

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



BAKALÁŘSKÁ PRÁCE

Výukový model regulované soustavy se
systémy PLC a SCADA

Praha, 2009

Autor: Martin Kopal

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne _____

podpis

Abstrakt

Na českých technických vysokých školách se již několik let v oblasti řídicí techniky ve velké míře využívá produkt Simulink, umožňující simulaci a analýzu dynamických systémů. Tento program by jistě našel uplatnění i na elektrotechnických středních školách, jenže takzvaná "College" licence Matlabu platí pouze pro vysoké školy.

Cílem mé práce tedy je vytvořit jednoduché modely soustav s vizualizací umožňující simulaci, analýzu systému či možnost návrhu a ladění regulátorů. Tyto modely s řízením jsou řešeny programem běžícím na simulovaném PLC produktu Mosaic, který je v základní verzi k dispozici zdarma a je tedy jednou z alternativ pro střední školy.

Abstract

At the Czech technical universities the product Simulink enabling simulation and analysis of dynamic systems has been used there for several years in a large extent in the sphere of control engineering. This programme could surely find its exercise also on the electrotechnical secondary schools, but so called "College" Matlab licence is valid only for universities.

The aim of my work is to construct simple models with visualization enabling simulation and analysis of system or an opportunity for controllers design and tuning. This models with regulation have been solved with a programme running on a simulated PLC product Mosaic that is gratis in the basic model and therefore is one of the options for the secondary schools.

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Martin Kopal**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Výukový model regulované soustavy se systémy PLC a SCADA**

Pokyny pro vypracování:

1. Seznamte se s programovatelnými automaty Tecomat Foxtrot, s jejich vývojovým systémem Mosaic, s knihovnami uživatelských bloků pro regulaci, s vizualizačním systémem Reliance a se zásadami programování PLC podle standardu IEC EN 61 131-3.
2. Zvolte typického představitele řízené soustavy, vhodné pro realizaci názorného modelu, s obdobnými vlastnostmi, jako tepelná soustava v technice budov.
3. Analyzujte funkce zvolené soustavy a navrhnete její algoritmičtý popis, jako podklad pro vytvoření modelu.
4. Realizujte model soustavy programem systémů PLC Tecomat a SCADA Reliance 4, včetně ovládacího panelu.

Seznam odborné literatury:

Šmejkal, L.: PLC a automatizace 2. Sekvenční logické funkce a minimum o fuzzy logice. BEN, Praha 2005.
Martinásková, M. - Šmejkal, L.: Řízení programovatelnými automaty III. Softwarové vybavení, ČVUT, Praha 2003.
kol.: Programování PLC podle normy IEC 61 131-3, Teco, Kolín 2007
Cicvárek, M.: Knihovna funkčních bloků pro analýzu tvaru a predikci průběhu číslicových signálů v PLC (bakalářská práce), ČVUT FEL, Praha, 2008.
www.tecomat.cz

Vedoucí: Ing. Martin Hlinovský, Ph.D.

Platnost zadání: do konce zimního semestru 2009/10

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry



doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 27. 2. 2009

Obsah

Seznam obrázků	vii
1 Úvod	1
1.1 PLC Tecomat Foxtrot	1
1.1.1 Struktura systému	1
1.1.2 Výkon systému	2
1.1.3 Komunikace systému	2
1.2 Mosaic	3
1.2.1 Dodávka programu	3
1.2.2 Programování	3
1.2.3 Nástroje programu	4
1.3 SCADA/HMI Reliance	5
1.3.1 Funkce programu	5
1.3.2 Modulární struktura systému	5
1.3.3 Vývojové prostředí	6
1.4 Standard IEC EN-61131-3	6
1.4.1 Společné prvky	7
1.4.2 Programovací jazyky	8
2 Modely	9
2.1 Model soustavy 1. řádu	9
2.1.1 Matematický model soustavy	9
2.1.1.1 Nelineární model	10
2.1.1.2 Linearizace	11
2.1.2 Implementace	12
2.2 Model soustavy 2. řádu	13
2.2.1 Matematický model soustavy	13

2.2.1.1	Nelineární model	14
2.2.1.2	Linearizace	15
2.2.2	Implementace	16
3	Regulace	18
3.1	PID regulátor	18
3.1.1	P složka	19
3.1.2	I složka	19
3.1.3	D složka	19
3.1.4	Anti-Windup	20
3.2	Dvoustavový regulátor	20
3.3	Návrhy regulátorů	21
3.3.1	Regulátor pro soustavu 1. řádu	21
3.3.2	Regulátor pro soustavu 2. řádu	22
4	Vizualizace	23
4.1	Komunikace s PLC	23
4.2	Grafický návrh	24
4.3	Logický návrh	26
4.3.1	Použité komponenty	27
5	Závěr	28
A	Zdrojové kódy	I
A.0.1	Main	I
A.0.2	Soustava 1. řádu	IV
A.0.3	Soustava 2. řádu	VI
A.0.4	PID regulátor	VII
A.0.5	Dvoustavový regulátor	IX
B	Obsah příloženého CD	XI

Seznam obrázků

1.1	PLC Tecomat Foxtrot	2
1.2	Vývojové prostředí Mosaic	3
2.1	Model soustavy prvního řádu	10
2.2	Přechodová charakteristika linearizovaného systému	12
2.3	Model soustavy druhého řádu	14
2.4	Přechodová charakteristika linearizovaného systému druhého řádu	16
3.1	Princip dvoustavového regulátoru	20
3.2	Srovnání reakcí simulinkového modelu a modelu navrženém v prostředí Mosaic na referenci skoku při použití PI regulátoru	21
3.3	Srovnání reakcí simulinkového modelu a modelu navrženém v prostředí Mosaic na referenci skoku při použití PI regulátoru	22
4.1	Model prvního řádu s řízením ve vizualizačním prostředí	24
4.2	Model druhého řádu s řízením ve vizualizačním prostředí	25

Kapitola 1

Úvod

1.1 PLC Tecomat Foxtrot

PLC (Programmable Logic Controller), neboli programovatelný logický automat je elektronický systém určený k řízení průmyslových procesů. Periferie PLC automatu jsou přizpůsobeny pro napojení na technologické procesy a těmito PLC předává a získává informace do a z řízeného procesu.

Řídicí algoritmus je zapsán v paměti uživatelského programu jako posloupnost instrukcí, které jsou cyklicky vykonávány. Po vykonání všech instrukcí jednoho cyklu programu provede centrální jednotka aktualizaci výstupních proměnných a aktualizaci stavů ze vstupních modulů do paměti. Tento postup zabraňuje vzniku hazardních stavů při běhu programu.

1.1.1 Struktura systému

Tecomat Foxtrot je malý modulární systém firmy Teco a. s. navržený pro snadnou instalaci do rozvaděče s montáží na U-lištu. Moduly řídicího systému jsou propojeny sběrnici TCL2, která je tvořena linkou RS-485. Pomocí této sběrnice lze k centrálnímu modulu připojit až deset rozšiřujících I/O modulů.

Moduly dovolují připojení analogových či binárních signálů ve standardních rozsazích. Základní modul obsahuje 8 binárních vstupů a 4 analogové vstupy.

1.1.2 Výkon systému

Výpočetní výkon systému je zajištěn 32bitovým procesorem RISC s frekvencí 166 MHz. Paměť pro program je 192 kB pro program samotný a 64 kB pro tabulky. Paměť programu i tabulek je zálohována lithium-iontovým akumulátorem. Pro ukládání dat je možné využít přídatné paměti Databox. Poprvé u systému Tecomat je také možné PLC osadit kartou MMC/SD.

1.1.3 Komunikace systému

Centrální jednotka obsahuje dva sériové kanály a jedno rozhraní Ethernet. Jeden ze sériových kanálů má pevné rozhraní RS-232, druhý umožňuje volbu rozhraní osazením příslušného submodulu do základního modulu. Sériová rozhraní jsou určena pro připojení inteligentních snímačů nebo např. operátorských panelů.

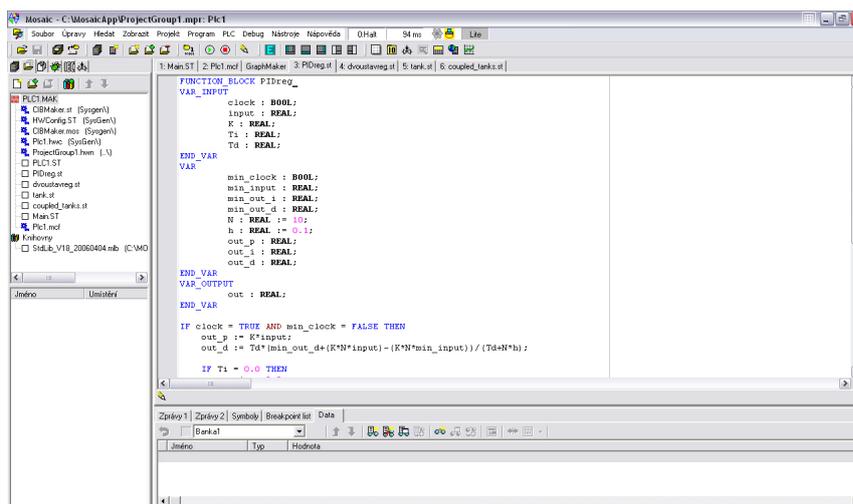
Tecomat Foxtrot podporuje komunikaci standardními protokoly, jako je Profibus-DP nebo CAN. Obsahuje integrovaný webový server který dává možnost vytvořit pro každý modul webovou stránku v jazyce XML, přes kterou lze proces sledovat i řídit z internetu.



