2.1 Schéma a opis sústavy

Hydraulický systém jednohladinový sa skladá z jednej nádoby, do ktorej je načerpávaná voda pomocou čerpadla, ktoré čerpá vodu zo zásobníka uloženého pod nádobou a čerpadlom.

Opis sústavy pomocou diferenciálnej rovnice:

$$\frac{dh_1}{dt} = \frac{1}{S_1} - \beta_1 \cdot \sqrt{h_1} \quad (3.2)$$

Ostatné známe parametre sústavy:

$$\begin{aligned} \beta_1 &= \alpha_1 / S_1 \\ \alpha_1 &= 2,08 * 10^{-3} \\ S_1 &= 2,025 * 10^{-3} [m^2] \end{aligned} \quad (3.3)$$

3.2.2 Realizácia vizualizácie

V rámci tejto vizualizácie má používateľ možnosť ovládať prítok pomocou JQuery UI objektu – “slider”.



Obr. 2. Ovládanie prítoku

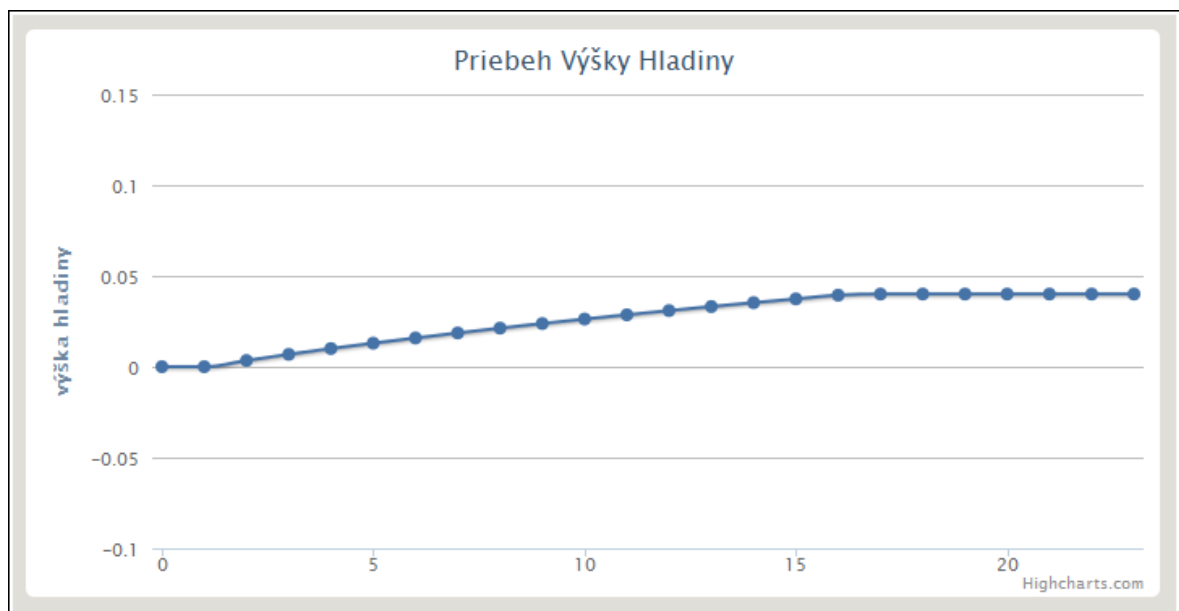
JavaScriptová funkcia na ovládanie „slidera“:


```

$(function() {
    $( "#slider-range-min" ).slider({
        range: "min",
        value: 50,
        min: 0,
        max: 50,
        slide: function( event, ui ) {
            $( "#amount" ).val(
                Math.round(ui.value*0.00000016*1000000)/1000000
            );
        }
    });
    $( "#amount" ).val( 0.000008 );
});

```

Ďalšou možnosťou je regulácia výšky hladiny sústavy, kde po zadaní požadovanej hodnoty a zaškrtnutí zaškrťavacieho políčka regulácie sa nám začne výška hladiny regulovať na základe vzťahu:

$$q_1 = \beta_{a1} \cdot \text{Math.sqrt} (\$ (\# \text{reg_hlad}) . \text{val} ()) * s_1$$


