

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ TECHNOLOGIE

Využitie MATLAB Builder JA na tvorbu virtuálneho laboratória

DIPLOMOVÁ PRÁCA

FCHPT-5414-21498

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ TECHNOLOGIE

Využitie MATLAB Builder JA na tvorbu virtuálneho laboratória

DIPLOMOVÁ PRÁCA

FCHPT–5414–21498

Študijný program: Automatizácia a informatizácia v chémii a potravinárstve

Číslo a názov študijného odboru: 5.2.14 Automatizácia

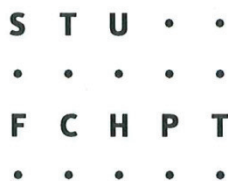
Školiace pracovisko: Oddelenie informatizácie a riadenia procesov

Vedúci záverečnej práce: Ing. Ľuboš Čirka, PhD.

Konzultant: Ing. Martin Kalúz

Bratislava 2011

Bc. Dávid Schmidt



ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Bc. Dávid Schmidt**

ID študenta: 21498

Študijný program: automatizácia a informatizácia v chémii a potravinárstve

Študijný odbor: 5.2.14 automatizácia

Vedúci práce: Ing. Ľuboš Čirka, PhD.

Konzultant: Ing. Martin Kalúz

Názov práce: **Využitie Matlab Builder JA na tvorbu virtuálneho laboratória**

Špecifikácia zadania:

Cieľom diplomovej práce je dobudovanie virtuálneho laboratória založeného na Matlab Builder JA.

Rozsah práce: 50

Riešenie zadania práce od: 14. 02. 2011

Dátum odovzdania práce: 21. 05. 2011

Bc. Dávid Schmidt
Študent



prof. Ing. Miroslav Fikar, DrSc.
Vedúci pracoviska

prof. Ing. Miroslav Fikar, DrSc.
Garant študijného programu

Pod'akovanie

Touto cestou by som sa chcel pod'akovať vedúcemu práce Ing. Ľubošovi Čirkovi, PhD. a konzultantovi Ing. Martinovi Kalúzovi za neoceniteľné rady a odbornú pomoc pri vypracovávaní tejto práce.

Čestné prehlásenie

Čestne prehlasujem, že som diplomovú prácu vypracoval samostatne, podľa pokynov vedúceho práce a s použitím zdrojov uvedených v zozname literatúry.

V Bratislave, 10. mája. 2011

Bc. Dávid Schmidt

Abstrakt

Práca sa zaoberá vytvorením on-line virtuálneho laboratória v prostredí MATLAB Builder JA (MBJA). Virtuálne laboratórium obsahuje riadenie výmenníkov tepla a zásobníkov kvapaliny. Teoretická časť opisuje možnosti uverejňovania napísaných m-funkcií na internete a použité metódy. Praktická časť sa zaoberá inštaláciou a konfiguráciou potrebných komponentov ako Java Development Kit (JDK), Java Runtime Environment (JRE), MATLAB a MATLAB Compiler Runtime (MCR), webového servera Apache Tomcat, kompiláciou vytvorených funkcií a ich umiestnením na internetový portál.

Kľúčové slová: MATLAB Builder, Java aplikácie, Virtuálne laboratórium

Abstract

Thesis deals with creation of on–line virtual laboratory in MATLAB Builder JA (MBJA) environment. Virtual laboratory contain heat exchange control and control of fluid tank system. Theoretical part describes various opportunities of publishing written m–functions and used methods. Practical part describes how to install and configure components like Java Development Kit (JDK), Java Runtime Environment (JRE), MATLAB and MATLAB Compiler Runtime (MCR), web server Apache Tomcat, compilation of created functions and their publishing on internet portal.

Keywords: MATLAB Builder, Java Applications, Virtual Laboratory

Obsah

Zoznam obrázkov	9
Zoznam symbolov a skratiek	10
Zoznam príloh.....	11
1 Úvod	12
1.1 Cieľ práce	12
2 Teoretický úvod	13
2.1 Metóda Runge–Kutta.....	13
2.2 PID regulátor	15
2.2.1 Základy regulačného obvodu.....	15
2.2.2 Diskrétny tvar PID regulátora.....	17
2.3 Možnosti zdieľania funkcií vytvorených v MATLABe	18
3 Praktická časť - príprava prostredia.....	20
3.1 Java Development Kit	21
3.1.1 Java Runtime Environment.....	23
3.2 MATLAB Compiler Runtime (MCR)	25
3.3 Apache Tomcat.....	28
3.4 Nastavenie systémových premenných.....	30
3.5 Premenné programu MATLAB.....	32
3.5.1 Konfigurácia MATLAB Compilera.....	32
4 Vytváranie virtuálneho laboratória	34
4.1 Zásobníky kvapaliny.....	34
4.1.1 Matematický model zásobníkov kvapaliny s interakciou.....	34
4.1.2 Matematický model zásobníkov kvapaliny bez interakcie	39
4.1.3 M–funkcia zásobníkov kvapaliny	44
4.2 Výmenníky tepla.....	48
4.2.1 Matematický model výmenníkov tepla.....	48
4.2.2 M–funkcia výmenníkov tepla	52
4.3 MATLAB Builder JA	53
4.3.1 Kompilácia funkcie zásobníkov kvapaliny	53
4.3.2 Kompilácia funkcie výmenníkov tepla	54
4. 4 Webové rozhranie.....	55
4.4.1 Implementácia javabuilder.jar do Apache Tomcat	55

4.4.2 Základné časti .jsp stránky	56
4.4.2 Zásobníky kvapaliny - internetové rozhranie	58
4.4.3 Výmenníky tepla - internetové rozhranie	59
5 Záver	60
6 Zoznam použitej literatúry	61

Zoznam obrázkov

Obr. 1 Schéma aproximácie y_{n+1} pre krok h	13
Obr. 2 Schéma uzavretého regulačného obvodu	16
Obr. 3 Možnosti distribúcie algoritmov programu MATLAB.....	18
Obr. 4 Úvodná obrazovka inštalačného programu JDK.....	21
Obr. 5 Možnosť vybratia destinácie inštalácie JDK	22
Obr. 6 Extrahovanie súborov do zadanej destinácie.....	22
Obr. 7 Možnosť vybratia destinácie inštalácie JRE.....	23
Obr. 8 Úprava inštalačného adresára.....	24
Obr. 9 Extrahovanie súborov JRE na disk	24
Obr. 10 Okno ukončenia inštalácie balíka JDK	25
Obr. 11 Okno výberu jazyka pre inštaláciu MCR	25
Obr. 12 Zoznam chýbajúcich komponentov	26
Obr. 13 Úvod inštalácie MCR	26
Obr. 14 Okno výberu inštalačného adresára	27
Obr. 15 Ukončenie inštalácie MCR	27
Obr. 16 Výber komponentov inštalácie	28
Obr. 17 Základné pred-inštalačné nastavenia	29
Obr. 18 Zvolenie destinácie pre nakopírovanie súborov	29
Obr. 19 Pridanie premennej JAVA_HOME.....	30
Obr. 20 Pridanie premennej path pre príkaz javac	31
Obr. 21 Overenie funkčnosti prostredia cez systémový príkazový riadok	31
Obr. 22 Premenná Path operačného systému	32
Obr. 23 Nastavenie kompilátora pre MATLAB.....	33
Obr. 24 Schéma zásobníkov kvapaliny s interakciou.....	34
Obr. 25 Schéma zásobníkov kvapaliny bez interakcie.....	39
Obr. 26 Schéma výmenníkov tepla	48
Obr. 27 Nastavenie triedy s vloženou funkciou v Deployment Tool.....	54
Obr. 28 Testovacia stránka.....	56
Obr. 29 Internetové rozhranie pre riadenie zásobníkov kvapaliny.....	58
Obr. 30 Internetové rozhranie pre riadenie výmenníkov tepla	59

Zoznam symbolov a skratiek

MBJA	MATLAB Builder for Java Applications
MWS	MATLAB Web Server
MBNE	MATLAB Builder for Microsoft .NET Framework
PID	Proporcionálno-Integračno-Derivačný
MBEX	MATLAB Builder for Microsoft Excel
JDK	Java Developer Kit
JRE	Java Runtime Environment
MCR	Matlab Compiler Runtime
JAR	Java ARchív
J2EE	Java 2 Platforma, Enterprise Edition
HTTP	HyperText Transfer Protocol
DMM	Dynamický Matematický Model
MMRS	Matematický Model v Rovnovážnom Stave
%matlabroot%	Inštalačný priečinok programu Matlab (napr.: c:\MATLAB\R2009b\)
%apacheroot%	Inštalačný priečinok Apache Tomcat (napr.: C:\ApacheSoftwareFoundation\Tomcat6.0)
JSP	JavaServer Page
WAR	Web Application aRchive
URL	Uniform Resource Locator

Zoznam príloh

Príloha č. 1:

CD obsahujúce vytvorené aplikácie, zdrojové súbory, obrázky a elektronickú verziu práce.

1 Úvod

[1]

MATLAB Builder JA (MBJA) je súčasť programového balíčka MATLAB. MATLAB samotný slúži na rôzne matematické výpočty, vykresľovanie grafov funkcií (2D a 3D) a simulácie procesov potrebných pri technických odvetviach dnešnej doby. Možno v ňom napísať funkcie, ktoré sa dajú neskôr použiť na výpočet s určitými vstupmi. MBJA je komponent, ktorý takúto funkciu obalí do Javy a umožní ju tak ďalej používať.

Pôvodné virtuálne laboratórium použité v Moodle, je vytvorené v prostredí MATLAB Web Server (MWS). Ten slúžil na prístup MATLAB aplikácií cez internetový prehliadač. Avšak od verzie 2006b je podpora pre MWS zrušená. Nahradený je niekoľkými možnosťami ako sprístupniť tieto aplikácie cez web:

- MATLAB Builder NE (využíva technológiu .NET)
- MATLAB Builder JA (využíva technológiu Java)

Všetky súbory v tejto práci sú kompilované pod operačným systémom Windows XP (32bit) a MATLAB s verziou 2009b.

1.1 Cieľ práce

Cieľom práce je vytvorenie virtuálneho laboratória v prostredí MATLAB Builder JA. Jednotlivé m-funkcie budú zabalené v Java kóde čím vzniknú Java archívy, ktoré budú umiestnené na e-learningový portál Moodle. Ich výstup potom bude zobrazený cez webový prehliadač bez potreby mať nainštalované súčasti MATLABu. Práca má opísať postup inštalácie jednotlivých súčastí, konfiguráciu systému a programov, ako správne skompilovať a umiestniť vzniknuté archívy na webový portál.

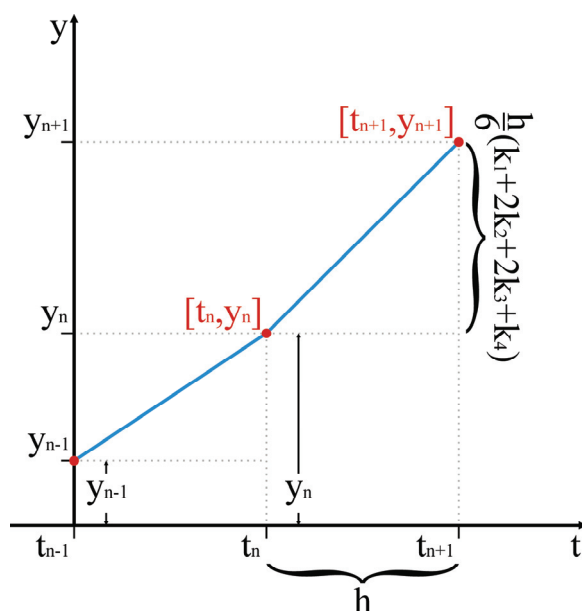
2 Teoretický úvod

Veľa diferenciálnych rovníc nie je možné vyriešiť analyticky. K výsledku sa dá dopracovať aproximáciou za pomoci rôznych algoritmov. Metódy Runge–Kutta sú dôležitým zoskupením aproximačných iteračných metód pre riešenie diferenciálnych rovníc. V roku 1895 bola publikovaná prvá Runge–Kutta metóda Carlom Runge. O 10 rokov, Martin Kutta opísal metódu Runge–Kutta štvrtého rádu.

2.1 Metóda Runge–Kutta

[2, 3, 4, 5]

Táto metóda je založená na rozvoji funkcie do Taylorovho radu a berie do úvahy aj vyššie členy rozvoja. Derivácie funkcie $f(t, y)$ sú rátané zložitejšou diferenčnou metódou za pomoci ďalších bodov medzi jednotlivými uzlami v sieti. Najpoužívanejšia z týchto metód je metóda Runge–Kutta štvrtého rádu označovaná tiež ako $RK4$, alebo klasická metóda Runge–Kutta (resp. len metóda Runge–Kutta).



Obr. 1 Schéma aproximácie y_{n+1} pre krok h

Majme diferenciálnu rovnicu so začiatočnou podmienkou:

$$y' = f(t, y) \quad (1)$$

$$y(t_0) = y_0 \quad (2)$$

Riešenie rovnice (1) pomocou metódy Runge–Kutta bude v tvare:

$$y_{n+1} = y_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \quad (3)$$

$$t_{n+1} = t_n + h \quad (4)$$

kde y_{n+1} je aproximácia $y(t_{n+1})$, h je krok (obr. 1) a jednotlivé konštanty sú definované nasledovne:

Hodnota na začiatku intervalu h (t_n):

$$k_1 = f(t_n, y_n) \quad (5)$$

Hodnota v strede intervalu h , bližšie k y_n , hodnota prírastku je daná k_1 :

$$k_2 = f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) \quad (6)$$

Hodnota v strede intervalu h , bližšie k y_{n+1} , hodnota prírastku je daná k_2 :

$$k_3 = f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2\right) \quad (7)$$

Na konci intervalu h , hodnota prírastku je daná k_3 :

$$k_4 = f(t_n + h, y_n + hk_3) \quad (8)$$

Konečná hodnota y_{n+1} je daná súčtom hodnoty y_n , intervalom h a vypočítaných konštánt (čiastkových prírastkov). Prírastok je váženým priemerom jednotlivých konštánt, pričom väčšia váha je prisudzovaná konštantám v strede, teda k_2 a k_3 :

$$Pr = \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \quad (9)$$

Metóda RK4 je štvrtého rádu a preto chyba každého kroku je h^5 (v tejto práci je používaná hodnota kroku $h = 0,05$ a teda chyba $O(0,05^5) = 3,125 \cdot 10^{-7}$). Túto chybu treba brať s rezervou, pretože skutočná chyba závisí od komplexnosti reálnej trajektórie daného systému.

2.2 PID regulátor

[6, 7, 8, 9, 10]

Proporcionálno–integračno–derivačný regulátor je v priemysle jedným z najviac využívaných regulátorov. PID regulátor spracúva regulačnú odchýlku (e) ako rozdiel medzi výstupnou veličinou (y) a žiadanou hodnotou (w). Úpravou vstupov do procesu sa regulátor snaží zmenšiť regulačnú odchýlku.

Algoritmus PID regulátora je zložený z troch rôznych zložiek – proporcionálnej (P) zložky, integračnej (I) a derivačnej (D) zložky. Z časového hľadiska P závisí od súčasných chýb, I ako akumulácia predošlých chýb a D ako predpoklad budúcich chýb. Tieto koeficienty vplyvajú na proces cez riadiaci element (otvorenie ventilu, množstvo dodaného prúdu alebo tepla).

Bez dostatočnej znalosti riadeného procesu je PID regulátor ten najlepší. Úpravou spomínaných parametrov v PID regulátore, je možné riadiť proces podľa požiadaviek. Odozva regulátora môže byť tiež opísaná ako rýchlosť odozvy na regulačnú odchýlku, preregulovanie vzhľadom na žiadanú hodnotu a osciláciu systému. PID regulátor však nezaručuje optimálne riadenie systému či jeho stabilitu.

Prítomnosť všetkých troch zložiek regulátora nie je nevyhnutná pre všetky procesy. Ak niektorá zložka nie je potrebná, jej konštanta je nulová a preto sa PID regulátor označuje tiež ako P, I, PI alebo PD. Najčastejšie používaným regulátorom je PI, pretože chýbajúca derivačná zložka je náchylná na šum a integračná zložka eliminuje regulačnú odchýlku.

2.2.1 Základy regulačného obvodu

[8]

Jednoduchým príkladom regulačného obvodu je riadenie teploty vody za pomoci otvorenia teplého vodovodného kohútika.

Zmeraná teplota na výstupe je výstupná veličina (y). Žiadaná hodnota (w) je teplota vody, ktorú chceme dosiahnuť. Vstup do procesu (pootvorenie ventilu teplej vody) je riadiaca veličina (u). Rozdiel medzi žiadanou a výstupnou veličinou je regulačná odchýlka (e), ktorá určuje či je voda studená alebo teplá, a ako veľmi.