Obrázek 1.1: PLC Tecomat Foxtrot

1.2 Mosaic

Mosaic je vývojové prostředí pro tvorbu a ladění programů pro programovatelné logické automaty od firmy Teco a. s. Kolín. Prostředí bylo vyvinuto tak, aby vyhovovalo normě IEC EN-61131-3, která definuje strukturu programu a programovací jazyky pro PLC.



Obrázek 1.2: Vývojové prostředí Mosaic

1.2.1 Dodávka programu

Instalace programu Mosaic obsahuje všechny nástroje, které jsou v daný čas k dispozici. Pokud při instalaci programu není přítomen HW klíč, funguje program Mosaic v tzv. Lite verzi. I ve verzi Lite jsou použitelné všechny nástroje a jejich funkce není nijak omezena.

WH klíč je potřeba, pokud chceme programovat větší typy PLC a je nutné deklarovat velké množství I/O modulů. K dispozici je česká, anglická, německá a ruská verze jazyka programovacího prostředí.

1.2.2 Programování

Mosaic dovoluje programovat všechny typy PLC dodávané firmou Teco.

Program psaný v jazyce podle normy IEC EN-61131-3 se skládá s částí nazývaných POU (Program Organisation Unit). Těmito programovými organizačními jednotkami jsou funkce, funkční bloky a samotný program.

Programovat je možné jak v textových, tak v grafických jazycích. Při použití grafického jazyka se jednotlivé bloky vybírají z nástrojové lišty a umisťují na plochu.

Při použití textových jazyků je možné využít podporu programování. Například v jazyce strukturovaného textu ST lze při psaní využít asistenta, který nabízí automatické doplnění rozepsaného příkazu. V programu je možné jazyky kombinovat, je tedy možné programovat každou POU v jiném jazyku. Mosaic také dovoluje vytvářet a využívat knihovní bloky.

1.2.3 Nástroje programu

Součástí programu Mosaic jsou i nástroje, pomocí kterých lze zjednodušit programování nebo analyzovat činnost programu.

- Inspektor POU

Umožňuje sledovat chování jednotlivých POU. U zobrazené POU jsou zobrazeny stavy vstupních a výstupních proměnných. U grafických editorů je barevně označena aktivní linie a zobrazeny hodnoty proměnných u vstupů a výstupů jednotlivých bloků.

- SimPLC

Umožňuje ladit programy bez nutnosti připojení reálného PLC. Tento nástroj dovoluje simulovat všechny vyráběné typy PLC Tecomat. Unikátní je možnost komunikace mezi prostředím Mosaic a vizualizačním systémem. Lze tedy ladit nejen program pro PLC, ale i vizualizaci a jejich vzájemnou interakci, přičemž oba programy mohou běžet na jednom počítači.

- GraphMaker

Nástroj se chová jako osciloskop se šestnácti kanály. Dovoluje sledovat průběhy všech typů proměnných v reálném čase. Průběhy je následně možné přímo analyzovat, uložit na disk, případně exportovat do jiných programů. Nástroj GraphMaker obsahuje vše potřebné k analýze průběhů jako jsou kurzory či lupa.

- PIDMaker

Nástroj je určen k snadné implementaci a ladění regulačních algoritmů. Nástroj automaticky generuje kód algoritmu který je zařazen do programu uživatele. Dokáže

také simulovat jednoduché procesy bez nutnosti jejich implementace v programu uživatele. V nabídce nástroje jsou lineární soustavy do třetího řádu s možností přidání dopravního zpoždění.

1.3 SCADA/HMI Reliance

Reliance je český vizualizační systém SCADA/HMI(Supervisory Control and Data Acquisition/ Human-Machine Interface) pro řízení a monitoring procesů v reálném čase.

1.3.1 Funkce programu

- Monitorování procesů v reálném čase
- Ovládání procesů v reálném čase
- Zobrazování grafů průběhů veličin
- Zobrazování a archivace alarmů
- Podpora multimédií
- Podpora vícemonitorových systémů

1.3.2 Modulární struktura systému

- Vývojové prostředí Reliance Design

Reliance Design je program pro tvorbu vizualizačních projektů. Existuje ve verzích Desktop a Enterprise, přičemž verze Enterprise dovoluje vytvářet síťové aplikace s libovolným množstvím ovládacích pracovišť.

- Run-time moduly Reliance Runtime a Server

Run-time modul je program, který zajišťuje běh vizualizace na počítači koncového uživatele.

- Webový klient Reliance J

Webový klient Reliance J je applet napsaný v jazyce java, který umožňuje spustit projekt ve webovém prohlížeči.

- Komunikační ovladače

Komunikační ovladače zajišťují přenos dat ze stanic do vizualizačního systému a přenos povelů opačným směrem.

1.3.3 Vývojové prostředí

Reliance poskytuje výkonné vývojové prostředí RAD (Rapid Application Development). Po založení projektu je nutné definovat stanice (PLC), které budou s vizualizací komunikovat a importovat proměnné které se budou využívat. Grafické rozhraní je tvořeno okny, do kterých se vkládají jednotlivé komponenty (např. displeje, tlačítka, obrázky). Každé komponentě lze přiřadit jiné vlastnosti.

Komponenty se dělí na statické a dynamické. Dynamické komponenty mohou být vázány na proměnnou a s její změnou mohou měnit své parametry. Statickou komponentou je např. obrázek. Dynamickou komponentou je např. displej.

Pro správu objektů slouží tzv. správci, kteří umožňují přehledné uspořádání objektů či hromadnou změnu vlastností.

Pro základní funkce není potřeba psát žádný kód, čímž se omezuje riziko vzniku chyb. Pokud je potřeba vytvořit složitější funkci, je možné vytvořit skript v jazyce VB Script.

1.4 Standard IEC EN-61131-3

Norma IEC EN-61131 pro programovatelné řídicí systémy se dělí na pět částí, které jsou věnovány jak technickému, tak programovému vybavení těchto systémů. Třetí část této normy se zabývá programovacími jazyky.

Norma IEC EN-61131-3 je výsledkem práce několika mezinárodních společností z oboru průmyslové automatizace a byla přijata jako směrnice u většiny významných výrobců PLC.

Dělí se na dvě základní části:

- Společné prvky
- Programovací jazyky

1.4.1 Společné prvky

Typy dat

Definování datových typů omezuje riziko vzniku chyb. Běžné datové typy jsou BOOL, INT, REAL, STRING atd.

Proměnné

Dělí se na lokální a globální. Oblast působnosti lokálních proměnných je omezena na POU kde byly definovány. Jejich jména poté mohou být použita v jiných částech programu. Pokud definujeme proměnnou jako globální, má platnost v celém projektu. Proměnným může být přiřazena počáteční hodnota při startu.

Konfigurace

V rámci konfigurace můžeme definovat jedno nebo více zařízení vykonávající IEC programy (Resource). Uvnitř zdroje můžeme definovat jednu nebo více úloh (Task).

Programové organizační jednotky

V rámci normy IEC EN-61131 jsou jako programové organizační jednotky nazývány společně funkce, funkční bloky a programy.

Funkce

Norma definuje standardní a uživatelem definované funkce. Funkce mohou být v programu používány opakovaně.

Funkční bloky

Obsahují algoritmy i data, mohou tedy uchovávat informaci z minulosti. Mají definované rozhraní a skryté vnitřní proměnné. Funkční bloky mohou být v programu používány opakovaně.

Programy

Program je sítí funkcí a funkčních bloků.

1.4.2 Programovací jazyky

V normě jsou definovány čtyři programovací jazyky.

LD - Ladder Diagram

Jazyk příčkového diagramu. Je založen na grafické reprezentaci reléové logiky. Síť v jazyku LD je zleva i zprava ohraničena svislými čarami, které se nazývají levá a pravá napájecí sběrnice. Mezi nimi je tzv. příčka, která může být rozvětvena. Do příček mohou být vloženy kontakty, funkce či funkční bloky.

FBD - Function Block Diagram

Jazyk funkčního blokového schématu. Vyjadřuje chování funkcí, funkčních bloků a programů jako sadu provázaných grafických bloků.

ST - Structured Text

Jazyk strukturovaného textu. Jedná se o vyšší programovací jazyk, který obsahuje prvky jako jsou smyčky či větvení.

IL - Instruction List

Jazyk seznamu instrukcí. POU je složena ze sekvence instrukcí.

Kapitola 2

Modely

Při tvorbě modelu soustavy tepelné techniky budov jsem vycházel z předpokladu, že tepelnou techniku budov lze modelovat také jako soustavu spojených nádob. V tepelné technice budov se můžeme často setkat s výraznými nelinearitami, které se vyskytují také u modelu spojených nádob. Práce by měla být využita jako výukový model, proto jsem se rozhodl vytvořit model s takovými parametry, aby časové konstanty byly mnohem menší než v tepelné technice budov.

