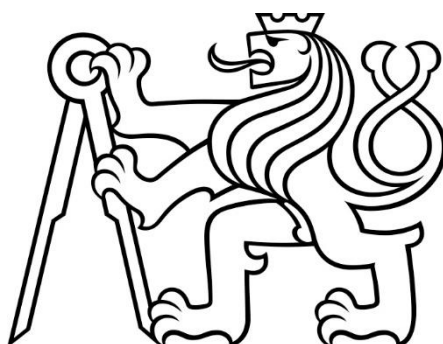


**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

**FAKULTA
STROJNÍ**



**BAKALÁŘSKÁ
PRÁCE**

2018

**DUŠAN
RICHTÁR**

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Richtár** Jméno: **Dušan** Osobní číslo: **459609**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Teoretický základ strojního inženýrství**
Studijní obor: **bez oboru**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Reléové řízení tepelných soustav s nesymetrickou dynamikou

Název bakalářské práce anglicky:

Relay control of thermal systems with asymmetric dynamics

Pokyny pro vypracování:

- Proveďte rešerši reléového řízení tepelných soustav s pasivním chlazením.
- Navrhněte reléové řízení vhodné pro řízení tepelných soustav s nesymetrickou dynamikou a pasivním chlazením.
- Simulačně v prostředí Matlab/Simulink ověřte navržené řízení.
- Na zvoleném laboratorním modelu ověřte navržené řízení.

Seznam doporučené literatury:

Balátě, J. : Automatické řízení. Praha : BEN ? technická literatura, 2003.
Šulc, B.; Vítečková, M.: Teorie a praxe návrhu regulačních obvodů. Praha : Vydavatelství ČVUT, 2004.
Hofreiter, M.: Základy automatického řízení, Praha : Vydavatelství ČVUT, 2018.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

prof. Ing. Milan Hofreiter, CSc., U12110.3

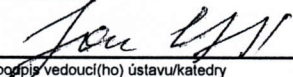
Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:


Datum zadání bakalářské práce: **18.04.2018**

Termín odevzdání bakalářské práce: **15.06.2018**

Platnost zadání bakalářské práce:


prof. Ing. Milan Hofreiter, CSc.
podpis vedoucí(ho) práce


podpis vedoucí(ho) ústavu/katedry


prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem předloženou bakalářskou práci zpracoval samostatně a použil jsem podklady uvedené v příloženém seznamu. Souhlasím s tím, že její výsledky mohou být dále využity dle uvážení vedoucího této práce prof. Ing. Milana Hofreitera, CSc..

Datum.....

Podpis.....

Poděkování

Děkuji mému vedoucímu práce panu prof. Ing. Milanu Hofreiterovi, CSc. za výběr zajímavého tématu bakalářské práce a za následné vedení, rady a konzultace při jeho zpracovávání. Dále bych rád poděkoval mým přátelům za jejich rady při práci s výpočetním softwarem Matlab a konzultace při vytváření algoritmu.

Abstrakt

Obsah této práce je zaměřen na vytvoření inteligentního, univerzálního dvoupolohového regulátoru, který by byl schopen regulace soustav s nesymetrickou dynamikou s minimálním zadáváním vstupních parametrů uživatelem. Možnost aplikací takto vytvořeného regulátoru je více. V této práci je využití zaměřeno na tepelnou techniku budov, kde má za úkol zmenšení překmitů teploty nad dovolenou hranici. Tímto dochází k zmenšení nákladů na vytápění. Pro vytvoření algoritmu byl použit výpočetní program MATLAB/Simulink. Následně byl algoritmus otestován na teoretickém modelu soustavy a na reálné laboratorní úloze.

Klíčová Slova

Dvoupolohová regulace, tepelná technika budov, Matlab, nelineární regulátor, nesymetrická soustava, tepelná soustava

Abstract

Focus of this bachelor's thesis is on creating smart, universal two position controller, that would be able to regulate a system with asymmetrical dynamic. Minimal intervention of the user is desired. There are many applications for the controller. The bachelor's thesis focus on use in thermal engineering of buildings, where it should be used to reduce oscillation over allowed temperature, which leads to energy savings. MATLAB/Simulink was used to create the algorithm for the controller. Subsequently the smart controller was tested on theoretical system and on real laboratory system.

Key words

Two position control, thermal engineering of buildings, nonlinear regulation, asymmetric systems, thermal system

Obsah

Obsah	6
1 Úvod	8
2 Regulace soustav	10
2.1 Regulační obvod	10
2.2 Dvoupolohová regulace	11
2.3 Symetrická a nesymetrická soustava.....	15
3 Regulovaná soustava.....	17
3.1 Programové prostředí MATLAB/SIMULINK	17
3.2 Reálná soustava a její náhrada	17
3.3 Časová konstanta T	18
3.4 Model soustavy	19
3.4.1 Nesymetrický model soustavy.....	20
4 Soustava a regulátor	23
4.1 Obyčejné relé.....	24
4.2 Optimalizace překmitů inteligentním regulátorem.....	25
4.2.1 Regulace překmitů následující první překmit.....	26
5 S – function	28
5.1 Jak S-Function funguje	28
5.2 Callback Method	29
6 Inteligentní regulátor.....	30
6.1 Regulace prvotního překmitu.....	30
6.1.1 Popis způsobu regulace prvního překmitu způsobem 1.....	30
6.1.2 Popis způsobu regulace prvního překmitu způsobem 2.....	33
7 Reálný experiment	35

7.1	Realizace experimentu na kaskádových nádobách.....	36
7.2	Obyčejné relé.....	37
7.3	Regulace překmitů následující první překmit.....	38
7.4	Soustava s inteligentním regulátorem (1 způsob).....	39
7.5	Soustava s inteligentním regulátorem (2 způsob).....	44
8	Závěr.....	45
9	Přílohy.....	46
9.1	Příloha 1.....	46
9.2	Příloha 2.....	47
9.3	Příloha 3.....	49
9.4	Příloha 4.....	50
9.5	Příloha 5.....	51
9.6	Příloha 6.....	52
10	Zdroje.....	54

1 Úvod

V dnešní době je trendem ve strojírenství automatizace. Od automatického řízení výrobních linek, strojírenských podavačů, robotů a ramen je možnost ji využít i k optimalizaci procesů, které vedou k zefektivnění výroby, úspory času nebo k úsporám energie. Tlak Evropské unie a zvýšený zájem koncových uživatelů o úsporu energie v budovách nás dovedl k novému trendu *inteligentních budov*. Projektanti budov se ve svých projektech zaměřují na sbírání dat z různých zdrojů, jejich kombinaci a následné řízení a regulaci, které má za následek úsporu energie a zkvalitnění pobytu v budově. Již dnes je běžnou praxí být schopen řízení monitorovaného objektu, přes uživatelské rozhraní na telefonu nebo počítači samotného uživatele. Následná chytrá regulace na základě naměřených dat je obzvláště důležitá. Bakalářská práce je zaměřena na využití automatického řízení v tepelné technice budov, kde je využita k regulaci teploty v soustavě, kterou si můžeme představit jako místnost, výrobní halu nebo celý dům.

Cílem této práce je navrhnout inteligentní dvou-polohový regulátor, který by udržoval teplotu v místnosti v nastaveném intervalu teplotní pohody. Následné ověření jeho funkčnosti na praktickém případě, který bude analogií k vytápěné místnosti, je dobrým předpokladem nasazení regulátoru do reálných systémů. Regulace je zaměřená na minimalizování překmitů regulované veličiny soustavy nad a pod požadovanou hodnotu. Překmity soustavy jsou způsobené setrvačnými účinky soustavy. Na rozdíl od složitějších spojitých PID regulátorů, je síla binárních (dvou-polohových regulátorů) v jejich konstrukční a technické nenáročnosti a v neposlední řadě přijatelnou cenou.

V praxi se setkáváme s obrovským počtem problémů, které komplikují regulaci samotné soustavy. Na rozdíl od ilustračních příkladů, které jsou uvedeny v učebnicích automatického řízení, je popis reálné soustavy v praxi velký problém a to zejména z důvodů, že zkoumaná soustava není lineární a má rozdílnou dynamiku. V učebnicích popsané způsoby linearizování kolem pracovního bodu jsou těžce aplikovatelné a z praxe víme, že vytápěná místnost se chová jinak při aktivním topení a při pasivním chladnutí. V reálném světě jsou tyto dva děje popsány zcela jinými matematickými modely. Neznalost přenosové funkce soustavy, podle které bychom mohli regulátor nastavit, je další komplikací. Experimentální zjišťování přenosových funkcí v realitě může být u velkých soustav s velkými koeficienty přenosové funkce velmi zdlouhavý, nepřesný a finančně náročný proces. V takovém případě se dá pochopit i určitá

neochota vlastníka k zprostředkování zkoumaného systému pro výzkum a následná ztráta jeho zájmu.

V teoretické části bude v softwaru MATLAB/Simulink znázorněná nelineární dynamická soustava, na které bude postupně znázorněn proces vedoucí k úpravě dvoupolohové regulace.