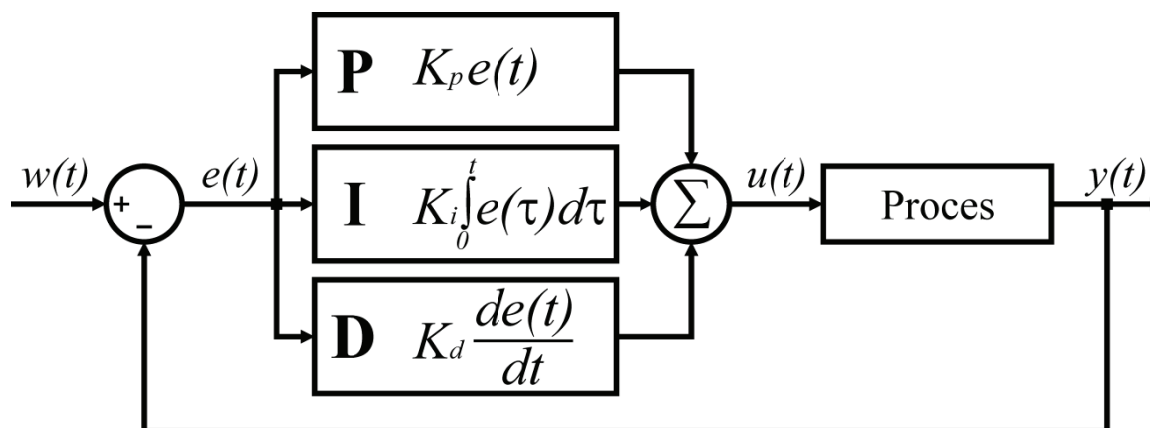
Po zmeraní teploty (y) a vypočítaní regulačnej odchýlky, regulátor vypočíta či je nutné zmeniť polohu ventilu a o koľko. Regulátor môže pootvoriť ventil menej ak má byť voda teplá, alebo viac ak je požadovaná horúca voda. To je príklad

proporcionálneho riadenia. Ak by voda nebola dostatočne horúca, regulátor by sa pokúšal urýchliť proces a otvoril by ventil ešte viac. To je príklad integračného riadenia.

Ak je zosilnenie veľké a regulačná odchýlka malá, dochádza k preregulovaniu. Pri opakovanom preregulovaní, v okolí rovnovážnej hodnoty, vplyvom zmien regulátora hovoríme o oscilácii. Tieto oscilácie môžu byť ustálené, zväčšujúce alebo zmenšujúce sa. Systém, ktorého kmity sa zväčšujú s časom sa nazýva nestabilným, kým systém ktorého kmity sa zmenšujú sa nazýva stabilným. Systém je na hranici stability ak sú jeho kmity ustálené.

Regulátor kmity tlmí, aby systém v konečnom dôsledku konvergoval k žiadanej hodnote – teplote (w). To je príklad derivačného riadenia.

V teórii riadenia je možné riadiť akýkoľvek proces, ktorého výstup (y) sa dá zmerať, je známa ideálna žiadaná hodnota (w), na ktorú je potrebné riadiť a vstupnú hodnotu (u), ktorá bude vhodne ovplyvňovať (y).



Obr. 2 Schéma uzavretého regulačného obvodu

Ako je zobrazené na obr. 2, PID regulátor je pomenovaný podľa svojich súčastí, ktorých suma udáva vstup do procesu (u). Algoritmus PID regulátora pri definovaní $u(t)$, ako výstupu z regulátora (a vstupu do procesu) je potom:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (10)$$

kde:

K_p – konštanta zosilnenia

K_i – integračná konštanta

K_d – derivačná konštanta

e – regulačná odchýlka ($e = w - y$)

t – čas

Existuje niekoľko spôsobov ako získať parametre regulátora:

– analyticky

- Metóda štandardných tvarov
- Naslinova metóda
- Strejcova metóda

– experimentálne

- Metóda pokus – omyl
- Ziegler–Nicholsová metóda
- Metóda trvalých kmitov

2.2.2 Diskrétny tvar PID regulátora

[8]

V tejto práci je použitá diskretizovaná forma PID regulátora, ktorá postačuje pre aproximáciu jednoduchých diferenciálnych rovníc, s periódou vzorkovania Δt .

Diskretizovaná I zložka:

$$\int_0^{t_k} e(\tau) d\tau = \sum_{i=1}^k e(t_i) \Delta t \quad (11)$$

Diskretizovaná D zložka:

$$\frac{de(t_k)}{dt} = \frac{e(t_k) - e(t_{k-1})}{\Delta t} \quad (12)$$

Po úpravách rovnice (10) je konečný tvar PID regulátora pre k -ty krok v tvare:

$$u(t_k) = u(t_{k-1}) + K_p \left[\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right) e(t_k) + \left(-1 - \frac{2T_d}{\Delta t} \right) e(t_{k-1}) + \frac{T_d}{\Delta t} e(t_{k-2}) \right] \quad (13)$$

kde:

T_i – integračná konštanta

T_d – derivačná konštanta

Δt – rovné kroku h pri RK4

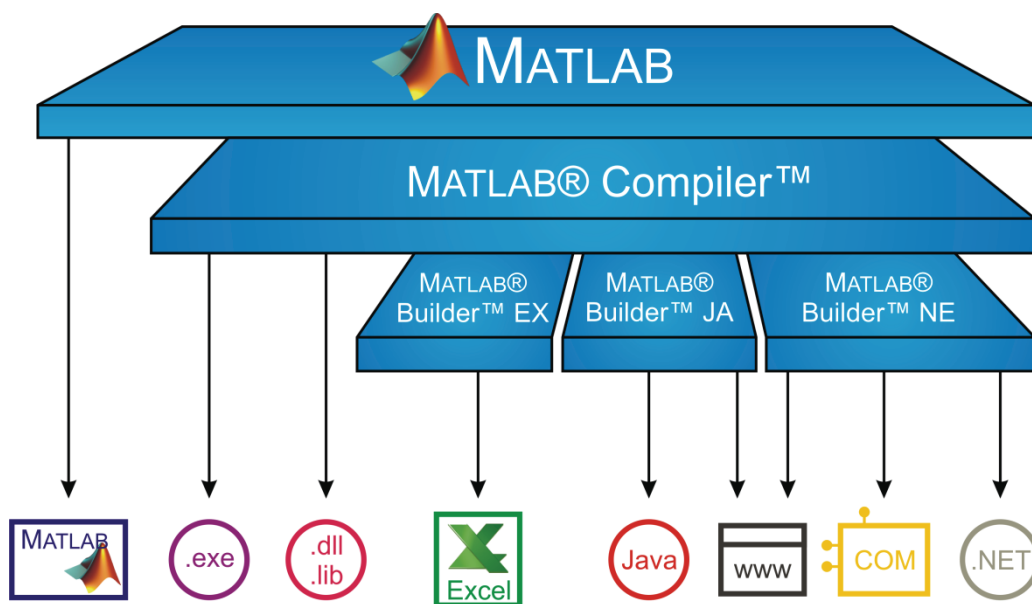
$u(t_k)$ – výstup z regulátora v k -tom kroku

$e(t_k)$ – regulačná odchýlka vstupujúca do regulátora v k -tom kroku

2.3 Možnosti zdieľania funkcií vytvorených v MATLABe

[11]

Najjednoduchším spôsobom, ako sa podeliť o napísané funkcie v MATLABe, je sprístupnenie samotného zdrojového kódu. Na to, aby tento kód bolo možné použiť, je potrebné mať nainštalovaný MATLAB. Pomocou MATLAB Compiler je možno zabaliť jednotlivé funkcie do spustiteľných súborov (*.exe) alebo knižníc (*.dll, *.lib), použiteľné iným programom. Pri kombinácii MATLAB Compiler a MATLAB Buildera možno funkcie zabaliť tak, aby boli použiteľné napríklad v programe Microsoft Excel (pomocou MATLAB Builder EX). Pomocou MATLAB Builder JA sa funkcie zabalia do *.jar súborov, ktoré obsahujú skompilované Java triedy a MATLAB Builder NE je schopný zabaliť funkcie do objektov COM alebo .NET komponentov. [12, 13] Na spustenie jednotlivých skompilovaných funkcií je nutné mať nainštalovaný MATLAB, alebo MATLAB Compiler Runtime (je priložený k distribúcii MATLABu).



Obr. 3 Možnosti distribúcie algoritmov programu MATLAB [14]

Ako je z obr. 3 vidieť, funkcie skompilované pomocou MBEA a MBNE možno umiestniť na web. Webovému serveru bude stačiť, aby obsahoval MATLAB Compiler Runtime spolu s podporou pre použitú technológiu (Java alebo .NET) a funkcie sú pripravené pre užívateľa. Samotný užívateľ tak nepotrebuje mať nainštalovanú žiadnu súčasť MATLABu, pretože všetky výpočty sú vykonávané na serverovej strane a užívateľovi sa ukáže len výstup (napríklad graf). Aby sa v prehliadači zobrazil výstup správne, treba mať nainštalovanú Javu, alebo .NET Framework.

V tejto práci je zvolená technológia Java, pretože je open–source a platforma nehrá veľkú úlohu. Na domovskej stránke Java.com v sekcii Download si možno stiahnuť distribúciu pre Windows, Solaris či Linux (Mac dodáva vlastnú verziu Java spolu so zvyškom aktualizácií). Napriek rôznym možnostiam implementácie .NET na iné operačné systémy (napr.: Project Mono www.mono-project.com [15]), je spoločnosťou Microsoft plne podporovaný len pre systémy Windows. [16] Jednou z nevýhod Javy je, že z celkového hľadiska zaťažuje systémové prostriedky viac ako C++ programy, no od prvej verzie sa výkon zreteľne zlepšil. [17]

3 Praktická časť - príprava prostredia

[18]

Aby bolo možné z jednotlivých MATLAB funkcií vytvoriť Java triedy je potrebné nainštalovať niektoré programové balíky, ktoré sú dostupné na internete, alebo sú súčasťou MATLABu.

Základné programy k inštalácii:

- MATLAB
- MATLAB Builder JA
- Java Development Kit (JDK) a Java Runtime Environment (JRE)

JRE je súčasťou JDK, ale JDK nie je súčasťou inštalačného balíka JRE.

Táto programová výbava umožní vytvoriť a spúšťať triedy na lokálnom počítači. Aby sa vytvorené aplikácie dali spúšťať aj na iných počítačoch, je potrebné nainštalovať:

- MATLAB alebo MATLAB Compiler Runtime (MCR)
- Java Runtime Environment

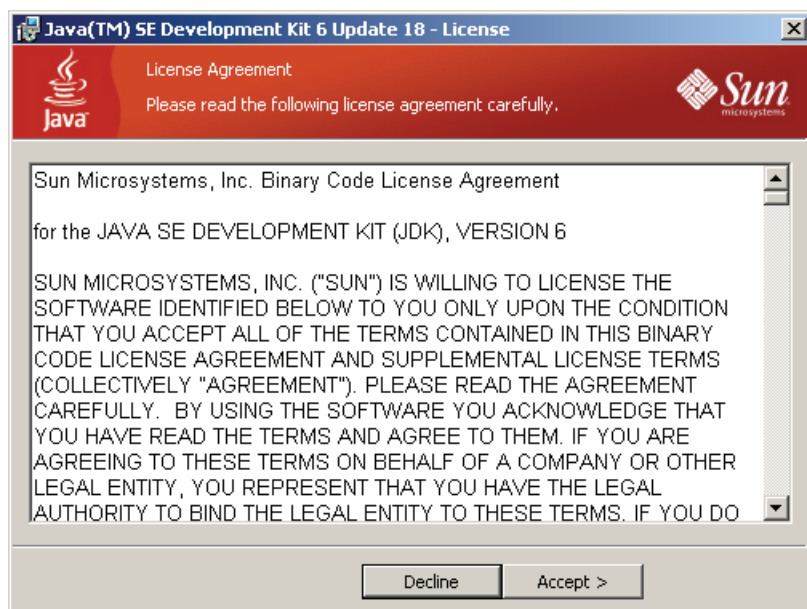
Ďalším krokom je nakonfigurovanie systémových premenných.

Je nutné podotknúť, že vytvorené triedy v určitej verzii MATLABu budú fungovať s komponentmi dodávanými spolu s touto verziou (pri zmene verzie MATLABu stačí nanovo skompilovať funkcie a nakopírovať potrebné súčasti ako je opísané v práci).

3.1 Java Development Kit

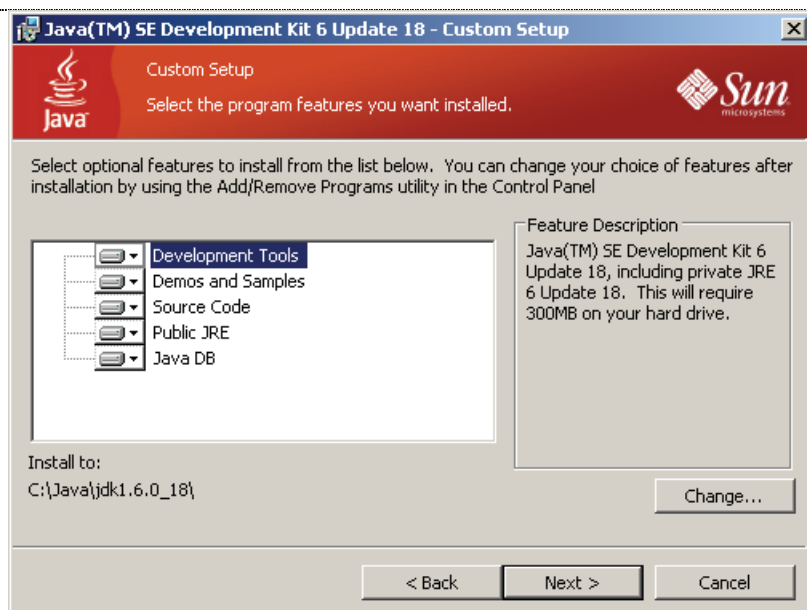
Program je dostupný na <http://java.sun.com/>, kde sa v hornej časti nachádza rozbaľovacia ponuka *Downloads* a *Java SE* (Standard Edition). Otvorí sa nová stránka, kde sú priame odkazy na JDK a JRE. Ako je napísané v predchádzajúcej stati, inštalačný balíček JDK obsahuje aj JRE a teda jeho inštalácia bude inicializovaná hneď po nainštalovaní JDK. JRE je možné nainštalovať ako prvý a potom JDK, je len nutné si dávať pozor, kam sa program nainštaluje, pretože medzera v ceste môže spôsobiť problémy pri spúšťaní vytvorených Java archívov.

Po stiahnutí a otvorení súboru JDK sa inštalácia začína potvrdením licenčnej zmluvy, ktorú si môžeme prečítať a stlačiť [*Accept >*] pre pokračovanie (obr. 4).



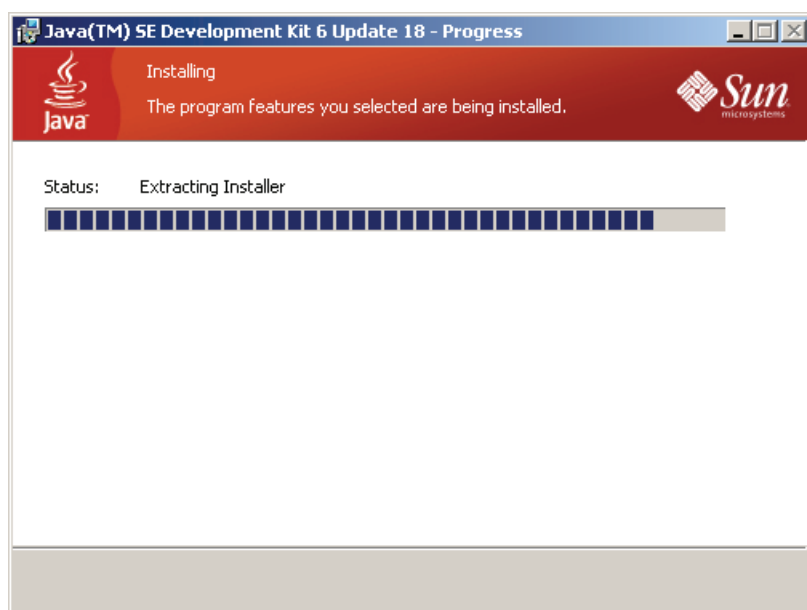
Obr. 4 Úvodná obrazovka inštalačného programu JDK

Po zvolení [*Accept >*] môžeme vybrať adresár kam sa program nainštaluje. Cestu je možné zmeniť stlačením [*Change...*] (obr. 5), upravíme umiestnenie na *C:\Java\jdk_verzia* (napr.: "*C:\Java\jdk1.6.0_18*") a potvrdíme zmeny stlačením [*OK*]. Dôvod zmeny cesty je ten, že pri spúšťaní vytvorených tried, Java nemusí pracovať správne kvôli medzere v názve priečinku *Program Files* a vypíše chybu namiesto vykonania príkazu.



Obr. 5 Možnosť vybratia destinácie inštalácie JDK

Po skontrolovaní vybraných súčastí JDK stlačíme [Next >] čím sa spustí inštalácia (obr. 6).