V práci jsem se zabýval návrhem dvou typů soustav a to soustavy 1. řádu a soustavy 2. řádu.

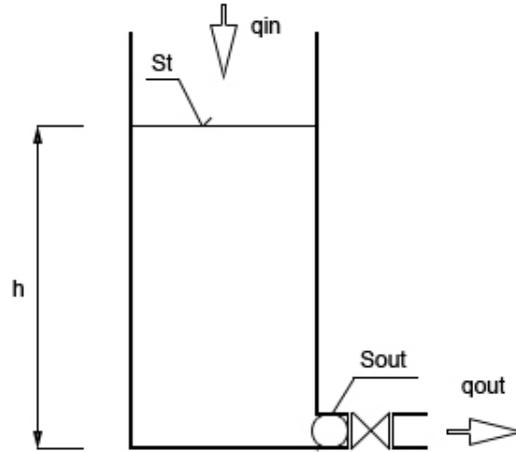
2.1 Model soustavy 1. řádu

Jako model soustavy prvního řádu jsem zvolil nádobu s přítokem shora s jedním uzavíratelným ventilem, kterým kapalina odtéká. Tento model by se dal v tepelné technice budov přirovnat např. k místnosti vytápěné přímotopem ve které dochází ke ztrátám tepla.

2.1.1 Matematický model soustavy

Parametry modelu:

- Plocha průřezu válce $S_t = 3 \cdot 10^{-3} \text{m}^2$
- Konstanta čerpadla $k_p = 1 \cdot 10^{-3}$
- Plocha průřezu výstupního potrubí $S_{out} = 25 \cdot 10^{-4} \text{m}^2$



Obrázek 2.1: Model soustavy prvního řádu

2.1.1.1 Nelineární model

Model soustavy jedné nádoby s přítokem shora a s jedním odtokovým potrubím, které je možné uzavírat lze popsat diferenciální rovnicí

$$\frac{dh}{dt} = \frac{1}{S_t} (q_{in} - q_{out}) \quad (2.1)$$

kde h [m] je výška hladiny ve válci, S_t [m²] průřez válce, q_{in} [m³s⁻¹] označuje přítok do válce a q_{out} [m³s⁻¹] odtok z válce. Tuto rovnici lze dále upravit na tvar

$$\frac{dh}{dt} = \frac{1}{S_t} (uk_p - S_{out}\sqrt{2gh}) \quad (2.2)$$

Tento popis soustavy předpokládá nulovou délku odtokového potrubí k ventilu, nádoba je umístěna na zemi a že použitou kapalinou je voda. Zanedbává také viskozitu kapaliny a různá tření.

Diferenciální rovnici

$$\frac{dh}{dt} = \frac{uk_p}{S_t} - \frac{S_{out}\sqrt{2g}}{S_t} \sqrt{h(t)}$$

můžeme zjednodušit pomocí konstant

$$k_0 = \frac{k_p}{S_t}$$

$$k_b = \frac{S_{out}\sqrt{2g}}{S_t}$$

kde k_0 je vstupní konstanta a k_b konstanta výstupní. Výsledný tvar diferenciální rovnice pro nelineární systém je tedy

$$\frac{dh}{dt} = k_0 u - k_b \sqrt{h(t)} \quad (2.3)$$

po dosazení číselných hodnot

$$\frac{dh}{dt} = 0.333 q_{in} - 0.369 \sqrt{h(t)} \quad (2.4)$$

Stavový popis modelu:

$$\begin{aligned} x &= h \\ u &= q_{in} \\ y &= h \end{aligned}$$

$$\begin{aligned} [\dot{x}] &= [k_0 u - k_b \sqrt{x}] \\ [y] &= [1] [x] + [0] [u] \end{aligned} \quad (2.5)$$

2.1.1.2 Linearizace

Model soustavy je nelineární, je tedy nutné jej pro získání přenosu linearizovat v pracovním bodě. Využil jsem stavový popis nelineární soustavy a upravil ho pomocí parciálních derivací na tvar

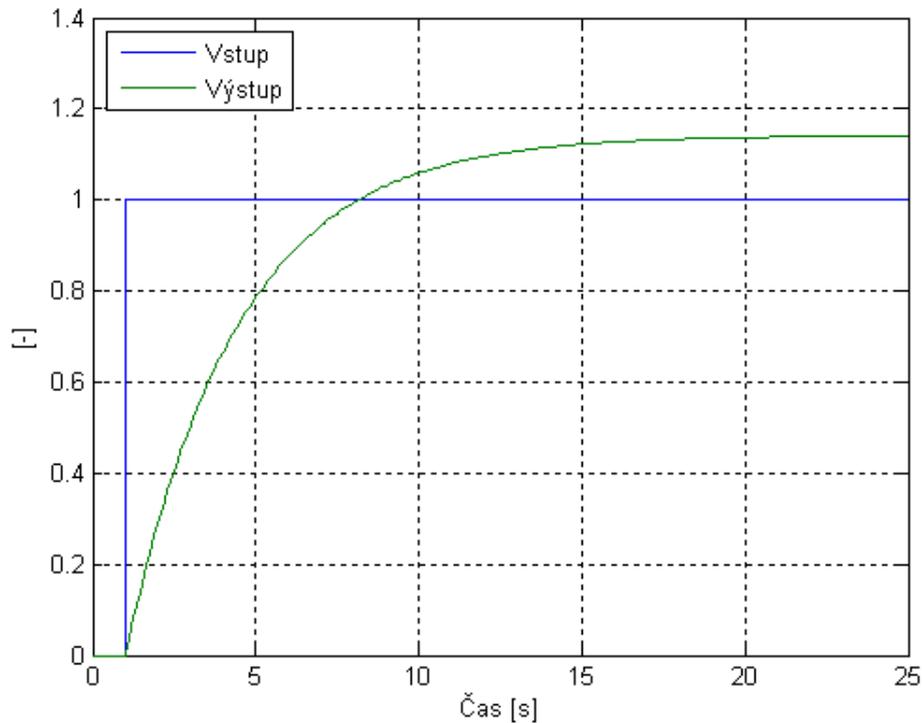
$$\begin{aligned} [\dot{x}] &= \left[-\frac{1}{2} \frac{k_b}{\sqrt{x}} \right] [\Delta x] + [k_0] [\Delta u] \\ [y] &= [1] [\Delta x] + [0] [\Delta u] \end{aligned}$$

Jako pracovní bod jsem zvolil rovnovážný stav, kdy $u = 0,7$ a $h = 0,4\text{m}$. Výsledný stavový popis systému v pracovním bodě po dosazení hodnot

$$\begin{aligned} [\dot{x}] &= [-0.2917]_{\substack{u=0.7 \\ x=0.4}} [\Delta x] + [0.333]_{\substack{u=0.7 \\ x=0.4}} [\Delta u] \\ [y] &= [1]_{\substack{u=0.7 \\ x=0.4}} [\Delta x] + [0]_{\substack{u=0.7 \\ x=0.4}} [\Delta u] \end{aligned} \quad (2.6)$$

Přenos systému v pracovním bodě

$$G(s) = \frac{0.333}{s + 0.2917} \quad (2.7)$$



Obrázek 2.2: Přechodová charakteristika linearizovaného systému

2.1.2 Implementace

Soustavu jsem se rozhodl implementovat v jazyce ST (Structured text) jako samostatný funkční blok. Původní myšlenkou bylo realizovat soustavu linearizovanou v pracovním bodě, tedy jako lineární systém daný přenosem. Toto řešení by ale nedovolovalo měnit za běhu programu některé parametry soustavy, jako např. míru otevření ventilů. Proto jsem se rozhodl realizovat soustavu nelineární. Toto řešení navíc umožňuje lépe vidět problematiku regulace nelineárních soustav a je podle mne jako učební pomůcka názornější. V okolí výše definovaného pracovního bodu platí pro přenos soustavy rovnice (2.7) a je tedy možné program využít při návrhu regulátorů.

Vstupy bloku jsou vstup do soustavy a hodnota koeficientu k_b . Výstup bloku odpovídá výstupu soustavy, tedy hladině h . Protože je soustava diskretní, je třeba jako vstup do bloku přidat ještě hodinové pulzy.

Jako metodu pro diskretizaci jsem zvolil metodu zpětné difference. Rovnice (2.4) se tedy změnila na

$$\frac{y(k) - y(k-1)}{h} = k_0 u - k_b \sqrt{y(k)}$$

Tato rovnice je ale pro $y(k)$ neřešitelná, protože výsledná rovnice je nekauzální. Roz-

hodl jsem se tedy nahradit hodnotu výstupu $y(k)$ pod odmocninou hodnotou výstupu v minulém kroku, tedy hodnotou $y(k-1)$, protože tato úprava změní chování soustavy pouze málo. Výsledná rovnice tedy odpovídá

$$\frac{y(k) - y(k-1)}{h} = k_0 u - k_b \sqrt{y(k-1)} \quad (2.8)$$

z ní

$$y(k) = y(k-1) + k_0 u h - k_b h \sqrt{y(k-1)} \quad (2.9)$$

Funkční blok realizuje výpočet výstupní hodnoty podle rovnice (2.9). Koeficient k_b je dán jedním ze vstupů bloku, stejně tak i vstup soustavy. Koeficient k_0 je pevně daný. Vzorkovací perioda h odpovídá periodě pulzů na vstupu bloku, tedy 0,1s. Výstupem bloku je výška hladiny v nádobě v metrech.

