

Návrh SW k ovládání a řízení laboratorního tepelného modelu

Design of Operating and Control SW for a Laboratory Heating System

Jaromír Kupka

Bakalářská práce
2018



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jaromír Kupka**
Osobní číslo: **A14116**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Návrh SW k ovládání a řízení laboratorního tepelného modelu**
Téma anglicky: **The Design of Operating and Control SW for a Laboratory Heating Model**

Zásady pro vypracování:

1. Uvedte strukturu, zapojení a funkce laboratorního tepelného modelu.
2. Popište prostředí Matlab/Simulink s důrazem na prostředky pro tvorbu grafického uživatelského rozhraní.
3. Vytvořte SW uživatelské rozhraní pro monitoring, ovládání a řízení laboratorního tepelného modelu implementující vybrané funkce a umožňující jeho snadné rozšiřování.
4. Ověřte jeho funkcionalitu v laboratorních podmínkách.
5. Zpracujte podrobnou uživatelskou a programátorskou příručku pro práci s vytvořeným SW.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **DOSTÁLEK, Petr.** Výukový model tepelné soustavy s dopravním zpožděním. Uživatelský manuál. Zlín: UTB ve Zlíně, 2006. 20 s.
2. **MATLAB GUI.** MATLAB [online]. MathWorks [cit. 2016-12-22]. Dostupné z: <https://www.mathworks.com/discovery/matlab-gui.html>
3. **PERŮTKA, Karel.** MATLAB – Základy pro studenty automatizace a informačních technologií, 1. vyd. Zlín: UTB ve Zlíně, 2005. 304 s. ISBN 80-7318-355-2.
4. **ŠULC, Bohumil a Miluše VÍTEČKOVÁ.** Teorie a praxe návrhu regulačních obvodů. Praha: Vydavatelství ČVUT, 2004. ISBN 80-01-03007-5.
5. **ŠVEHLÍKOVÁ, Martina.** Ověření matematického modelu tepelné soustavy. Simulační modely vybraných řízených systémů. Zlín, 2009. Diplomová práce. UTB ve Zlíně. Vedoucí práce Libor Pekař.

Vedoucí bakalářské práce:

Ing. Libor Pekař, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

15. prosince 2017

Termín odevzdání bakalářské práce:

15. května 2018

Ve Zlíně dne 15. prosince 2017



doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

.....
podpis diplomanta

ABSTRAKT

Tato bakalářská práce se zabývá vytvořením softwarového uživatelského rozhraní v programovém prostředí MATLAB/Simulink a to za účelem monitoringu, ovládání a řízení laboratorního tepelného modelu. Toto rozhraní je postaveno na základě propojení modelu s řídicím počítačem vyhotoveno jako diplomová práce panem Bc. Jakubem Krčmářem. První část této práce popisuje jak laboratorní tepelný model, tak i použité programovací prostředí zatímco druhá část vylicuje výsledné uživatelské rozhraní a jeho používání.

Klíčová slova: laboratorní tepelná soustava, MATLAB/Simulink, grafického uživatelského rozhraní, PID regulátor, Smithův prediktor

ABSTRACT

This bachelor's thesis addresses creation of a software user interface in programming environment MATLAB/Simulink and that for a purpose of monitoring, operation and control of a laboratory heat system. This interface is build on basis of connecting the model with control computer created as master thesis by Mr. Jakub Krčmář. In first part of this thesis I'm going to describe both the laboratory heat model and used programming environment while second part depicts resulting user interface and it's use.

Keywords: laboratory heat system, MATLAB/Simulink, graphical user interface, PID controller, Smith predictor

Rád bych poděkoval všem, kdo se podílel na dokončení této práce a zejména bych chtěl poděkovat mému vedoucímu, panu Ing. Liboru Pekařovi, Ph.D. za odborné vedení, ochotu a trpělivost jakož i za rady a spolupráci.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 LABOLATORNÍ TEPELNÝ MODEL	10
1.1 POPIS MODELU	10
1.2 UVEDENÍ MODELU DO PROVOZU	12
1.3 ÚDRŽBA MODELU	13
1.4 ZAPOJENÍ MODELU	14
1.5 DOPORUČENÍ PROVOZU	14
2 PROSTŘEDÍ MATLAB/SIMULINK	15
2.1 MATLAB	15
2.2 SIMULINK.....	16
2.3 GUIDE	17
II PRAKTICKÁ ČÁST	19
3 UŽIVATELSKÝ MANUÁL	20
3.1 POPIS PROGRAMU	20
3.2 GRAFICKÝ VZHLED.....	22
3.3 FUNKCE.....	23
3.3.1 Soubor InterfaceBoxGUI_B.m	23
3.3.1 Soubor moje.m	26
ZÁVĚR	27
ZÁVĚR V ANGLIČTINĚ	28
SEZNAM POUŽITÉ LITERATURY	29
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	30
SEZNAM OBRÁZKŮ	31
SEZNAM PŘÍLOH	32

ÚVOD

Existuje lidská vlastnost, která nás neustále pohání v technickém vývoji a tou je lenost. Ač se to může zdát nesmyslné, byla to lenost, která přiměla pračlověka vymyslet kolo a tím si usnadnit práci sobě a těm co přijdou po něm. Touha usnadnit si život nutí lidstvo zkoumat a zlepšovat. Úmysl této práce je stejný, usnadnění využívání laboratorního tepelného modelu skrz vytvoření jednoduše použitelného uživatelského rozhraní, jenž se bude dít dále rozšířit o další funkce.

Laboratorní model tepelné soustavy s dopravním zpožděním byl navržen a postaven jako výukový nástroj na Fakultě aplikované informatiky [1], [5], [6], [7]. Tento model měří teploty na třech místech a uživatel může nastavit tři různé prvky, výkon čerpadla, topení a ventilátoru [1], [5]. Původní propojení modelu a řídicího počítače bylo řešeno prostřednictvím jednotky CTRL V3 a komunikačního protokolu RS 232, byl vylepšen vyměněním starého propojení za nové, provedené pomocí multifunkční vstupně – výstupní technologické karty MF 624, při zachování původní metody připojení stále funkční [6]. Díky této úpravě je možno řídit a ovládat tepelný model přes programovací prostředí MATLAB/Simulink.

Nicméně, toto řešení, ač mnohem pohodlnější než předchozí konfigurace, se dá ještě zjednodušit a to s využitím MATLAB aplikace GUI Development Environment (zkráceno na GUIDE) který umožňuje snadné a intuitivní vytvoření uživatelského rozhraní [2].

Tato bakalářská práce je rozdělena do dvou částí, teoretické a praktické. V teoretické části proberu laboratorní tepelnou soustavu a programovací prostředí MATLAB/Simulink. V praktické části naleznete popis programu a uživatelský manuál.

I. TEORETICKÁ ČÁST

1 LABORATORNÍ TEPELNÝ MODEL

Laboratorní model tepelné soustavy byl navržen a postaven jako výukový nástroj na Fakultě aplikované informatiky University Tomáše Bati ve Zlíně, za účelem testování řídicích algoritmů na systému, kde významnou roli představuje dopravní zpoždění [1], [5], [6], [7].