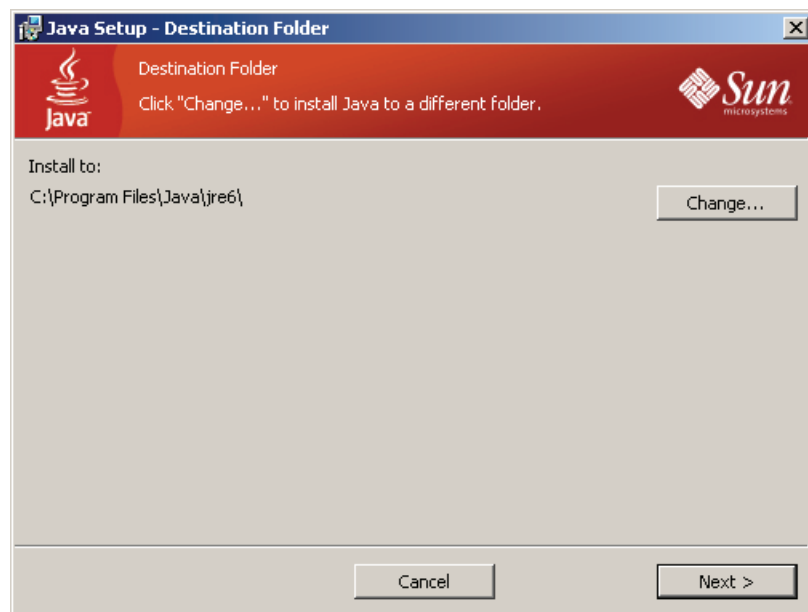
Obr. 6 Extrahovanie súborov do zadanej destinácie

Po skončení kopírovania súborov sa automaticky spustí inštalátor JRE. Ak by bol JRE nainštalovaný ako prvý, objavilo by sa okno pre ukončenie inštalácie.

3.1.1 Java Runtime Environment

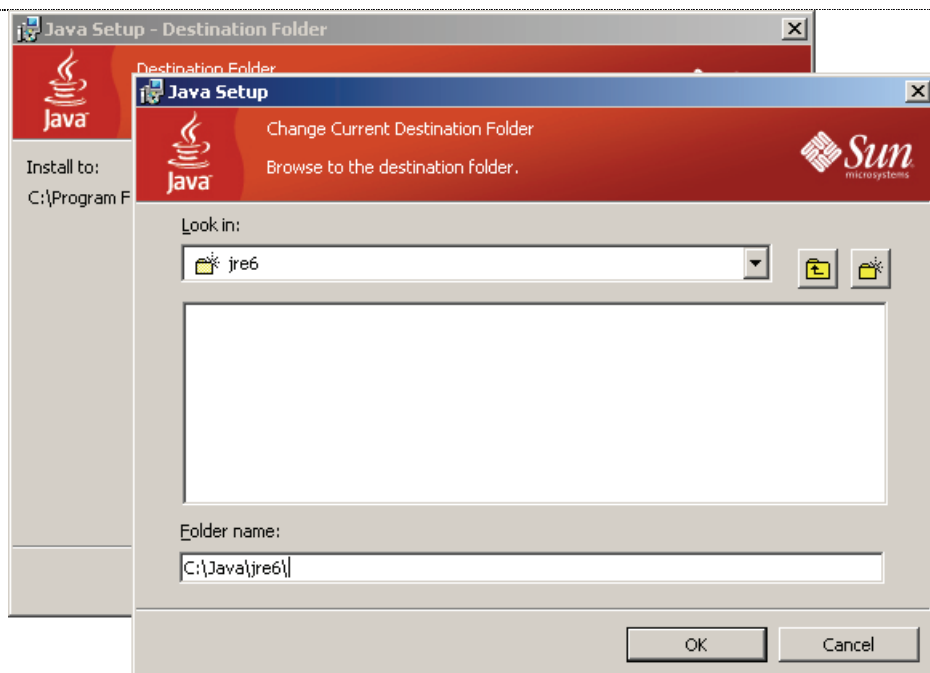
JRE je možné stiahnuť rovnakým spôsobom ako JDK. Bez tejto súčasti by sme neboli schopní vytvorené triedy spúšťať a je tiež potrebný pre prehliadanie niektorých internetových stránok založených na technológii Java.

Ak by inštalácia bola inicializovaná samostatne a nie automaticky, na začiatku by bolo treba potvrdiť licenčnú zmluvu a dobré je aj zaškrtnúť zmenu inštalačného adresára a stlačiť [*Install >*] pre pokračovanie.



Obr. 7 Možnosť vybratia destinácie inštalácie JRE

V ďalšom okne je potrebné zmeniť adresár kam sa súbory nakopírujú (obr. 7). Stlačíme [*Change...*] a zmeníme prednastavenú cestu na *C:\Java\jre6* (obr. 8). Priechinky si inštalátor vytvorí sám.



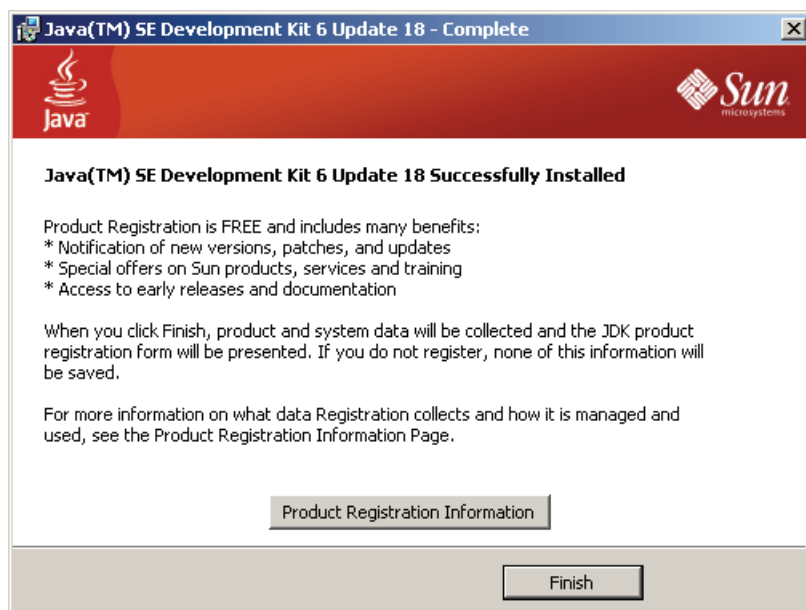
Obr. 8 Úprava inštalačného adresára

Stlačíme [OK] a potom [Next >] a začnú sa kopírovať súbory (obr. 9).



Obr. 9 Extrahovanie súborov JRE na disk

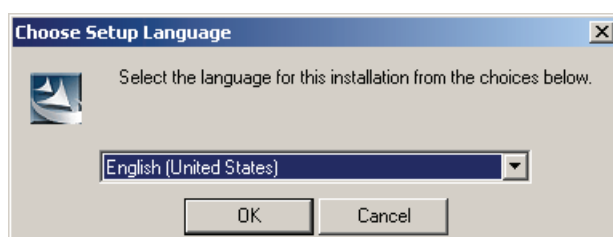
Nakoniec sa zobrazí okno ukončenia úspešnej inštalácie (obr. 10) a stlačením [Finish] sa ukončí inštalácia.



Obr. 10 Okno ukončenia inštalácie balíka JDK

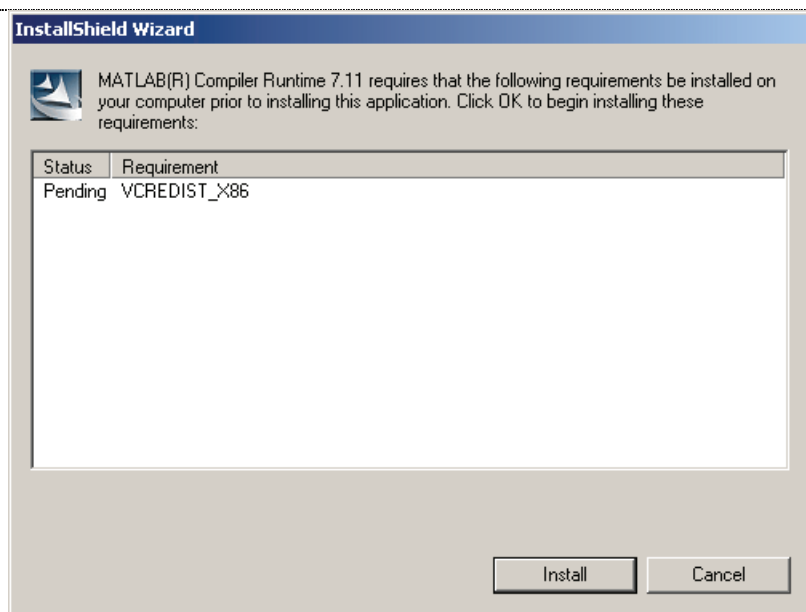
3.2 MATLAB Compiler Runtime (MCR)

Inštalačný súbor tejto pomocnej aplikácie sa nachádza v priečinku `%matlabroot%\toolbox\compiler\deploy\win32\` a je dodávaný s každou novšou verziou MATLABu. Po spustení si vyberieme jazyk (obr. 11) a stlačíme [OK].



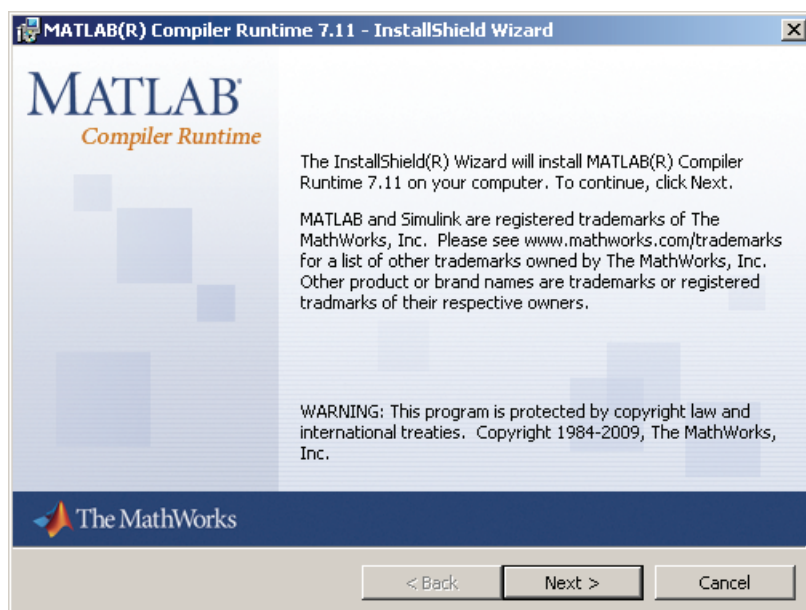
Obr. 11 Okno výberu jazyka pre inštaláciu MCR

MCR si skontroluje či sú prítomné všetky potrebné súčasti, ktoré musia byť nainštalované predtým ako sa začnú kopírovať jeho vlastné súbory.



Obr. 12 Zoznam chýbajúcich komponentov

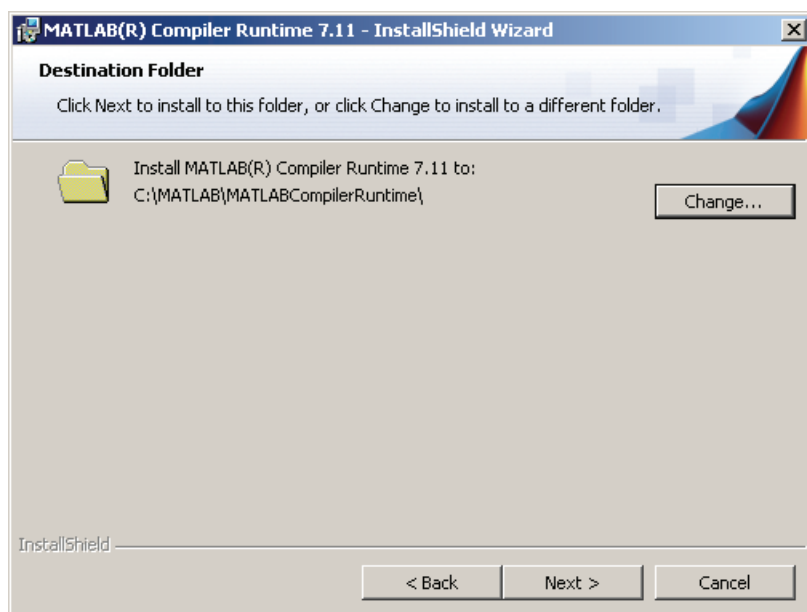
Chýbajúcou súčasťou je *Microsoft Visual C++ 2005 Redistributable* (obr. 12), ktorý možno získať zo stránok spoločnosti Microsoft a nainštalovať ho ručne, alebo stlačíme [Install] pre jeho inštaláciu. Ďalej sa už pokračuje v inštalácii MCR (obr. 13).



Obr. 13 Úvod inštalácie MCR

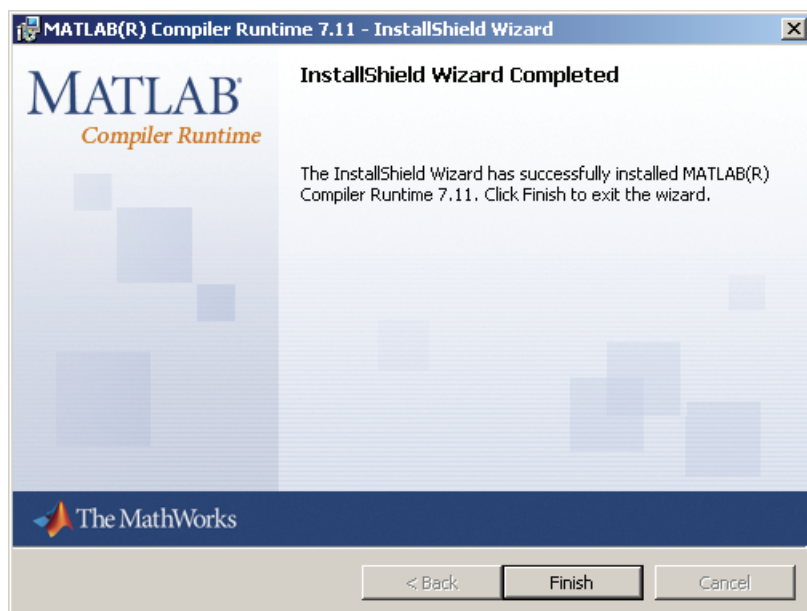
Kliknutím na [Next >] sa dostaneme na okno, kde vyplníme meno používateľa a názov organizácie a klikneme na [Next >]. V ďalšom kroku je možné zmeniť cestu, kam sa nakopírujú súbory inštalácie, stlačíme [Change...] a prepíšeme názov priečinku napríklad na *C:\MATLAB\MATLABCompilerRuntime* (obr. 14).

Skontrolujeme cestu a potvrdíme kliknutím na [OK] a potom na [Next >].



Obr. 14 Okno výberu inštalačného adresára

Posledným krokom je oznámenie, že program je pripravený na inštaláciu, kde stačí už len kliknúť na [Install]. Po skopírovaní všetkých súborov a nakonfigurovaní systému stačí kliknúť na [Finish] a inštalácia je ukončená (obr. 15).