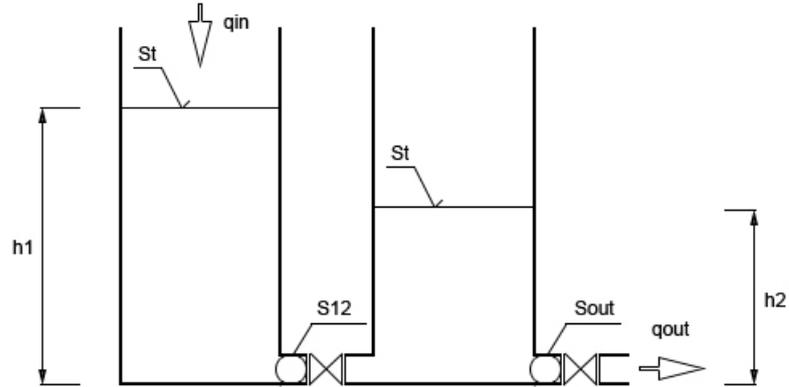
2.2 Model soustavy 2. řádu

Jako model soustavy druhého řádu jsem zvolil dvě nádoby propojené uzavíratelným ventilem. Tekutina přitéká shora do první nádrže, z druhé nádrže odtéká také přes uzavíratelný ventil. Tento model by se dal v tepelné technice budov přirovnat např. ke dvěma místnostem, kde je první místnost vytápěná přímotopem a dochází k přenosu tepla do druhé místnosti např. dveřmi. V druhé místnosti poté dochází k tepelným ztrátám.

2.2.1 Matematický model soustavy

Parametry modelu:

- Plocha průřezu válců $S_t = 3 \cdot 10^{-3} \text{m}^2$
- Konstanta čerpadla $k_p = 1 \cdot 10^{-3}$
- Plocha průřezu spojovacího potrubí $S_{12} = 25 \cdot 10^{-4} \text{m}^2$
- Plocha průřezu výstupního potrubí $S_{out} = 25 \cdot 10^{-4} \text{m}^2$



Obrázek 2.3: Model soustavy druhého řádu

2.2.1.1 Nelineární model

Model soustavy dvou spojených nádob s přítokem shora do první nádoby a s jedním odtokovým potrubím z nádoby druhé je popsán diferenciálními rovnicemi

$$\begin{aligned}\frac{dh_1}{dt} &= \frac{1}{S_t} (q_{in} - q_{12}) \\ \frac{dh_2}{dt} &= \frac{1}{S_t} (q_{12} - q_{out})\end{aligned}\quad (2.10)$$

kde h_1 [m] je výška hladiny v prvním válci, h_2 [m] je výška hladiny v druhém válci, S_t [m²] průřez válců, q_{in} [m³s⁻¹] označuje přítok do prvního válce, q_{12} [m³s⁻¹] průtok mezi válci a q_{out} [m³s⁻¹] odtok z druhého válce. Tuto rovnici lze dále upravit na tvar

Diferenciální rovnice

$$\begin{aligned}\frac{dh_1}{dt} &= \frac{uk_p}{S_t} - \frac{S_{12}\sqrt{2g}}{S_t} \sqrt{h_1 - h_2} \\ \frac{dh_2}{dt} &= \frac{S_{12}\sqrt{2g}}{S_t} \sqrt{h_1 - h_2} - \frac{S_{out}\sqrt{2g}}{S_t} \sqrt{h_2}\end{aligned}\quad (2.11)$$

můžeme zjednodušit pomocí konstant

$$\begin{aligned}k_0 &= \frac{k_p}{S_t} \\ k_b &= \frac{S_{out}\sqrt{2g}}{S_t} \\ k_{12} &= \frac{S_{12}\sqrt{2g}}{S_t}\end{aligned}$$

kde k_0 je vstupní konstanta, k_{12} konstanta průchodu mezi válci a k_b konstanta výstupní. Výsledný tvar diferenciálních rovnic pro nelineární systém je tedy

$$\begin{aligned}\frac{dh_1}{dt} &= uk_0 - k_{12}\sqrt{h_1 - h_2} \\ \frac{dh_2}{dt} &= k_{12}\sqrt{h_1 - h_2} - k_b\sqrt{h_2}\end{aligned}\quad (2.12)$$

po dosazení číselných hodnot

$$\begin{aligned}\frac{dh_1}{dt} &= 0.333k_0 - 0.369\sqrt{h_1 - h_2} \\ \frac{dh_2}{dt} &= 0.369\sqrt{h_1 - h_2} - 0.369\sqrt{h_2}\end{aligned}\quad (2.13)$$

Stavový popis modelu:

$$\begin{aligned}x_1 &= h_1 \\ x_2 &= h_2 \\ u &= q_{in} \\ y &= h_2\end{aligned}$$

$$\begin{aligned}\dot{x}_1 &= 0.333u - 0.369\sqrt{x_1 - x_2} \\ \dot{x}_2 &= 0.369\sqrt{x_1 - x_2} - 0.369\sqrt{x_2}\end{aligned}$$

$$\begin{aligned}\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} uk_0 - k_m\sqrt{x_1 - x_2} \\ k_m\sqrt{x_1 - x_2} - k_b\sqrt{x_2} \end{bmatrix} \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} [u]\end{aligned}\quad (2.14)$$

2.2.1.2 Linearizace

Využil jsem stavový popis nelineární soustavy a upravil ho pomocí parciálních derivací na tvar

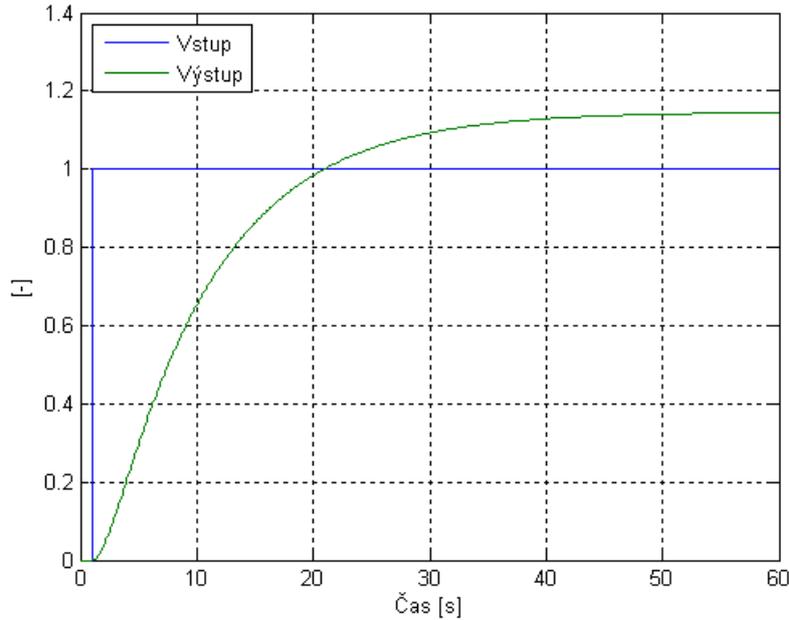
$$\begin{aligned}\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -\frac{1}{2} \frac{k_m}{\sqrt{x_1 - x_2}} & \frac{1}{2} \frac{k_m}{\sqrt{x_1 - x_2}} \\ \frac{1}{2} \frac{k_m}{\sqrt{x_1 - x_2}} & -\frac{1}{2} \frac{k_m}{\sqrt{x_1 - x_2}} - \frac{1}{2} \frac{k_b}{\sqrt{x_2}} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} k_0 \\ 0 \end{bmatrix} [\Delta u] \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} [\Delta u]\end{aligned}$$

Jako pracovní bod jsem zvolil rovnovážný stav, kdy $u = 0,7$, $h_1 = 0,8\text{m}$ a $h_2 = 0,4\text{m}$. Výsledný stavový popis systému v pracovním bodě po dosazení hodnot

$$\begin{aligned}\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -0.2917 & 0.2917 \\ 0.2917 & -0.5834 \end{bmatrix}_{\substack{u=0.7 \\ x_1=0.8 \\ x_2=0.4}} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} 0.333 \\ 0 \end{bmatrix}_{\substack{u=0.7 \\ x_1=0.8 \\ x_2=0.4}} [\Delta u] \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{\substack{u=0.7 \\ x_1=0.8 \\ x_2=0.4}} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}_{\substack{u=0.7 \\ x_1=0.8 \\ x_2=0.4}} [\Delta u]\end{aligned}\quad (2.15)$$

Přenos systému v pracovním bodě

$$G(s) = \frac{0.09722}{s^2 + 0.8741s + 0.0848} \quad (2.16)$$



Obrázek 2.4: Přechodová charakteristika linearizovaného systému druhého řádu

2.2.2 Implementace

Také tuto soustavu jsem se rozhodl implementovat jako nelineární ve formě samostatného funkčního bloku. V okolí výše definovaného pracovního bodu platí pro přenos soustavy rovnice (2.16).