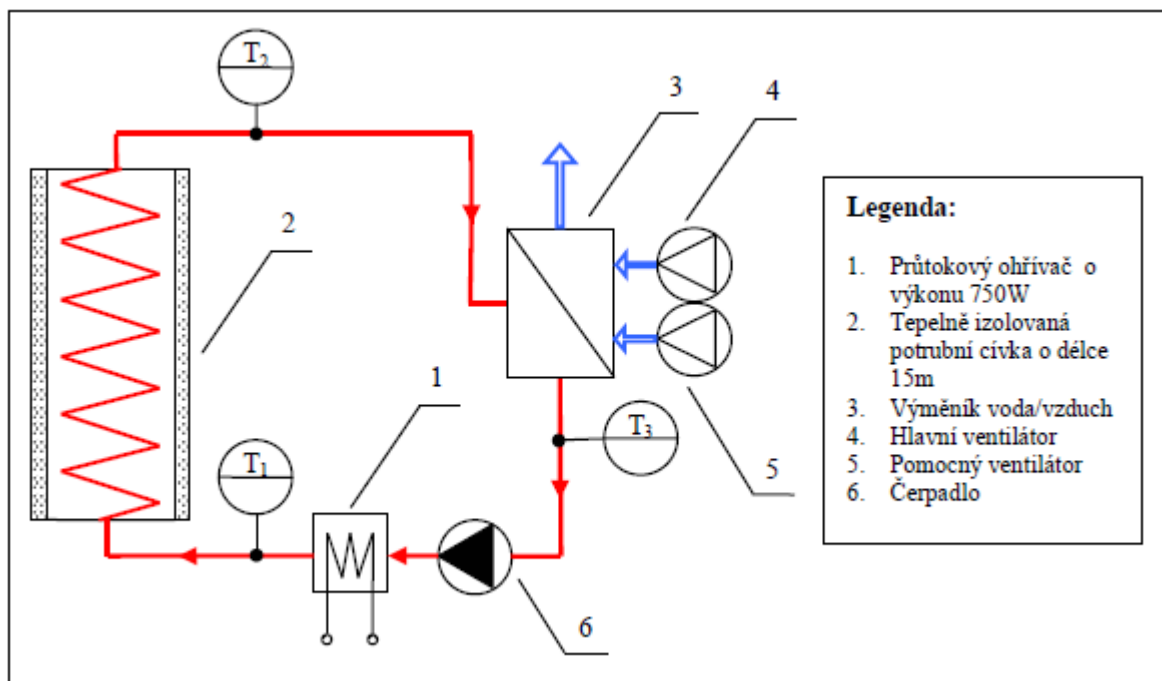


Obr. 1. Laboratorní tepelný model.

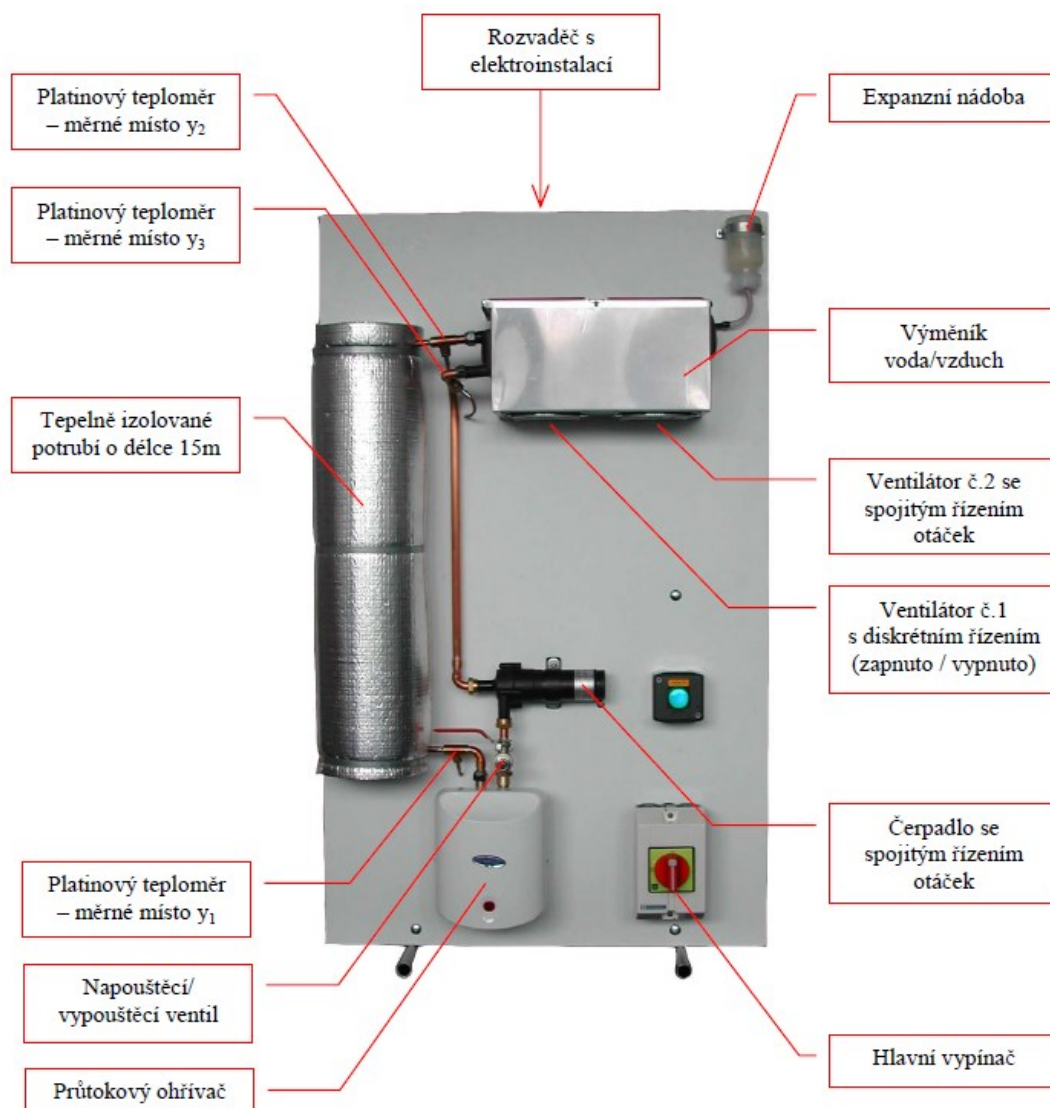
1.1 Popis modelu

Výukový tepelný model s dopravním zpožděním je založen na principu přenosu tepla pomocí teplotonosného média prostřednictvím potrubního systému. Blokové schéma modelu tepelné soustavy s dopravním zpožděním je uvedeno na obrázku (Obr. č. 2). Teplotonosné médium je transportováno pomocí spojitě regulovatelného čerpadla (6) do průtokového ohříváče (1) o výkonu 750W. Teplota vody vystupující z ohříváče je měřena platinovým teploměrem T_1 . Ohřátá voda dále vstupuje do tepelně izolované potrubní cívky (2), jenž je,

tvořena měděným potrubím o délce 15 m. Zde vzniká v závislosti na zvolených otáčkách čerpadla dopravní zpoždění v rozmezí 50 až 200 s. Spotřebič tepelné energie představuje výměník tepla typu voda/vzduch (3), který předává tepelnou energii teplotnosného média do okolního prostředí. Úroveň spotřeby tepla lze nastavit pomocí dvou regulovatelných ventilátorů (4), (5). Ventilátor (5) je možno ovládat pouze dvoustavově (tedy stav zapnuto/vypnuto), ventilátor (4) je řízen spojitě a slouží primárně pro generování poruchové veličiny, zatímco ověřujeme vlastnosti navržených regulátorů. Teplota vody vstupující do výměníku a z něj vystupující se měří teploměry T_2 a T_3 . Z výměníku se voda vrací zpět do čerpadla a celý koloběh teplotnosného média se opakuje znovu. Vliv tepelné roztažnosti vody je kompenzována expanzní nádobou, umístěnou na nejvyšším bodu tepelného modelu [1], [5], [6].



Obr. 2. Blokové schéma tepelného modelu s dopravním zpožděním.



Obr. 3. Popis komponentů laboratorního tepelného modelu.