[6] [8] [9]

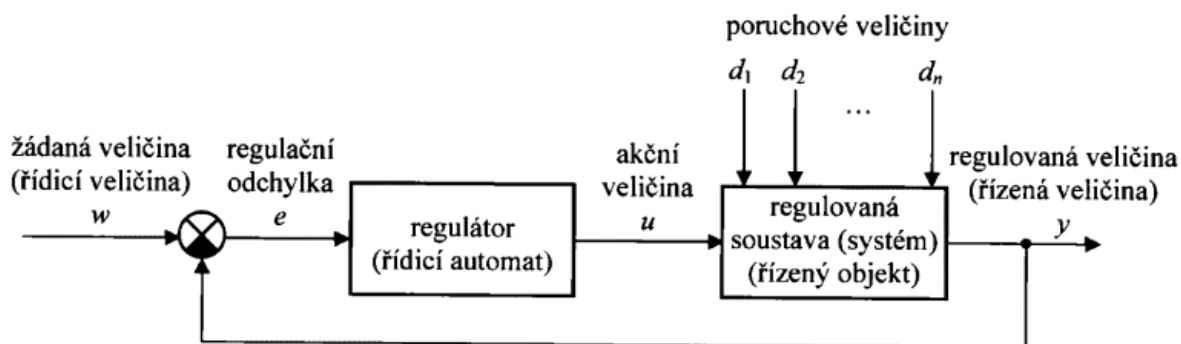
2 Regulace soustav

2.1 Regulační obvod

Regulace neboli zpětnovazebné řízení je založena na porovnávání regulované veličiny $y(t)$ a řídicí veličiny $w(t)$ v čase t přes součtový člen. Hodnota řídicí veličiny $w(t)$ (požadovaná teplota v místnosti, velikost otáček motoru) je nastavena uživatelem. Cílem regulace je, aby rozdíl mezi řídicí veličinou a regulovanou veličinou byl co nejmenší, nejlépe roven 0. Rovnicí (1) získáme regulační odchylku $e(t)$, která následně vstupuje do regulátoru, regulátor vyhodnotí regulační odchylku $e(t)$ a určí hodnotu akční veličiny (zásahu) $u(t)$. Akční zásah $u(t)$ může nabývat v závislosti na velikosti regulační odchylky a na použitém typu regulátoru různých hodnot.

$$e(t) = w(t) - y(t) \quad (1)$$

Na obr. 1 je ilustrováno blokové schéma regulačního modelu s popisem všech jeho veličin. Z obrázku je zřejmé, že hodnota regulační odchylky $e(t)$ je závislá na řídicí veličině $w(t)$ a poruchových veličinách $d_i(t)$.



Obr. 1 - Regulační obvod (převzato z [2] str. 38)

Jakoukoli odchylku regulované a řídicí veličiny můžeme považovat za nežádoucí a snažíme se je pomocí regulace odstranit. Z toho vyplývají dva základní body, které charakterizují úkoly regulačního obvodu soustavy [2]

- Zajistit, aby regulovaná veličina co nejlépe sledovala průběh řídicí veličiny.
- Eliminovat působení poruchových veličin tak, aby se jejich vliv v obvodu na regulované veličině projevil co nejméně.

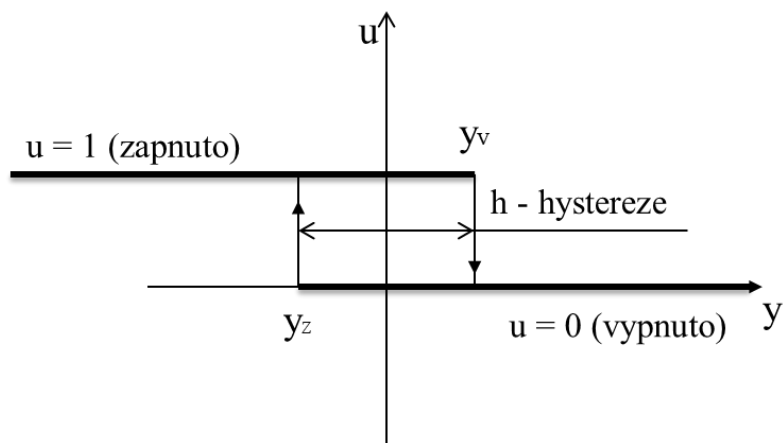
2.2 Dvoupolohová regulace

Dvoupolohové nebo třípolohové regulátory se zařazují do kategorie nespojitých regulátorů, které se vyznačují zejména tím, jak už jejich pojmenování napovídá, že jejich akční veličina $u(t)$ nabývá pouze určitého počtu hodnot. Akční veličina se v takovém případě mění z jedné hodnoty na další skokem. V případě dvoupolohových regulátorů akční veličina nabývá dvou hodnot, logické 0 nebo 1 viz obr 2. Hodnota akční veličiny $u(t)$ závisí na velikosti regulační odchylky $e(t)$, je-li $e(t) > 0$ \parallel $e(t) < 0$, nebo v případě s hysterezí $e(t) > y_{\text{v}}(\text{horní hranice hystereze})$ \parallel $e(t) < y_{\text{z}}(\text{spodní hranice hystereze})$. Oproti složitějším spojitém regulátorům nám velikost regulační odchylky neovlivňuje velikost akčního zásahu $u(t)$. Když se zaměříme na praxi, zjistíme, že je dvoupolohový regulátor hojně zastoupen na příklad v termostatu žehliček, varných konvicích nebo u čerpadel. Časté využití dvoupolohových regulátoru je zejména kvůli jeho konstrukční jednoduchosti, příznivé pořizovací ceně a spolehlivosti řízení. Můžeme se dokonce setkat s použitím čistě mechanických termostatů, bimetalu, u kterého se k regulaci využívá pouze jeho tepelné roztažnosti materiálu.

$$u(t) = 1 \text{ (otevřeno, zapnuto)} \vee 0 \text{ (zavřeno, vypnuto)}$$

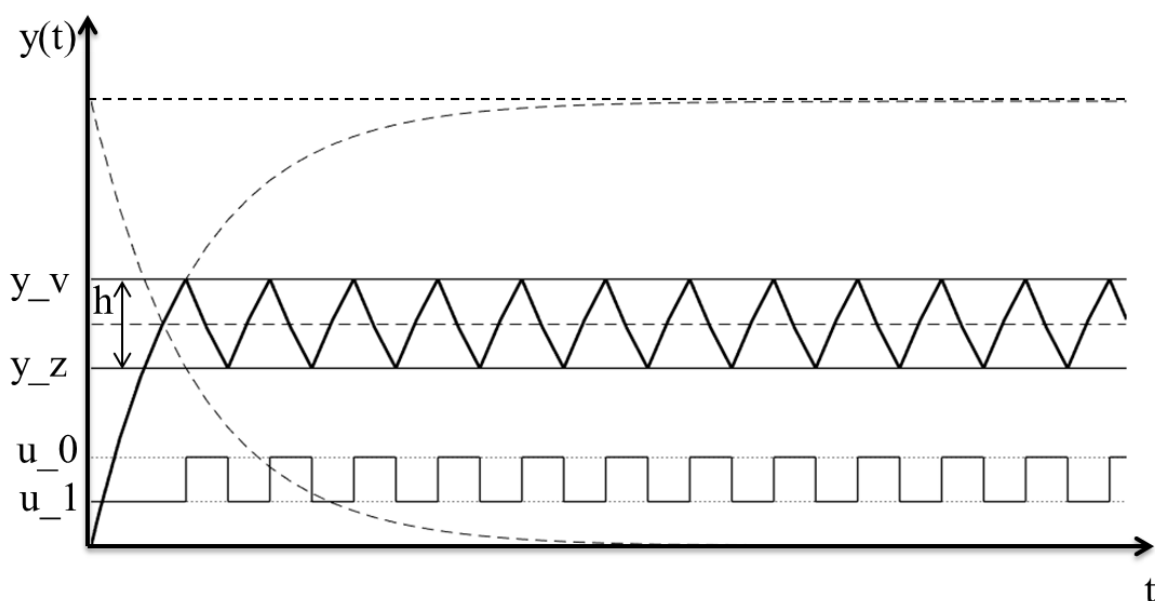
Obr 2 - hodnoty akční veličiny

Aby nedocházelo k častému přepínání mezi hodnotami akční veličiny $u(t)$, logickou nulou a jedničkou, při udržování hodnoty regulované veličiny $y(t)$ na řídicí veličině $w(t)$, je regulátoru nastavována hystereze. Hysterezi chápeme jako interval kolem řídicí veličiny, ve kterém se regulovaná veličina ještě může pohybovat a je uživatelem akceptovatelná. Tímto způsobem se za cenu zvětšení odchylek mezi $y(t)$ a $w(t)$ a určité ztráty přesnosti regulace zabraňuje rychlému mechanickému opotřebení regulátoru. Princip spínání dvoupolohového regulátoru pro $y(t) = 0$ je graficky znázorněn na obr. 3.