Vstupy bloku jsou vstup do soustavy, hodnota koeficientu k_b a hodnota koeficientu k_{12} . Výstupy bloku odpovídají výstupům soustavy, tedy hladinám h_1 a h_2 . Protože je soustava diskrétní, je třeba jako vstup do bloku přidat ještě hodinové pulzy.

Jako metodu pro diskretizaci jsem zvolil metodu zpětné diference. Rovnice (2.12) se tedy změní na

$$\begin{aligned} \frac{y_1(k) - y_1(k-1)}{h} &= k_0 u - k_{12} \sqrt{y_2(k) - y_1(k)} \\ \frac{y_2(k) - y_2(k-1)}{h} &= k_{12} \sqrt{y_2(k) - y_1(k)} - k_b \sqrt{y_2(k)} \end{aligned}$$

Tyto rovnice jsou ale pro $y_1(k)$ a $y_2(k)$ neřešitelné. Rozhodl jsem se tedy nahradit hodnoty výstupů $y_1(k)$ a $y_2(k)$ pod odmocninami hodnotami výstupů v minulém kroku, tedy hodnotami $y_1(k-1)$ a $y_2(k-1)$. Výsledné rovnice tedy odpovídají

$$\begin{aligned}\frac{y_1(k)-y_1(k-1)}{h} &= k_0 u - k_{12} \sqrt{y_2(k-1) - y_1(k-1)} \\ \frac{y_2(k)-y_2(k-1)}{h} &= k_{12} \sqrt{y_2(k-1) - y_1(k-1)} - k_b \sqrt{y_2(k-1)}\end{aligned}\quad (2.17)$$

z nich

$$\begin{aligned}y_1(k) &= y_1(k-1) + k_0 u h - k_{12} \sqrt{y_2(k-1) - y_1(k-1)} \\ y_2(k) &= y_2(k-1) + k_{12} \sqrt{y_2(k-1) - y_1(k-1)} - k_b h \sqrt{y_2(k-1)}\end{aligned}\quad (2.18)$$

Funkční blok realizuje výpočet výstupních hodnot podle rovnic (2.18). Koeficient k_b je dán jedním ze vstupů bloku, stejně tak i koeficient k_{12} a vstup soustavy. Koeficient k_0 je pevně daný. vzorkovací perioda h odpovídá periodě pulzů na vstupu bloku, tedy 0,1s. Výstupy bloku odpovídají výškám hladin v nádobách v metrech.

Kapitola 3

Regulace

V programu jsem se rozhodl implementovat dva typy regulátorů a to typ PID a typ dvoustavový.

3.1 PID regulátor

Prosředí Mosaic obsahuje nástroj PIDMaker, který dovoluje automaticky implementovat regulační algoritmus, který lze poté zařadit do programu uživatele. Rozhodl jsem se však, že PID regulátor navrhnu jako samostatný funkční blok. Blok realizuje funkci diskretní verze PID regulátoru popsaného rovnicí

$$u(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (3.1)$$

K převodu do diskretního tvaru jsem využil zpětných diferencí, kterými jsem nahradil derivace. Vstupem bloku je vstup do regulátoru, tedy regulační odchylka, dále zesílení regulátoru K a časové konstanty T_i a T_d . Výstupem bloku je výstup regulátoru, tedy akční zásah.

PID regulátor se skládá ze tří složek a to ze složky proporciální (P), integrační (I) a složky derivační (D). Složka proporciální realizuje potřebné zesílení, složka integrační nulovou regulační odchylku na skok řídicí veličiny a složka derivační omezuje velikost překmitu způsobeném složkou integrační.

3.1.1 P složka

Diskrétní realizace zesílení je poměrně snadná, jde o pouhé vynásobení současné hodnoty vstupu konstanou K (zesílení).

$$P(t_k) = Ke(t_k) \quad (3.2)$$

3.1.2 I složka

Výraz

$$\frac{K}{T_I} \int_0^t e(\tau) d\tau$$

jsme upravil na tvar

$$\frac{dI}{dt} = \frac{K}{T_I} e$$

Derivaci ve výrazu jsem nahradil zpětnou diferencí

$$\frac{I(t_k) - I(t_{k-1})}{h} = \frac{K}{T_I} e(t_k) \quad (3.3)$$

Z výrazu po úpravě dostaneme hodnotu integrační složky v čase t_k

$$I(t_k) = \frac{Kh}{T_I} e(t_k) + I(t_{k-1}) \quad (3.4)$$

3.1.3 D složka

Derivační složku PID regulátoru jsem navrhl s filtrací, protože ideální derivace příliš zesiluje vysokofrekvenční šumy.

Přenos derivační složky s filtrací je dán vztahem

$$\begin{aligned} \frac{Y(s)}{U(s)} &= \frac{sKT_d}{1 + \frac{sT_d}{N}} \\ Y(s)(1 + s\frac{T_d}{N}) &= sT_dU(s) \end{aligned} \quad (3.5)$$

Vztah jsem pomocí zpětné difference upravil na

$$y(t_k) + \frac{T_d}{N} [y(t_k) - y(t_{k-1})] = KT_d [u(t_k) - u(t_{k-1})] \quad (3.6)$$

Po úpravách a přejmenování dostaneme výsledný vztah pro hodnotu derivační složky v čase t_k

$$D(t_k) = \frac{[D(t_{k-1}) + (KNe(t_k)) - (KNe(t_{k-1}))]}{T_d + Nh} \quad (3.7)$$

3.1.4 Anti-Windup

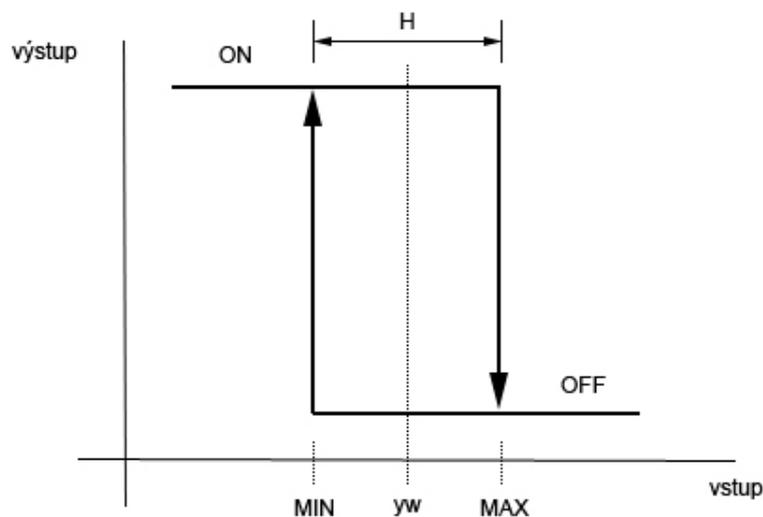
Každý akční člen má omezený rozsah, např. čerpadlo v mém programu má rozsah 0 až 1. Když dojde k saturaci, řídicí signál neroste/neklesá, smyčka je v podstatě otevřená. Výstup integračního členu regulátoru stále zvyšuje svou hodnotu, ale zbytečně. Když se změni znaménko regulační odchylky, začne výstup klesat, ale dlouho trvá než se dostane pod úroveň saturace.

Řešením je Anti-Windup, který po dosažení saturace nastaví integrální akci rovnou nule. Tuto funkci jsem vyřešil pomocí IF-ELSE funkcí.

3.2 Dvoustavový regulátor

Dvoustavový regulátor jsem navrhl jako blok se třemi vstupy a jedním výstupem. Dva ze vstupů dovolují zadat požadovanou hodnotu výstupu a velikost hysterze, třetí vstup slouží jako vstup do regulátoru. Výstupem bloku je akční zásah regulátoru, tedy 0 nebo 1.

Funkci regulátoru jsem implmentoval pomocí if-else funkcí.



Obrázek 3.1: Princip dvoustavového regulátoru

3.3 Návrhy regulátorů

Soustavy jsou v pracovním bodě a jeho okolí popsány přenosem, lze tedy pro tento bod navrhnout regulátory, které dokážou řídit soustavu.

3.3.1 Regulátor pro soustavu 1. řádu

Rozhodl jsem se pro regulátor typu PI (proporciálně-integrační). Tento regulátor zajistí dostatečnou rychlost reakce a nulovou regulační odchylku na skok řídicí veličiny. Regulátor jsem navrhl tak, aby překmit regulované veličiny nepřesáhl 10%. Návrh jsem provedl v MATLABu pomocí metody geometrického místa kořenů (GMK).