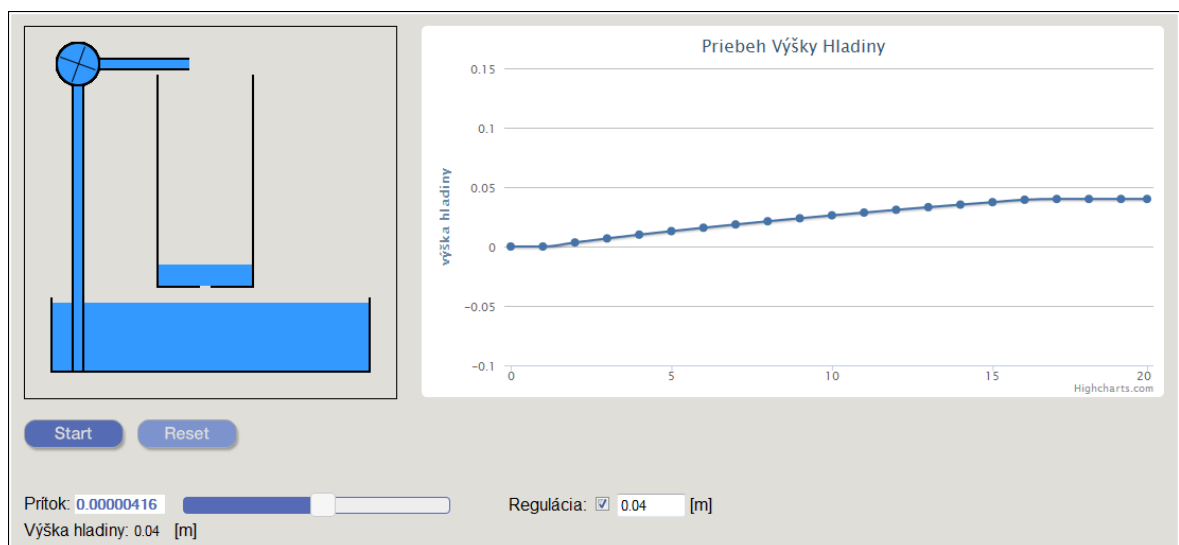
Obr. 3. Regulovaná hladina

Pomocou funkcie `setInterval()` voláme každú desatinu sekundy funkciu `draw()`, ktorá si vo svojom tele vypočítava jednotlivé koeficienty pomocou RK4 metódy a následne vypočítava výšku hladiny:

```
setInterval(draw,100);

function rungel() {
    regulacia();
    if(ypred != 0){
        k1 = 1/s1*q1 - betal*Math.sqrt(ypred);
        k2 = 1/s1*q1 - betal*Math.sqrt(ypred) + h/2*k1;
        k3 = 1/s1*q1 - betal*Math.sqrt(ypred) + h/2*k2;
        k4 = 1/s1*q1 - betal*Math.sqrt(ypred) + h*k3;
    }else{
        k1 = 1/s1*q1;
        k2 = 1/s1*q1 + h/2*k1;
        k3 = 1/s1*q1 + h/2*k2;
        k4 = 1/s1*q1 + h*k3;
    }
}

function draw()
{
    rungel();
    y=ypred + h/6*(k1 + 2*k2 + 2*k3 +k4);
    ...
}
```



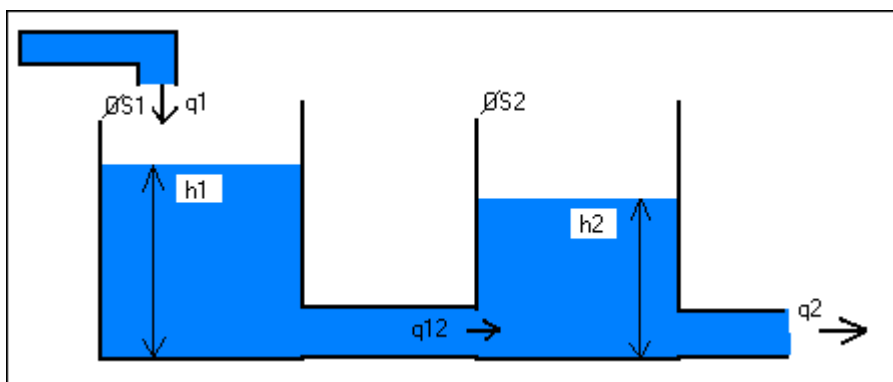
Obr. 4. Model: Hydraulický systém jednohladinový

3.3 Hydraulický systém dvojhladinový

Hydraulická sústava pozostáva z dvoch prepojených nádrží s rozmermi 45x45x350 mm. Táto sústava takisto obsahuje čerpadlo, ktoré načerpáva vodu z misky umiestnenej pod nádobami. Medzi prvou a druhou nádobou sa nachádza ventil, ktorý môžeme v ovládaní uzavrieť alebo ponechať otvorený a tým ovládať prítok do druhej nádoby. Takisto môžeme ovládať ventil medzi druhou nádobou a miskou pod sústavou a ním ovládať odtok vody z celej sústavy.

Takisto ako pri predchádzajúcom modeli môžeme ovládať prítok do sústavy, až kým nezapneme reguláciu. Po tomto momente sa prítok už nedá ovládať. Pri tejto sústave zaznamenávame priebeh obidvoch výšok hladín do grafu pričom môžeme regulovať výšku hladín jednotlivých nádrží.

3.3.1 Schéma a opis sústavy



Obr. 5. Parametre hydraulického systému

Sústava popísaná pomocou diferenciálnych rovníc:

$$\begin{aligned}\frac{dh_1}{dt} &= \frac{1}{S_1} - \beta_1 \cdot \sqrt{h_1 - h_2} \\ \frac{dh_2}{dt} &= \beta_2 \cdot \sqrt{h_1 - h_2} - \beta_3 \cdot \sqrt{h_2}\end{aligned}\tag{3.4}$$

Hodnoty parametrov opisujúcich sústavu:

$$\alpha_1 = 0.0000208 \quad \alpha_2 = 0.0000163$$
$$S_1 = 0.002025 \text{ m}^2 \quad S_2 = 0.002025 \text{ m}^2$$

Kde:

$$\beta_1 = \frac{\alpha_1}{S_1} \quad \beta_2 = \frac{\alpha_1}{S_2} \quad \beta_3 = \frac{\alpha_2}{S_2}$$

Potom pre hodnoty:

$$\beta_1 = 0,01027160494$$

$$\beta_2 = 0,01027160494$$

$$\beta_3 = 0,008049382716$$

dostaneme opis pre systém hydraulickej sústavy:

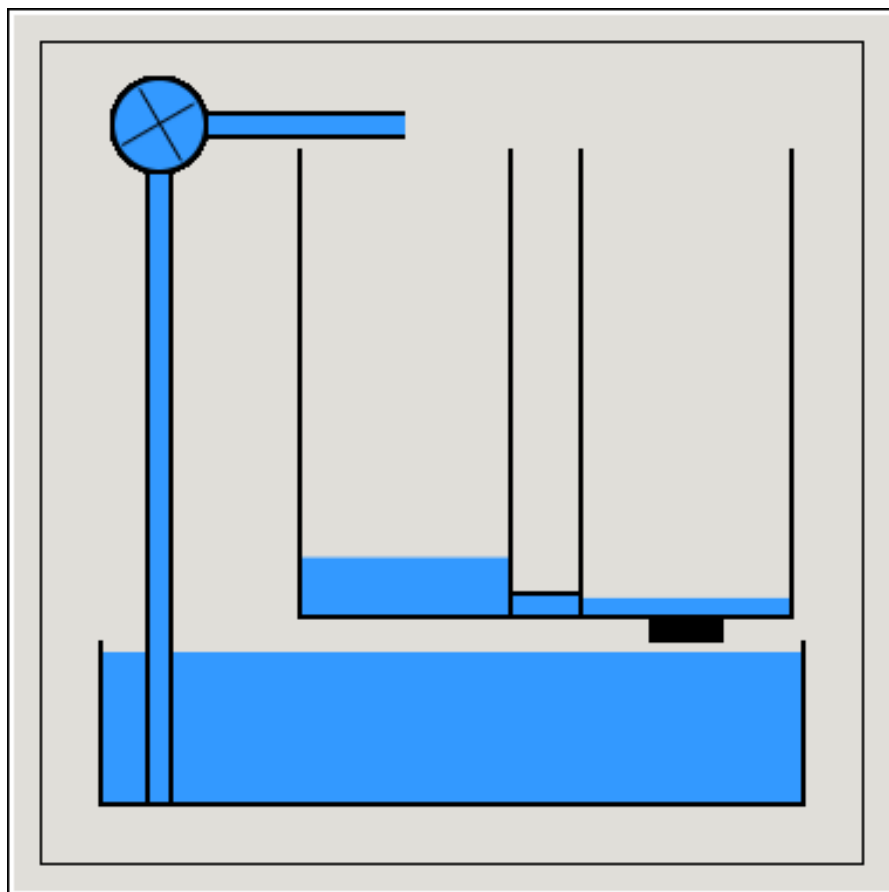
$$\begin{aligned} dh_1 &= 0,4938271605 - 0,01207160494 \cdot \sqrt{h_1 - h_2} \\ dh_2 &= 0,01207160494 \cdot \sqrt{h_1 - h_2} - 0,008049382716 \cdot \sqrt{h_2} \end{aligned} \quad (3.5)$$

3.3.2 Realizácia vizualizácie

Pri realizácii tejto vizualizácie sme takisto použili JQuery UI “slider” na ovládanie prítoku vody do prvej nádrže. Máme možnosť uzatvárať a otvárať ventily medzi jednotlivými nádobami pomocou zaškrtávacích políčk pod vizualizácie, kde je umiestnené všetko ovládanie.

Ukážka prevedenia odčítavania stavu ventilov:

```
if( $('#ventil1').attr('checked') ){
    beta1 = 0;
    beta2 = 0;
}else{
    beta1 = alfa1/s1;
    beta2 = beta1;}
```



Obr. 6. Hydraulický systém dvojhladinový s uzatvorením druhým ventilom

Ventil č.:1 uzavrieť: <input checked="" type="checkbox"/>	Pritok: <input type="text" value="0.000008"/>	<input type="range" value="100"/>
Ventil č.:2 uzavrieť: <input checked="" type="checkbox"/>		
Výška1: 0.038 m	Regulácia: nádoba č.1: <input checked="" type="checkbox"/> <input type="text" value="0.5"/> m	
Výška2: 0 m	nádoba č.2: <input type="checkbox"/> <input type="text"/> m	

Obr. 7 Ovládanie a regulácia dvojhladinového hydraulického systému

Vykreslenie samotnej vizualizácie sme dosiahli klasickým postupom ako aj pri predchádzajúcich modeloch. Pristúpenie ku canvas značke a následne kresliace funkcie:

```
var canvas = document.getElementById("myCanvas");
context= canvas.getContext('2d');
context.rect(25,260,300,65);
context.fillStyle="rgb(51,153,255)";
context.fill();
```

Pre tento krát sme boli nútení použiť dvojité RK4 metódu, nakoľko sme riešili sústavu diferenciálnych rovníc:

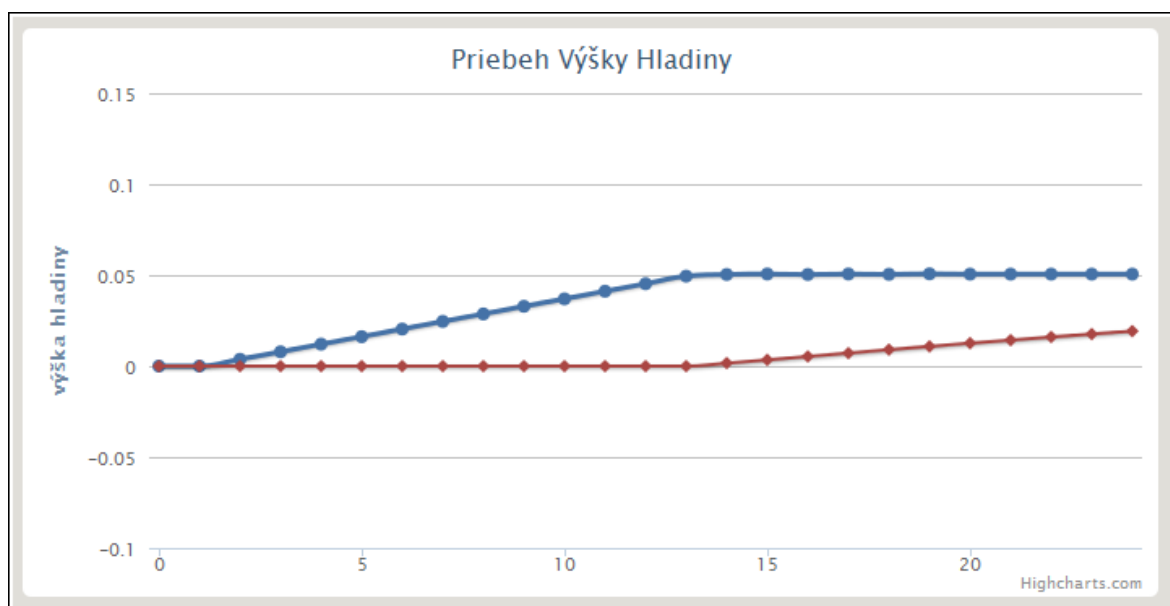
```
if(y2pred > 0){
    k1=1/s1*q1-beta1*Math.sqrt(ypred - y2pred);
    k2=1/s1*q1-beta1*Math.sqrt(ypred - y2pred)+ h/2*k1;
    k3=1/s1*q1-beta1*Math.sqrt(ypred - y2pred)+ h/2*k2;
    k4=1/s1*q1-beta1*Math.sqrt(ypred - y2pred)+ h*k3;
}else{
    k1=1/s1*q1-beta1*Math.sqrt(ypred);
    k2=1/s1*q1-beta1*Math.sqrt(ypred)+ h/2*k1;
    k3=1/s1*q1-beta1*Math.sqrt(ypred)+ h/2*k2;
    k4=1/s1*q1-beta1*Math.sqrt(ypred)+ h*k3;
}

if(y2pred > 0){
    k5=beta2*Math.sqrt(ypred-y2pred) - beta3*Math.sqrt(y2pred);
    k6=beta2*Math.sqrt(ypred-y2pred) - beta3*Math.sqrt(y2pred)+ h/2*k5;
    k7=beta2*Math.sqrt(ypred-y2pred) - beta3*Math.sqrt(y2pred)+ h/2*k6;
    k8=beta2*Math.sqrt(ypred-y2pred) - beta3*Math.sqrt(y2pred)+ h*k7;
}else{
    k5=beta2*Math.sqrt(ypred);
    k6=beta2*Math.sqrt(ypred) + h/2*k5;
    k7=beta2*Math.sqrt(ypred) + h/2*k6;
    k8=beta2*Math.sqrt(ypred) + h*k7;
}
```

Regulácia bola v tomto prípade náročnejšia pretože sme naraz museli zohľadňovať výšku hladín dvoch nádob a prítok do nich. Preto sme si navrhli vlastnú regulačnú funkciu, ktorá zachytáva všetky stavy hladín a žiadaných regulácií:

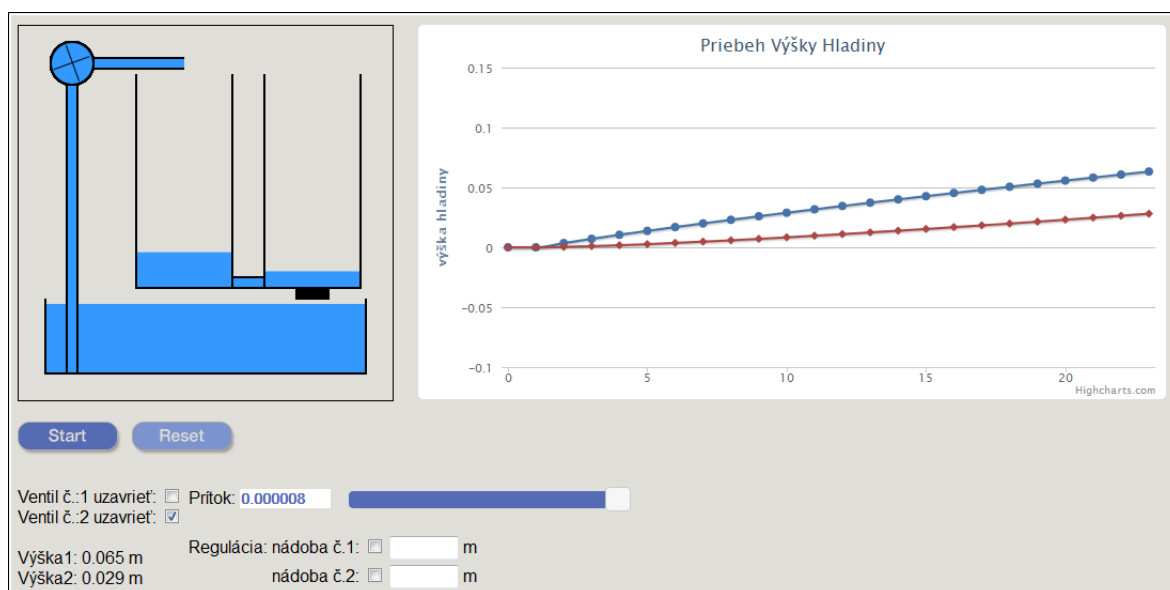
```
if( $('#regulacia1').attr('checked') &&
    $('#regulacia2').attr('checked') ){
    if( vyska < $('#reg_hlad1').val() ){
        if( vyska2 < $('#reg_hlad2').val() ){
            $('#ventil2').attr('checked',true);
            q1 = 0.000008;
        }else if( vyska2 > $('#reg_hlad2').val() ){
            $('#ventil2').attr('checked',false);
            q1 = 0; //funkcia sa rozvetvuje dalej
        }
    }
}
```

Príklad priebehu výšky hladín so zapnutou reguláciou:



Obr. 8. Graf regulácie hladín

Ukážka kompletnej vizualizácie:



Obr. 9. Model: Hydraulický systém dvojhladinový

3.4 Magnetická levitácia

Štvrtou vizualizáciou je sústava magnetickej levitácie. Na opis tejto sústavy sme využili linearizáciu diferenciálnych rovníc.