1.2 Uvedení modelu do provozu

Před prvním spuštěním modelu tepelné soustavy je nutno nejdříve naplnit celý potrubní systém destilovanou vodou. K tomuto účelu slouží napouštěcí / vypouštěcí ventil umístěný na vstupu průtokového ohřivače (Obr. č. 4). Na tento ventil připojte externí čerpadlo, otevřete ventil a pomalu doplňujte systém vodou tak, aby vzduch mohl volně unikat přes odvzdušňovací otvor v expanzní nádobě. Jakmile je expanzní nádoba do jedné třetiny zaplněná vodou, zavřete napouštěcí ventil a následně proveďte odvzdušnění. **Odvzdušňování se provádí při zapnutém tepelném modelu, neboť je nutné, aby běželo oběhové čerpadlo.** Vzduchové bublinky se postupně začnou hromadit v nejvyšším bodě potrubního systému, neboli v expanzní nádobě. Dokonalého odvzdušnění se docílí tak, že stisknete ex-

panzní nádobu, následně ucpete prstem odvzdušňovací otvor a nádobku uvolníte. Tím se vytvoří v systému podtlak a vzduchové bublinky rychleji proudí do expanzní nádoby. Tento postup je nutné tak dlouho opakovat, dokud je v potrubí vzduch. Vlivem odvzdušnění poklesne hladina vody v expanzní nádobě. Na požadovanou výši ji doplňte přímo přes odvzdušňovací otvor pomocí injekční stříkačky. Doplnění přes hlavní napouštěcí / vypouštěcí ventil není vhodné, protože by mohlo způsobit další zavzdušnění systému. Pokud je model dokonale odvzdušněn, odmontujte páčku vypouštěcího ventilu. Zamezí se tím nechtěnému vypuštění teplonosného média z potrubního systému. Nyní se již může přikročit k propojení modelu s řídicím počítačem [1], [5], [6].

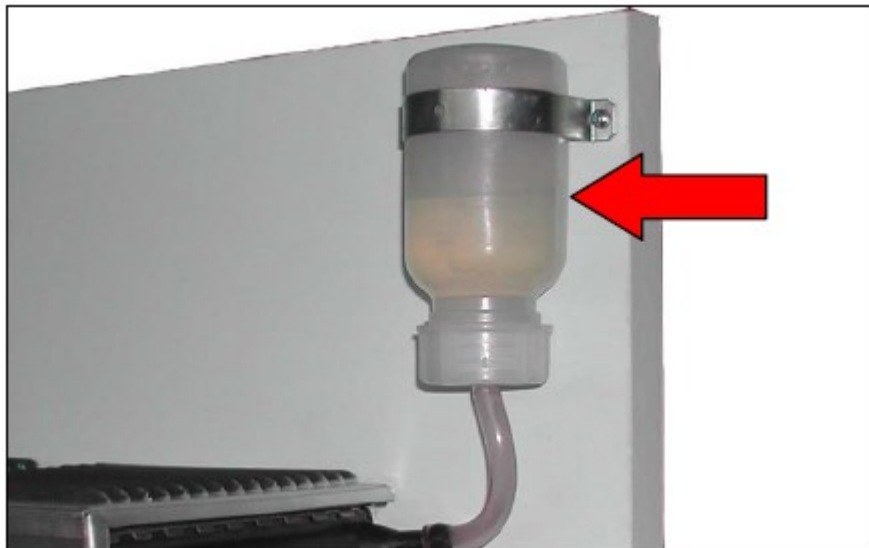


Obr. 4. Ventil pro napouštění a vypouštění vody.

1.3 Údržba modelu

Zařízení nevyžaduje žádnou zvláštní údržbu. Nicméně, je ale zapotřebí **před každým měřením zkontrolovat hladinu destilované vody v expanzní nádržce**. Ta by se měla udržovat v jedné třetině výšky nádoby, jak je naznačeno na (Obr. č. 5). Kontrola se provádí zásadně při běžné pokojové teplotě tepelného média. Doplnění většího množství destilova-

né vody může způsobit během měření přetečení expanzní nádoby přes odvzdušňovací otvor [1], [5], [6].



Obr. 5. Kontrola hladiny vody v expanzní nádobě.

1.4 Zapojení modelu

Ovládání tepelného systému je realizováno pomocí multifunkční vstupně - výstupní měřicí karty MF624 od firmy Humusoft. Laboratorní tepelný systém je zařízení se čtyřmi vstupy a třemi výstupy. Ovládací prostředí bylo vytvořeno v programu MATLAB/SIMULINK (verze 2009b) panem Krčmářem [6].

Alternativou je původní jednotka pro sbírání dat CTRL V3, využívající standardní sériové rozhraní RS232 [1].

1.5 Doporučení provozu

- Referenční hodnotu tepelné ochrany nenastavovat na hodnoty přesahující 80 °C, doporučená hodnota je 75 °C [6].
- Výkon ventilátoru č. 1 je doporučen nastavit na 100 %. V případě snížení jeho výkonu, dojde ke zmenšení velikosti pracovního rozsahu tepelného modelu z hlediska regulace [6].
- Periodu vzorkování ovládacího bloku, vyjádřené proměnnou $T0model$, je doporučeno volit 0.1 s. Proměnnou $T0model$ používat pouze v bloku "Ovládání tepelného modelu", nikde jinde [6].

2 PROSTŘEDÍ MATLAB/SIMULINK

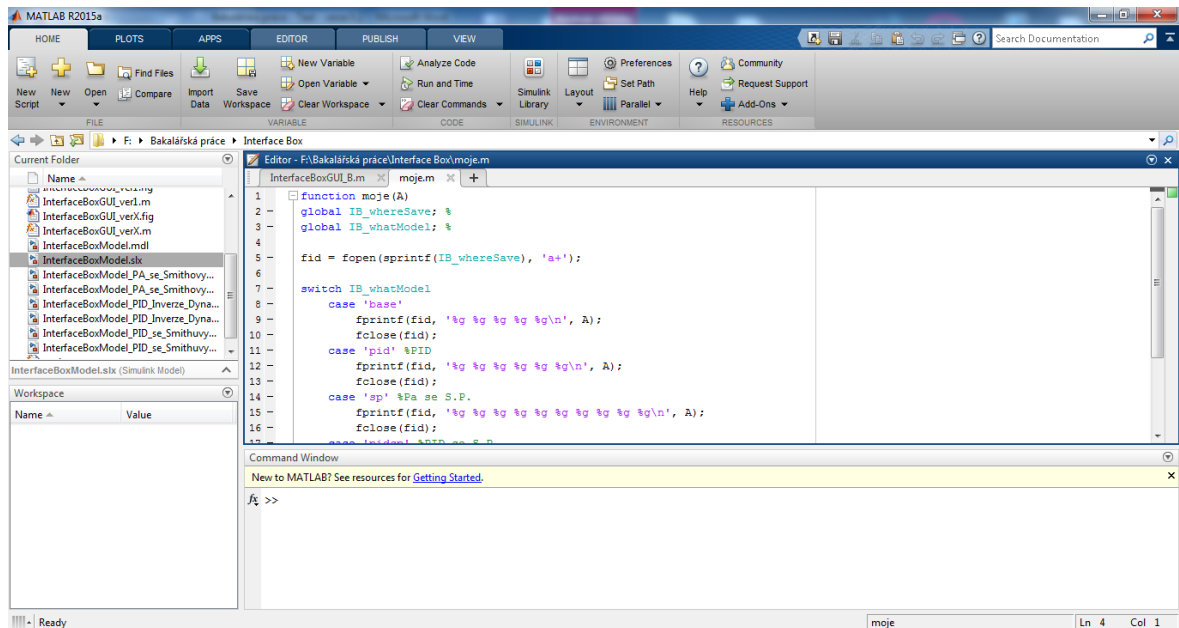
Programovací prostředí MATLAB a Simulink byli vytvořeny společností MathWorks, s.r.o. sídlící ve městě Natick, státu Massachusetts, Spojené státy americké [2]. První verze MATLABU byla vydána kolem roku 1985 pro platformu PC XT. Verze pro Windows vyšla přibližně v roce 1994 a o rok později se ukončil vývoj pro operační systém MS DOS [3]. V roce 2002 byl uveden na trh MATLAB verze 6.5 pyšící se propracovaným grafickým zpracováním a verzí Simulink 5.0.2. Ve chvíli publikování této práce je MATLAB i Simulink ve verzi 9.1 nazvané R2018a [2].

K provozování MATLAB/Simulink je třeba zpoplatněná licence. Nejznámější konkurent je Wolfram Mathematica.

2.1 MATLAB

MATLAB, jehož název je zkratka Matrix Laboratory, je integrované programovací prostředí využívané zejména pro výpočty vědeckotechnického charakteru. Základem MATLABu jsou operace s maticemi, umožňující mimo jiné násobení matic, jejich inverze a výpočet determinantu, ať už jsou obsahem matic reálná či komplexní čísla. Toto byl počáteční koncept jak je patrné z plného názvu, prostý maticový kalkulátor, ke kterému postupně přibývaly další a další funkce, dokud se z něj nestalo dnešní všestranné programovací prostředí [3].