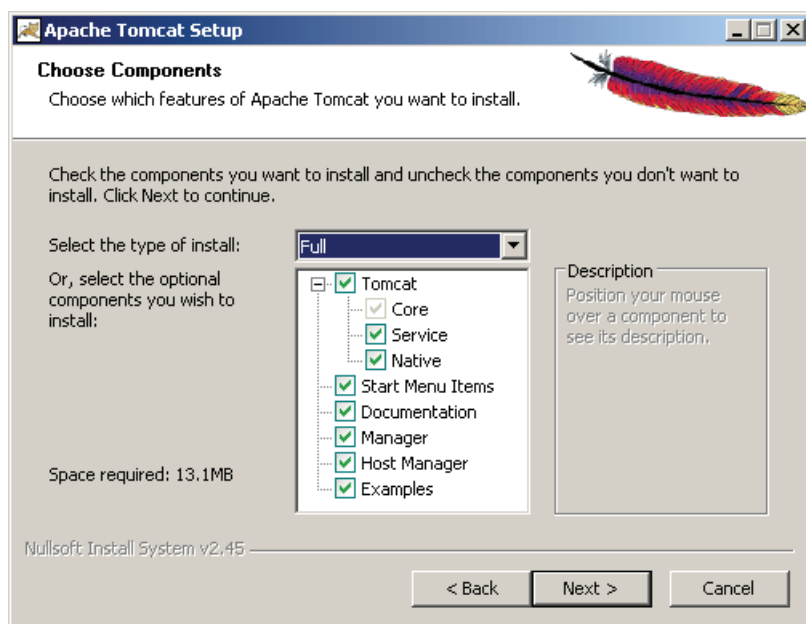


Obr. 15 Ukončenie inštalácie MCR

3.3 Apache Tomcat

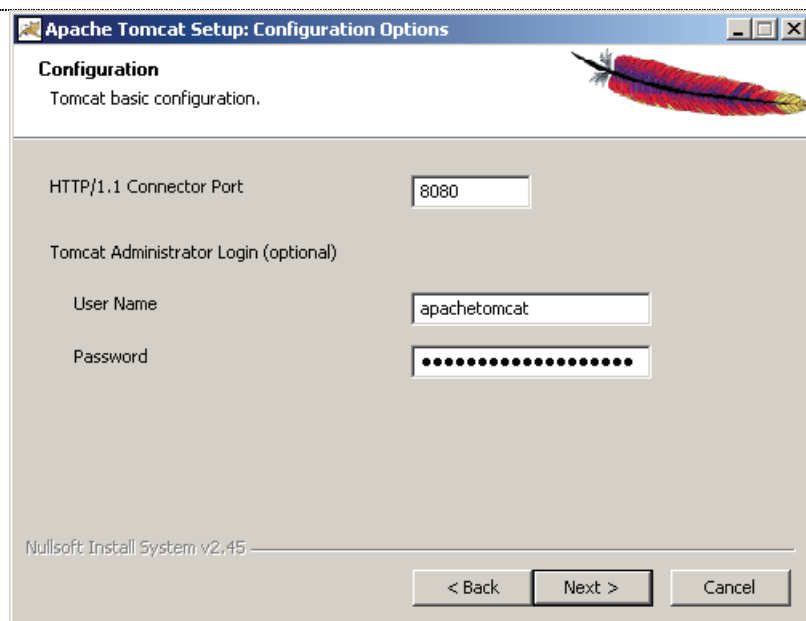
Aby bolo možné publikovať skompilované Java triedy (**Java ARchívy – JAR**) na internete, je potrebné mať webový server s podporou pre J2EE, ako je napríklad Apache Tomcat. Apache Tomcat je open–source software s implementáciou podpory pre Javu a zobrazovaním JavaServer Pages. Distribúcia Tomcatu je voľne dostupná na <http://tomcat.apache.org/> v sekcii *Download*. V práci je použitý inštalačný balíček s verziou 6.0.26. Jeho inštalácia je opísaná nižšie.

Po stiahnutí a spustení inštalácie programu sa zobrazí uvítacie okno. Kliknutím na [Next >] sa zobrazí licenčná zmluva, ktorú treba akceptovať stlačením [I Agree], inak program nebude nainštalovaný. V ďalšom kroku je možné vybrať jednotlivé komponenty, ktoré bude program obsahovať (obr. 16). Ako prevenciu pred možnými problémami bola zvolená plná inštalácia program (*Full*).



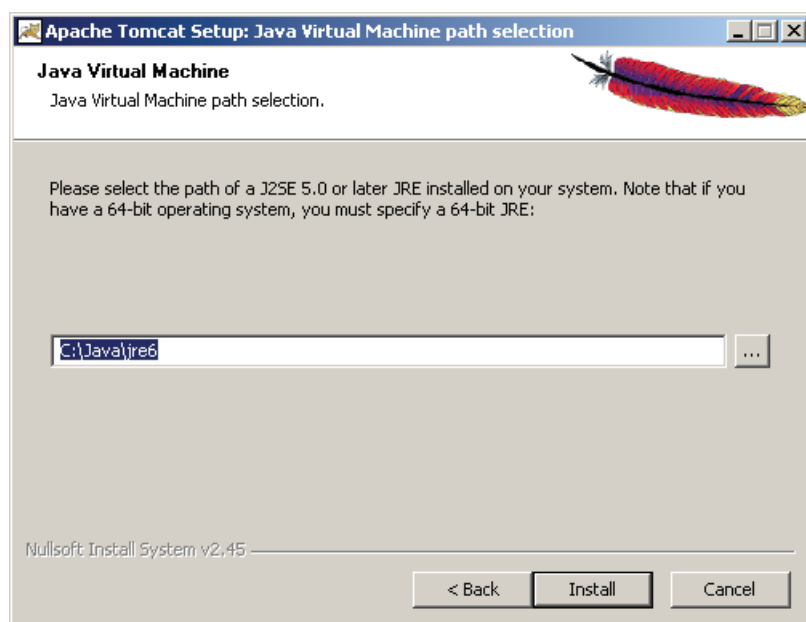
Obr. 16 Výber komponentov inštalácie

Stlačením [Next >] sa inštalácia posunie do ďalšieho kroku. Po zvolení úložnej destinácie zvolených komponentov (napr.: *C:\ApacheSoftwareFoundation\Tomcat6.0*) a stlačením [Next >] je možné nakonfigurovať port (základný port pre HTTP je 80, alebo 8080), cez ktorý bude užívateľ komunikovať so serverom a správcovský účet, ktorého konfigurácia je voliteľná (obr. 17).



Obr. 17 Základné pred-inštalačné nastavenia

Po opätovnom stlačení [Next >] sa inštalačný program dostáva do posledného kroku pred inštaláciu, kde je nutné nastaviť cestu (obr. 18) k súborom JRE (C:\Java\jre6\).



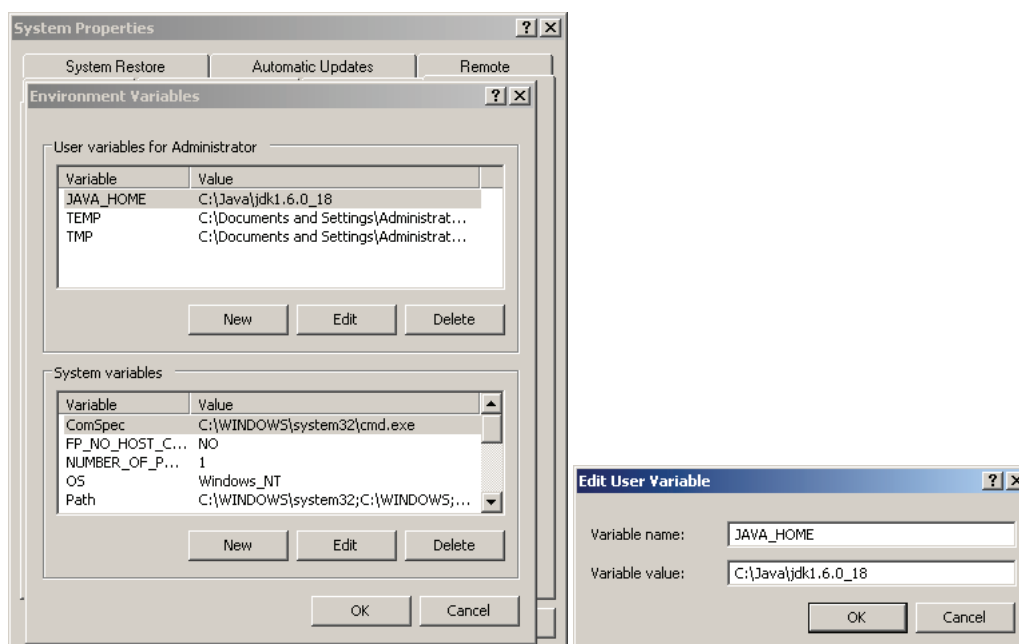
Obr. 18 Zvolenie destinácie pre nakopírovanie súborov

Po stlačení [Install] sa vybrané súčasti extrahujú do zadaného adresára. Inštalátor tiež nakonfiguruje proces (service – službu), ktorý sa bude spúšťať automaticky pri spustení počítača. Po stlačení [Finish] sa inštalácia programu Apache Tomcat končí.

3.4 Nastavenie systémových premenných

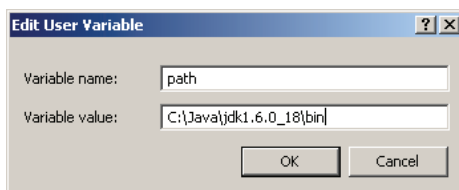
Systémové premenné ovplyvňujú správanie jednotlivých programov a tiež chod samotného systému [19]. V systémoch Windows poznáme systémové premenné, ktoré platia pre všetkých užívateľov a užívateľské, ktoré ovplyvňujú len prostredie daného užívateľa. Nastavujú sa tak, že stlačíme pravým tlačidlom myši na *My Computer* (*Tento Počítač*) a vyberieme *Properties* (*Vlastnosti*). Na karte *Advanced* (*Spresnenie*) sa v dolnej časti nachádza tlačidlo *Environment Variables* (*Premenné prostredia*), klikneme naň a zobrazia sa možnosti nastavenia užívateľských a systémových premenných.

Postačí nastaviť premenné užívateľa, na ktorého sme práve prihlásení. Ako prvú nastavíme premennú, ktorú potrebuje MATLAB pri kompilácii *m-file* do súboru *jar*. V hornej časti stlačíme tlačidlo [*New*]. Do názvu premennej napíšeme *JAVA_HOME* (obr. 19) a do hodnoty napíšeme cestu kam sme nainštalovali JDK a teda *C:\Java\jdk1.6.0_18*, ak je všetko v poriadku, stlačíme [*OK*].



Obr. 19 Pridanie premennej *JAVA_HOME*

Druhú premennú pridáme podobne. Názov premennej bude *path* a cesta *C:\Java\jdk1.6.0_18\bin* (obr. 20). Nachádzajú sa tam binárne súbory JDK no hlavne *javac.exe*. Nastavenie tejto premennej nie je nutné pre správny chod kompilácií.

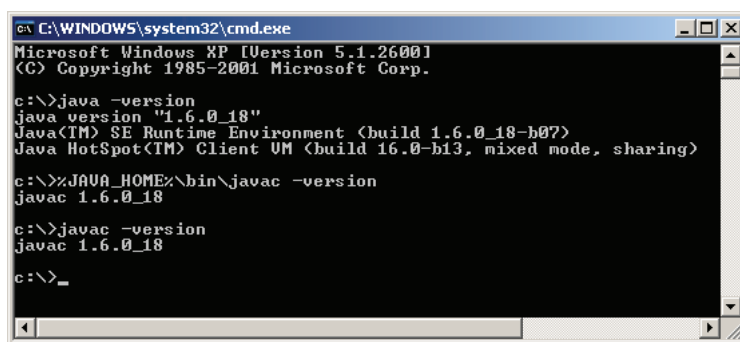


Obr. 20 Pridanie premennej *path* pre príkaz *javac*

Systém je potrebné reštartovať, aby systém akceptoval vykonané zmeny. Po reštartovaní treba skontrolovať, či funguje všetko tak, ako by malo. Do systémového príkazového riadku zadáme `java -version` a stlačíme klávesu *Enter*. Výstupom sú verzie jednotlivých súčastí JRE (obr. 21).

Ďalej je potrebné overiť funkčnosť premennej *JAVA_HOME* pomocou príkazu `javac`.

Upravená premenná *path* umožňuje priamo zadať príkaz `javac -version` do príkazového riadku bez zadávania celej cesty alebo použitia premennej *JAVA_HOME*. Výstup bude rovnaký.



Obr. 21 Overenie funkčnosti prostredia cez systémový príkazový riadok

3.5 Premenné programu MATLAB

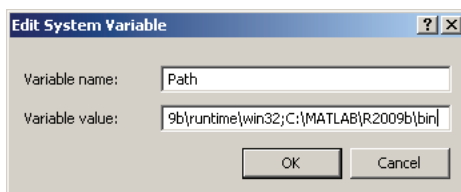
Pri inštalácii MATLABu sa automaticky do premennej *Path* (obr. 22) pridajú tieto hodnoty:

- `C:\MATLAB\R2009b\bin`
- `C:\MATLAB\R2009b\runtime\win32`

Podobne pri inštalácii MCR sa pridá nasledujúca hodnota:

- `C:\MATLAB\MATLABCompilerRuntime\v711\runtime\win32`

Viaceré hodnoty pre tu istú premennú sú oddelené bodkočiarkou. Samotná hodnota bude vyzeráť podľa toho, kde je MATLAB nainštalovaný.



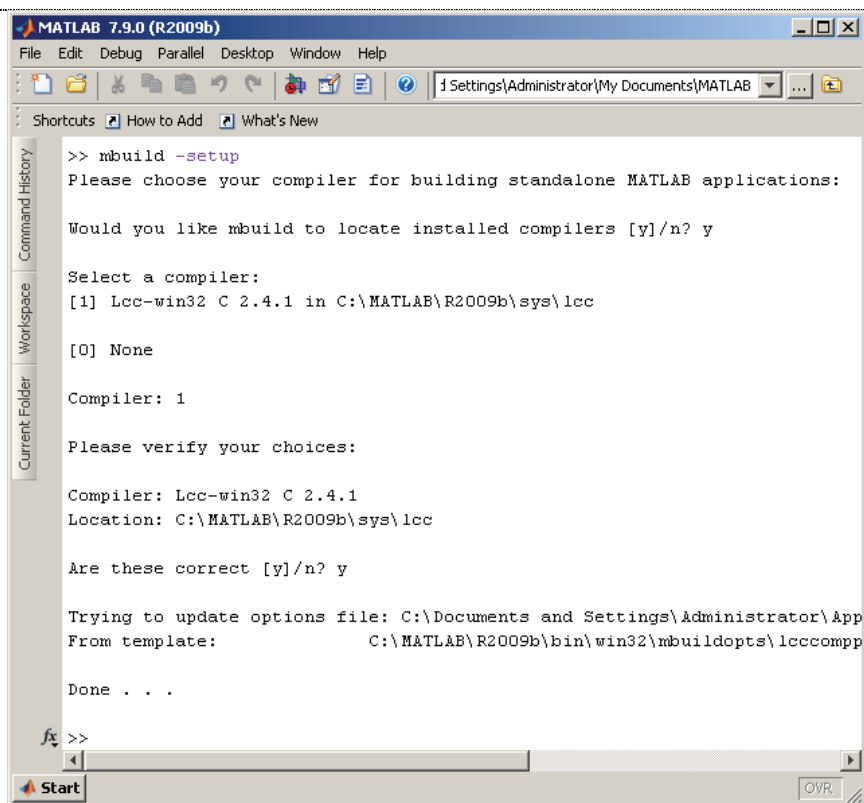
Obr. 22 Premenná Path operačného systému

Je veľmi dôležité, aby tieto cesty boli nastavené správne. V opačnom prípade samotné programy nebudú fungovať správne.

3.5.1 Konfigurácia MATLAB Compiler

[18]

Aby bolo možné pomocou MATLABu kompilovať naše funkcie, musíme mu určiť, aký kompilátor má použiť. Do príkazového riadku zadáme `mbuild -setup` (obr. 23) a potvrdíme. Spustí sa konfigurácia, ktorá sa opýta, či má MATLAB nájsť všetky nainštalované kompilátory. Zadáme "y" a potvrdíme klávesou *Enter*. MATLAB skontroluje prostredie a vypíše zoznam nájdených kompilátorov. Samotná inštalácia programu MATLAB obsahuje jeden, ktorý je automaticky dostupný. Jednotlivé kompilátory sú označené číslami na ľavej strane názvu. Vyberieme z ponuky "1" a potvrdíme. Prebehne kontrola výberu, a ak je všetko v poriadku, *mbuild* sa nastaví a konfigurácia je tak ukončená.



Obr. 23 Nastavenie kompilátora pre MATLAB

4 Vytváranie virtuálneho laboratória

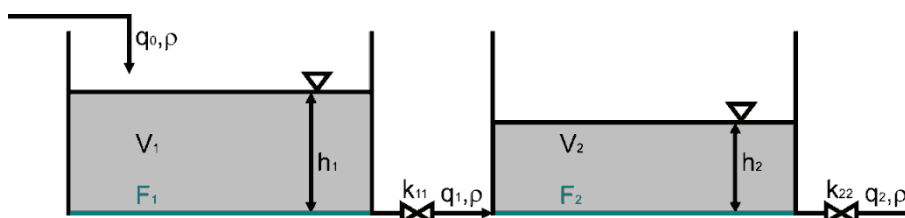
Táto kapitola sa zaoberá matematickými modelmi riadených technologických procesov. Obsahuje tiež opis m-funkcií a ich kompiláciu. Opísaný je tiež základný kód webových stránok a výsledný vzhľad stránok. Všetky súbory sú na priloženom CD.

4.1 Zásobníky kvapaliny

Úlohou je riadiť hladinu v druhom zásobníku kvapaliny. Riadia sa zásobníky kvapaliny s interakciou (obr. 24) a bez interakcie (obr. 25). Kvapalina je privádzaná do prvého zásobníka, kde odteká v spodnej časti prvého zásobníka a prechádza cez ventil do druhého zásobníka. Kvapalina z tohto zásobníka je tiež odvádzaná v spodnej časti zásobníka, kde je opäť ventil. Pri zásobníkoch s interakciou je potrubie z prvého zásobníka priamo spojené s druhým zásobníkom, a tak výška hladiny z druhého zásobníka ovplyvňuje prítok kvapaliny, pričom pri zásobníkoch bez interakcie kvapalina z prvého zásobníka prepadá do druhého zásobníka z hornej časti.