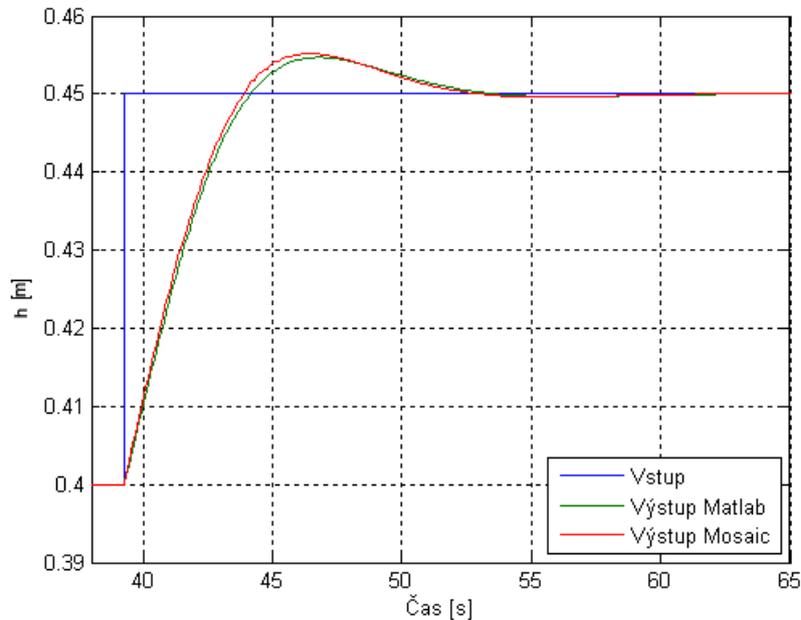
Přenos regulátoru

$$C(s) = 0.8 + \frac{0.571}{s} \quad (3.8)$$

z něj

$$K = 0.8 \quad (3.9)$$

$$T_I = 1.4$$



Obrázek 3.2: Srovnání reakcí simulinkového modelu a modelu navrženém v prostředí Mosaic na referenci skoku při použití PI regulátoru

Z obr. (3.2) je vidět, že se výstup regulované soustavy simulované v MATLABu od výstupu soustavy simulované v mém programu příliš neliší. Odchylka může být způsobena drobnou odlišností návrhu soustav.

3.3.2 Regulátor pro soustavu 2. řádu

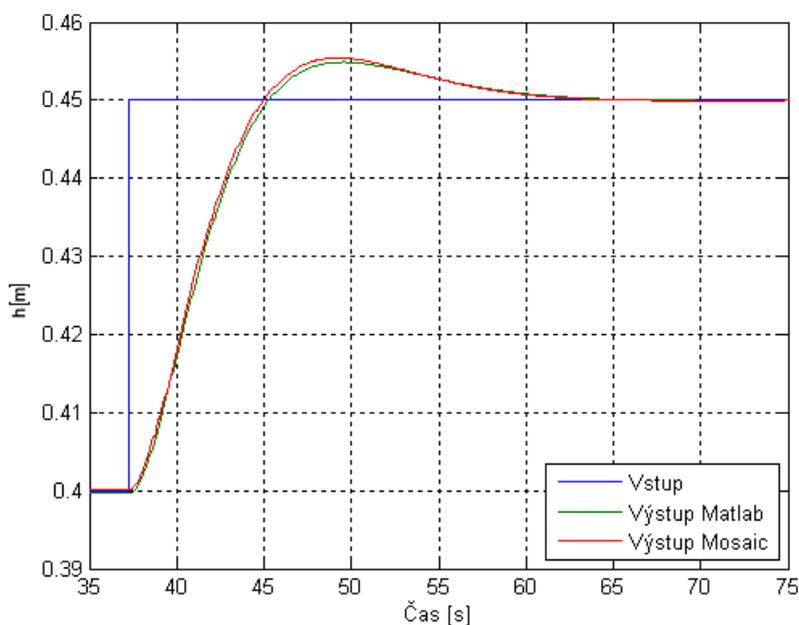
Také u soustavy druhého řádu jsem se rozhodl pro regulátor typu PI (proporciálně-integrační). Regulátor jsem opět navrhl tak, aby překmit regulované veličiny nepřesáhl 10%. Návrh jsem provedl v MATLABu pomocí metody geometrického místa kořenů (GMK).

Přenos regulátoru

$$C(s) = 1.8 + \frac{0.312}{s} \quad (3.10)$$

z něj

$$\begin{aligned} K &= 1.8 \\ T_I &= 5.9 \end{aligned} \quad (3.11)$$



Obrázek 3.3: Srovnání reakcí simulinkového modelu a modelu navrženém v prostředí Mosaic na referenci skoku při použití PI regulátoru

Kapitola 4

Vizualizace

Vizualizaci soustav a jejich řízení jsem navrhl v prostředí SCADA/HMI Reliance. Tento vizualizační systém je k dispozici ve verzi Lite zdarma s omezením maximálně pětadvaceti datových bodů. Tímto počtem jsem tedy byl ve své práci limitován.

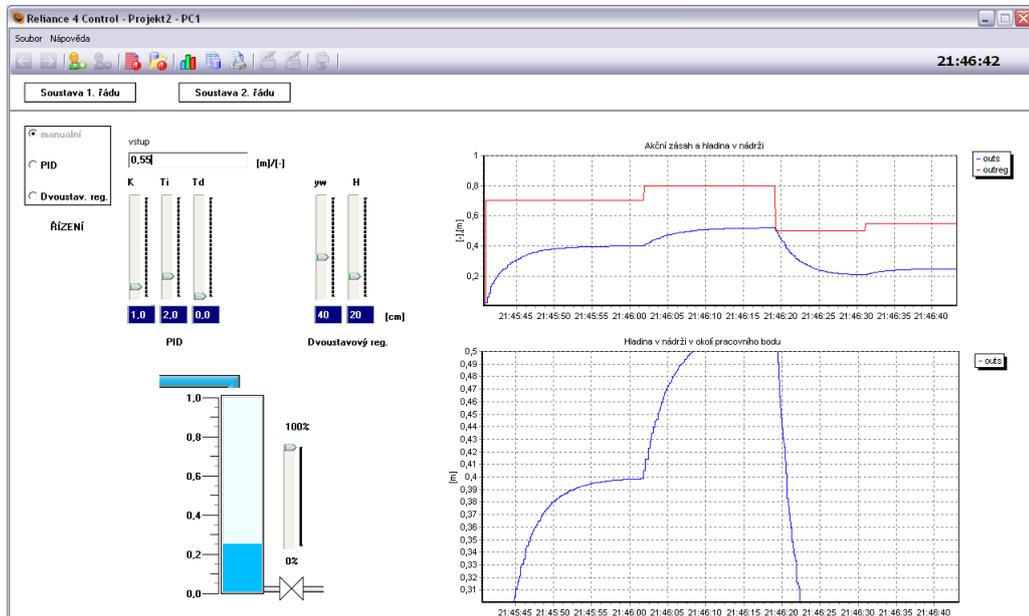
4.1 Komunikace s PLC

Každá stanice (PLC) má svoji unikátní IP adresu pomocí které je v síti jednoznačně identifikovatelná. V nastavení projektu vizualizace je třeba určit cílovou stanici, tedy zadat její IP adresu. Vývojové prostředí Mosaic umožňuje pracovat s tzv. simulovaným PLC, kdy k počítači není připojeno fyzické PLC, ale jeho chod je v PC simulován. Volbu simulovaného PLC jsem se rozhodl ve své práci využít.

Volba Mosaic PLC v nastavení projektu prostředí Mosaic povoluje možnost připojení vizualizace přímo k simulátoru v Mosaicu přes Ethernet. Je-li vizualizace spuštěna ve stejném počítači, pak adresa simulátoru je na interní adrese IP 127.0.0.1.

Po spuštění vizualizace by mělo dojít k automatickému spojení se stanicí. Pokud se spojení z nějakého důvodu nezdaří, zobrazí se varovné hlášení. V mé práci vyjímečně dojde k situaci kdy vizualizační program hlásí po spuštění chybu spojení ačkoli je simulované PLC připojeno a program v něm běží. Po restartování komunikace PLC se však spojení naváže. Bohužel se mi nepodařilo nalézt a vyřešit příčinu tohoto problému.

4.2 Grafický návrh



Obrázek 4.1: Model prvního řádu s řízením ve vizualizačním prostředí

Modely soustav jsou vytvořeny v jednom vizualizačním projektu. Vizualizace soustav jsou vyřešeny v samostatných oknech mezi kterými lze přepínat.

Výběr typu soustavy

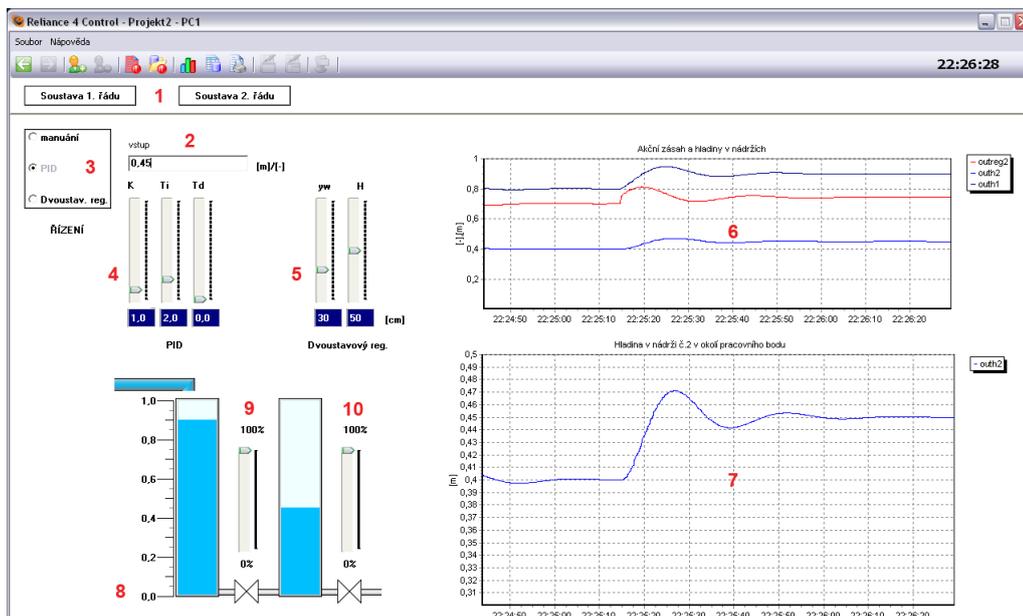
Výběr typu soustavy lze provést pomocí tlačítek (1) v obr. (4.2). Lze zvolit mezi soustavou prvního nebo soustavou druhého řádu.