Velkou výhodou MATLABu jsou flexibilní datové struktury které umožňují ukládání proměnných různých typů. Jelikož jsou v MATLABu proměnné uloženy v datových řadách neboli v polích aniž by jim byl přidělen datový typ, mohou být vedle sebe v buňkách pole prvky zcela jiných datových typů. Díky tomu je přetypování z jednoho datového typu na druhý jednoduché a můžeme skládáním vytvořit složité datové struktury bez problémů. Můžeme tedy do proměnné vložit textový řetězec string a následně ho přepíšeme integerem aniž by to MATLABu vadilo. Nevýhodou této metody je nutnost uživatele hlídat si typovou kontrolu [2], [3].

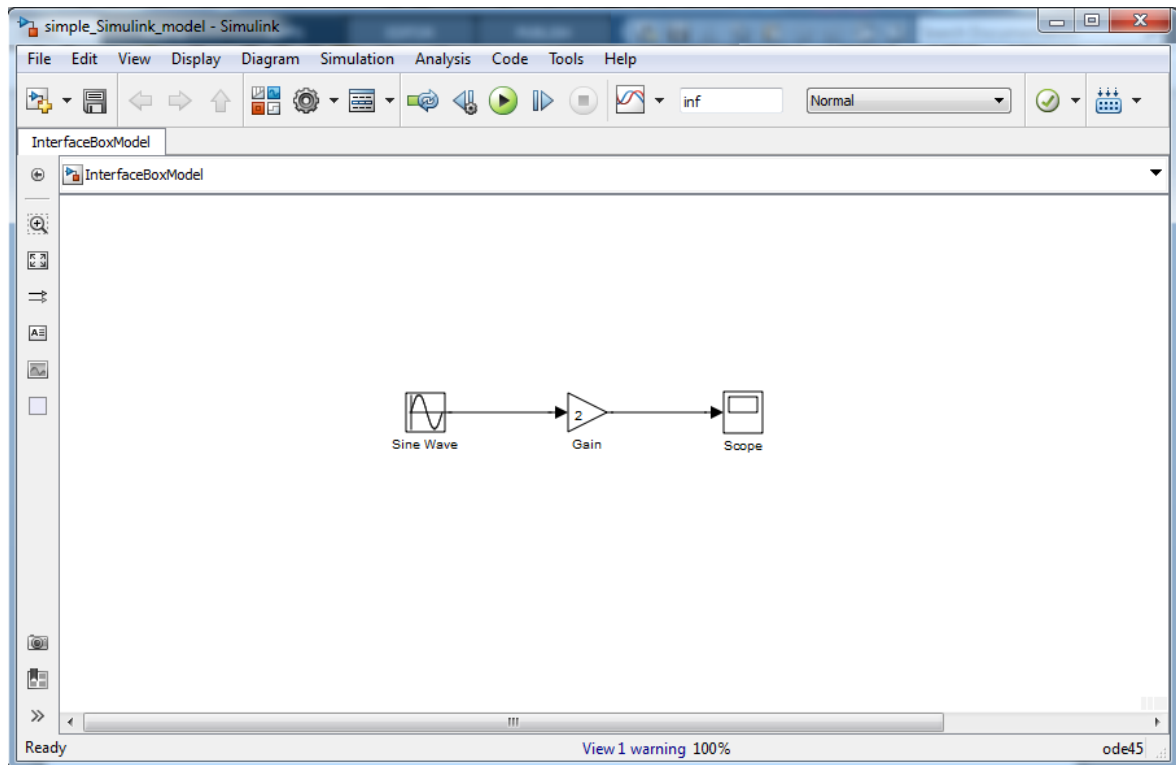


Obr. 6. Uživatelské rozhraní MATLABu.

Programovací prostředí MATLAB má svůj vlastní skriptovací programovací jazyk založený na programovacím jazyku Fortran, jmenující se taktéž MATLAB na kterém je prostředí postaveno. Lze jej používat stylem zadávání příkazů do příkazové řádky. Alternativou je vytváření nejrůznějších Skript a Funkcí které se ukládají do souborů, které mají koncovku *.m, tzv. M-files. Je to multiparadigmatický programovací jazyk a tudíž podporuje objektové-orientované programování [2], [3].

2.2 Simulink

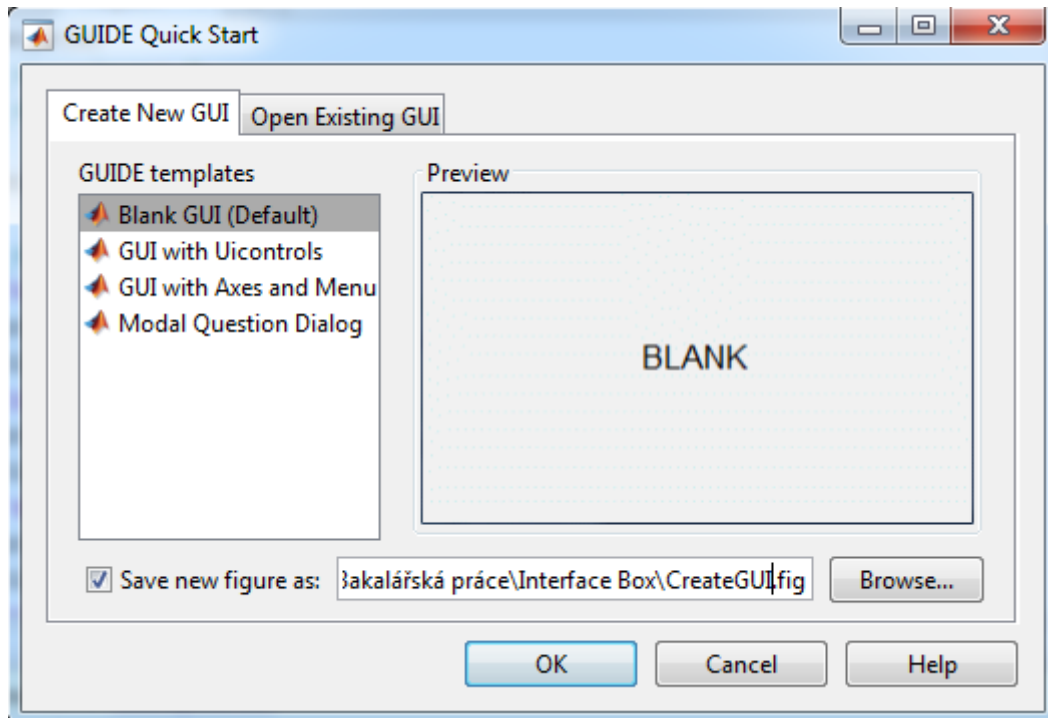
Simulink je grafické programovací prostředí úzce spjata s prostředím MATLAB, sloužící k simulaci a modelování dynamických systémů. V tomto prostředí se pracuje s předdefinovanými bloky různých druhů vybírané z knihovny, každý provádějící jiný úkon, které umístíme na pracovní plochu a propojíme do celistvého modelu. V případě že žádný blok v knihovně se nehodí do uživatelem zamýšleného modelu, lze vytvořit vlastní blok. Model Simulinku je v podstatě virtuální projekcí reálného systému, tyto soubory se ukládají do souborů s koncovkou *.mdl v případě starších verzí nebo *.slx u novějších verzí MATLABu. Model po spuštění simuluje v reálném čase [2].