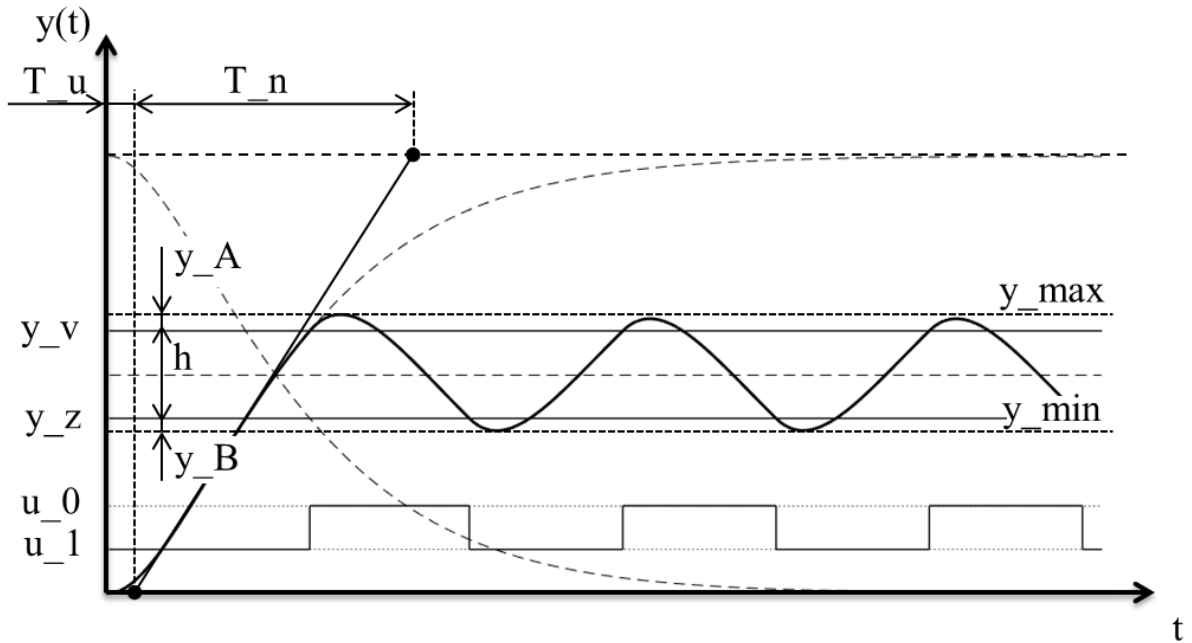


Obr. 3 - princip spínání dvoupolohového regulátoru s hysterezí

Obr. 3 můžeme nazvat statickou charakteristikou regulátoru. Vidíme, že se akční zásah $u(t)$ mění skokově. Dle obrázku pozorujeme, že se akční zásah $u(t)$ rovná hodnotě 1 (zapnuto) až do okamžiku, než regulovaná veličina $y(t)$ dosáhne své vypínací úrovně y_v , znázorňující horní hranu nastavené hystereze. Následně se hodnota akčního zásahu $u(t)$ přepne na 0 (vypnuto), v které setrvá až do okamžiku, v němž se hodnota regulované veličiny dostane do zapínací úrovně y_z , neboli spodní hrany hystereze. Pásmo hystereze můžeme také definovat jako interval, v kterém může akční zásah regulátoru nabývat jak logické 0, tak 1 podle toho, zda byl předem vypnut nebo zapnut.



Obr. 4 - přechodová a impulsní charakteristika soustavy popsanou diferenciální rovnicí prvního řádu



Obr. 5 - přechodová charakteristika druhého řádu

y_v	vypínací úroveň (horná hrana hystereze)
y_z	zapínací úroveň (spodní hrana hystereze)
u_0	regulátor nastaven na hodnotu u_0 , vypnuto
u_1	regulátor nastaven na hodnotu u_1 , zapnuto
y_A	překmit
y_B	podkmit
T_u	doba průtahu
T_n	doba nábehu
y_{max}	maximální hodnota regulované veličiny
y_{min}	minimální hodnota regulované veličiny

Tabulka 1 - legenda k obr. 4 a obr. 5

Na obr. 4 je graficky znázorněna přechodová charakteristika symetrické soustavy prvního řádu. Zda se jedná o soustavu prvního nebo vyššího řádu rozpoznáme tak, že regulovaná veličina reaguje na změnu akční veličiny okamžitě, nedochází k žádným průtahům. V době skokové změny protíná tečna rychlosti změny regulované veličiny vypínací úroveň regulátoru. Stejný jev vidíme, když se nám hodnota regulované veličiny dostane na vypínací úroveň, a poté zpět zapínací úroveň. Nedochází k žádnému překmitu této veličiny a její směr se okamžitě změní stejně tak se změní znaménko a hodnota její derivace. Z přechodové charakteristiky regulace je zřejmé, že se soustava při regulaci neustálí na určité hodnotě $y(t)$, ale dosáhne pouze svého *limitního cyklu* ve svém hysterezním intervalu. V praxi, vztaženou na tepelnou techniku budov, se se soustavou prvního řádu téměř nesetkáme, není taky zcela běžné, že bychom narazili na zcela symetrickou úlohu, takové úlohy se používají zejména v učebnicových a ilustračních příkladech.

U reálných soustav dochází vlivem setrvačnosti a dopravního zpoždění k překmitům nad y_v a pod y_z . Tyto reálné soustavy zjednodušíme a popisujeme diferenciálními rovnicemi druhého a vyššího řádu. Reakce regulované veličiny na skokovou změnu se liší oproti soustavě prvního řádu tím, že změnu rychlosti regulované veličiny na začátku můžeme považovat za nulovou. V počátečním momentu je směrnice tečny k přechodové charakteristice nulová. Při samotném simulování soustavy zpravidla předpokládáme linearitu a rozdělujeme poruchové veličiny s vlastními přenosovými funkcemi na jednotlivé části. Kvůli setrvačnosti celého systému a jeho překmitům regulované veličiny za hranici hystereze dochází k prodloužení časového intervalu přepínání akční veličiny $u(t)$ mezi 0 a 1. Zmenšením pásma hystereze v takovém případě docílíme přesnější regulace, aniž bychom se museli obávat rychlého opotřebení mechanické části regulátoru. Výsledná diferenciální rovnice systému závisí na spoustě faktorů. Jako jednou z nejčastějších příčin dopravního zpoždění v tepelné technice budov si můžeme představit proudění ohřívací kapaliny v trubkách, zpožděná reakce čidla na zvýšení teploty v místnosti, vzdálenost čidla od topení nebo nepravidelné proudění ohřátého vzduchu v místnosti. Poruchové veličiny pro nás potom mohou představovat, počet lidí a spotřebičů v dané místnosti nebo kvalita zateplení a jiné.

Chování nasimulované symetrické soustavy vyššího řádu můžeme sledovat na obr. 5. Překmit nad vypínací úroveň je označen y_{-A} a pod zapínací úroveň y_{-B} . Velikost y_{-A} získáváme z rovnice (2) a naopak y_{-B} z (3). Jelikož se s regulovanou

veličinou dostáváme mimo hysterezi, jsme schopni následnou úpravou vypínací a zapínací úrovně dosáhnout požadovaného průběhu regulované veličiny. Přechodové charakteristiky pro soustavy druhého a vyššího řádu jsou charakteristické inflexním bodem (přechod z konvexního průběhu do konkávního). Vycházíme z toho, že průběhy v okolí inflexního bodu jsou téměř přímkové a určení tečny v tomto bodě považujeme za relativně přesné. Prodloužením tečny vycházejícího z inflexního bodu do hodnoty statické citlivosti K_u a do ustáleného stavu přechodové charakteristiky, kdy $y(t) = 0$, získáváme na ose času dobu průtahu T_u a dobu náběhu T_n . Tyto veličiny jsou závislé na vlastnostech řízené soustavy, které nám následně ovlivňují statickou citlivost, a tedy i T_u a T_n . Velikost hodnot T_u a T_n nám zejména slouží k vyhodnocování soustav z grafického zápisu. Tyto veličiny lze využít k nastavování složitějších regulátorů na příklad metodou Zieglera a Nicholse z přechodové charakteristiky nebo můžeme v určité míře předpovídat chování soustavy, čehož bude dále v teoretické části využito. Jelikož neznáme přechodové funkce reálných soustav, je toto předpovídání velmi nepřesné, ale v mále míře může pomoci regulaci systému. [1] [2] [5] [6] [12]