Zadání vstupu

V případě manuálního režimu řízení je pro zadání hodnoty vstupu do soustavy určeno textové pole (2) v obr. (4.2). V případě, že je soustava řízena PID regulátorem, slouží toto pole pro zadání požadované hodnoty výstupu soustavy.

Volba typu řízení

Soustavu lze řídit třemi typy řízení a to manuálně, tedy bez regulátoru, pomocí PID regulátoru či pomocí dvoustavového regulátoru. Výběr typu řízení lze provést výběrem příslušného rádiového tlačítka (3) v obr. (4.2).



Obrázek 4.2: Model druhého řádu s řízením ve vizualizačním prostředí

Nastavení PID regulátoru

Pomocí tří posuvníků (4) v obr. (4.2) lze nastavit zesílení, hodnotu integrační časové konstanty a hodnotu derivační časové konstanty PID regulátoru. Zesílení lze nastavit v rozmezí 0 až 10, časové konstanty v rozmezí 0 až 10s s krokem 0,1s.

Nastavení dvoustavového regulátoru

Pomocí dvou posuvníků (5) v obr. (4.2) lze nastavit požadovanou výšku hladiny a velikost hysterze.

Graf průběhu hladin v nádržích a akční veličiny

Graf (6) v obr. (4.2) zobrazuje velikost akčního zásahu a výšky hladin v nádržích v rozsazích 0 až 1 a 0 až 100cm. Tento graf kvůli své velikosti nepodává detailní informace, je určen zejména jako orientační.

Graf průběhu výšky regulované hladiny

Graf (7) v obr. (4.2) zobrazuje výšku regulované hladiny v nádrži v rozsahu 0,3 až 0,5m. Tento graf je určen pro vizualizaci řízení hladiny v okolí pracovního bodu 0,4m. Zejména se uplatní při návrhu PID regulátorů pro tento pracovní bod.

Animace soustavy

Vizualizace (8) v obr. (4.2) znázorňuje animací aktuální výšky hladin v nádržích.

Ventil mezi nádržemi

Posuvník (9) v obr. (4.2) umožňuje uzavírat ventil spojující nádrže.

Výstupní ventil

Posuvník (10) v obr. (4.2) umožňuje uzavírat výstupní odtokový ventil z druhé nádrže.

4.3 Logický návrh

Některým komponentám vizualizace je nutné přiřadit proměnné programu běžícího v PCL. Program Mosaic dovoluje označit zvolené proměnné při deklaraci klíčovým slovem PUBLIC. Při překladu projektu se názvy a adresy těchto proměnných uloží do souboru typu *.bak*. Tento soubor je poté možné v nastavení projektu Reliance otevřít a s načtenými proměnnými okamžitě pracovat. Pokud ovšem dojde k úpravě programu, která má vliv na adresy proměnných, je nutné soubor *.bak* v nastavení opětovně načíst. Bez tohoto kroku by komponenty ve vizualizaci odkazovaly na jiná místa registru než se nachází cílové proměnné, což má většinou fatální následky.

U každé proměnné lze nastavit interval aktualizace hodnoty, případně její různé korekce.

Neregistrovaný vizualizační systém Reliance, který jsem použil pro tvorbu vizualizace, omezuje počet takto načtených proměnných na 25. Abych nepřesáhl tento limit, byl jsem nucen sloučit některé ovládací prvky obou typů soustav. Těmito prvky jsou nastavení typu regulace a nastavení dvoustavového regulátoru. Pokud se tedy například v okně soustavy prvního řádu změní způsob regulace na manuální, dojde ke změně způsobu regulace na manuální i v okně soustavy druhého řádu. Podobně u nastavení mezi dvoustavového regulátoru dojde při nastavení u jedné soustavy k nastavení stejných hodnot i u soustavy druhé. Toto omezení je sice nepříjemné, ale nemá žádný vliv na funkci soustav či regulace.

4.3.1 Použité komponenty

K vizualizaci průběhů akční veličiny a výšky hladin v nádržích jsem použil komponenty typu Plovoucí graf. Tyto komponenty umožňují snadnou realizaci a správu grafů. Nevýhodou grafů je, že zobrazují aktuální hodnotu v reálném čase, průběh nelze zastavit či prohlížet.

K animaci vody v nádržích jsem použil komponenty typu Indikátor průběhu svázané s proměnnými výstupů soustav v projektu programu Mosaic.

K nastavování hodnot jsem použil komponenty typu Posuvník. Nevýhodou těchto komponent je minimální krok 1. Velikost posuvníku jsem proto při zadávání desetinných čísel musel desetkrát zvětšit a hodnotu proměnné následně v programu deseti dělit.

Kapitola 5

Závěr

V práci jsem navrhl dva typy simulovaných soustav, které odpovídají soustavám tepelné techniky budov. Soustavy jsem pro získání přenosu linearizoval v pracovních bodech a pro dané přenosy navrhl proporciálně-integrační regulátory. V jazyce strukturovaného textu jsem v prostředí Mosaic naprogramoval modely těchto soustav, diskrétní verzi PID regulátoru a dvoustavový regulátor. Nedílnou součástí práce je vizualizace, která umožňuje analýzu soustav, návrh a testování PID regulátorů na soustavách, návrh a testování dvoustavového regulátoru či možnost vyzkoušet ruční řízení soustav.

Myslím, že tento výukový model splňuje požadavky na jednoduchost ovládání a zároveň nabízí velký prostor pro experimenty s řízením soustav a mohl by tedy být k výuce na středních školách reálně využit.

Literatura

FRANKLIN, Gene F., POWELL, J. David, EMAMI-NAEINI, Abbas. *Feedback Control of Dynamic Systems : Fifth Edition*. [s.l.] : [s.n.], c2006. 910 s. ISBN 0-13-149930-0.

ASTÖM, K., HÄGGLUNT, T. *PID Controllers: Theory, Design, and Tuning*. [s.l.] : [s.n.], c1995. 343 s. 2.. ISBN 978-1-55617-516-9.

HORÁČEK, Petr. *Systémy a modely*. Praha : Vydavatelství ČVUT, 2001. 232 s.

CICVÁREK, Martin. *Knihovna funkčních bloků pro analýzu tvaru a predikci průběhu číslicových signálů v PLC*. [s.l.], 2008. 41 s. ČVUT v Praze, Fakulta elektrotechnická. Bakalářská práce.

Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic. 10. vyd. [s.l.] : [s.n.], 2007. 101 s. Dostupný z WWW: www.tecomat.cz.

Realiance 3 Design. [s.l.] : [s.n.], [200-]. 254 s. Dostupný z WWW: www.tecomat.cz.