4.1.1 Matematický model zásobníkov kvapaliny s interakciou

[9, 20, 21]



Obr. 24 Schéma zásobníkov kvapaliny s interakciou

Legenda:

q_0 – prítok na vstupe do 1. zásobníka [$m^3 \cdot s^{-1}$]

q_1 – prítok na výstupe z 1. zásobníka [$m^3 \cdot s^{-1}$]

q_2 – prítok na výstupe z 2. zásobníka [$m^3 \cdot s^{-1}$]

ρ – hustota kvapaliny [$kg \cdot m^{-3}$]

$h_{1,2}$ – výška hladiny v 1. a 2. zásobníku [m]

$k_{11,22}$ – konštanta ventilu na výstupe z 1. a 2. zásobníka [$m^{2,5} \cdot s^{-1}$]

$V_{1,2}$ – objem 1. a 2. zásobníka [m^3]

$F_{1,2}$ – plocha podstavy zásobníkov [m^2]

Matematický model je zostavený z materiálovej (hmotnostnej) bilancie, kde súčet vstupujúcich hmotnostných tokov sa rovná súčtu vystupujúcich hmotnostných tokov a rýchlosti akumulácii hmotnosti v systéme.

Materiálové bilancie dynamického systému:

Pre prvý zásobník:

$$\dot{m}_0(t) = \dot{m}_1(t) + \frac{dm_1(t)}{dt} \quad (14)$$

Pre druhý zásobník:

$$\dot{m}_1(t) = \dot{m}_2(t) + \frac{dm_1(t)}{dt} \quad (15)$$

kde:

$$\dot{m}_0(t) = q_0(t)\rho \quad (16)$$

$$\dot{m}_1(t) = q_1(t)\rho \quad (17)$$

$$\dot{m}_2(t) = q_2(t)\rho \quad (18)$$

$$m_1(t) = V_1(t)\rho \quad (19)$$

$$m_2(t) = V_2(t)\rho \quad (20)$$

Po dosadení rovníc (16) až (20) do materiálových bilancií (14) a (15):

$$q_0(t)\rho = q_1(t)\rho + \frac{d[V_1(t)\rho]}{dt} \Rightarrow q_0(t) = q_1(t) + \frac{dV_1(t)}{dt} \quad (21)$$

$$q_1(t)\rho = q_2(t)\rho + \frac{d[V_2(t)\rho]}{dt} \Rightarrow q_1(t) = q_2(t) + \frac{dV_2(t)}{dt} \quad (22)$$

Pre q_1, q_2 a V_1, V_2 platí nasledovné:

$$q_1(t) = k_{11}\sqrt{h_1(t) - h_2(t)} \quad (23)$$

$$q_2(t) = k_{22}\sqrt{h_2(t)} \quad (24)$$

$$V_1(t) = F_1 h_1(t) \quad (25)$$

$$V_2(t) = F_2 h_2(t) \quad (26)$$

Nelineárny matematický model po dosadení (23) až (26) do rovníc (21) a (22):

$$\begin{aligned} q_0(t) &= k_{11}\sqrt{h_1(t) - h_2(t)} + \frac{d[F_1 h_1(t)]}{dt} \\ &= k_{11}\sqrt{h_1(t) - h_2(t)} + F_1 \frac{dh_1(t)}{dt} \end{aligned} \quad (27)$$

$$\begin{aligned}
k_{11}\sqrt{h_1(t) - h_2(t)} &= k_{22}\sqrt{h_2(t)} + \frac{d[F_2 h_2(t)]}{dt} \\
&= k_{22}\sqrt{h_2(t)} + F_2 \frac{dh_2(t)}{dt}
\end{aligned} \tag{28}$$

Klasifikácia veličín v zásobníkoch kvapaliny:

Stavové veličiny – $h_1(t), h_2(t)$

Vstupná veličina – $q_0(t)$

Výstupné veličiny – $h_1(t), h_2(t)$

Rovnice (27) a (28) s deriváciou na ľavej strane:

$$F_1 \frac{dh_1(t)}{dt} = q_0(t) - k_{11}\sqrt{h_1(t) - h_2(t)} \tag{29}$$

$$F_2 \frac{dh_2(t)}{dt} = k_{11}\sqrt{h_1(t) - h_2(t)} - k_{22}\sqrt{h_2(t)} \tag{30}$$

V rovnovážnom stave nedochádza k akumulácii $dh/dt = 0$ a súčet vstupujúcich hmotnostných tokov sa rovná súčtu vystupujúcich hmotnostných tokov:

$$0 = q_0^s - k_{11}\sqrt{h_1^s - h_2^s} \Rightarrow q_0^s = k_{11}\sqrt{h_1^s - h_2^s} \tag{31}$$

$$0 = k_{11}\sqrt{h_1^s - h_2^s} - k_{22}\sqrt{h_2^s} \Rightarrow k_{11}\sqrt{h_1^s - h_2^s} = k_{22}\sqrt{h_2^s} \tag{32}$$

Matematický model v rovnovážnom stave (31), (32) sa dá využiť na výpočet výšok hladín v rovnovážnom stave, alebo na výpočet objemového prietoku vstupného prúdu do zásobníka, z ktorých je možné zistiť výšky hladín v zásobníkoch.

Výška hladiny v druhom zásobníku:

$$q_0^s = k_{22}\sqrt{h_2^s} \Rightarrow h_2^s = \left(\frac{q_0^s}{k_{22}} \right)^2 \tag{33}$$

Výška hladiny v prvom zásobníku:

$$q_0^s = k_{11}\sqrt{h_1^s - h_2^s} \Rightarrow \left(\frac{q_0^s}{k_{11}} \right)^2 = h_1^s - h_2^s \Rightarrow h_1^s = \left(\frac{q_0^s}{k_{11}} \right)^2 + h_2^s \tag{34}$$

Odvedenie linearizovaného dynamického modelu:

Dynamický matematický model (DMM):

$$F_1 \frac{dh_1(t)}{dt} = q_0(t) - k_{11}\sqrt{h_1(t) - h_2(t)}, \quad h_1(0) = h_1^s \tag{35}$$

$$F_2 \frac{dh_2(t)}{dt} = k_{11}\sqrt{h_1(t) - h_2(t)} - k_{22}\sqrt{h_2(t)}, \quad h_2(0) = h_2^s \tag{36}$$

Matematický model rovnovážneho stavu (MMRS):

$$F_1 \frac{dh_1^s}{dt} = q_0^s - k_{11}\sqrt{h_1^s - h_2^s} \quad (37)$$

$$F_2 \frac{dh_2^s}{dt} = k_{11}\sqrt{h_1^s - h_2^s} - k_{22}\sqrt{h_2^s} \quad (38)$$

Linearizovaný DMM sa získa rozdielom DMM (35), (36) a MMRS (37), (38):

$$F_1 \frac{dh_1(t)}{dt} - F_1 \frac{dh_1^s}{dt} = q_0(t) - q_0^s - k_{11}\sqrt{h_1(t) - h_2(t)} + k_{11}\sqrt{h_1^s - h_2^s} \quad (39)$$

$$F_1 \frac{d[h_1(t) - h_1^s]}{dt} = q_0(t) - q_0^s - k_{11}\sqrt{h_1(t) - h_2(t)} + k_{11}\sqrt{h_1^s - h_2^s} \quad (40)$$

$$F_2 \frac{dh_2(t)}{dt} - F_2 \frac{dh_2^s}{dt} = k_{11}\sqrt{h_1(t) - h_2(t)} - k_{11}\sqrt{h_1^s - h_2^s} - k_{22}\sqrt{h_2(t)} + k_{22}\sqrt{h_2^s} \quad (41)$$

$$F_2 \frac{d[h_2(t) - h_2^s]}{dt} = k_{11}\sqrt{h_1(t) - h_2(t)} - k_{11}\sqrt{h_1^s - h_2^s} - k_{22}\sqrt{h_2(t)} + k_{22}\sqrt{h_2^s} \quad (42)$$

Definovanie odchýlkových veličín:

Stavové veličiny:

$$x_1(t) = h_1(t) - h_1^s \quad (43)$$

$$x_2(t) = h_2(t) - h_2^s \quad (44)$$

Vstupná veličina:

$$u(t) = q_0(t) - q_0^s \quad (45)$$

Výstupné veličiny:

$$y_1(t) = x_1(t) \quad (46)$$

$$y_2(t) = x_2(t) \quad (47)$$

Linearizácia nelineárnych členov:

Linearizácia člena $\sqrt{h_1 - h_2}$:

$$\begin{aligned} \sqrt{h_1 - h_2} \Big|_{h_1=h_1^s, h_2=h_2^s} &= \sqrt{h_1^s - h_2^s} \\ &+ \frac{d\sqrt{h_1 - h_2}}{dh_1} \Big|_{h_1=h_1^s, h_2=h_2^s} (h_1 - h_1^s) \\ &+ \frac{d\sqrt{h_1 - h_2}}{dh_2} \Big|_{h_1=h_1^s, h_2=h_2^s} (h_2 - h_2^s) \end{aligned} \quad (48)$$

$$\sqrt{h_1 - h_2}|_{h_1=h_1^s, h_2=h_2^s} = \sqrt{h_1^s - h_2^s} + \frac{1}{2\sqrt{h_1^s - h_2^s}}x_1 - \frac{1}{2\sqrt{h_1^s - h_2^s}}x_2 \quad (49)$$

Linearizácia člena $\sqrt{h_2}$:

$$\sqrt{h_2}|_{h_2=h_2^s} = \sqrt{h_2^s} + \frac{d\sqrt{h_2}}{dh_2}\bigg|_{h_2^s} (h_2 - h_2^s) \quad (50)$$

$$\sqrt{h_2}|_{h_2=h_2^s} = \sqrt{h_2^s} + \frac{1}{2\sqrt{h_2^s}}x_2 \quad (51)$$

Dosadenie rovníc linearizovaných členov (49), (51) a odchýlkových veličín (43) až (47) do rovníc (40) a (42):

$$F_1 \frac{dx_1(t)}{dt} = u(t) - k_{11} \left(\sqrt{h_1^s - h_2^s} + \frac{1}{2\sqrt{h_1^s - h_2^s}}x_1(t) - \frac{1}{2\sqrt{h_1^s - h_2^s}}x_2(t) \right) + k_{11}\sqrt{h_1^s - h_2^s} \quad (52)$$

$$F_2 \frac{dx_2(t)}{dt} = k_{11} \left(\sqrt{h_1^s - h_2^s} + \frac{1}{2\sqrt{h_1^s - h_2^s}}x_1(t) - \frac{1}{2\sqrt{h_1^s - h_2^s}}x_2(t) \right) - k_{11}\sqrt{h_1^s - h_2^s} - k_{22} \left(\sqrt{h_2^s} + \frac{1}{2\sqrt{h_2^s}}x_2(t) \right) + k_{22}\sqrt{h_2^s} \quad (53)$$

Definovanie k_1 a k_2 :

$$k_1 = k_{11} \frac{1}{2\sqrt{h_1^s - h_2^s}} \quad (54)$$

$$k_2 = k_{22} \frac{1}{2\sqrt{h_2^s}} \quad (55)$$

Po substitúcii k_1 , k_2 a úprave rovníc (52) a (53):

$$F_1 \frac{dx_1(t)}{dt} = u(t) - k_1 x_1(t) + k_1 x_2(t) \quad (56)$$

$$F_2 \frac{dx_2(t)}{dt} = k_1 x_1(t) - k_1 x_2(t) - k_2 x_2(t) \quad (57)$$

Začiatkové podmienky:

$$x_1(0) = h_1(0) - h_1^s = h_1^s - h_1^s = 0 \quad (58)$$

$$x_2(0) = h_2(0) - h_2^s = h_2^s - h_2^s = 0 \quad (59)$$

Výsledné linearizované dynamické modely:

Pre prvý zásobník:

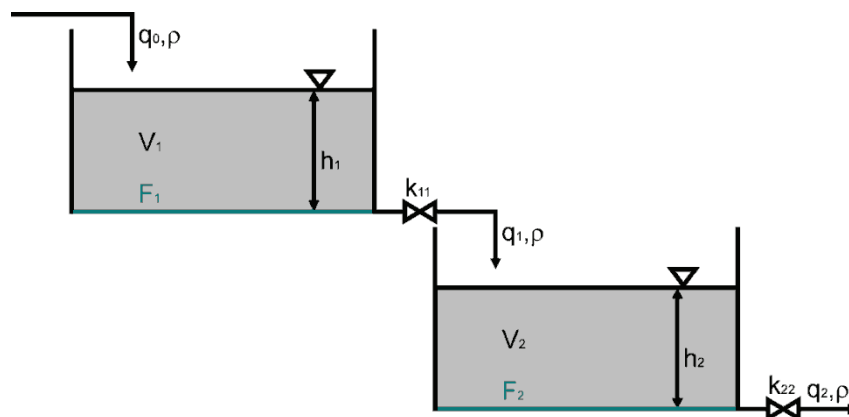
$$\frac{dx_1(t)}{dt} = \frac{1}{F_1} u(t) - \frac{k_1}{F_1} x_1(t) + \frac{k_1}{F_1} x_2(t), \quad x_1(0) = 0 \quad (60)$$

Pre druhý zásobník:

$$\frac{dx_2(t)}{dt} = \frac{k_1}{F_2} x_1(t) - \frac{k_1}{F_2} x_2(t) - \frac{k_2}{F_2} x_2(t), \quad x_2(0) = 0 \quad (61)$$

4.1.2 Matematický model zásobníkov kvapaliny bez interakcie

[9, 20, 21]



Obr. 25 Schéma zásobníkov kvapaliny bez interakcie

Legenda:

q_0 – prietok na vstupe do prvého zásobníka [$m^3 \cdot s^{-1}$]

q_1 – prietok na výstupe z prvého zásobníka [$m^3 \cdot s^{-1}$]

q_2 – prietok na výstupe z druhého zásobníka [$m^3 \cdot s^{-1}$]

ρ – hustota kvapaliny [$kg \cdot m^{-3}$]

$h_{1,2}$ – výška hladiny v 1. a 2. zásobníku [m]

$k_{11,22}$ – konštanta ventilu na výstupe z 1. a 2. zásobníka [$m^{2,5} \cdot s^{-1}$]

$V_{1,2}$ – objem 1. a 2. zásobníka [m^3]

$F_{1,2}$ – plocha podstavy zásobníkov [m^2]

Odvodenie linearizovaných matematických modelov prebieha podobne ako u zásobníkov kvapaliny s interakciou:

Materiálové bilancie dynamického systému:

Pre prvý zásobník:

$$\dot{m}_0(t) = \dot{m}_1(t) + \frac{dm_1(t)}{dt} \quad (62)$$

Pre druhý zásobník:

$$\dot{m}_1(t) = \dot{m}_2(t) + \frac{dm_1(t)}{dt} \quad (63)$$

kde,

$$\dot{m}_0(t) = q_0(t)\rho \quad (64)$$

$$\dot{m}_1(t) = q_1(t)\rho \quad (65)$$

$$\dot{m}_2(t) = q_2(t)\rho \quad (66)$$

$$m_1(t) = V_1(t)\rho \quad (67)$$

$$m_2(t) = V_2(t)\rho \quad (68)$$

Po dosadení rovníc (64) až (68) do materiálových bilancií (62) a (63):

$$q_0(t)\rho = q_1(t)\rho + \frac{d[V_1(t)\rho]}{dt} \Rightarrow q_0(t) = q_1(t) + \frac{dV_1(t)}{dt} \quad (69)$$

$$q_1(t)\rho = q_2(t)\rho + \frac{d[V_2(t)\rho]}{dt} \Rightarrow q_1(t) = q_2(t) + \frac{dV_2(t)}{dt} \quad (70)$$

Pre q_1, q_2 a V_1, V_2 platí nasledovné:

$$q_1(t) = k_{11}\sqrt{h_1(t)} \quad (71)$$

$$q_2(t) = k_{22}\sqrt{h_2(t)} \quad (72)$$

$$V_1(t) = F_1 h_1(t) \quad (73)$$

$$V_2(t) = F_2 h_2(t) \quad (74)$$

Nelineárny matematický model po dosadení rovníc (71) až (74) do rovníc (69) a (70):

$$q_0(t) = k_{11}\sqrt{h_1(t)} + \frac{d[F_1 h_1(t)]}{dt} = k_{11}\sqrt{h_1(t)} + F_1 \frac{dh_1(t)}{dt} \quad (75)$$

$$k_{11}\sqrt{h_1(t)} = k_{22}\sqrt{h_2(t)} + \frac{d[F_2 h_2(t)]}{dt} = k_{22}\sqrt{h_2(t)} + F_2 \frac{dh_2(t)}{dt} \quad (76)$$

Klasifikácia veličín v zásobníkoch kvapaliny:

Stavové veličiny – $h_1(t), h_2(t)$

Vstupná veličina – $q_0(t)$

Výstupné veličiny – $h_1(t), h_2(t)$

Rovnice (75) a (76) s deriváciou na ľavej strane:

$$F_1 \frac{dh_1(t)}{dt} = q_0(t) - k_{11}\sqrt{h_1(t)} \quad (77)$$

$$F_2 \frac{dh_2(t)}{dt} = k_{11}\sqrt{h_1(t)} - k_{22}\sqrt{h_2(t)} \quad (78)$$