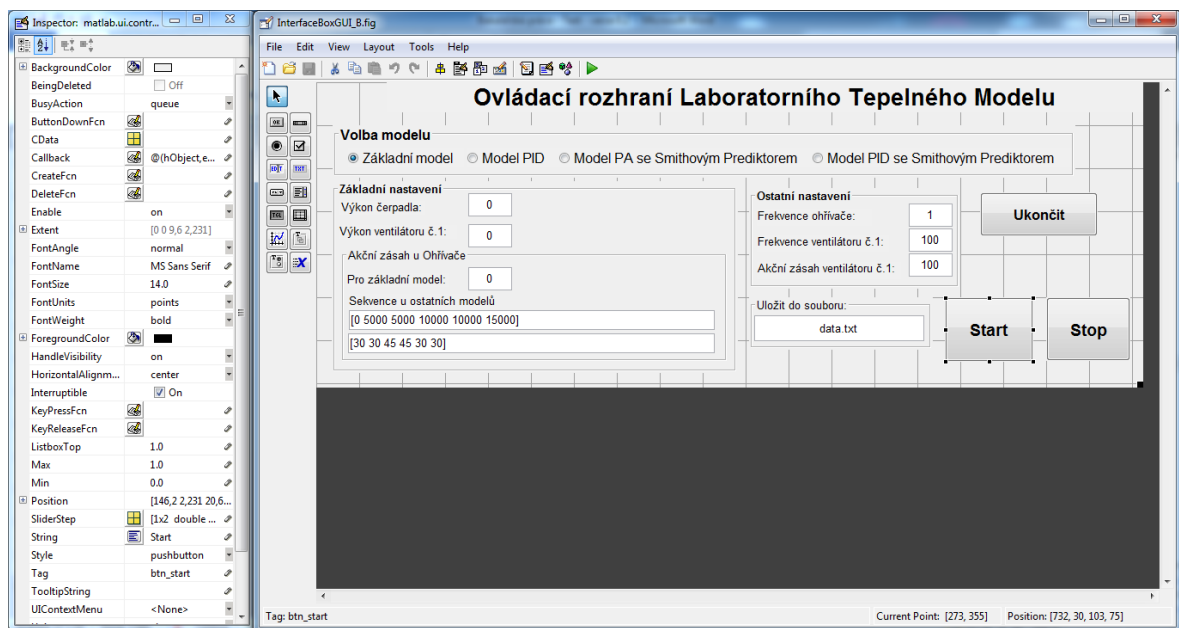
Obr. 7. Uživatelské rozhraní Simulinku.

2.3 GUIDE

Tvorba grafického uživatelského rozhraní se provádí v MATLAB aplikaci Graphical User Interface Development Environment (zkráceno na GUI Development Environment nebo GUIDE). Tato aplikace otevře jednoduchý editor, ve kterém se vytvoří grafická podoba rozhraní. V editoru, můžeme na plochu jenž znázorňuje okno budoucího rozhraní, vkládat prvky jako jsou tlačítka, popisky, boxy s editovatelným textem, checkboxy, různá menu, atd., tyto prvky rozmístíme do layoutu dle přání a potřeby. Každý tento prvek má několik parametrů které můžeme upravovat. Kromě grafické podoby ukládané do souboru s koncovkou *.fig, se také vytvoří M-soubor, do kterého se naprogramuje logika rozhraní.



Obr. 8. Create new GUI menu.



Obr. 9. GUIDE editor.

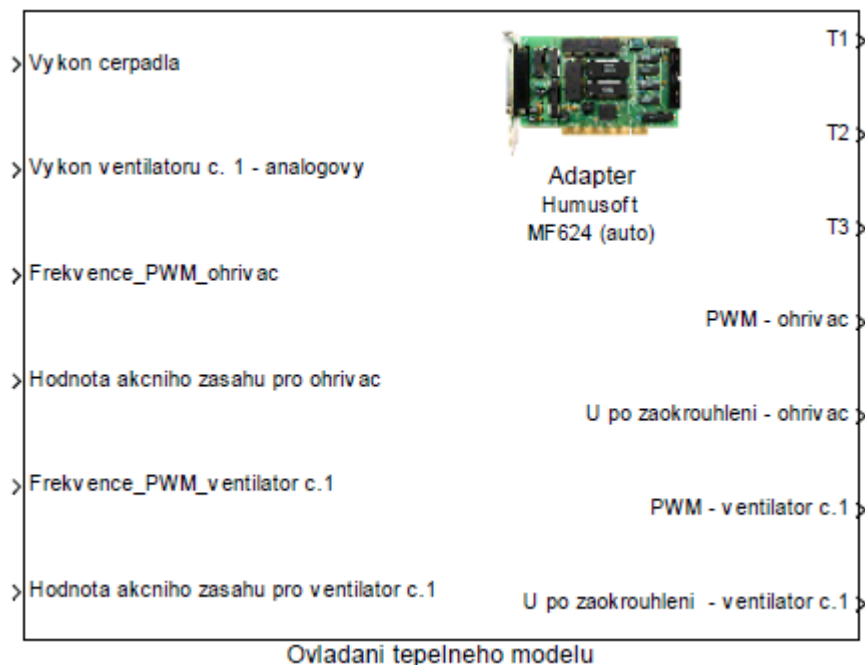
II. PRAKTICKÁ ČÁST

3 UŽIVATELSKÝ MANUÁL

text

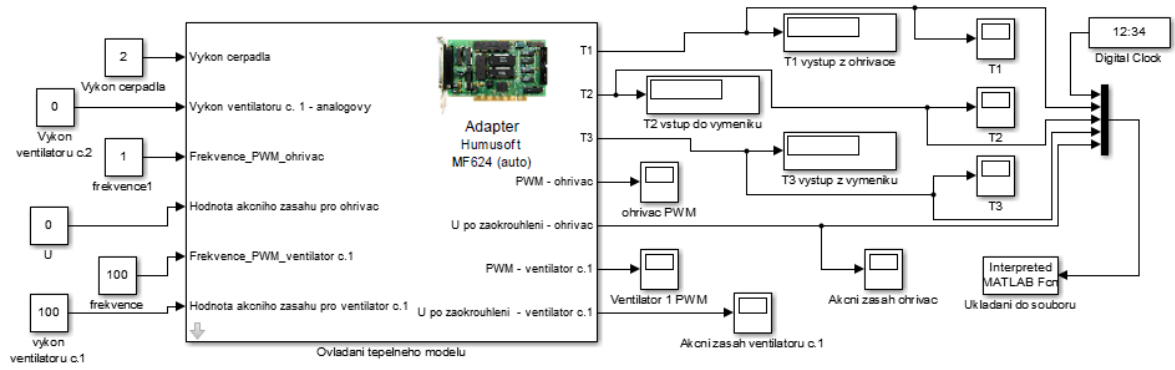
3.1 Popis programu

Program je vytvořen standardně pomocí GUIDE nástroje v prostředí MATLAB/Simulink. Využívá propojení laboratorního tepelného modelu s řídicím počítačem od pana Krčmáře, a také používá jeho Simulinkový blok nazvaný „Ovládání tepelného modelu“ jenž je virtuální podobou reálného systému. Tento blok, jenž má šest vstupních a sedm výstupních parametrů, je připojen skrz adaptér a ovládá tepelný model [6].

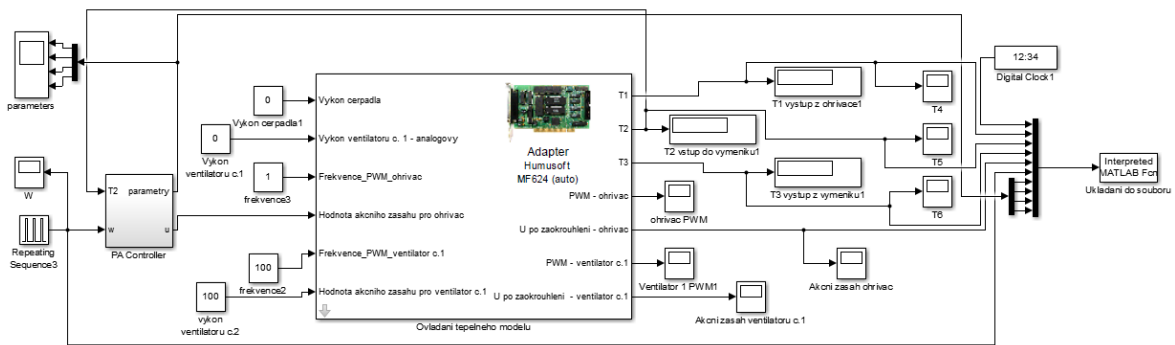


Obr. 10. Simulink blok „Ovládání tepelného modelu“.