$$y_A = y_{max} - y_v \quad (2)$$

$$y_B = |y_{min} - y_z| \quad (3)$$

2.3 Symetrická a nesymetrická soustava

Na obr. 4 a obr. 5 byly znázorněny systémy se symetrickou dynamikou. V praxi, kterou vztáhneme na tepelnou techniku budov, se setkáváme s výraznou nesymetrií pro dvě hodnoty akční veličiny u_1 a u_0 . Uvažujeme – li soustavu v letních měsících, předpokládáme, že akční člen (klimatizace) slouží jen k tomu, aby soustavu chladil. Bylo by iracionální se domnívat, že by soustava při dosažení požadované teploty a následném vypnutí klimatizační jednotky měla stejnou dynamiku jako při zapnutém chlazení. Proces oteplování místnosti bývá zpravidla oproti vychlazování velmi dlouhý s velkými konstantami přechodové funkce. Kvůli velkému množství poruchových veličin, jako jsou prostup tepla stěnami nebo okny, pohyb osob v místnosti, množství a rozmístění spotřebičů, venkovní teplota, míra oslunění v daný čas atd., je velmi složité přesně dopočítat kýžené konstanty přechodové funkce. Vzhledem k vyspělosti výpočetní techniky a znalosti teorie tepelných procesů se k těmto konstantám můžeme

přiblížit výpočty *prostupu tepla rovinnou stěnou, sdílením tepla vedením nebo prouděním a jiné*. [12] Jiným, extrémnějším a snadno představitelným příkladem může být varná konvice. Vyhřátí vody ve varné konvici proběhne v řádu několika minut. Její zchlazení na původní teplotu může trvat v rámci hodin v závislosti na kvalitě izolace. Z uvedených příkladu je zřejmé, že v praxi často pracujeme se systémy s nesymetrickou dynamikou.

3 Regulovaná soustava

3.1 Programové prostředí MATLAB/SIMULINK

Program Matlab a jeho skriptovací jazyk vyvíjený firmou MathWorks je programovatelné prostředí, které pomáhá uživateli při simulaci a analýze soustav a matematických modelů. MATLAB/SIMULINK je program a rovněž uživatelské rozhraní, které je instalováno zároveň s Matlabem a umožní uživateli odstoupit od textového programovacího jazyka k blokovým schémátům s předem naprogramovatelnými funkcemi. Tyto bloky slouží zejména k usnadnění programování dynamických systémů, které by bylo textově těžké naprogramovat, a vnáší nám do modelu určitou vizuální stránku napomáhající k lepší přehlednosti a orientaci.

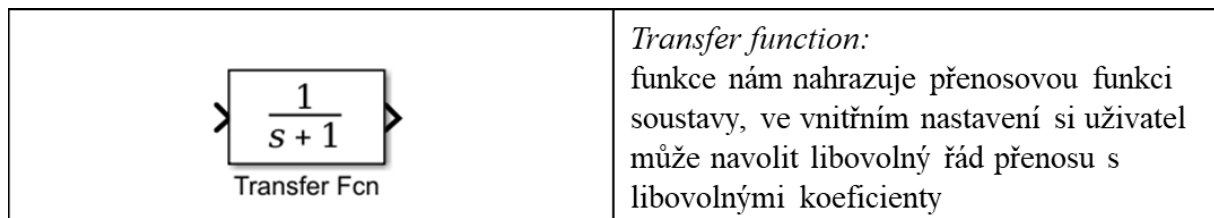
3.2 Reálná soustava a její náhrada

Jak již bylo zmíněno, reálné soustavy popisujeme diferenciálními rovnicemi. Pro zjednodušení řešení převádíme lineární diferenciální rovnice pomocí *Laplacovy transformace* na algebraické rovnice, které nám výrazně usnadňuje řešení. Laplacova transformace je nám zejména nápomocná při popisu reálných i teoretických soustav, které popisujeme jejichmi přenosovými funkcemi $G(s)$. Přenosovou funkci systému zjišťujeme v praxi analytickou nebo experimentální identifikací. Jedná-li se o učebnicový příklad, tak výpočtem. Tyto znalosti o systému jsou dále využívány k správnému nastavování regulátorů. Přenosovou funkci systémů chápeme jako podíl Laplacova obrazu výstupní veličiny na Laplacově obrazu vstupního signálu, viz ((4)). Tímto dojde k zjednodušení popisu soustavy. Soustavu dále můžeme rozdělit na přenosové funkce jednotlivých částí, které nám soustavu ovlivňují. Následným spojením těchto dílčích částí bychom dostali výslednou přenosovou funkci. Tato možnost nám dovoluje udržovat v modelu soustavy určitou přehlednost a možnosti přidání jakýchkoli dalších poruchových veličin, aniž bychom museli upravovat jednu komplexní diferenciální rovnici.

V Matab/Simulink se nám nabízí použití bloku *Transfer Fcn* ze simulinkové knihovny *Continuous*, který nahrazuje přenosovou funkci soustavy. Ve vnitřním

nastavení *Transfer Fcn* jsme jednoduchým způsobem schopni nastavit výsledný Laplaceův přenos. Viz. obr. 6.

$$G(s) = \frac{Y(s)}{U(s)} \quad (4)$$



Obr. 6 - Transfer Function

Spojení bloků Simulinku provádíme dle zásad blokové algebry. Spojovací linie mají přesně daný směr, který nám určuje, v jaké pořadí bude Matlab provádět dané funkce systému. [2][10]

3.3 Časová konstanta T

Důležitým parametrem přechodové funkce je její časová konstanta T sloužící k ohodnocení jejích dynamických vlastností. Časová konstanta nám udává čas, za který soustava v reakci na jednotkový skok urazí přibližně 63,2% dráhy své přechodové charakteristiky. Uděláme-li tečnu k přechodové charakteristice, která bude zároveň přeponou pravoúhlého trojúhelníku, vytyčí její subtangenta na časové ose úsek rovnající se časové konstantě T nezávisle na tom, z jakého bodu bude tečna přechodové charakteristiky vycházet. Ve skutečnosti trvá přechodový děj soustavy nekonečně dlouho. Když si uvědomíme, že rychlost přechodového děje s rostoucím časem exponenciálně klesá, můžeme děj po uplynutí několikanásobku časové konstanty považovat za ukončený. V praxi předpokládáme, že za dobu $3T$ dosáhne sledovaná veličina přibližně 95% své ustálené hodnoty a po $5T$ uvažujeme, že dostáhla 99% své ustálené hodnoty. [2][3][6]

3.4 Model soustavy

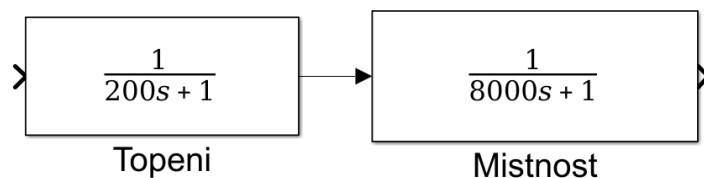
Soustava, na kterou budeme připojovat regulátor a postupně optimalizovat regulaci, se skládá z modelu místnosti, na kterou působí ztráty z okolního prostředí. Soustava byla navržena pro zimní období, ve kterém se venkovní teploty pohybují pod teplotami v místnosti, dochází tedy k nežádoucím ztrátám, příklady byly již zmíněny výše. Velikosti časových konstant ztrát a poruchových veličin budou mít velmi různé hodnoty. Časové konstanty ztrát přes stěny, okna nebo strop budou závislé především na kvalitě tepelné izolace budovy a budou mnohonásobně větší než časová konstanta čidla snímajícího teplotu v místnosti. Čidlo v takovém případě považujeme za poruchovou veličinu.

Pro popis soustavy, jsme potřebovali vytvořit diferenciální rovnici druhého nebo vyššího řádu. K tomu jsme použili sériové zapojení přenosové funkce místnosti, topení a ztrát. K modelu soustavy je možné připojit další bloky reprezentující nové části ovlivňující celkovou přenosovou funkci místnosti, jako jsou na příklad čidlo, které snímá teplotu místnosti, teplotní vliv spotřebičů na místnost, nebo výskyt osob v místnosti. Těchto přídavných ztrát může být nekonečně mnoho. Přidávání dalších parametrů ovlivňující soustavu se můžeme dostat k přesnějším výsledkům, problémem je určení těchto dílčích parametrů. Přidání dalších vlivů by mělo za následek zvýšení řádu přechodové funkce a došlo by ke změnám dopravního zpoždění a zvýšení setrvačnosti soustavy. My jsme se omezili pro zjednodušení a ilustraci na topení a místnost, ke kterým jsme následně připojili ztráty. Část reprezentující vytápěnou místnost soustavy je symetrická a je znázorněná na obr. 7. V tabulka 2 jsou vypsány hodnoty časových konstant, které jsme zvolili pro naši experimentální soustavu.