V rovnovážnom stave nedochádza k akumulácii $dh/dt = 0$ a súčet vstupujúcich hmotnostných tokov sa rovná súčtu vystupujúcich hmotnostných tokov:

$$0 = q_0^s - k_{11}\sqrt{h_1^s} \Rightarrow q_0^s = k_{11}\sqrt{h_1^s} \quad (79)$$

$$0 = k_{11}\sqrt{h_1^s} - k_{22}\sqrt{h_2^s} \Rightarrow k_{11}\sqrt{h_1^s} = k_{22}\sqrt{h_2^s} \quad (80)$$

Výšky hladín v rovnovážnom stave:

$$q_0^s = k_{11}\sqrt{h_1^s} \Rightarrow h_1^s = \left(\frac{q_0^s}{k_{11}} \right)^2 \quad (81)$$

$$q_0^s = k_{22}\sqrt{h_2^s} \Rightarrow h_2^s = \left(\frac{q_0^s}{k_{22}} \right)^2 \quad (82)$$

Odvodenie linearizovaného dynamického modelu:

Dynamický matematický model (DMM):

$$F_1 \frac{dh_1(t)}{dt} = q_0(t) - k_{11}\sqrt{h_1(t)}, \quad h_1(0) = h_1^s \quad (83)$$

$$F_2 \frac{dh_2(t)}{dt} = k_{11}\sqrt{h_1(t)} - k_{22}\sqrt{h_2(t)}, \quad h_2(0) = h_2^s \quad (84)$$

Matematický model rovnovážneho stavu (MMRS):

$$F_1 \frac{dh_1^s}{dt} = q_0^s - k_{11}\sqrt{h_1^s} \quad (85)$$

$$F_2 \frac{dh_2^s}{dt} = k_{11}\sqrt{h_1^s} - k_{22}\sqrt{h_2^s} \quad (86)$$

Linearizovaný DMM sa získa rozdielom DMM (83), (84) a MMRS (85), (86):

$$F_1 \frac{dh_1(t)}{dt} - F_1 \frac{dh_1^s}{dt} = q_0(t) - q_0^s - k_{11}\sqrt{h_1(t)} + k_{11}\sqrt{h_1^s} \quad (87)$$

$$F_1 \frac{d[h_1(t) - h_1^s]}{dt} = q_0(t) - q_0^s - k_{11}\sqrt{h_1(t)} + k_{11}\sqrt{h_1^s} \quad (88)$$

$$F_2 \frac{dh_2(t)}{dt} - F_2 \frac{dh_2^s}{dt} = k_{11}\sqrt{h_1(t)} - k_{11}\sqrt{h_1^s} - k_{22}\sqrt{h_2(t)} + k_{22}\sqrt{h_2^s} \quad (89)$$

$$F_2 \frac{d[h_2(t) - h_2^s]}{dt} = k_{11}\sqrt{h_1(t)} - k_{11}\sqrt{h_1^s} - k_{22}\sqrt{h_2(t)} + k_{22}\sqrt{h_2^s} \quad (90)$$

Definovanie odchýlkových veličín:

Stavové veličiny:

$$x_1(t) = h_1(t) - h_1^s \quad (91)$$

$$x_2(t) = h_2(t) - h_2^s \quad (92)$$

Vstupná veličina:

$$u(t) = q_0(t) - q_0^s \quad (93)$$

Výstupné veličiny:

$$y_1(t) = x_1(t) \quad (94)$$

$$y_2(t) = x_2(t) \quad (95)$$

Linearizácia nelineárnych členov:

Linearizácia člena $\sqrt{h_1}$:

$$\sqrt{h_1}|_{h_1=h_1^s} = \sqrt{h_1^s} + \left. \frac{d\sqrt{h_1}}{dh_1} \right|_{h_1^s} (h_1 - h_1^s) \quad (96)$$

$$\sqrt{h_1}|_{h_1=h_1^s} = \sqrt{h_1^s} + \frac{1}{2\sqrt{h_1^s}} x_1 \quad (97)$$

Linearizácia člena $\sqrt{h_2}$:

$$\sqrt{h_2}|_{h_2=h_2^s} = \sqrt{h_2^s} + \left. \frac{d\sqrt{h_2}}{dh_2} \right|_{h_2^s} (h_2 - h_2^s) \quad (98)$$

$$\sqrt{h_2}|_{h_2=h_2^s} = \sqrt{h_2^s} + \frac{1}{2\sqrt{h_2^s}} x_2 \quad (99)$$

Dosadenie rovníc linearizovaných členov (97), (99) a odchýlkových veličín (91) až (95) do rovníc (40) a (42):

$$F_1 \frac{dx_1(t)}{dt} = u(t) - k_{11} \left(\sqrt{h_1^s} + \frac{1}{2\sqrt{h_1^s}} x_1(t) \right) + k_{11}\sqrt{h_1^s} \quad (100)$$

$$F_2 \frac{dx_2(t)}{dt} = k_{11} \left(\sqrt{h_1^s} + \frac{1}{2\sqrt{h_1^s}} x_1(t) \right) - k_{11} \sqrt{h_1^s} - k_{22} \left(\sqrt{h_2^s} + \frac{1}{2\sqrt{h_2^s}} x_2(t) \right) + k_{22} \sqrt{h_2^s} \quad (101)$$

Definovanie k_1 a k_2 :

$$k_1 = k_{11} \frac{1}{2\sqrt{h_1^s}} \quad (102)$$

$$k_2 = k_{22} \frac{1}{2\sqrt{h_2^s}} \quad (103)$$

Po substitúcii k_1 , k_2 a úprave rovníc (100) a (101):

$$F_1 \frac{dx_1(t)}{dt} = u(t) - k_1 x_1(t) \quad (104)$$

$$F_2 \frac{dx_2(t)}{dt} = k_1 x_1(t) - k_2 x_2(t) \quad (105)$$

Začiatkové podmienky:

$$x_1(0) = h_1(0) - h_1^s = h_1^s - h_1^s = 0 \quad (106)$$

$$x_2(0) = h_2(0) - h_2^s = h_2^s - h_2^s = 0 \quad (107)$$

Výsledné linearizované dynamické modely (104) a (105):

Pre prvý zásobník:

$$\frac{dx_1(t)}{dt} = \frac{1}{F_1} u(t) - \frac{k_1}{F_1} x_1(t), \quad x_1(0) = 0 \quad (108)$$

Pre druhý zásobník:

$$\frac{dx_2(t)}{dt} = \frac{k_1}{F_2} x_1(t) - \frac{k_2}{F_2} x_2(t), \quad x_2(0) = 0 \quad (109)$$

4.1.3 M–funkcia zásobníkov kvapaliny

Rovnice (3), (5) až (8), (33), (34), (44), (54), (55), (60), (61), (81), (82), (92), (102), (103), (108) a (109) boli použité v m–funkcií vytvorenej v MATLABe. Spolu s diskretizovanou formou PID regulátora (13) nahradili simulinkovú schému pre riadenie zásobníkov kvapaliny.

Parametre, ktoré sa zadávajú ako vstup pre funkciu:

- h – krok výpočtu (zadané fixne v kóde na hodnotu 0,05)
- Tf – čas simulácie
- q0s – ustálený prietok kvapaliny do 1. zásobníka
- k11 – konštanta 1. ventilu
- k22 – konštanta 2. ventilu
- F1 – plocha podstavy 1. zásobníka
- F2 – plocha podstavy 2. zásobníka
- Zr – konštanta zosilnenia pre regulátor
- Ti – integračná konštanta pre regulátor
- Td – derivačná konštanta pre regulátor
- w – žiadaná hodnota
- ina – prepínač medzi zásobníkmi s interakciou a zásobníkmi bez interakcie
- gr – možnosť zapnutia mriežky v zobrazenom grafe
- intpr – interval presnosti (vytvorenie horného a dolného ohraničenia v grafe)

Definovanie funkcie *riadZas*, výstupu *out*, ktorý bude zobrazený na webovej stránke a vstupných premenných:

```
function out = riadZas(h,Tf,q0s,k11,k22,F1,F2,Zr,Ti,Td,w,ina,gr,intpr)
```

Definovanie premennej času, kroku, v ktorom sa práve cyklus nachádza a začiatočné stavy odchýlkových veličín. Premenná *i* je použitá pre zapisovanie vypočítaných hodnôt na konci každého cyklu do poľa, pre vykreslenie grafu:

```
cas=0  
i=1;  
x1=0;  
x2=0;
```

Premennou *ina* sa rozlišuje medzi výpočtom zásobníkov kvapaliny s interakciou a bez interakcie.

```
if ina == 1
```

Podmienkou *if* sa rozhoduje, ktorá časť patrí zásobníkom kvapaliny s interakciou. Ak *ina* nadobudne hodnotu 1, funkcia použije vstupy na výpočet zásobníkov kvapaliny s interakciou. Pre hodnotu inú ako 1, funkcia bude pokračovať od *else*, čím začína cyklus pre výpočet zásobníkov kvapaliny bez interakcie.

Výpočet výšky hladín v rovnovážnom stave (33), (34) v druhom a prvom zásobníku, a výpočet konštánt zásobníkov (54), (55). Posledný riadok prevádza žiadanú hodnotu h_2 do odchýlkového tvaru.

```
h2s=(q0s/k22)^2;  
h1s=(q0s/k11)^2+h2s;  
k1=k11/(2*sqrt(h1s-h2s));  
k2=k22/(2*sqrt(h2s));  
ziadanaOdch=w-h2s;
```

Začiatok cyklu výpočtu riadenia, pričom výpočet prebieha po krokoch 0.05. Premenná *cas* je zadefinovaná na počiatku ako 0, teda pri zadanom čase simulácie $T_f = 10$ s sa výpočet vykoná 200 krát, kým sa cyklus zastaví, a funkcia bude pokračovať.

```
while cas<=Tf
```

Zadefinovanie začiatočných stavov pre výpočet regulátora. V 1. kroku je regulačná odchýlka rovná 0 ako aj regulačná odchýlka v $k-1$ a $k-2$ kroku. Akčný zásah v $k-1$ kroku je tiež rovný 0, pretože sa jedná o prvý krok výpočtu.

```
if cas == 0  
    q0_m1=0;  
    e=0;  
    e_m1=0;  
    e_m2=0;  
end
```

Zákon riadenia, prepísaná rovnica (13).

```
q0=q0_m1+Zr*((1+h/Ti+Td/h)*e+(-1-2*Td/h)*e_m1+Td/h*e_m2);
```

Prepísanie regulačnej odchýlky a akčného zásahu pre nasledujúci krok (k -ty krok sa presúva do $k-1$ kroku).

```
e_m2=e_m1;  
e_m1=e;  
q0_m1=q0;
```

Akčný zásah zo zákona riadenia sa prevedie na odchýlkový tvar.

```
u=q0-q0s;
```

V simuláciách sa nachádza spodné aj horné ohraničenie akčného zásahu. Minimálny prítok môže byť $0 \text{ m}^3 \cdot \text{s}^{-1}$ a maximálny prítok je daný ako 10 násobok vstupného prítoku v ustálenom stave.

```

if u < 0
    u = 0;
end
if u > q0s*10
    u = q0s*10;
end

```

Riešenie prvej rovnice dynamiky (60), pomocou metódy Runge–Kutta a získanie nového stavu x_1 .

```

p1=u/F1-k1/F1*x1+k1/F1*x2;
p2=u/F1-k1/F1*(x1+h/2*p1)+k1/F1*x2;
p3=u/F1-k1/F1*(x1+h/2*p2)+k1/F1*x2;
p4=u/F1-k1/F1*(x1+h*p3)+k1/F1*x2;
x1=x1+h/6*(p1+2*p2+2*p3+p4);

```

Riešenie druhej rovnice dynamiky (61), pomocou metódy Runge–Kutta a získanie nového stavu x_2 .

```

q1=k1/F2*x1-k2/F2*x2-k1/F2*x2;
q2=k1/F2*x1-k2/F2*(x2+h/2*q1)-k1/F2*(x2+h/2*q1);
q3=k1/F2*x1-k2/F2*(x2+h/2*q2)-k1/F2*(x2+h/2*q2);
q4=k1/F2*x1-k2/F2*(x2+h*q3)-k1/F2*(x2+h*q3);
x2=x2+h/6*(q1+2*q2+2*q3+q4);

```

Na základe novej hodnoty x_2 sa vypočíta regulačná odchýlka pre ďalší krok.

```

e=ziadanaOdch-x2;

```

Prevodom stavu v 2. zásobníku z ochýlkového tvaru na normálny tvar, sa vypočíta skutočná výška hladiny kvapaliny v 2. zásobníku v aktuálnom kroku.

```

h2=x2+h2s;

```

Ukladanie výsledkov zásobníkov kvapaliny s interakciou do poľa. Hodnoty budú na konci funkcie použité pre vykreslenie grafu.

```

data(1,i)=cas;
data(2,i)=h2;
data(3,i)=w;
data(4,i)=w+intpr;
data(5,i)=w-intpr;

```

Posun výpočtu o ďalší krok (krok je 0.05) a tiež pripočítanie kroku do premennej i , pre zapisovanie hodnôt do poľa.

```

cas=cas+h;
i=i+1;

```

Koniec cyklu výpočtu zásobníkov s interakciou.

```

end

```

V prípade, že premenná *ina* nenadobúda hodnotu 1, je zvolený zásobník kvapaliny bez interakcie.

```

else

```

Cyklus je veľmi podobný, ako pre zásobníky s interakciou. Zmenili sa rovnice pre výpočet rovnovážnych stavov (81), (82), rovnice konštant zásobníkov (102), (103) a rovnice dynamiky (108), (109) pre získanie stavu x_1 a x_2 . Celý cyklus pre výpočet zásobníkov bez interakcie je potom ukončený.

end

Hodnoty uložené v poli je potrebné vykresliť do grafu, ktorý sa zobrazí na webovej stránke. Najprv treba vytvoriť *figure*, do ktorého sa graf vloží.

```
f = figure;  
set(f,'Visible','off');
```

Potom vykresliť samotný graf, pričom sa definujú aj formáty čiar a ich farby.

```
plot(data(1,:),data(3:),'-.',data(1,:),data(2,:),data(1,:),data(4:),'k--',data(1,:),data(5:),'k--');
```

Pomenovanie grafu je podľa toho, aké zásobníky boli zvolené. Pomenované sú aj osi. Umiestnenie legendy je automatické a do grafu je možné vkresliť aj s mriežku.

```
if ina == 1  
    title('Riadenie vysky hladiny v 2. zasobniku [zasobniky s interakciou]');  
else  
    title('Riadenie vysky hladiny v 2. zasobniku [zasobniky bez interakcie]');  
end  
xlabel('Cas [s]'); ylabel('Vyska hladiny v 2. zasobniku [m]');  
legend('w – Ziadana vyska hladiny','h_2 – Riadena vyska hladiny','Location','Best');  
if gr == 1  
    grid on;  
end
```

Celý graf je potom odoslaný ako výstup na webovú stránku a funkcia zatvorí *figure* čím uvoľní prostriedky.

```
out = webfigure(f);  
close(f);
```

Koniec samotnej funkcie.

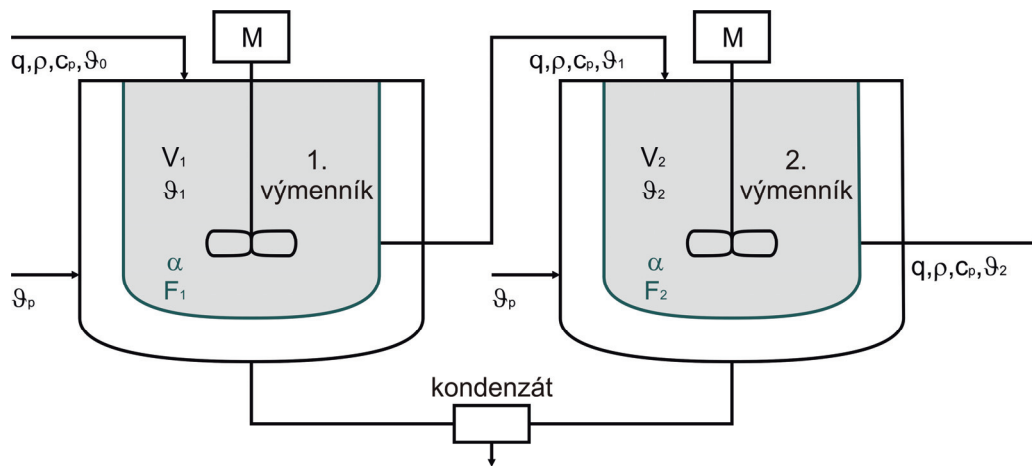
end

4.2 Výmenníky tepla

Úlohou je riadiť dva sériovo zapojené plášťové výmenníky tepla (obr. 26). Kvapalina je ohrievaná vodnou parou. Kondenzát je odvádzaný v spodnej časti zariadenia. Distribúcia tepla v celom objeme výmenníkov je zabezpečená miešadlami. Ohrievaná kvapalina vstupuje do prvého, a potom do druhého výmenníka tepla. Riadenie teploty na výstupe z druhého výmenníka je realizované reguláciou teploty ohrevnej pary.