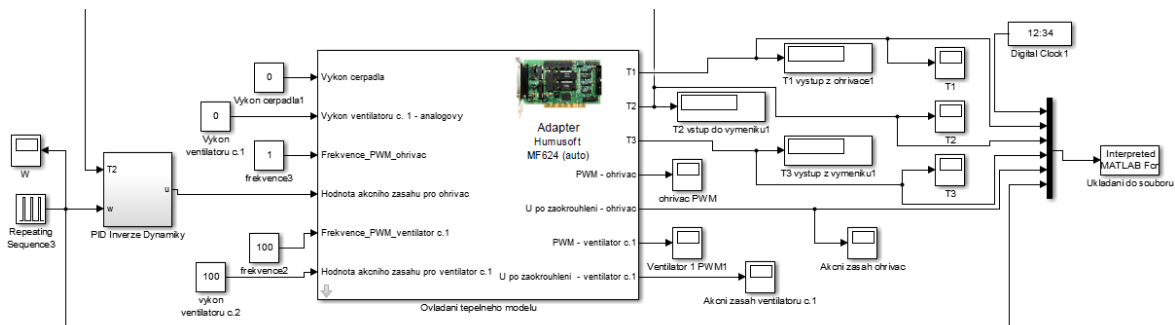
Pana Krčmář pomocí tohoto ovladače vytvořil čtyři různé varianty modelu, každý se svým Simulinkovým souborem. Na tyto soubory se napojuje toto uživatelské rozhraní, hodnoty zadané v rozhraní budou předány do Simulinkového modelu načež se model spustí. Rozhraní také umožňuje výběr mezi těmito čtyřmi variantami a volbu kam se z něho mají ukládat data [6].



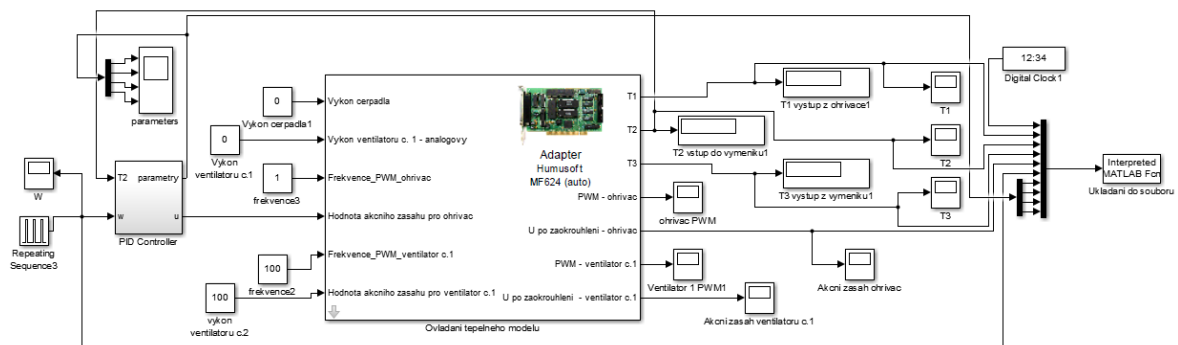
Obr. 11. Simulink model „InterfaceBoxModel“.



Obr. 12. Simulink model „InterfaceBoxModel
_PA_se_Smithovym_Prediktorem“.

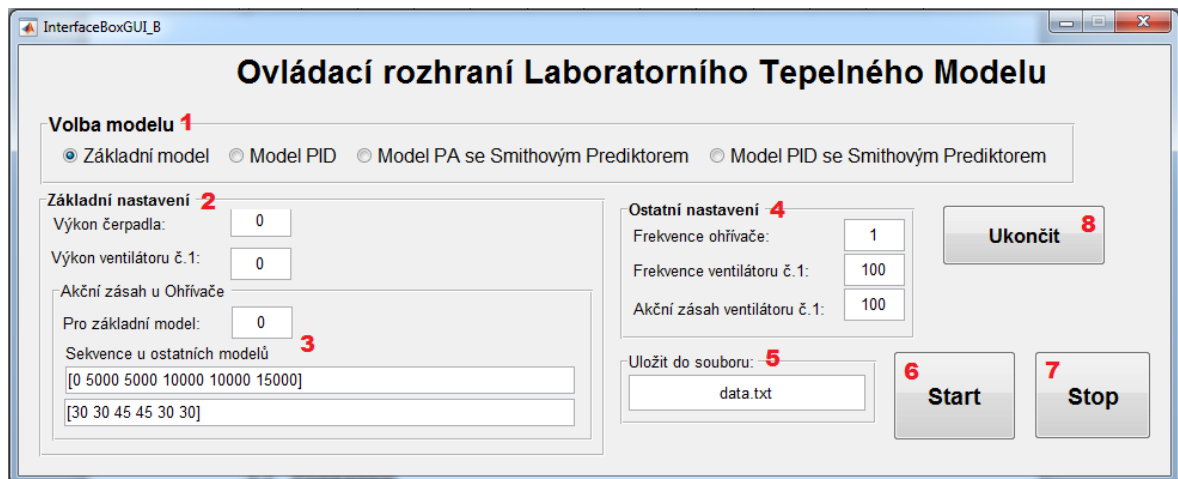


Obr. 13. Simulink model „InterfaceBoxModel_PID_Inverze_Dynamiky“.



Obr. 14. Simulink model „InterfaceBoxModel_PID_se_Smithuvym_Prediktorem“.

3.2 Grafický vzhled



Obr. 15. Ovládací rozhraní Laboratorního Tepelného Modelu.

Zde vidíme grafickou podobu uživatelského rozhraní.

- 1) Tlačítková skupina nazvaná Volba modelu obsahuje čtyři přepínače, jenž volí, který ze čtyř Simulinkových modelů použít.
- 2) Panel Základní nastavení umožňuje vložit hodnoty dvou vstupních parametrů a to Výkon čerpadla a Výkon ventilátoru č. 1.
- 3) Panel Akční zásah u ohřivače umožňuje zadat tento vstupní parametr. Jelikož tento parametr má rozdílnou podobu záleží na použitém modelu, Základní model má konstantu, zatímco ostatní modely používají blok Repeating Sequence.
- 4) V Panelu Ostatní nastavení udáváme zbylé 3 vstupní parametry.
- 5) Zde určíme, do kterého souboru se mají ukládat data z laboratorní tepelné soustavy.
- 6) Tlačítko Start, kterým pošleme parametry do zvoleného modelu a spustíme jej.

- 7) Tlačítko Stop, kterým zastavíme spuštěný model.
- 8) Tlačítko Ukončit, kterým ukončíme Ovládací rozhraní.

3.3 Funkce

Funkce jsou ve dvou souborech, InterfaceBoxGUI_B.m který obsluhuje grafické uživatelské rozhraní a moje.m, obsluhující ukládání dat do poznámkového bloku, jenž bylo převzato od pana Krčmáře a upraveno.