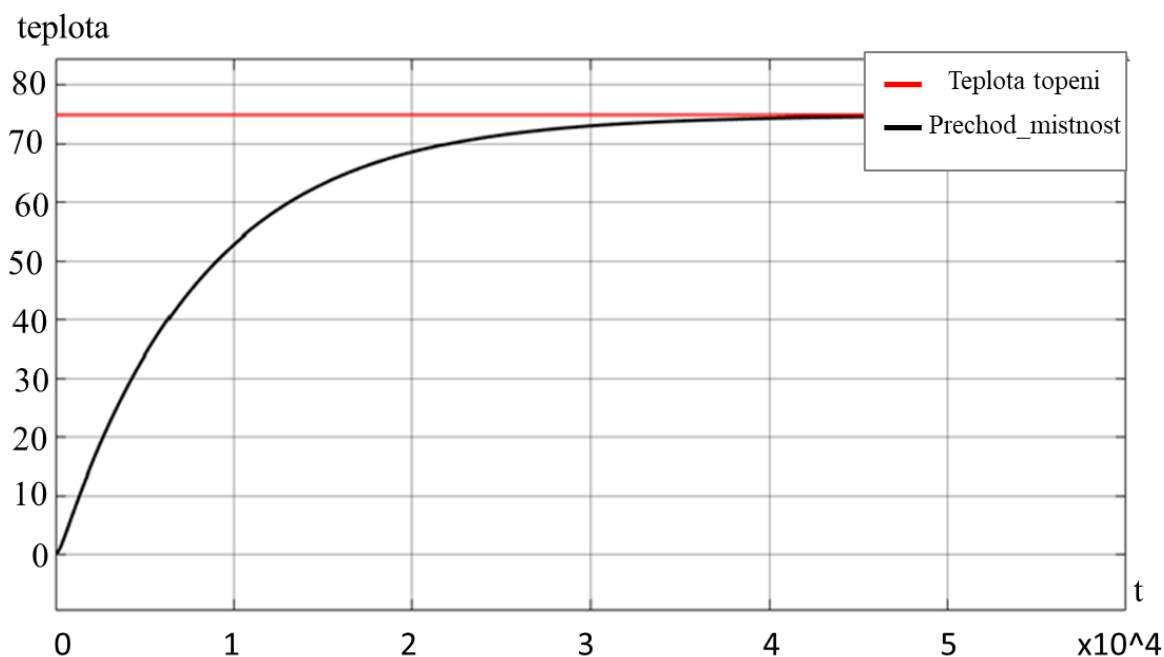
$T_1 = 200s$	Časová konstanta topení
$T_2 = 8000s$	Časová konstanta místnosti
$T_3 = 10000s$	Časová konstanta ztrát

Tabulka 2 - Hodnoty časových konstant

Obr. 8 ilustruje přechodovou charakteristiku místnosti s přechodovou funkcí určenou z obr. 7, jako odezvu na jednotkový skok roven teplotě topení 75°C pro případ, že by teplota v místnosti nebyla nijak regulovaná a nepůsobily by na ni žádné ztráty, ustálí se teplota v místnosti na teplotě topného tělesa.



Obr. 7 - Blokové schéma modelu místnosti



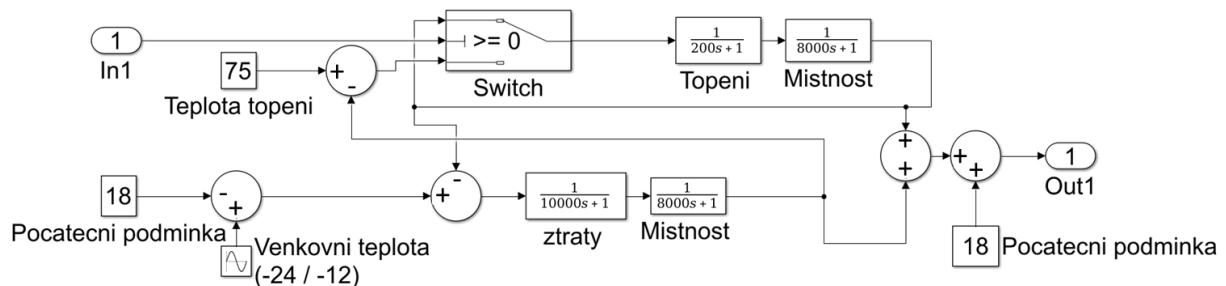
Obr. 8 - přechodová charakteristika místnosti

Výsledný model soustavy místnosti se ztrátami dostaneme tak, že připojíme paralelně novou větev reprezentující ztráty místnosti na již stávající větev určující chování vytápěné místnosti.

3.4.1 Nesymetrický model soustavy

V realitě jsou tepelné soustavy velmi nesymetrické a setkáváme se u nich s rozdílnou dynamikou pro topení a chladnutí. V ilustrované soustavě počítáme s tím, že je třeba soustavu aktivně vytápět. To znamená, že akční veličina $u(t)$, reprezentující teplotu topení, je vždy větší než regulovaná veličina $y(t)$, která značí teplotu místnosti. Značnou nesymetrii mezi topením a chladnutím můžeme sledovat tehdy, kdy soustavu připojíme na dvoupolohový regulátor s předem danou hysterezí. Poté, co teplota

v místnosti dosáhne do velikosti horní meze zadané hystereze, přestane topné těleso vytápět a soustava nám pomalu přejde z jednoho děje do druhého. Proces chladnutí je v takovém případě závislý pouze na ztrátách místnosti a trvá několika násobně déle, než vytápění místnosti.



Obr. 9 - blokové schéma nesymetrického modelu soustavy (převzato a upraveno z [6])

Na obr. 9 je znázorněn model nesymetrické soustavy místnosti. Zda se soustava vytápí nebo je topení vypnuto nám určuje hodnota $In1$, jakých hodnot může $In1$ nabývat je principiálně popsán v kapitole 2.2, dDvoupolohová regulace. Na obrázcích obr. 4 a obr. 6 můžeme sledovat hodnotu u_1 a u_2 v soustavě prvního a druhého řádu, které reprezentují hodnotu $In1$. Vztah 1 určuje možné logické hodnoty vstupu $In1$, které může $In1$ nabývat.

$$In1 = \begin{cases} 1 & (\text{topení zapnuto}) \\ -1 & (\text{topení vypnuto}) \end{cases}$$

Vztah 1

Na obr. 9 vidíme spojení dvou větví. Ztrát, které působí na místnost (spodní polovina) a samotné místnosti (horní polovina). Jelikož jsou bloky jednotlivých částí spojeny sériově, získáváme diferenciální rovnici druhého řádu. Z toho plyne, že má soustava určité setrvačné účinky závislé na řádu diferenciální rovnice a na hodnotách časových konstant jednotlivých částí. V realitě si tuto skutečnost představíme tak, že máme topení o teplotě 80 °C nebo 75°C. V případě, že požadujeme teplotu v místnosti 21°C a topení na této teplotě vypneme, bude nám topení kvůli jeho teplotní kapacitě nadále místnost vytápět až do doby, než se teplota v místnosti a teplota topení vyrovnají. Překmit, ke kterému dochází, můžeme požadovat za ztráty. Tato skutečnost byla motivací k vytvoření inteligentního regulátoru, který by byl schopen úpravy vypínací úrovně topení tak, aby snížil velikost překmitů nad požadovanou úroveň.

Inteligentní regulátor nemá za úkol překmity úplně eliminovat, ale jenom je co nejvíce přiblížit k požadované teplotě a tím snížit energetické ztráty celého systému. Eliminace překmitů není vzhledem k rozmanitosti poruch, jednoduchosti regulátoru a neschopnosti identifikace soustavy možná.

Pro správné fungování celého systému jej musíme připojit do regulačního obvodu viz. obr. 1. Celkový model, ve kterém je soustava místnosti připojena na regulátor, je uveden příloha 1. Kromě soustavy místnosti, se v celkovém modelu nachází ještě dva další subsystémy *subsystém 1* a *subsystém 2*. *Subsystém 2* nahrazuje vytvořený regulátor, který připojujeme na soustavu. Jak již bylo zmíněno, regulátor je řešen pomocí *S-funkce* zdrojovým kódem psaném v Matlabu. Tento zdrojový kód s popisem je uveden v příloha 2. *S-funkce* realizující regulátor pracuje s mnoha parametry, proto je vnitřní struktura *subsystému 2* uvedena v příloha 3. *Subsystémem 1* získáváme hodnotu překmitu teploty nad horní hranici hystereze. Tento subsystém byl opětovně vyřešen *S-funkcí* kvůli absenci vhodného bloku v knihovně Matlab/Simulink. Jelikož subsystém má také složitou strukturu, je uvedena jeho podoba v příloze 5 a jeho zdrojový kód v příloze 4.