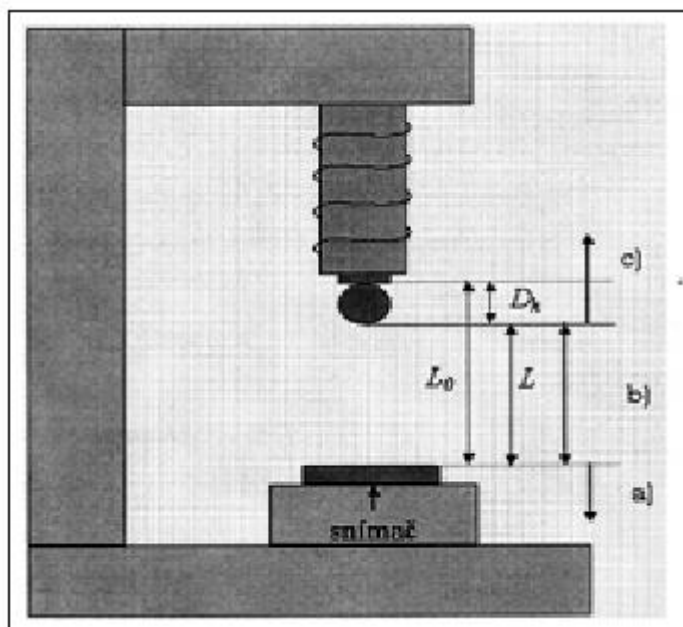
3.4.1 Schéma a opis sústavy

Štvrtou vizualizáciou, ktorú sme spracovali je sústava viacerých podsystemov, ktoré v súčinnosti vytvárajú magnetickú levitáciu.

Jedná sa o tieto:

- Gulička s cievkou
- Zosilňovač
- D/A prevodník
- A/D prevodník
- Senzor polohy

Kompletnú sústavu máme znázornenú na obrázku č. 10. Rozmery tejto sústavy sú iba ilustračné, nakoľko nezodpovedajú pomeru voči reálnej sústave.



Obr. 10 Nákres sústavy magnetickej levitácie

Zjednodušené by sme mohli model opísať diferenciálnou rovnicou:

$$m_k \ddot{x} = \frac{i^2 k}{(x - x_0)^2} - m_k g \quad (3.6)$$

Ak by sme brali do úvahy aj pružnosť materiálu, obmedzenie pohybu v horizontálnom smere a tlmenie pohybu, potom by sme mohli model opísať rovnicou:

$$m_k \ddot{x} + k_{fv} \dot{x} = \frac{i^2 k}{(x - x_0)^2} - m_k g \quad (3.7)$$

Vhodný opis pre linearizáciu vyzerá nasledovne:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{i^2 k_c}{m_k (x - x_0)^2} - g - k_{fv}/m_k x_2 \end{aligned} \quad (3.8)$$

Linearizovaním sústavy dostávame nasledujúcu sústavu:

$$\begin{aligned} \dot{x} &= A_{sys} x + B_{sys} u \\ \dot{y} &= C_{sys} x \end{aligned} \quad (3.9)$$

Matice sústavy majú nasledovný tvar:

$$\begin{aligned} A_{sys} &= \begin{bmatrix} 0 & 1 \\ -102,7332705 & -2,389 \end{bmatrix} \\ B_{sys} &= \begin{bmatrix} 0 \\ 0,001680609913 \end{bmatrix} \\ C_{sys} &= \begin{bmatrix} 1 & 0 \end{bmatrix} \end{aligned} \quad (3.10)$$

Zložky vektora \dot{x} :

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + b_1u \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + b_2u \end{aligned} \quad (3.11)$$

3.4.2 Realizácia vizualizácie

Vizualizáciu magnetickej levitácie sme navrhli proporcionálne t.j. rozmery reálnej sústavy sú v proporcionálnom pomere voči vizualizácii.