3.3.1 Soubor InterfaceBoxGUI_B.m

První a nejdůležitější funkce je obsluha Tlačítka Start:

```
% --- Executes on button press in btn_start.
function btn_start_Callback(hObject, eventdata, handles)

var IB_V1 %Výkon čepřadla
var IB_V2 %výkon ventilátoru
var IB_V3 %Hodnota akčního zásahu pro ohřivač
var IB_F1 %Frekvence ohřivač
var IB_F2 %Frekvence ventilátoru
var IB_F3 %Hodnota akčního zásahu ventilátor
var IB_RSt %Repeating sequence - time
var IB_RSo %Repeating sequence - output
global IB_whereSave %
global IB_whatModel %
%global IB_V1 IB_V2 IB_V3 IB_F1 IB_F2 IB_F3 IB_RSt IB_RSo IB_whereSave
IB_whatModel
```

Zde definuji lokální proměnné používané v této funkci a dvě globální proměnné které budou použité nejenom v této funkci ale i v souboru moje.m.

```
IB_V1 = str2double(get(handles.prom_vykoncerpadla, 'string'));
IB_V2 = str2double(get(handles.prom_vykonventl, 'string'));
IB_F1 = str2double(get(handles.prom_, 'string'));
IB_F2 = str2double(get(handles.prom_, 'string'));
IB_F3 = str2double(get(handles.prom_, 'string'));
IB_whereSave = get(handles.prom_savehere, 'string');
```

Tento kód vezme hodnoty zadané v Ovládacím rozhraní a dosadí je do příslušných proměnných ve vhodném datovém typu, s využitím příkazu `str2double`. Výjimkou je parametr akčního zásahu ventilátoru, jenž bude obslužen v následující sekci.

```
if get(handles.ui_bg_pickmodel.SelectedObject, 'tag') == rb_basemodel
    IB_whatModel = 'base';
    IB_V3 = str2double(get(handles.prom_zasahohr_base, 'string'));
elseif get(handles.ui_bg_pickmodel.SelectedObject, 'tag') == rb_modelpid
    IB_whatModel = 'pid';
    IB_RSt = get(handles.prom_zasahohr_time, 'string');
    IB_RSo = get(handles.prom_zasahohr_output, 'string');
elseif get(handles.ui_bg_pickmodel.SelectedObject, 'tag') == rb_modelsp
```

```

    IB_whatModel = 'sp';
    IB_RSt = get(handles.prom_zasahohr_time, 'string');
    IB_RSo = get(handles.prom_zasahohr_output, 'string');
elseif get(handles.ui_bg_pickmodel.SelectedObject, 'tag') ==
rb_modelpidsp
    IB_whatModel = 'pidsp';
    IB_RSt = get(handles.prom_zasahohr_time, 'string');
    IB_RSo = get(handles.prom_zasahohr_output, 'string');
else
    disp('Error! Radio box choice error. ');
    IB_whatModel = 'base';
    IB_V3 = str2double(get(handles.prom_zasahohr_base, 'string'));
end

```

Tato část kódu obsluhuje přepínače v Tlačítkové skupině Volba modelu. Zároveň dosadí správnou variantu parametru akčního zásahu ventilátoru.

```

switch IB_whatModel
    case 'base'
        set_param('InterfaceBoxModel/Vykon cernadla', 'Value', 'IB_V1');
        set_param('InterfaceBoxModel/Vykon ventilatoru
c.2', 'Value', 'IB_V2');
        set_param('InterfaceBoxModel/U', 'Value', 'IB_V3');
        set_param('InterfaceBoxModel/frekvence1', 'Value', 'IB_F1');
        set_param('InterfaceBoxModel/frekvence', 'Value', 'IB_F2');
        set_param('InterfaceBoxModel/vykon ventilatoru
c.1', 'Value', 'IB_F3');
        sim('InterfaceBoxModel');
    case 'pid' %PID
        set_param('InterfaceBoxModel_PID_Inverze_Dynamiky/Vykon cerna-
dla1', 'Value', 'IB_V1');
        set_param('InterfaceBoxModel_PID_Inverze_Dynamiky/Vykon ventila-
toru c.1', 'Value', 'IB_V2');
        set_param('InterfaceBoxModel_PID_Inverze_Dynamiky/Repeating
Sequence3', 'rep_seq_t', 'IB_RSt', 'rep_seq_y', 'IB_RSo');
        set_param('InterfaceBoxModel_PID_Inverze_Dynamiky/frekvence3',
'Value', 'IB_F1');
        set_param('InterfaceBoxModel_PID_Inverze_Dynamiky/frekvence2',
'Value', 'IB_F2');
        set_param('InterfaceBoxModel_PID_Inverze_Dynamiky/vykon ventila-
toru c.2', 'Value', 'IB_F3');
        sim('InterfaceBoxModel_PID_Inverze_Dynamiky');
    case 'sp' %Pa se S.P.
        set_param('InterfaceBoxModel_PA_se_Smithovym_Prediktorem/Vykon
cernadla1', 'Value', 'IB_V1');
        set_param('InterfaceBoxModel_PA_se_Smithovym_Prediktorem/Vykon
ventilatoru c.1', 'Value', 'IB_V2');
        set_param('InterfaceBoxMo-
del_PA_se_Smithovym_Prediktorem/Repeating Sequence3',
'rep_seq_t', 'IB_RSt', 'rep_seq_y', 'IB_RSo');
        set_param('InterfaceBoxMo-
del_PA_se_Smithovym_Prediktorem/frekvence3', 'Value', 'IB_F1');
        set_param('InterfaceBoxMo-
del_PA_se_Smithovym_Prediktorem/frekvence2', 'Value', 'IB_F2');
        set_param('InterfaceBoxModel_PA_se_Smithovym_Prediktorem/vykon
ventilatoru c.2', 'Value', 'IB_F3');
        sim('InterfaceBoxModel_PA_se_Smithovym_Prediktorem');
    case 'pidsp' %PID se S.P.

```



```

        set_param('InterfaceBoxModel_PID_se_Smithuvym_Prediktorem/Vykon
cerpadla', 'Value', 'IB_V1');
        set_param('InterfaceBoxModel_PID_se_Smithuvym_Prediktorem/Vykon
ventilatoru c.1', 'Value', 'IB_V2');
        set_param('InterfaceBoxMo-
del_PID_se_Smithuvym_Prediktorem/Repeating Sequence3',
'rep_seq_t', 'IB_RSt', 'rep_seq_y', 'IB_RSo');
        set_param('InterfaceBoxMo-
del_PID_se_Smithuvym_Prediktorem/frekvence3', 'Value', 'IB_F1');
        set_param('InterfaceBoxMo-
del_PID_se_Smithuvym_Prediktorem/frekvence2', 'Value', 'IB_F2');
        set_param('InterfaceBoxModel_PID_se_Smithuvym_Prediktorem/vykon
ventilatoru c.2', 'Value', 'IB_F3');
        sim('InterfaceBoxModel_PID_se_Smithuvym_Prediktorem');
    otherwise
        disp('Error! Global variable IB_whatModel has been changed from
outside.');
```

```

        set_param('InterfaceBoxModel/Vykon cerpadla', 'Value', 'IB_V1');
        set_param('InterfaceBoxModel/Vykon ventilatoru
c.2', 'Value', 'IB_V2');
        set_param('InterfaceBoxModel/U', 'Value', 'IB_V3');
        set_param('InterfaceBoxModel/frekvence1', 'Value', 'IB_F1');
        set_param('InterfaceBoxModel/frekvence', 'Value', 'IB_F2');
        set_param('InterfaceBoxModel/vykon ventilatoru
c.1', 'Value', 'IB_F3');
        sim('InterfaceBoxModel');
end
```

Zde pomocí přepínače `switch` vložíme parametry do zvoleného modelu pomocí příkazu `set_param()` a spustíme model příkazem `sim()`.

Druhá funkce je obsluha Tlačítka Stop:

```

% --- Executes on button press in btn_stop.
function btn_stop_Callback(hObject, eventdata, handles)

switch IB_whatModel
    case 'base'
        set_param('InterfaceBoxModel', 'SimulationCommand', 'stop');
    case 'pid' %PID
        set_param('InterfaceBoxModel_PID_Inverze_Dynamiky', 'Simulation-
Command', 'stop');
    case 'sp' %Pa se S.P.
        set_param('InterfaceBoxModel_PA_se_Smithovym_Prediktorem', 'Simu-
lationCommand', 'stop');
    case 'pidsp' %PID se S.P.
        set_param('InterfaceBoxModel_PID_se_Smithuvym_Prediktorem', 'Si-
mulationCommand', 'stop');
    otherwise
        disp('Error! Global variable IB_whatModel has been changed from
outside.');
```

```

        set_param('InterfaceBoxModel', 'SimulationCommand', 'stop');
        set_param('InterfaceBoxModel_PID_Inverze_Dynamiky', 'Simulation-
Command', 'stop');
        set_param('InterfaceBoxModel_PA_se_Smithovym_Prediktorem', 'Simu-
lationCommand', 'stop');
        set_param('InterfaceBoxModel_PID_se_Smithuvym_Prediktorem', 'Si-
mulationCommand', 'stop');
end
```

Tato funkce vybere správný model a ukončí jeho chod, stejným principem jak byl spuštěn v předchozí sekci kódu.