4 Soustava a regulátor

Regulace je rozdělena na více fází, proto budeme na soustavu místnosti z obr. 9 postupně připojovat regulátory a ilustrovat, jaký mají na průběh teploty vliv. V každé fázi bude vysvětlena řídicí logika regulátoru. Obdobným způsobem se postupovalo při vytváření inteligentního regulátoru, kdy ze začátku bylo na soustavu připojeno pouze obyčejné relé, podle kterého jsme zjistili, jak se soustava chová a co musíme při regulaci zohlednit. Následně jsme upravovali algoritmus relé až do podoby inteligentního regulátoru. Regulátor bude pro lepší ilustraci postupně připojen na dvě soustavy s různými hodnotami časových konstant. Připojení regulátoru na dvě různé soustavy nám navíc odráží jeho univerzálnost. Ve všech ilustračních případech chceme docílit teploty v místnosti 20 °C. Regulátorům jsme ve všech případech nastavili dovolenou hodnotu hystereze +/- 0,5 °C. Okolní teplota, která nám realizuje proces ztrát, osciluje v podobě sinusové křivky v tepelném rozmezí -24/-12 °C, aby lépe simulovala proměnlivost okolního prostředí. Kmitu realizující okolní teplotu je dána úhlová rychlost popsaná v rovnici (5), která se rovná jednomu kmitu za den, symbolizující proměnlivost teploty ve dne a v noci. V tabulka 3 jsou uvedeny hodnoty časových konstant pro jednotlivé případy.

$$T_{ok} = \frac{2\pi}{86400} = 7,2722 * 10^{-5} \text{ rad/s} \quad (5)$$

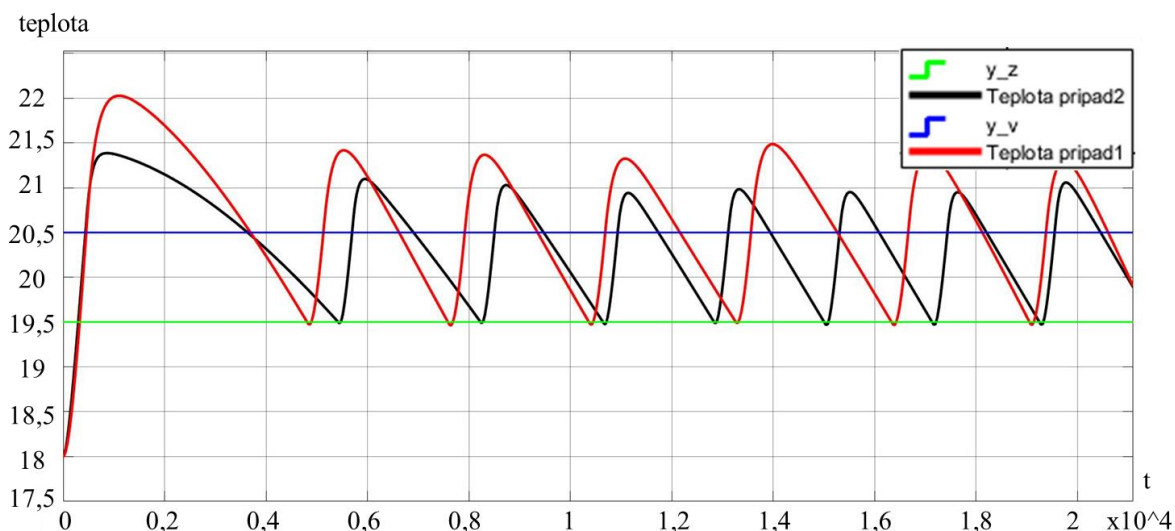
T_1 = 200s	Časová konstanta topení (případ 1)
T_2 = 8000s	Časová konstanta místnosti (případ 1)
T_3 = 10000s	Časová konstanta ztrát (případ 1)
T_4 = 100s	Časová konstanta topení (případ 2)
T_5 = 10000s	Časová konstanta místnosti (případ 2)
T_6 = 15000s	Časová konstanta ztrát (případ 2)

Tabulka 3 - hodnoty časových konstant pro případ 1 a 2

4.1 Obyčejné relé

Vytápíme-li místnost z počáteční teploty nižší, než je sporní hranice hystereze, je velikost prvotního překmitu vyšší než ostatní překmity. Důvodem je délka otevření topného ventilu způsobující prodloužení setrvačnosti systému. Poté co se teplota pohybuje v pásmu hystereze, dosáhne klesající teplota pouze na její spodní hranici. To má za následek, že při dalším regulování teploty bude topný ventil otevřený na mnohem kratší dobu, než to bylo u prvního překmitu, kdy jsme místnost vytápěli z určitých počátečních podmínek daných počáteční teplotou v místnosti. Porovnání délky vypnutí a zapnutí topného média lze pozorovat na hodnotách u_{-1} a u_{-0} na obr. 11.

Obyčejné relé reguluje soustavu podle principu popsaného v kapitole 2.2. Regulátor nijak neupravuje vypínací úroveň. Topení se vypne na horní hranici hystereze y_v a po vychladnutí na spodní hranici y_z se opět zapíná. Na obr. 10 jsou ilustrovány dvě varianty soustavy s rozdílnými délkami časových konstant rozdělených dle tabulka 3 na *případ 1* (červená křivka) a na *případ 2* (černá křivka). Vidíme, že překmity nad vypínací úroveň y_v jsou velmi značné a můžeme je považovat za energetické, tudíž i finanční, ztráty. Z obr. 10 je jasné, že nás při regulaci budou zajímat právě kmity, které překračují vypínací úroveň. Vzhledem k velké velikosti časové konstanty chladnutí má soustava setrvační účinky velmi malé. Dochází k velmi malým podkmitům zapínací úrovně. Takové ztráty považujeme za akceptovatelné a nebudeme se jimi v algoritmu zabývat. Jejich regulace by byla možná stejným principem, jako je použit k regulování překmitů soustavy.



Obr. 10 - teplota v místnosti regulovaná reléovým zařízením

4.2 Optimalizace překmitů inteligentním regulátorem

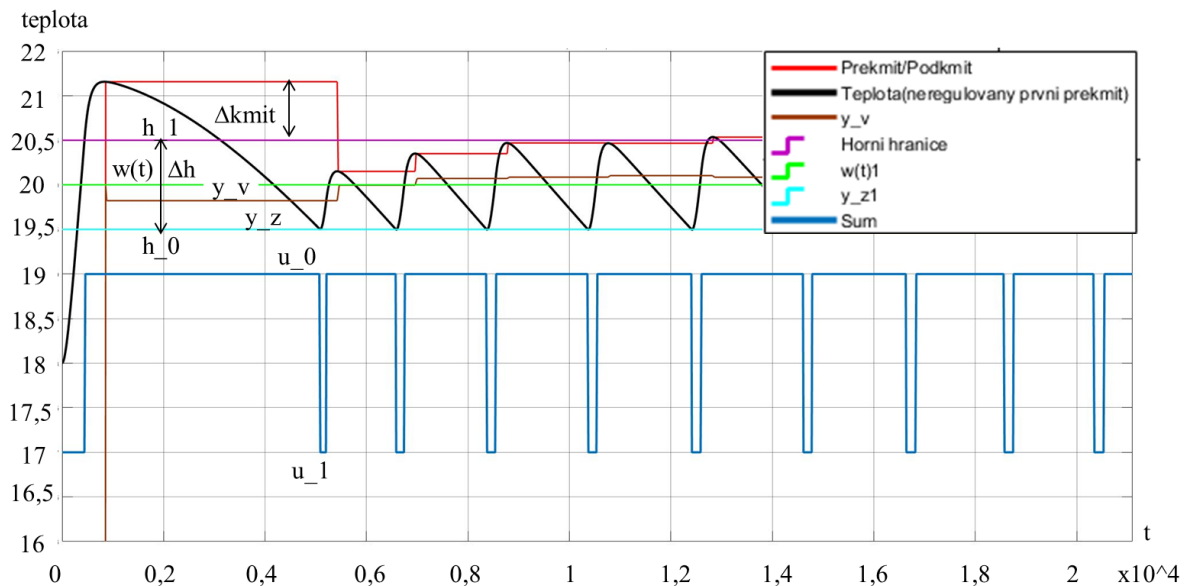
Při vytváření algoritmu chytrého regulátoru jsme rozdělili teplotní křivku na dvě části, na optimalizaci prvního překmitu a na regulaci následujících kmitů. Realizace regulace prvního kmitu je velmi složitá, protože při spouštění jakékoli soustavy nám chybí základní parametry k popisu soustavy. Neznáme působení okolních vlivů, samotnou velikost soustavy nebo teplotu topného tělesa. Výhoda inteligentního regulátoru je v tom, že je navržený tak, aby zvládal regulaci bez jakéhokoli zadávání hodnot samotným uživatelem. Samotný uživatel si volí požadovanou teplotu v místnosti, které chce dosáhnout a případnou velikost intervalu hystereze ve které se má teplota pohybovat.

Diplomová práce pana inženýra Straky [6] nám posloužila jako inspirace při tvorbě algoritmu překmitů následující první překmit. Myšlenka, která byla v práci uvedena, byla upravena a zrealizována v jiné formě. Hlavním rozdílem byla realizace algoritmu pomocí nadčasové funkce zvané Matlab 2 S-function a ne pomocí bloků z programu MATLAB/Simulink. Realizace složitých algoritmů pomocí bloků se jeví jako zastaralý a velmi složitý způsob. Nová funkce Matlab 2 S-function nám nabízí možnost vytvořit algoritmus jako textový skript, který je implementovatelný do modelu soustavy a kompatibilní s dalšími bloky Simulinku. Vytvořený program se při nahrávání do mikropočítače řídicího regulaci překládá překladačem do skriptovacího jazyka s ním kompatibilním.