4.2.1 Matematický model výmenníkov tepla

[5, 20, 22]



Obr. 26 Schéma výmenníkov tepla

Legenda:

- q – objemový prietok ohrievanej kvapaliny [$m^3 \cdot s^{-1}$]
- ρ – hustota ohrievanej kvapaliny [$kg \cdot m^{-3}$]
- c_p – špecifická tepelná kapacita ohrievanej kvapaliny [$J \cdot kg^{-1} \cdot K^{-1}$]
- θ_0 – teplota na vstupe do 1. výmenníka [K]
- θ_1 – teplota na vstupe do 2. výmenníka (teplota kvapaliny v 1. výmenníku) [K]
- θ_2 – teplota na výstupe z 2. výmenníka (teplota kvapaliny v 2. výmenníku) [K]
- θ_p – teplota ohrevnej pary [K]
- $V_{1,2}$ – objem 1. a 2. výmenníka tepla [m^3]
- $F_{1,2}$ – teplovýmenná plocha (1. výmenníka, 2. výmenníka) [m^2]
- α – koeficient prestupu tepla teplovýmennej plochy [$J \cdot s^{-1} \cdot m^{-2} \cdot K^{-1}$]
- M – miešadlo

Zjednodušujúce predpoklady pri odvodzovaní matematického modelu:

- zanedbanie tepelných kapacít stien výmenníka tepla
- zanedbanie strát tepla do okolia (výmenník je dokonale izolovaný)
- kvapalina vo výmenníku je dokonale premiešavaná
- prietok a objem ohrievanej kvapaliny je konštantný
- parametre sú konštantné (hustota, špecifická tepelná kapacita, úhrnný koeficient prechodu tepla (prestupu tepla prúdením), element plochy prestupu tepla, element objemu)
- tlak v systéme je konštantný
- referenčná teplota je rovnaká pre všetky bilancie

Matematický model je zostavený z tepelnej (entalpickej) bilancie dynamického systému, kde súčet vstupujúcich tepelných tokov sa rovná súčtu vystupujúcich tepelných tokov a rýchlosti akumulácie tepla v systéme.

Dynamický matematický model:

Prvý výmenník tepla:

$$\dot{Q}_0(t) + \dot{Q}_{p,1}(t) = \dot{Q}_1(t) + \frac{dQ_1(t)}{dt} \quad (110)$$

Druhý výmenník tepla:

$$\dot{Q}_1(t) + \dot{Q}_{p,2}(t) = \dot{Q}_2(t) + \frac{dQ_2(t)}{dt} \quad (111)$$

Jednotlivé členy sú reprezentované nasledovnými vzťahmi:

$$\dot{Q}_0(t) = q\rho c_p \vartheta_0(t) \quad (112)$$

$$\dot{Q}_1(t) = q\rho c_p \vartheta_1(t) \quad (113)$$

$$\dot{Q}_2(t) = q\rho c_p \vartheta_2(t) \quad (114)$$

$$Q_1(t) = V_1 \rho c_p \vartheta_1(t) \quad (115)$$

$$Q_2(t) = V_2 \rho c_p \vartheta_2(t) \quad (116)$$

$$\dot{Q}_{p,1}(t) = \alpha F_1 (\vartheta_p(t) - \vartheta_1(t)) \quad (117)$$

$$\dot{Q}_{p,2}(t) = \alpha F_2 (\vartheta_p(t) - \vartheta_2(t)) \quad (118)$$

Po dosadení rovníc (112) až (118) do entalpických bilancií (110) a (111):

1. výmenník

$$q\rho c_p \vartheta_0(t) + \alpha F_1 (\vartheta_p(t) - \vartheta_1(t)) = q\rho c_p \vartheta_1(t) + \frac{d[V_1 \rho c_p \vartheta_1(t)]}{dt} \quad (119)$$

$$q\rho c_p \vartheta_0(t) + \alpha F_1 \vartheta_p(t) - \alpha F_1 \vartheta_1(t) = q\rho c_p \vartheta_1(t) + V_1 \rho c_p \frac{d\vartheta_1(t)}{dt} \quad (120)$$

$$q\rho c_p \vartheta_0(t) + \alpha F_1 \vartheta_p(t) = (q\rho c_p + \alpha F_1) \vartheta_1(t) + V_1 \rho c_p \frac{d\vartheta_1(t)}{dt} \quad (121)$$

$$\frac{q\rho c_p}{q\rho c_p + \alpha F_1} \vartheta_0(t) + \frac{\alpha F_1}{q\rho c_p + \alpha F_1} \vartheta_p(t) = \vartheta_1(t) + \frac{V_1 \rho c_p}{q\rho c_p + \alpha F_1} \frac{d\vartheta_1(t)}{dt} \quad (122)$$

Začiatková podmienka – $\vartheta_1(0)$

Klasifikácia veličín v modeli výmenníka tepla:

Stavová veličina – $\vartheta_1(t)$

Vstupné veličiny – $\vartheta_0(t), \vartheta_p(t)$

Výstupná veličina – $\vartheta_1(t)$

2. výmenník

$$q\rho c_p \vartheta_1(t) + \alpha F_2 (\vartheta_p(t) - \vartheta_2(t)) = q\rho c_p \vartheta_2(t) + \frac{d[V_2 \rho c_p \vartheta_2(t)]}{dt} \quad (123)$$

$$\frac{q\rho c_p}{q\rho c_p + \alpha F_2} \vartheta_1(t) + \frac{\alpha F_2}{q\rho c_p + \alpha F_2} \vartheta_p(t) = \vartheta_2(t) + \frac{V_2 \rho c_p}{q\rho c_p + \alpha F_2} \frac{d\vartheta_2(t)}{dt} \quad (124)$$

Začiatková podmienka – $\vartheta_2(0)$

Klasifikácia veličín v modeli výmenníka tepla:

Stavová veličina – $\vartheta_2(t)$

Vstupné veličiny – $\vartheta_1(t), \vartheta_p(t)$

Výstupná veličina – $\vartheta_2(t)$

Definovanie časových konštánt a zosilnení z rovníc (122) a (124):

1. výmenník:

$$Z_{11} = \frac{\alpha F_1}{q\rho c_p + \alpha F_1} \quad (125)$$

$$Z_{12} = \frac{q\rho c_p}{q\rho c_p + \alpha F_1} \quad (126)$$

$$T_1 = \frac{V_1 \rho c_p}{q \rho c_p + \alpha F_1} \quad (127)$$

2. výmenník

$$Z_{21} = \frac{\alpha F_2}{q \rho c_p + \alpha F_2} \quad (128)$$

$$Z_{22} = \frac{q \rho c_p}{q \rho c_p + \alpha F_2} \quad (129)$$

$$T_2 = \frac{V_2 \rho c_p}{q \rho c_p + \alpha F_2} \quad (130)$$

S deriváciou na ľavej strane a po podelení časovou konštantou dostáva lineárny stavový model nasledovný tvar:

1. výmenník

$$T_1 \frac{d\vartheta_1(t)}{dt} = -\vartheta_1(t) + Z_{11}\vartheta_p(t) + Z_{12}\vartheta_0(t) \quad (131)$$

$$\frac{d\vartheta_1(t)}{dt} = -\frac{1}{T_1}\vartheta_1(t) + \frac{Z_{11}}{T_1}\vartheta_p(t) + \frac{Z_{12}}{T_1}\vartheta_0(t) \quad (132)$$

2. výmenník

$$T_2 \frac{d\vartheta_2(t)}{dt} = -\vartheta_2(t) + Z_{21}\vartheta_p(t) + Z_{22}\vartheta_1(t) \quad (133)$$

$$\frac{d\vartheta_2(t)}{dt} = -\frac{1}{T_2}\vartheta_2(t) + \frac{Z_{21}}{T_2}\vartheta_p(t) + \frac{Z_{22}}{T_2}\vartheta_1(t) \quad (134)$$

Matematický model rovnovážneho stavu sa dá určiť z dynamického modelu za predpokladu, že v rovnovážnom stave nedochádza k akumulácii tepla a veličiny $(\vartheta_0^s, \vartheta_1^s, \vartheta_2^s, \vartheta_p^s)$ sú konštantné.

Prvý výmenník po vynásobení rovnice (131) členom $1/T_1$:

$$\frac{d\vartheta_1^s}{dt} = -\frac{1}{T_1}\vartheta_1^s + \frac{Z_{11}}{T_1}\vartheta_p^s + \frac{Z_{12}}{T_1}\vartheta_0^s \quad (135)$$

$$0 = -\frac{1}{T_1}\vartheta_1^s + \frac{Z_{11}}{T_1}\vartheta_p^s + \frac{Z_{12}}{T_1}\vartheta_0^s \quad (136)$$

$$0 = -\vartheta_1^s + Z_{11}\vartheta_p^s + Z_{12}\vartheta_0^s \quad (137)$$

Teplota na výstupe z prvého výmenníka tepla v rovnovážnom stave je:

$$\vartheta_1^s = Z_{11}\vartheta_p^s + Z_{12}\vartheta_0^s \quad (138)$$

A teplota na výstupe z druhého výmenníka tepla v rovnovážnom stave je:

$$\vartheta_2^s = Z_{21}\vartheta_p^s + Z_{22}\vartheta_1^s \quad (139)$$

4.2.2 M–funkcia výmenníkov tepla

Rovnice (3), (5) až (8), (13), (125) až (130), (132), (134), (138) a (139) boli použité v m–funkcií vytvorenej v MATLABe. Funkcia je podobná ako pre zásobníky kvapaliny. Samotný kód funkcie je obsiahnutý v prílohe.

Parametre, ktoré je možno zadať ako vstup pre funkciu:

- h – krok výpočtu (zadané fixne v kóde na hodnotu 0,05)
- w – žiadaná hodnota
- theta0 – teplota na vstupe do 1. výmenníka
- thetaps – ustálená teplota ohrevnej pary
- q – prietok ohrievanej kvapaliny
- ro – hustota ohrievanej kvapaliny
- cp – špecifická tepelná kapacita
- alfa – koeficient prestupu tepla teplovýmennej plochy
- V1 – objem výmenníka
- V2 – objem výmenníka
- F1 – teplovýmenna plocha
- F2 – teplovýmenna plocha
- cas_sim – čas simulácie
- Zr – konštanta zosilnenia pre regulátor
- Ti – integračná konštanta pre regulátor
- Td – derivačná konštanta pre regulátor
- gr – možnosť zapnutia mriežky v zobrazenom grafe
- intpr – interval presnosti

4.3 MATLAB Builder JA

[18]

Kompiláciu do Java archívov vykonáva MATLAB Builder. Ten možno spustiť pomocou príkazu `deploytool`, alebo cez ponuku *Start* programu MATLAB (*Start* → *MATLAB* → *MATLAB Builder JA* → *Deployment Tool (deploytool)*).

Pred jeho spustením je dôležité mať:

- Nainštalovaný balíček JDK (kapitola 3.1)
- Nastavená systémová premenná *JAVA_HOME* (kapitola 3.4)
- Nakonfigurovaný MATLAB Compiler (kapitola 3.5.1)

4.3.1 Kompilácia funkcie zásobníkov kvapaliny

[18]


Pred spustením *Deployment tool*-u je dobré zmeniť pracovný priečinok do adresára, kde sa funkcia nachádza a to príkazom `cd` (napr.: `cd "c:\projekt\riadZas\"`), alebo cez navigačný panel. Po jeho spustení treba zadať názov, umiestnenie a typ projektu.

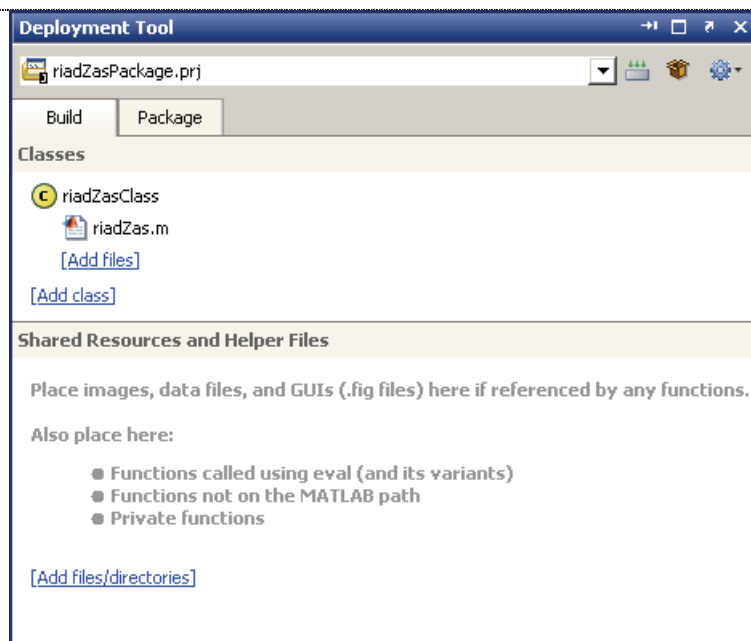
Názvy a súbory použité v tejto práci:

- Názov projektu – *riadZasPackage.prj*
- Adresár projektu – *c:\projekt\riadZas* (aktuálne nastavený adresár)
- Typ projektu – *Java Package*

Po stlačení [OK] sa otvorí nové okno (obr. 27), kde možno zmeniť názov triedy a vložiť vytvorenú m-funkciu.

- Názov triedy – *riadZasClass*
- Kompilovaný súbor (funkcia) – *riadZas.m*

Po vložení všetkých potrebných súborov a premenovaní triedy stačí stlačiť ikonu  (*Build*) a projekt sa začne kompilovať. Po skončení sa vytvorí niekoľko adresárov, ktoré obsahujú dočasné súbory (*.\riadZasPackage\src*) a skompilovanú funkciu (*.\riadZasPackage\distrib\riadZasPackage.jar*).



Obr. 27 Nastavenie triedy s vloženou funkciou v Deployment Tool


4.3.2 Kompilácia funkcie výmenníkov tepla

[18]

Podobne ako v kapitole 4.3.1, je dobré zmeniť pracovný priečinok, až potom spustiť *Deployment tool*.

Názvy a súbory použité v tejto práci:

- Názov projektu – *riadVymPackage.prj*
- Adresár projektu – *c:\projekt\riadVym* (aktuálne nastavený adresár)
- Typ projektu – *Java Package*
- Názov triedy – *riadVymClass*
- Kompilovaný súbor (funkcia) – *riadVym.m*

Po stlačení ikony  (*Build*) sa projekt skompiluje. Projekty majú rovnakú adresárovú štruktúru a vytvorí sa Java archív *.\riadZasPackage\distrib\riadZasPackage.jar*.

4. 4 Webové rozhranie

Pomocou webového rozhrania sa budú funkciám predávať vstupné parametre. Po úspešnom výpočte sa zobrazí výsledok v podobe grafu.

4.4.1 Implementácia *javabuilder.jar* do Apache Tomcat

[18]

Nevyhnutnou podmienkou fungovania skompilovaných m–funkcií je nakopírovanie súboru *javabuilder.jar* (knihnica funkcií MATLABu) do adresára k ostatným knižniciam webového servera.

Súbor *javabuilder.jar* sa nachádza v adresári nainštalovaného MATLABu (resp. MCR) *%matlabroot%\toolbox\Javabuilder\jar\javabuilder.jar*. Ten treba nakopírovať do *%apacheroot%\lib* (napr.: *C:\ApacheSoftwareFoundation\Tomcat6.0\lib*). Server je teraz nie len funkčný, ale je schopný zobrazovať *.jsp stránky, ktoré obsahujú súčasti z prostredia MATLAB.