Příloha A

Zdrojové kódy

A.0.1 Main

```
VAR_GLOBAL
  // nadrže
  nadrž : tank;
  nadrže : coupled_tanks;

  // PID regulatory
  pid : PIDreg;
  pid2 : PIDreg;

  // dvoustavove regulatory
  dvstav : dvoustavreg;
  dvstav2 : dvoustavreg;

  // takt hodin 0.1s
  pulz AT %S13.0 : BOOL;

  // definice PUBLIC promennych
  // soustava 1.rad
  ins {PUBLIC} : REAL :=0;
  outs {PUBLIC} : REAL :=0;

  //PID1
  kpid {PUBLIC} : REAL := 10.0;
```

```
kpidvis {PUBLIC} : REAL;
tipid {PUBLIC} : REAL :=20.0;
tipidvis {PUBLIC} : REAL;
tdpid {PUBLIC} : REAL :=0;
tdpidvis {PUBLIC} : REAL;
inpid : REAL := 0;
outreg {PUBLIC} : REAL;
kv1 {PUBLIC} : REAL := 369;

//dvoustav. reg.
yw {PUBLIC} : REAL := 40;
hyst {PUBLIC} : REAL := 20;

// soustava 2.rad
intanks {PUBLIC} : REAL := 0;
outh1 {PUBLIC} : REAL;
outh2 {PUBLIC} : REAL;

//PID2
kpid2 {PUBLIC} : REAL := 10.0;
kpid2vis {PUBLIC} : REAL;
tipid2 {PUBLIC} : REAL := 20.0;
tipid2vis {PUBLIC} : REAL;
tdpid2 {PUBLIC} : REAL :=0;
tdpid2vis {PUBLIC} : REAL;
inpid2 : REAL :=0;
outreg2 {PUBLIC} : REAL :=0;
kv2 {PUBLIC} : REAL := 369;
km {PUBLIC} : REAL := 369;

// rizeni soustav: 0-man, 1-PID, 2-dvoustav. reg.
rizeni {PUBLIC} : INT := 1;
END_VAR

PROGRAM Main
VAR_INPUT
END_VAR
```

```
VAR
END_VAR
VAR_OUTPUT
END_VAR
VAR_TEMP
END_VAR

// 1 valec
// PID rizeni
IF rizeni = 1 THEN
    inpid := ins-outs;
    pid(clock := pulz, input := inpid, K := kpid/10.0, Ti := tipid/10.0,
        Td := tdpid/10.0, out =>outreg);
END_IF;

// dvoustavove rizeni
IF rizeni = 2 THEN
    dvstav(clock := pulz, input := outs, yw := yw/100.0, hyst := hyst/100.0,
        out => outreg);
END_IF;

// manualni ovladani
IF rizeni = 0 THEN
    outreg := ins;
END_IF;

// soustava
nadrz(vstup := outreg, kv:=(kv1/1000.0), clock := pulz, vystup => outs);

kpidvis := kpid/10.0;
tipidvis := tipid/10.0;
tdpidvis := tdpid/10.0;

// 2 valce
// PID rizeni
IF rizeni = 1 THEN
```

```

    inpid2 := intanks-outh2;
    pid2(clock := pulz, input := inpid2, K := kpid2/10.0, Ti := tipid2/10.0,
        Td := tdpid2/10.0, out =>outreg2);
END_IF;

// dvoustavove rizeni
IF rizeni = 2 THEN
    dvstav2(clock := pulz, input := outh2, yw := yw/100.0, hyst := hyst/100.0,
        out => outreg2);
END_IF;

// manualni ovladani
IF rizeni = 0 THEN
    outreg2 := intanks;
END_IF;

nadrze(vstup := outreg2, clock := pulz, kv:= (kv2/1000.0), km:=(km/1000.0),
    vystup_h1 => outh1, vystup_h2 => outh2);

kpid2vis := kpid2/10.0;
tipid2vis := tipid2/10.0;
tdpid2vis := tdpid2/10.0;
END_PROGRAM

```

A.0.2 Soustava 1. řádu

```

FUNCTION_BLOCK tank
VAR_INPUT
    // vstup do soustavy
    vstup : REAL;
    // vystupni ventil
    kv : REAL;
    // hodinove pulzy
    clock : BOOL;
END_VAR
VAR
    // hodnota minuleho stavu hladiny

```

```
    minul_vystup : REAL;
    min_clock : BOOL;
    vystup_test : REAL;
END_VAR
VAR_OUTPUT
    // vystup soustavy, hladina h
    vystup : REAL;
END_VAR

// provadeni tela jednou za 0,1s
IF min_clock=FALSE AND clock=TRUE THEN
    IF vstup <0.0 THEN
        vstup := 0.0;
    END_IF;

    // vypocet vystupu soustavy
    vystup_test:=(minul_vystup + 0.333*0.1*vstup - (kv*sqrt(minul_vystup)*0.1));

    // omezeni vysky hladiny
    IF vystup_test < 0.0 THEN
        vystup := 0.0;
    ELSE
        IF vystup_test > 1.0 THEN
            vystup := 1.0;
        ELSE
            // vystup
            vystup := vystup_test;
        END_IF;
    END_IF;

    // ulozeni hodnoty vystupu
    minul_vystup := vystup;
END_IF;
min_clock := clock;
END_FUNCTION_BLOCK
```

A.0.3 Soustava 2. řádu

```
FUNCTION_BLOCK coupled_tanks
VAR_INPUT
    //vstup do soustavy
    vstup : REAL;
    // vystupni ventil
    kv : REAL;
    // propojovaci vantil
    km : REAL;
    // hodinova pulzy
    clock : BOOL;
END_VAR
VAR
    //hodnota minulych stavu hladiny h1
    minuly_vystup_h1 : REAL;
    //hodnota minulych stavu hladiny h2
    minuly_vystup_h2 : REAL;
    min_clock : BOOL;
    vystup_h1_test : REAL;
    vystup_h2_test : REAL;
END_VAR
VAR_OUTPUT
    // hladina h1
    vystup_h1 : REAL;
    // hladina h2
    vystup_h2 : REAL;
END_VAR

// provedeni tela jednou za 0,1s
IF min_clock=FALSE AND clock=TRUE THEN
    IF vstup<0.0 THEN
        vstup := 0.0;
    END_IF;

    // vypocet vystupu
    vystup_h1_test:=(minuly_vystup_h1 + 0.333*0.1*vstup -
```

```
(km*sqrt(minuly_vystup_h1-minuly_vystup_h2)*0.1));
vystup_h2_test:=(minuly_vystup_h2 +
(km*sqrt(minuly_vystup_h1-minuly_vystup_h2)*0.1) -
(kv*sqrt(minuly_vystup_h2)*0.1));

// omezeni vysek hladin
IF vystup_h1_test < 0.0 THEN
    vystup_h1 := 0.0;
ELSE
    IF vystup_h1 > 1.0 THEN
        vystup_h1 := 1.0;
    ELSE
        // vystup h1
        vystup_h1 := vystup_h1_test;
    END_IF;
END_IF;

IF vystup_h2_test < 0.0 THEN
    vystup_h2 := 0.0;
ELSE
    IF vystup_h2 > 1.0 THEN
        vystup_h2 := 1.0;
    ELSE
        // vystup h2
        vystup_h2 := vystup_h2_test;
    END_IF;
END_IF;

// ulozeni hodnot vystupu
minuly_vystup_h1 := vystup_h1;
minuly_vystup_h2 := vystup_h2;
END_IF;
min_clock:=clock;
END_FUNCTION_BLOCK
```

A.0.4 PID regulátor

```
FUNCTION_BLOCK PIDreg
```

```
VAR_INPUT
```

```
    clock : BOOL;  
    // vstup  
    input : REAL;  
    // zesileni  
    K : REAL;  
    // integracni casova konstanta  
    Ti : REAL;  
    //derivacni casova konstanta  
    Td : REAL;
```

```
END_VAR
```

```
VAR
```

```
    // minule stavy  
    min_clock : BOOL;  
    min_input : REAL;  
    min_out_i : REAL;  
    min_out_d : REAL;  
    // filtrace  
    N : REAL := 10;  
    // perioda vzorkovani  
    h : REAL := 0.1;  
    // vystupy jednotlivych slozek  
    out_p : REAL;  
    out_i : REAL;  
    out_d : REAL;
```

```
END_VAR
```

```
VAR_OUTPUT
```

```
    // vystup regulatoru  
    out : REAL;
```

```
END_VAR
```

```
// provadeni tela jednou za 0,1s
```

```
IF clock = TRUE AND min_clock = FALSE THEN
```

```
    // vypocet zesileni
```

```
    out_p := K*input;
```

```
    // vypocet derivacni slozky
```

```
    out_d := Td*(min_out_d+(K*N*input)-(K*N*min_input))/(Td+N*h);
```

```
// anti-windup
IF Ti = 0.0 THEN
    out_i := 0.0;
ELSE
    IF out <= 0.0 OR out >=1.0 THEN
        out_i := min_out_i;
    ELSE
        // vypocet integracni slozky
        out_i := min_out_i+((K*h/Ti)*input);
    END_IF;
END_IF;

// omezeni vystupu regulatoru
IF (out_p+out_i+out_d > 1.0) THEN
    out := 1.0;
ELSE
    // vystup regulatoru
    out := out_p+out_i+out_d;
END_IF;

// ulozeni hodnot
min_out_i := out_i;
min_out_d := out_d;
min_input := input;
END_IF;
min_clock := clock;
END_FUNCTION_BLOCK
```

A.0.5 Dvoustavový regulátor

```
FUNCTION_BLOCK dvoustavreg
VAR_INPUT
    clock : BOOL;
    // vstup
    input : REAL;
    // pozadovana hodnota
    yw : REAL;
```

```
        // hysterze
        hyst : REAL;
END_VAR
VAR
    // rust/klesani hladiny
    up : REAL := 0.0;
    min_clock : BOOL;
    // minimalni a maximalni pozadovana vyska hladiny
    hmin : REAL;
    hmax : REAL;
END_VAR
VAR_OUTPUT
    // vystup regulatoru
    out : REAL;
END_VAR

// provadeni tela jednou za 0,1s
IF clock = TRUE AND min_clock = FALSE THEN
    //vypocet min. a max. hladiny
    hmin := yw - hyst/2.0;
    hmax := yw + hyst/2.0;

    IF input < hmin THEN
        out := 1.0;
        up := 1.0;
    END_IF;
    IF input > hmax THEN
        out := 0.0;
        up := 0.0;
    END_IF;
    IF input > hmin AND input < hmax THEN
        out := up;
    END_IF;
END_IF;
min_clock := clock;
END_FUNCTION_BLOCK
```

Příloha B

Obsah přiloženého CD

K této práci je přiloženo CD, na kterém je uložena elektronická podoba této práce, zdrojové kódy a vizualizační projekt.

- Adresář 1: MosaicApp
- Adresář 2: RelianceApp