Postup, kterým jsme postupovali při regulaci prvotního překmitu, bude popsán v dalších kapitolách této práce.

4.2.1 Regulace překmitů následující první překmit

Zmenšení překmitů nad vypínací úroveň docílíme tak, že předčasně zavíráme ventil topného tělesa (topení). K jaké změně vypínací úrovně má dojít, zjišťujeme postupnou kalibrací závislou na velikosti předešlého překmitu.



Obr. 11 - průběh teploty při regulaci překmitů následující první překmit

Na obr. 11 je znázorněn průběh teploty s adaptibilní vypínací úrovní y_v (hnědá barva). K této vypínací úrovni se dostáváme rovnicí (6). Kde $\Delta kmit$ je rozdíl mezi extrémem teploty daného kmitu s h_1 (horní hranicí hystereze). Hodnota h_0 v našem případě značí spodní hranu hystereze, které je s časem neměnná. Velikost $\Delta kmit$ může nabývat jak kladných, tak i záporných hodnot. To se stane v takovém případě, kdy úpravou vypínací úrovně a současným vlivem okolní teploty nastane situace, že nám teplota v místnosti podkmitne horní hranici hystereze. Chytrý regulátor poté podle znaménka $\Delta kmit$ vyhodnotí, jestli se má vypínací úroveň posunout směrem nahoru ($-\Delta kmit$) nebo dolů ($+\Delta kmit$).

Aby nedošlo k rozkmitání celého systému, nemůžeme měnit velikost vypínací úrovně o celkovou velikost $\Delta kmit$, ale pouze o její zmenšenou hodnotu. Hodnotu $\Delta kmit$ přenásobíme koeficientem $1/4$ nebo $1/5$ v závislosti na požadované přesnosti a rychlosti kalibrace vypínací úrovně. Počáteční hodnotu y_v nastavujeme dle (7). Tato hodnota byla určena empiricky na základě pokusů s naší soustavou. Pro zcela

neznámou soustavu by bylo nejvhodnější nastavit y_v na stejnou velikost jako je požadována teplota místnosti $w(t)$. Hodnota y_v je závislá na řádu přechodové funkce a velikostí časových konstant systému, její optimální velikost by mohla být předmětem dalšího zkoumání.

$$y_{v_n} = y_{v_{n-1}} + \frac{\Delta kmit}{4} \quad (6)$$

$$y_{v_{n=0}} = 0,65\Delta h + h_0 + \frac{\Delta kmit}{4} \quad (7)$$

Srovnáme-li teploty na obr. 10 (soustava s reléovým přepínačem) a obr. 11 (připojení na inteligentní regulátor), vidíme značný rozdíl v jejich průběhu teplot. Na **Chyba! Nenalezen zdroj odkazů.** nedocházelo k žádným změnám vypínací úrovně soustavy a vidíme relativně ustálenou velikost překmitů nad y_v soustavy, která se s časem mění pouze nepatrně a to vlivem poruchových veličin. Naproti tomu na obr. 11 vidíme postupné kalibrování teploty až do doby, kdy se s jejím kmitáním dostáváme do požadovaného intervalu pásma hystereze. V obou případech je očividné, že jsou největší ztráty způsobené prvotním překmitem. Jelikož inteligentní regulátor počítá s velkým setrvačným účinkem soustavy, který způsobuje velké překmitnutí prvního překmitu, dochází v případě s adaptivní vypínací úrovní (obr. 11) k velikému poklesnutí vypínací úrovně a následnému podkmitnutí teploty soustavy do té doby, než se soustava vykalibruje. Z těchto důvodů je optimalizace prvotního překmitu nedílnou součástí optimální regulace celého systému a je potřeba se na jeho regulaci při vytváření algoritmu pro inteligentní regulátor zaměřit.

Algoritmus řídící optimalizaci vypínací úrovně překmitu následující první překmit se nahrává v textové formě do Simulinkového bloku *Matlab 2 S-function*, který po připojení na celý model soustavy řídí jeho regulaci. Program řídící regulaci je v textové formě uvedený v příloze 2.

5 S – function

Mimo samotnou logiku inteligentního regulátoru, je forma jeho zpracování brána, jako další přidaná hodnota. Zpracování bylo provedeno pomocí S-Funkce, nástroje, který se používá především pro složité algoritmy, které jsou pomocí bloků simulinku složitě naprogramovatelné.

S-Function nám umožňuje rozšířit možnosti programového prostředí MATLAB/Simulink. Představit si ji můžeme, jako textové popsání Simulinkového bloku, který jsme schopni napsat v textovém jazyce MATLAB®, C, C+ nebo Fortran. Matlab nám nabízí základní předpřipravenou formu, *S-function template*, kterou lze následnými úpravami přemodelovat do námi požadované podoby a vytvořit kontinuální, diskrétní nebo hybridní systém.

5.1 Jak S-Function funguje

Pro správnou implementaci *S-Funkce* do našeho modelu a jejího napsání v textovém editoru je potřeba znát, jak Matlab pracuje s bloky simulinkové knihovny a způsob simulace celé soustavy.

Práce s bloky je jednoduchá, když si uvědomíme, že každý blok má tři typy parametru. Vstupy, stavy a výstupy. Vstupy jsou hodnoty, které jsou do bloku přivedeny. Stavy jsou určeny matematickou funkcí uvnitř bloku. Hodnoty výstupu jsou určeny funkcí závislou na zbylých parametrech bloku a čase soustavy dle rovnice (8).

$$výstup = f(vstup, stav, t) \quad (8)$$

Simulace celé soustavy provádí Matlab ve dvou fázích. Inicializační fáze (*initialization phase*) ve které Matlab zařazuje do systému bloky ze Simulinkové knihovny, určí parametry těchto bloků, typ signálu, délku kroku a pořadí, v kterém se zařazené funkce mají provést. V případě, že je potřeba uložení dat, určí místo jejich uložení. Simulace poté přejde do takzvané simulační smyčky (*simulation loop*), která provede simulaci po časových krocích, nazvané *Simulation step*. V této fázi se ve

smyčce provedou všechny funkce systému v pořadí, které bylo určeno v první fázi inicializace.

5.2 Callback Method

S-funkce je komplexní nástroj, který je schopný plnit více matematických operací a algoritmů najednou. Proto, aby S-funkce správně fungovala, musíme nastavit metody, které jsou v S-funkci použity. Tyto metody Matlab přivolává v každém simulačním kroku (simulation step) a jsou esenciální pro správné fungování programu napsaného v textovém editoru. Tyto metody nazýváme „callback methods“. Bez nich Matlab není schopen napsaný kód přečíst a neví co se má s napsaným kódem provést. Existuje velké množství *Callback Methods*, které se do S-funkce implementují v závislosti na tom, jak složité operace provádíme. Námi použité Callback Methods s popisem její úloh jsou uvedeny v tabulka 4.

Outputs	Funkce přivolaná v každém časovém kroku výpočtu. Má za úkol vypočítat hodnoty všech výstupů vycházejících z S-funkce v daném okamžiku. Tyto hodnoty dále ukládá ve formě řad.
Terminate	Tato funkce slouží k uvolnění paměti potřebné k správnému fungování programu nebo je přivolaná v případech, kdy je určitá část programu smazána z modelu.
Setup	V této funkci nastavujeme všechny potřebné parametry, které jsou používány v S-funkci. Například, zda se jedná o spojitou funkci. Počet vstupů a výstupů a další.

Tabulka 4 - Callback Methods

6 Inteligentní regulátor

Jak již bylo zmíněno, algoritmus inteligentního regulátoru jsme rozdělili na dvě části. Část, která upravuje překmity následující první překmit, a na regulaci prvního překmitu. V kapitole 4.2.1 *Regulace překmitů následující první překmit* byla popsána myšlenka optimalizující překmity. Z grafu na obr. 11 je zřejmé, že nám regulace funguje a vychází z chování soustavy, které je udáno prvním neregulovaným překmitem. Jde tedy říct, že o regulované soustavě něco víme a můžeme předpokládat její další chování.

U regulace prvního překmitu je problém v tom, že vůbec nevíme, jak se regulovaná soustava chová, co ji ovlivňuje a jaká může být velikost prvního překmitu, z kterého jsme vycházeli při další regulaci. Jedna z podstat navrhovaného inteligentního regulátoru by měla být jeho univerzálnost. Myšlenka byla taková, že uživatel připojí regulátor na soustavu, aniž by musel zadávat nějaké vstupní parametry. Proto jsme při regulaci prvního překmitu museli vycházet z toho, co nám o sobě samotná soustava byla schopna „říci“ a ne z nepřesných odhadů lajka nebo empirických odhadů odborníka.