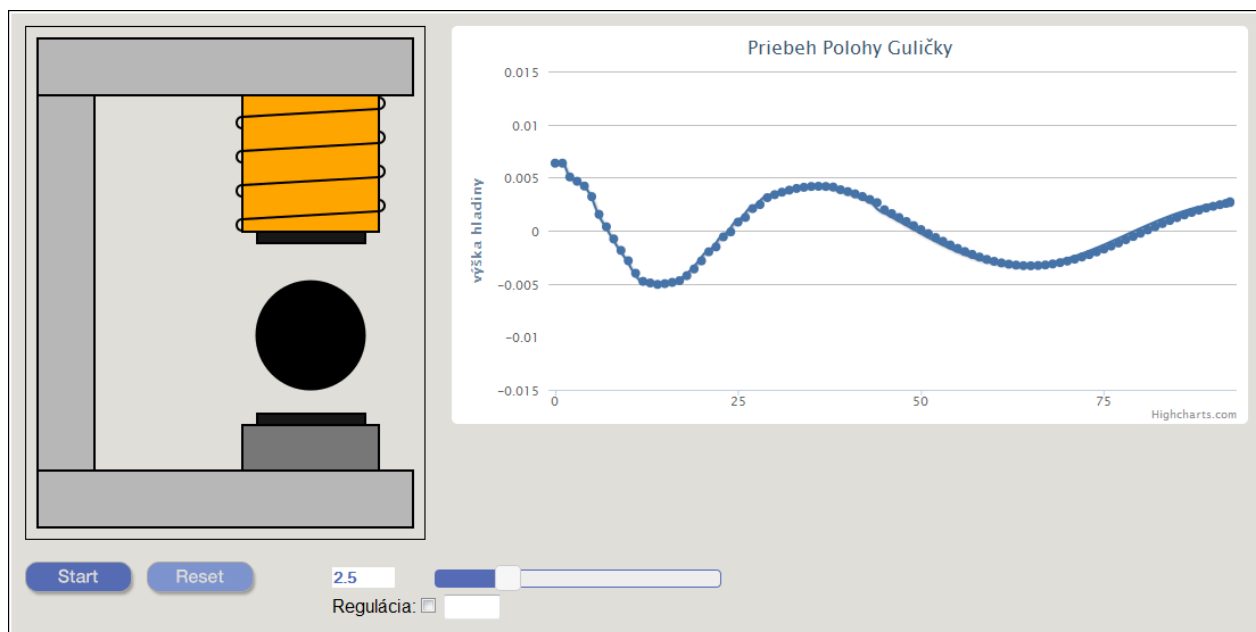
Ukážka vykreslenia vizualizácie:

```
//top snimac
context.beginPath();
context.rect(203, 180, 95, 10);
context.fillStyle = '#171717';
context.fill();
context.stroke();

//bottom snimac
context.beginPath();
context.rect(203, 340, 95, 10);
context.fillStyle = '#171717';
context.fill();
context.stroke();

//gulicka
context.beginPath();
context.arc(centerX, centerY, 47.5, 0, 2 * Math.PI, false);
context.fillStyle = 'black';
context.fill();
context.stroke();

//samotna cievka
//zavit
context.beginPath();
context.arc(310, 67, 5, 1.5*Math.PI, (1/2)*Math.PI, false);
context.lineTo(190, 79);
context.lineWidth = 2;
context.strokeStyle = "black";
context.stroke();
```



Obr. 11. Model: Magnetická levitácia

Užívateľ má možnosť ovládať napätie na cievke pomocou JQuery UI slidera:

```
u = $( "#amount" ).val();
```

Podobne môže danú sústavu aj regulovať. Zaškrtnutím zaškrťavacieho políčka aplikácia preberie hodnotu napätie z vedľajšieho textového políčka a následne s ním kalkulovať.

Rovnica regulácie má tvar:

$$u = -kxx + kww \quad (3.12)$$

V aplikácii vyzerá nasledovne:

```
u = -890906.7587*x1 - 46179.96903*x2 +  
952035.3218*$('#regulacia').val();
```

3.5 Inverzné kyvadlo

Inverzné kyvadlo je sústava, ktorá sa skladá z dvoch podsústav. Sú to vozík, ktorý sa pohybuje po vodorovnej osi a tyč so závažím uchytená pevným bodom na vozíku okolo ktorého môže rotovať. Pri tejto sústave sa riadi poloha vozíka tak aby tyč zvierala uhol 0 stupňov so zvislou osou.

3.5.1 Schéma a opis sústavy

Sústava je opísaná nelineárnymi diferenciálnymi rovnicami:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \beta(x_3)a_{23}\sin x_3 \cos x_3 + a_{22}x_2 + a_{24}\cos x_3 x_4 + a_{25}\sin x_3 x_4^2 + b_2 F \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \beta(x_3)a_{43}\sin x_3 + a_{42}\cos x_3 x_2 + a_{44}x_4 + a_{45}\cos x_3 \sin x_3 x_4^2 + b_4 \cos x_3 F\end{aligned}\tag{3.13}$$

kde:

$$\beta(x_3) = \left(1 + \frac{N^2 \sin^2 x_3}{N_{01}^2}\right)^{-1}$$

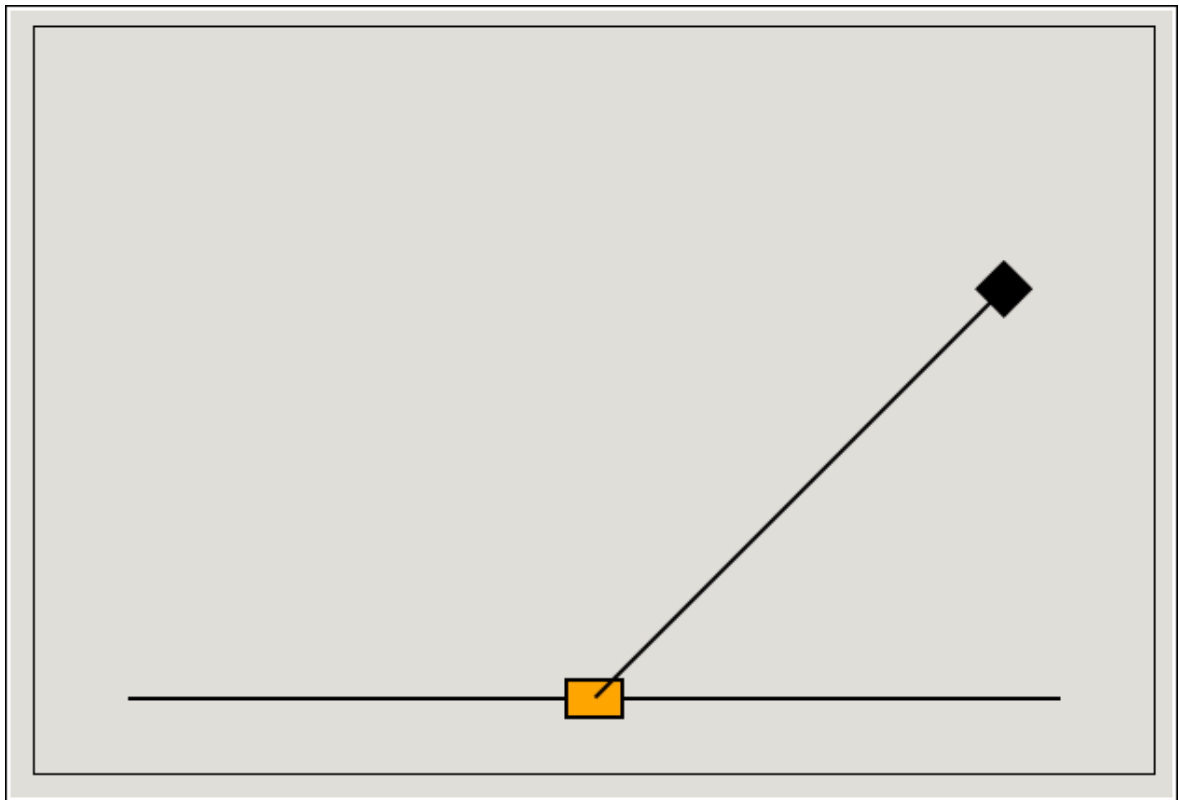
$$\begin{aligned}b_2 &= -\theta/N_{01}^2 \\ b_4 &= -N/N_{01}^2 \\ a_{22} &= -\theta F_r/N_{01}^2 \\ a_{23} &= -N^2 g/N_{01}^2 \\ a_{24} &= NC/N_{01}^2 \\ a_{25} &= \theta N/N_{01}^2 \\ a_{42} &= NF_r/N_{01}^2 \\ a_{43} &= MN g/N_{01}^2 \\ a_{44} &= -MC/N_{01}^2 \\ a_{45} &= -N^2/N_{01}^2\end{aligned}$$

Konštanty sústavy:

Tab. 2: Konštanty sústavy Inverzné kyvadlo

Konštantá	Hodnota	Jednotka
M_0	4	kg
M_1	0,36	kg
M	4,36	kg
I_s	0,451	m
theta	0,0843	kg.m ²
N	0,1624	kg.m
N_{01}	$\sqrt{0,3413}$	kg.m
F_r	10	kg/s
F	0	N
C	0,00145	kg.m ² /S
g	9,81	m.s ⁻²

3.5.2 Realizácia vizualizácie



Obr. 12 Vychýlené kyvadlo

Vykreslenie modelu sme dosiahli klasickými metódami ako pri predchádzajúcich modeloch:

```
//vozik
context.beginPath();
context.rect(285, 350, 30, 20);
context.fillStyle = '#FFA500';
context.fill();
context.stroke();

//kyvadlo
ctx.beginPath();
ctx.translate(555,105);
ctx.rotate(Math.PI / uhol);
ctx.moveTo(0,50);
ctx.lineTo(0,360);
ctx.lineWidth = 2;
ctx.strokeStyle = "black";
ctx.rect(-10, 40, 20, 20);
```

Pri tomto modeli má užívateľ možnosť regulovať vychýlenie vozíka podobne ako tomu bolo v predchádzajúcich modeloch. Ako prvé musí zadať hodnotu vychýlenia vozíka a potom zaškrtnúť zaškrťavacie políčko pre reguláciu. Riešená bola sústava diferenciálnych rovníc štvrtého rádu:

```
function runge1() {  
    k1 = x2;  
    k2 = x2 + h/2*k1;  
    k3 = x2 + h/2*k2;  
    k4 = x2 + h*k3;  
}  
  
function runge2() {  
    i = beta * ( a23*Math.sin(x3)*Math.cos(x3) + a22*x2 +  
        a24*Math.cos(x3)*x4 + a25*Math.sin(x3)*x4*x4 + b2*F );  
    k5 = i;  
    k6 = i + h/2*k5;  
    k7 = i + h/2*k6;  
    k8 = i + h*k7;  
}  
  
function runge3() {  
    k9 = x4;  
    k10 = x4 + h/2*k9;  
    k11 = x4 + h/2*k10;  
    k12 = x4 + h*k11;  
}  
  
function runge4() {  
    j = beta * ( a43*Math.sin(x3) + a42*Math.cos(x3)*x2 + a44*x4 +  
        a45*Math.cos(x3)*Math.sin(x3)*x4*x4 + b4*Math.cos(x3)*F );  
    k13 = j;  
    k14 = j + h/2*k13;  
    k15 = j + h/2*k14;  
    k16 = j + h*k15;  
}
```

Regulácia bola dosiahnutá pomocou linearizovanej rovnice pre regulator a jej tvar môžeme vidieť vo funkcii regulácie:

```
function regulacia(){  
    F = - ( - 0.01313482817 * x1 - 10.10509022 * x2 - 46.76016035 * x3  
            - 4.218843951 * x4 ) + (-0.01313482817*0.2);  
}
```


4 Kompatibilita aplikácie

Aplikácia je kompatibilná vo všetkých moderných internetových prehliadačoch, ktoré podporujú canvas značku jazyka HTML5.