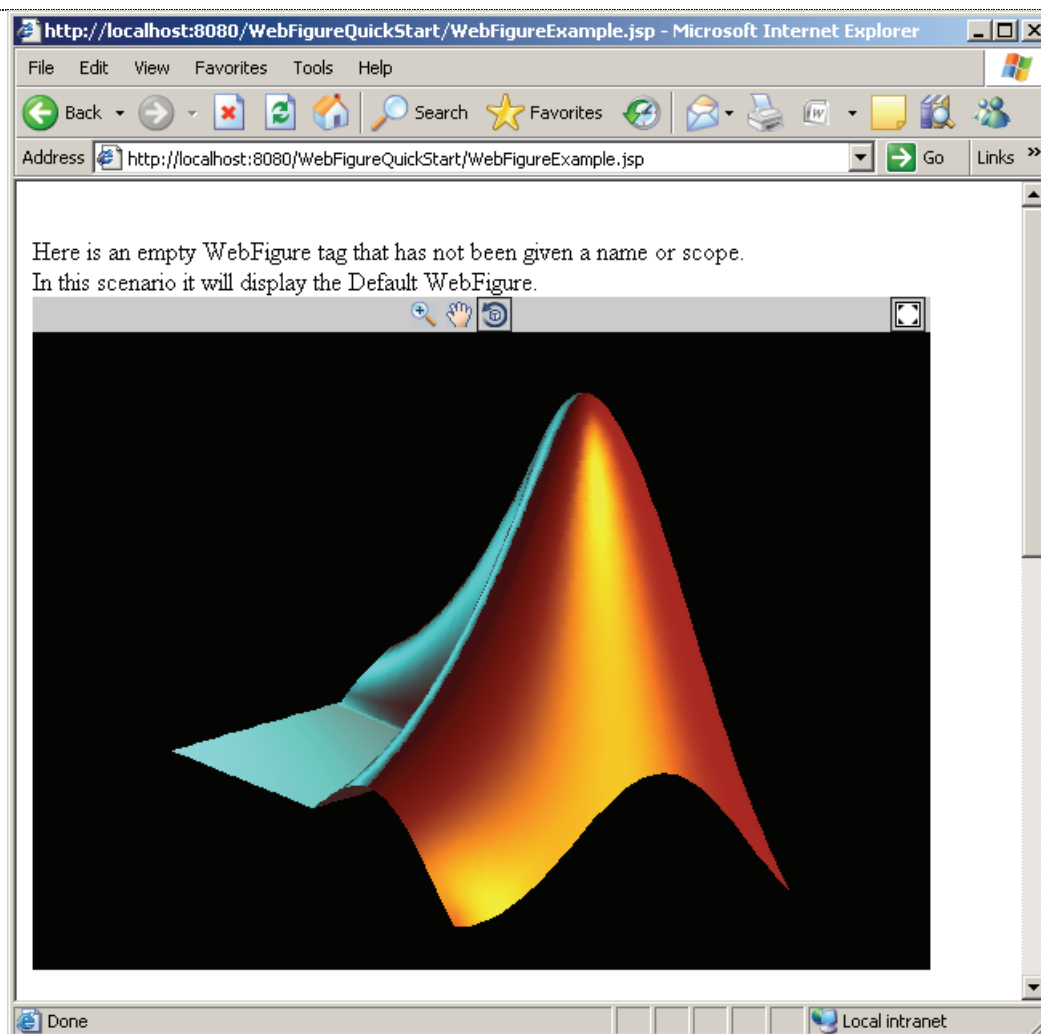
Varovanie: Je veľmi dôležité, aby sa tento súbor nachádzal len raz v cestách tried webového servera. Duplicity tohto súboru spôsobujú chyby.

Jednoduchým testom, či je server správne prepojený s MATLABom, je umiestnenie príkladu na webový server. Príklad sa nachádza v *%matlabroot%\toolbox\Javabuilder\jar\WebFigureQuickStart.war* (Web Application Archive [12]). Tento súbor možno rozbaľiť pomocou Apache Tomcat, alebo ho nakopírovať do koreňového adresára webového servera (napr.: *%apacheroot%\webapps*) a reštartovaním servera si archív Apache sám rozbalí.

URL adresa:

http://localhost:8080/WebFigureQuickStart/WebFigureExample.jsp

Ak sa v prehliadači otvorí stránka s načítaným logom MATLABu, webový server je pripravený a plne funkčný (obr. 28).



Obr. 28 Testovacia stránka

4.4.2 Základné časti .jsp stránky

[18]

Aby sa aplikácia zobrazila ako na obr. 28, treba aby súbor *web.xml* v podadresári vytvorenej stránky (*WEB-INF*) obsahoval nasledujúci kód (**tučným** je vyznačená časť kódu, ktorú možno ľubovoľne meniť a **tučné kurzíva** označujú časť kódu, ktorá je prepojená s názvami pri kompilácii funkcií):

```
<servlet>
  <servlet-name>WebFigures</servlet-name>
  <servlet-class>
    com.mathworks.toolbox.javabuilder.webfigures.WebFiguresServlet
  </servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>WebFigures</servlet-name>
  <url-pattern>/WebFigures/*</url-pattern>
</servlet-mapping>
```

V jsp súbore stránky treba pridať (na začiatok) odkaz na tag WebFigure pomocou nasledovného riadku:

```
<%@ taglib prefix="wf" uri=http://www.mathworks.com/builderja/webfigures.tld %>
```

V stránke tak bude možno použiť tag:

```
<wf:web-figure />
```

ten bude použitý pre zobrazenie grafu ako výstupu zo skompilovanej funkcie.

Na začiatku kódu .jsp stránky treba importovať nami vytvorený komponent a ďalšie potrebné súčasti:

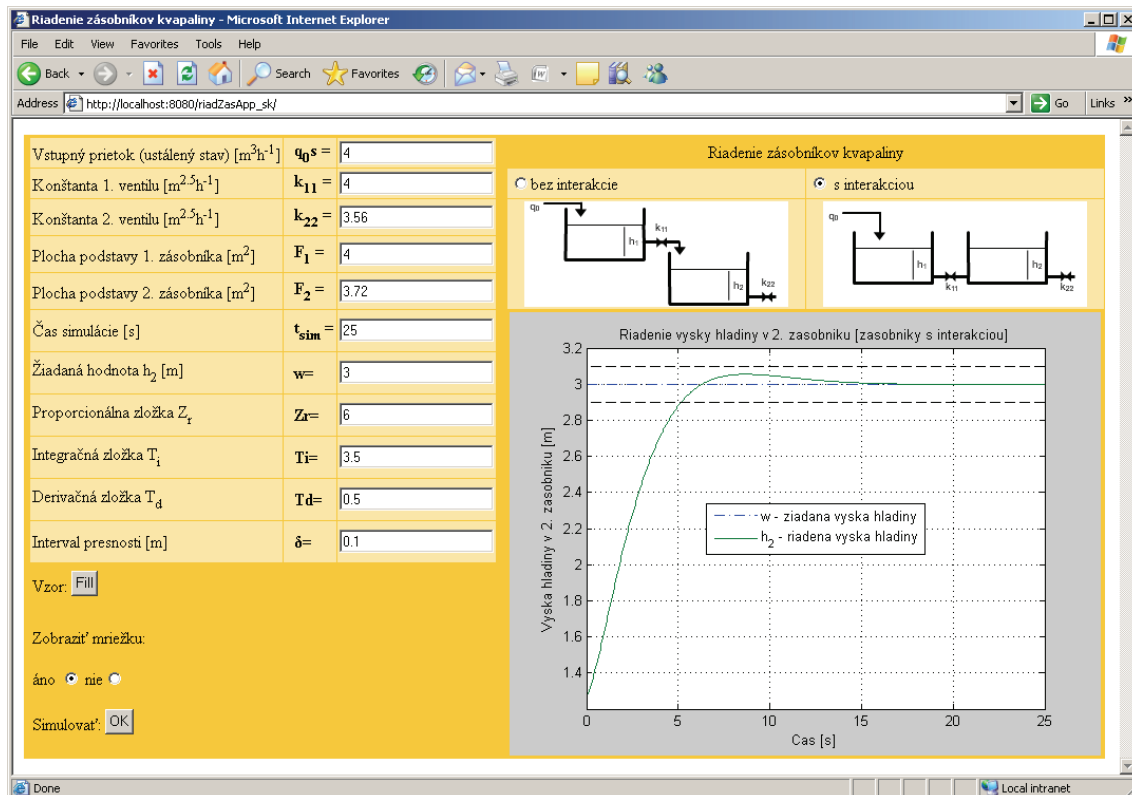
```
<%@ page import="yourComponentsPackage.YourComponentsClass" %>
<%@ page import="com.mathworks.toolbox.javabuilder.webfigures.WebFigure" %>
```

Na stránke potom treba inicializovať komponent a zavolať metódu, ktorá vráti výstup (graf):

```
<%
    YourComponentClass myDeployedComponent = null;
    try {
        myDeployedComponent = new YourComponentClass();
        try {
            WebFigure webFigure = (WebFigure)
                (MWJavaObjectRef)myDeployedComponent.getSurf(1)[0].get();
            request.getSession().setAttribute("YourFigure", webFigure);
        }
        catch(ClassCastException e) {
            throw new Exception
                ("Issue casting deployed components outputs to WebFigure", e);
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    finally {
        myDeployedComponent.dispose();
    }
%>
<wf:web-figure name= "YourFigure" scope="session"/>
```

4.4.2 Zásobníky kvapaliny - internetové rozhranie

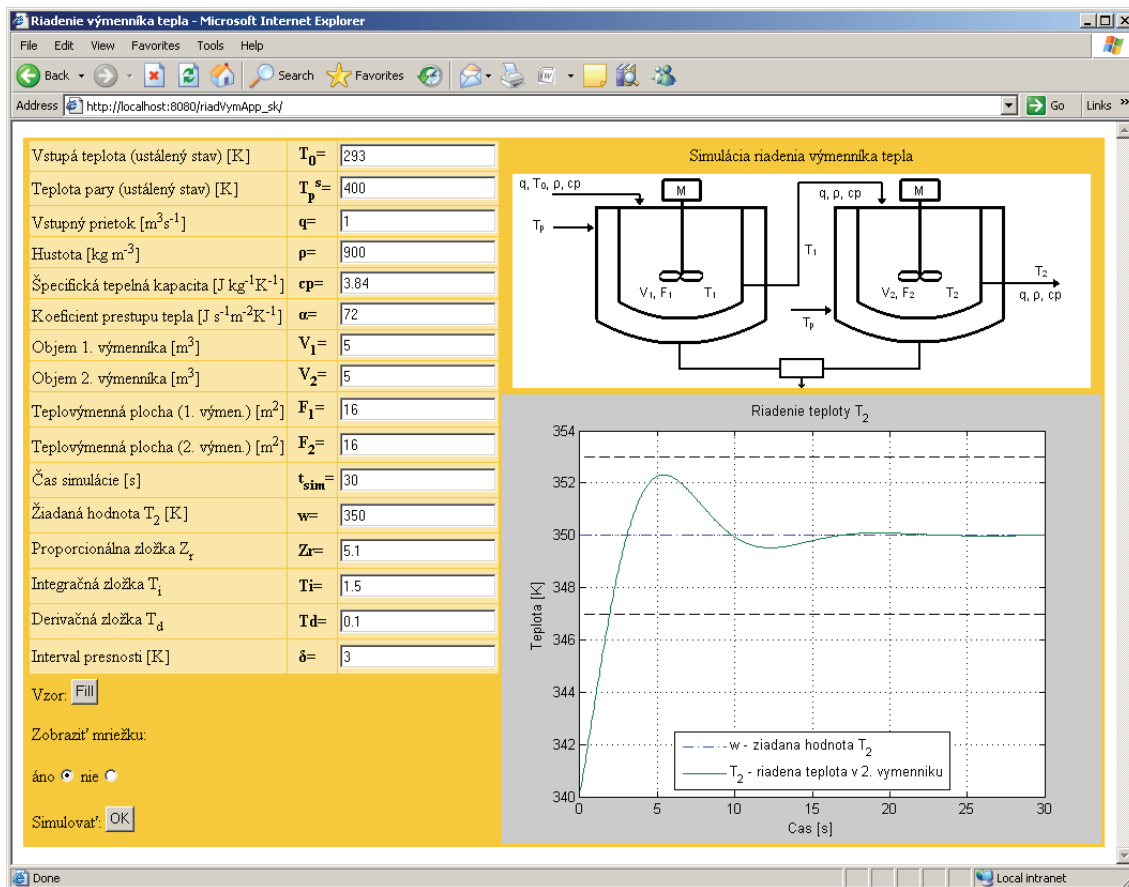
Stránka (obr. 29) obsahuje polia, ktoré užívateľ môže vyplniť, alebo použije tlačidlo [FILL] a polia sa vyplnia prednastavenými hodnotami. Nižšie je možnosť vykreslenia mriežky v grafe. V pravej hornej časti je možné si vybrať medzi zásobníkmi s interakciou, alebo bez interakcie. Ak sú všetky polia vyplnené, stačí kliknúť na [OK]. Po zbehnutí funkcie sa v pravom dolnom rohu zobrazí vykreslený graf:



Obr. 29 Internetové rozhranie pre riadenie zásobníkov kvapaliny

4.4.3 Výmenníky tepla - internetové rozhranie

Podobne ako pre zásobníky kvapaliny sú tu polia na vyplnenie (obr. 30), alebo tlačidlom [FILL] sa polia vyplnia prednastavenými hodnotami. Rozhranie tiež obsahuje voľbu mriežky. Po kliknutí na [OK] sa vypočíta funkcia a zobrazí sa graf:



Obr. 30 Internetové rozhranie pre riadenie výmenníkov tepla

5 Záver

Práca opisuje inštaláciu potrebných programov a konfiguráciu prostredia, vytvorenie Java archívov z napísaných m-funkcií, a ich správne implementovanie do kódu JSP stránky. Simulácia riadenia výmenníkov tepla a zásobníkov kvapaliny je založená na matematických modeloch zariadení. Vytvorené virtuálne laboratórium by malo pomôcť pri štúdiu predmetov vyučovaných na Oddelení informatizácie a riadenia procesov a pochopení riadenia základných technologických procesov.

Súčasti virtuálneho laboratória boli umiestnené na e-learningový portál Moodle. Toto laboratórium je možné veľmi ľahko rozšíriť o ďalšie simulácie a pomôcky pre študentov, a priblížiť im tak jednotlivé aspekty riadenia a automatizácie, keďže stačí skompilovať už raz napísané m-funkcie.

6 Zoznam použitej literatúry

- [1] *Options for Deploying MATLAB Applications via the Web* na mathworks.com,
Dostupné na internete:
<www.mathworks.com/products/new_products/webserver_discontinued.html>,
On-line, 5. 5. 2011
- [2] *Runge–Kutta methods* na Wikipedia.org, Dostupné na internete:
<http://en.wikipedia.org/wiki/Runge-Kutta_methods>,
On-line, 5. 5. 2011
- [3] *Numerické metódy* na VUTBR.cz, Dostupné na internete:
<<http://www.fce.vutbr.cz/studium/materialy/Dynsys/kap7/kap7.htm>>,
On-line, 5. 5. 2011
- [4] *Numerical ordinary differential equations* na Wikipedia.org,
Dostupné na internete:
<http://en.wikipedia.org/wiki/Numerical_ordinary_differential_equations>,
On-line, 5. 5. 2011
- [5] Kalúz, M., *Virtuálne laboratórium*, Diplomová práca, FCHPT STU v Bratislave, 2010
- [6] *PID regulátor* na Wikipedia.org, Dostupné na internete:
<http://cs.wikipedia.org/wiki/PID_regulátor>, On-line, 5. 5. 2011
- [7] *PID regulátor* na Wikipedia.org, Dostupné na internete:
<http://sk.wikipedia.org/wiki/PID_regulátor>, On-line, 5. 5. 2011
- [8] *PID controller* na Wikipedia.org, Dostupné na internete:
<http://en.wikipedia.org/wiki/PID_controller>, On-line, 5. 5. 2011
- [9] Mikleš, J., Fikar, M., *Modelovanie, identifikácia a riadenie procesov II*,
STU v Bratislave, 2008
- [10] Bakošová, M., Fikar, M., Čirka, L., *Základy automatizácie*,
FCHPT STU v Bratislave, 2002
- [11] *Desktop and Web Deployment* na MathWorks.com, Dostupné na internete:
<<http://www.mathworks.com/desktop-web-deployment/demos.html>>,
On-line, 5. 5. 2011
- [12] *JAR file* na Wikipedia.org, Dostupné na internete:
<http://en.wikipedia.org/wiki/JAR_file>, On-line, 5. 5. 2011

-
- [13] *Matlab Builder for .Net a ASP.NET web aplikácie* na TUKE.sk,
Dostupné na internete:
<<http://MATLAB.fei.tuke.sk/vos/NET/dotnetindex.html>>, On-line, 5. 5. 2011
- [14] *MATLAB Builder JA Demos and Webinars* na MathWorks.com,
Dostupné na internete:
<<http://www.mathworks.com/products/javabuilder/demos.html>>,
On-line, 5. 5. 2011
- [15] *Mono (software)* na Wikipedia.org, Dostupné na internete:
<[http://en.wikipedia.org/wiki/Mono_\(software\)](http://en.wikipedia.org/wiki/Mono_(software))>, On-line, 5. 5. 2011
- [16] *.NET Framework* na Wikipedia.org, Dostupné na internete:
<http://en.wikipedia.org/wiki/.NET_Framework>, On-line, 5. 5. 2011
- [17] *Java performance* na Wikipedia.org, Dostupné na internete:
<http://en.wikipedia.org/wiki/Java_performance>, On-line, 5. 5. 2011
- [18] *MATLAB Builder JA User's Guide* na MathWorks.com, Dostupné na internete:
<http://www.mathworks.com/help/pdf_doc/javabuilder/javabuilder.pdf>,
On-line, 5. 5. 2011
- [19] *Environment variable* na Wikipedia.org, Dostupné na internete:
<http://en.wikipedia.org/wiki/Environment_variable>, On-line, 5. 5. 2011
- [20] Mikleš, J., Fikar, M., *Modelovanie, identifikácia a riadenie procesov I*,
STU v Bratislave, 1999
- [21] Bakošová, M., FIKAR, M., *Matematické modely chemickotechnologických systémov, zásobníky kvapaliny s interakciou*, (pdf prednášky),
FCHPT STU v Bratislave
- [22] Bakošová, M., *Procesy s prestupom tepla, Plášťový výmenník tepla*,
(pdf prednášky), FCHPT STU v Bratislave