Poslední funkce v souboru InterfaceBoxGUI_B.m je obsluha Tlačítka Ukončit:

```
% --- Executes on button press in btn_close.
function btn_close_Callback(hObject, eventdata, handles)
% hObject    handle to btn_close (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(InterfaceBoxGUI_B); % zavřu okno
```

Tato funkce je naprosto jednoduchá, pouze provede příkaz k zavření okna.

3.3.1 Soubor moje.m

Tato funkce obsluhuje ukládání Výstupních parametrů laboratorního tepelného modelu do poznámkového bloku:

```
function moje(A)
global IB_whereSave; %
global IB_whatModel; %

fid = fopen(sprintf(IB_whereSave), 'a+');

switch IB_whatModel
    case 'base'
        fprintf(fid, '%g %g %g %g %g\n', A);
        fclose(fid);
    case 'pid' %PID
        fprintf(fid, '%g %g %g %g %g %g\n', A);
        fclose(fid);
    case 'sp' %Pa se S.P.
        fprintf(fid, '%g %g %g %g %g %g %g %g %g %g\n', A);
        fclose(fid);
    case 'pidsp' %PID se S.P.
        fprintf(fid, '%g %g %g %g %g %g %g %g %g %g\n', A);
        fclose(fid);
    otherwise
        disp('Error! Global variable IB_whatModel has been changed from
outside. ');
        fprintf(fid, '%g %g %g %g %g\n', A);
        fclose(fid);
end
```

Funkce moje načte poznámkový blok a využitím přepínače `switch` do něj uloží data z laboratorního tepelného modelu ve správném formátu.

ZÁVĚR

Vytvořené grafické uživatelské rozhraní slouží jako snadná metoda ovládání a řízení laboratorního modelu tepelné soustavy s dopravním zpožděním, který byl navržen a postaven jako výukový nástroj na Fakultě aplikované informatiky. Toto Ovládací rozhraní umožňuje používání tepelné soustavy bez jakékoliv manipulace v prostředí Simulink.

V teoretické části jsem laboratorní model názorně popsal, včetně způsobu propojení s řídicím počítačem a uvedl, jak funguje, jak jej zprovoznit a správně s ním zacházet. Také jsem představil programovací prostředí MATLAB/Simulink od společností MathWorks, s.r.o., zmínil jsem se o historii, začátcích a zvláštностech tohoto programu. Informoval jsem o Simulinku který využil pan Krčmář k řízení laboratorního modelu přes jím zkonstruované propojení modelu a počítače, o nástroji GUIDE pomocí kterého jsem vytvořil Ovládací rozhraní, jenž navazuje na snahu o zjednodušení ovládání daného modelu.

Praktická část této práce byla věnována Ovládacímu rozhraní Laboratorního Tepelného Modelu, založeném na kontrole souborů Simulinku obsahující blok "Ovládání tepelného modelu" který pro změnu ovládá tepelnou soustavu. Tam můžeme najít popis tohoto programu a zhotovenou uživatelskou a programátorskou příručku.

Měl jsem za úkol zjednodušit práci s laboratorním tepelným modelem a to se mi doufám podařilo. Jsem si jist, že další práce se stejným úmyslem budou následovat, jelikož touha usnadnit si život v lidech byla a bude.

ZÁVĚR V ANGLIČTINĚ

The Created graphical user interface serves as easy method of operating and controlling laboratory model heat system with time delay, which was design and build as teaching aid at the Faculty of applied informatics. This Control interface allows usage of heat system without any manipulation in Simulink environment.

In theoretical part I've illustratively described laboratory model, including method by which it's linked up with control computer and stated, how it works, how to put it into operation and correctly handle it. Also I've introduced programming environment MATLAB/Simulink by MathWorks , Inc., I've mentioned history, begginings and extraordinariness of this program. I've informed about Simulink which Mr. Krčmář used to operate the laboratory model through the connection created by him between model and computer, about the GUIDE tool with which I've created the Control interface, which continues the effort to simplify operation of the model.

Practical part of this thesis was byla devoted to the Control interface of Laboratory Heat Model, based on controlling Simulink files containing block called „Ovládání tepelného modelu" which in turn controls the heat model. There we can find description of this program and user and programmer manual.

My task was to make work with the laboratory heat model easier and I hope i was succesfulo. I'm sure, that another thesis with same intent will follow, baccouse desire to make one's life easier was in people and always will be.

SEZNAM POUŽITÉ LITERATURY

- [1] DOSTÁLEK, Petr. *Výukový model tepelné soustavy s dopravním zpožděním*. Uživatelský manuál. Zlín: UTB ve Zlíně, 2006. 20 s.
- [2] MATLAB GUI. *MATLAB* [online]. MathWorks [cit. 2016-12-22]. Dostupné z: <https://www.mathworks.com/discovery/matlab-gui.html>
- [3] PERŮTKA, Karel. *MATLAB - Základy pro studenty automatizace a informačních technologií*, 1. vyd. Zlín: UTB ve Zlíně, 2005. 304 s. ISBN 80-7318-355-2.
- [4] ŠULC, Bohumil a Miluše VÍTEČKOVÁ. *Teorie a praxe návrhu regulačních obvodů*. Praha: Vydavatelství ČVUT, 2004. ISBN 80-01-03007-5.
- [5] ŠVEHLÍKOVÁ, Martina. Ověření matematického modelu tepelné soustavy. *Simulační modely vybraných řízených systémů*. Zlín, 2009. Diplomová práce. UTB ve Zlíně. Vedoucí práce Libor Pekař.
- [6] KRČMÁŘ, Jakub. *Návrh a realizace hardwarového rozhraní mezi modelem tepelné soustavy a multifunkční vstupně – výstupní kartou*. Zlín, 2012. Diplomová práce. UTB ve Zlíně. Vedoucí práce prof. Ing. Vladimír Bobál, CSc.
- [7] DOSTÁLEK, Petr, VAŠEK, Vladimír and DOLINAY, Jan. Design and Implementation of Portable Data Acquisition Unit in Process Control and Supervision Applications. *WSEAS Transactions on Systems and Control*, 3, No. 9, 2008, 779-788. Available from: <http://www.wseas.us/e-library/transactions/control/2008/28-336.pdf>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

T_1	Platinový teploměr umístěný za ohřivačem.
T_2	Platinový teploměr umístěný před ventilátorem.
T_3	Platinový teploměr umístěný za ventilátorem a před čerpadlem.
W	Watt, jednotka výkonu.
m	Metr, jednotka délky.
s	Sekunda, jednotka času.
°C	Stupeň Celsia, jednotka teploty.
T_{0model}	Perioda vzorkování bloku "Ovládání tepelného modelu".

SEZNAM OBRÁZKŮ

Obr. 1. Laboratorní tepelný model.....	10
Obr. 2. Blokové schéma tepelného modelu s dopravním zpožděním.....	11
Obr. 3. Popis komponentů laboratorního tepelného modelu.	12
Obr. 4. Ventil pro napouštění a vypouštění vody.	13
Obr. 5. Kontrola hladiny vody v expanzní nádobě.	14
Obr. 6. Uživatelské rozhraní MATLABu.	16
Obr. 7. Uživatelské rozhraní Simulinku.	17
Obr. 8. Create new GUI menu.	18
Obr. 9. GUIDE editor.	18
Obr. 10. Simulink blok „Ovládání tepelného modelu“.....	20
Obr. 11. Simulink model „InterfaceBoxModel“.....	21
Obr. 12. Simulink model „InterfaceBoxModel_PA_se_Smithovym_Prediktorem“.....	21
Obr. 13. Simulink model „InterfaceBoxModel_PID_Inverze_Dynamiky“.....	21
Obr. 14. Simulink model „InterfaceBoxModel_PID_se_Smithuvym_Prediktorem“.....	22
Obr. 15. Ovládací rozhraní Laboratorního Tepelného Modelu.	22

SEZNAM PŘÍLOH

P I Disk CD s bakalářskou prací a programy.