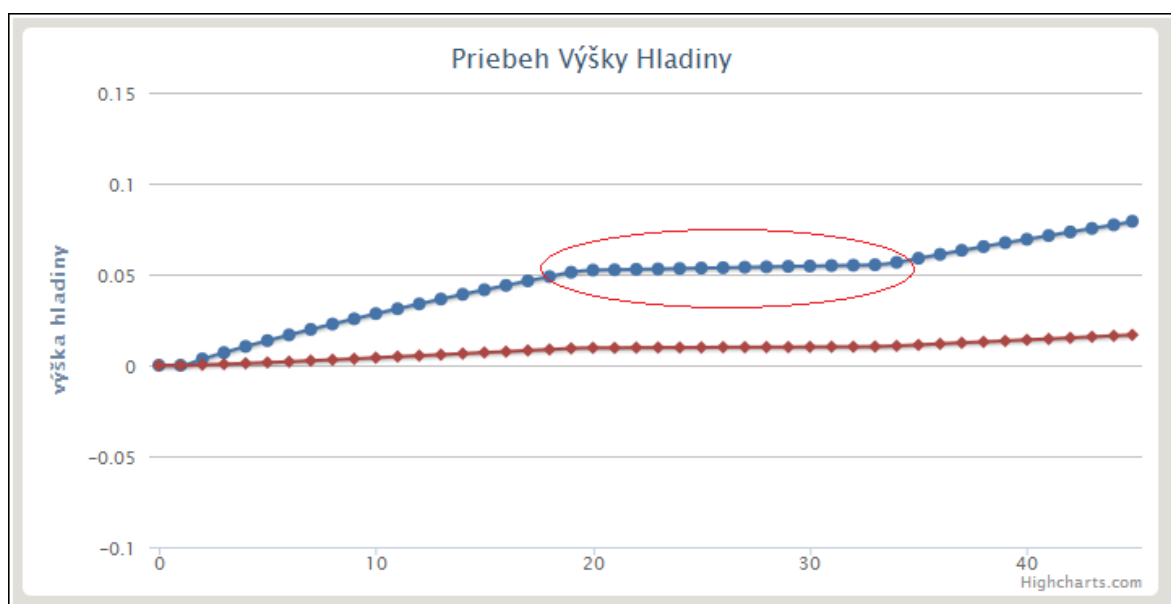
Tab. 3: Kompatibilita v internetových prehliadačoch

Prehliadač	Kompatibilita
Mozilla Firefox	áno
Google Chrome	áno
Internet Explorer	nie
Opera	áno

Táto aplikácia teda funguje vo väčšine internetových prehliadačov a na všetkých operačných systémoch a má teda výhodu voči ostatným aplikáciám renderujúcim dynamické sústavy. Tieto vo väčšine vyžadujú inštaláciu samotného samotnej aplikácie alebo zásuvného modulu potrebného na jej spustenie.

Aplikácia má jedno obmedzenie, ktoré sa týka iba dvoch internetových prehliadačov. Sú to Google Chrome a Mozilla Firefox, ktoré ako prostriedok proti zahlcovaniu zdrojov pamäte pri minimalizovaní okna alebo prepnutí na inú záložku prehliadača, redukovujú frekvenciu funkcie `setInterval()` a tým môže dochádzať ku chybám. V tomto prípade sa zmení počet krokov RK4 metódy voči pevne stanovenému krokovaniu.

Na nasledujúcom obrázku môžeme vidieť zmenu priebehu vyvolanú týmto obmedzením:



Obr. 13 Odchýlka

5 Inštalácia aplikácie

Inštalácia aplikácie je veľmi jednoduchá a spočíva iba v prostom skopírovaní hlavného priečinka aplikácie do adresára s html súbormi na Apache serveri. Všetky knižnice sa inicializujú priamo v kóde a nie sú teda nutné žiadne ďalšie zásahy.

Záver

Cieľom diplomovej práce bolo navrhnuť a realizovať interaktívne animácie dynamických sústav.

Toto sa nám podarilo pre 1 všeobecný a 4 reálne modely dynamických sústav. Všetky animácie boli realizované pomocou JavaScriptu a HTML5 značky “canvas”. Tieto vizualizácie sa nám podarilo navrhnuť pomerne verne a sú teda výraznou konkurenciou v oblasti multimediálnych aplikácií voči technológiám ako Flash alebo SVG.

Navrhovanie vizualizácií nemalo komplikovaný priebeh. Voči hardwarovo náročným aplikáciám nástroja Flash má teda aplikácia vypracovaná ako diplomová práca výhodu aj v stabilite.

Nedostatkom našej aplikácie je absencia 3D rozmeru, ktorá sa dá dosiahnuť v aplikáciách naprogramovaných pomocou jazyka ActionScript alebo Java. Na tomto nedostatku sa momentálne pracuje a v novších verziách internetových prehliadačov bude možné renderovať aj v tretej dimenzii.

Aj vďaka tomu zatiaľ nerozprávame o nahradení technológií Flash, Silverlight alebo SVG, avšak tieto technológie získali vážnu a rýchlo vzrastajúcu konkurenciu ktorá má potenciál vytlačiť tieto technológie z oblasti prehrávania multimediálneho obsahu.

Literatúra

- [1] <http://www.html5canvastutorials.com/>.
- [2] http://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods
- [3] <http://elearn.elf.stuba.sk/moodle/course/view.php?id=600>
- [4] <http://www.asgaard.co.uk/misc/html5canvas.php>
- [5] <http://www.w3schools.com/>
- [6] https://developer.mozilla.org/en/Canvas_tutorial
- [7] http://slovak.evlm.stuba.sk/elearning/elearning_files/Rukovat_studenta/kapitola10.pdf
- [8] <http://www.javascripter.net>
- [9] http://cyberneticsmws.fei.tuke.sk/CyberVirtLab/Magneticka_Levitacia/popis.html
- [10] http://www.kasr.elf.stuba.sk/predmety/rstp/Cvicenia/Navody%20na%20cvicenia/RS_TP_opis%20procesov.pdf
- [11] <http://billmill.org/static/canvastutorial/paddle.html>