6.1 Regulace prvotního překmitu

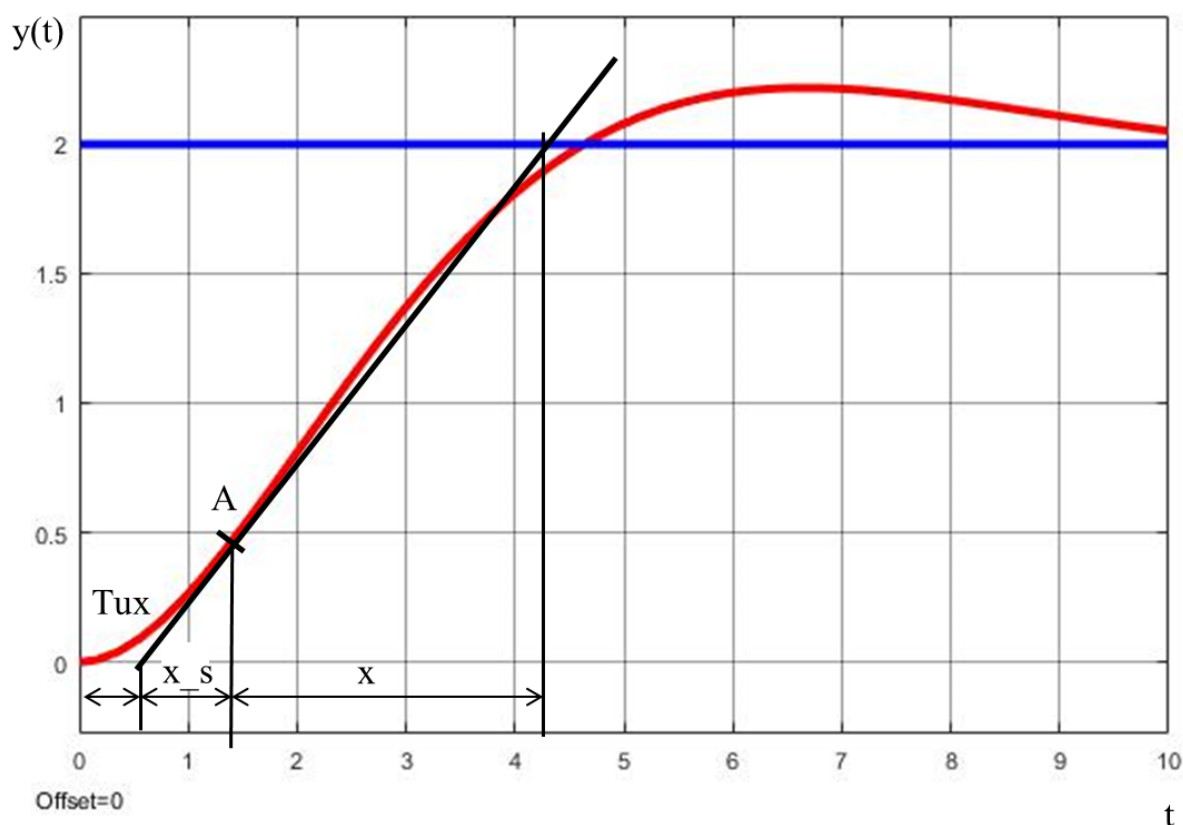
Regulaci prvotního překmitu jsme zkoušeli dvěma způsoby, které jsme navzájem teoreticky porovnávali.

6.1.1 Popis způsobu regulace prvního překmitu způsobem 1

Zjišťování důležitých parametrů proto, abychom mohli první překmit regulovat, jsme dostali podobným způsobem, jako získáváme hodnoty doby náběhu T_n a doby průtahu T_u , s tím rozdílem, že jsme veličiny nezkuovali pouze v inflexním bodě přechodové charakteristiky, ale v každém kroku, který program udělal hned od začátku spuštění simulace soustavy. Pro lepší představu se zaměříme na soustavu s přenosovou funkcí dle rovnice (9). Přenosová funkce je druhého řádu se statickou citlivostí 2. Poté co tuto soustavu připojíme na jednotkový skok, dostáváme průběh, který je znázorněn na obr. 12. Na to, abychom mohli upravovat vypínací úroveň soustav ještě předtím, než

nám překmitnou přes hodnoty, na kterých chceme, aby se nám výsledná teplota ustálila, potřebujeme časové hodnoty, x , x_s a T_{ux} , které jsou ilustrovány na obrázku. Tyto parametry jsou esenciální k tomu, abychom mohli vytvořit podmínky, podle kterých má inteligentní regulátor regulovat prvotní překmitnutí.

$$G(s) = \frac{2}{3s^2 + 2s + 1} \quad (9)$$



Obr. 12 - znázornění parametrů potřebných k regulaci prvního překmitu dle prvního způsobu

Bod A prochází každým výpočtovým krokem soustavy. V každém bodě, kterým A prochází, si představme tečnu, která nám vytyčí časové hodnoty x a x_s . Rovnicí přímky (10) v daném bodě A dopočítáme kýženou hodnotu x . Hodnotu x_s dostáváme analogickým způsobem z rovnice (11). T_{ux} získáme následným odečtením času, v kterém soustava provádí výpočet a vypočítané hodnoty x_s (rovnice(12)). Hodnota y značí teplotu, které soustava v daný moment nabývá, y_{v} v horní hrana hystereze (původní vypínací úroveň), y_{poc} reprezentuje teplotu, z které soustava vychází a dT je

hodnota derivace v raném výpočtovém kroku v čase t . Hodnoty i v rovnicích postupují od $i = t = 0 \rightarrow$ až do vypnutí soustavy.

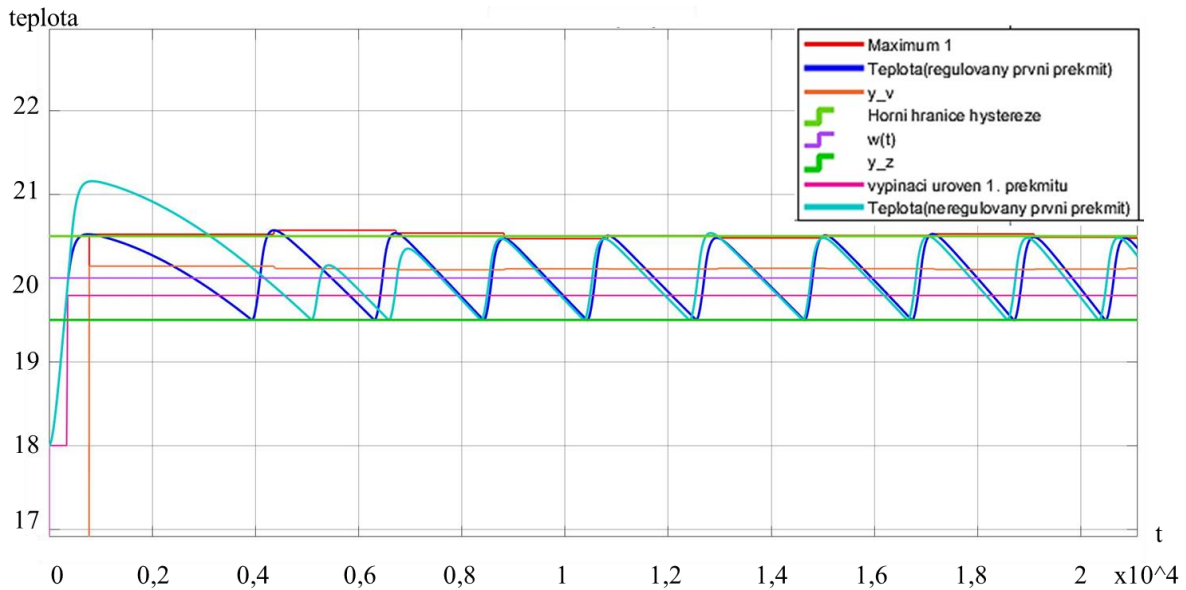
$$y_v = dT_i * x_i + y_i \quad (10)$$

$$y_i = x_{si} * dT_i + y_{poc} \quad (11)$$

$$Tux_i = t_i - x_{si} \quad (12)$$

Nejdůležitějším parametrem, který nám dokáže předem říci, jak se soustava chová, je naměřená hodnota Tux . Myšlenka inteligentního regulátoru je založena na porovnávání této hodnoty s hodnotou x . Topení má za úkol soustavu vytápět do momentu než se $x = Tux$. V tomto momentu se topení vypne. Předpokládáme podobný průběh setrvačnosti soustavy, jako byl průběh jejího pomyslného průtahu. Jelikož nám Tux udává tuto pomyslnou hodnotu doby průtahu, předpokládáme, že teplota soustavy doběhne do jeho maxima za podobný čas.

Jelikož doba průtahu a následná setrvačnost systému nemají stejný charakter, nedoběhne teplota v místnosti přesně na požadovanou hodnotu. Tímto způsobem regulace dosáhneme pouze zmenšení prvotního překmitu. Toto zmenšení nám ale výrazně pomůže v regulaci překmitů následujících první překmit. Jak bude přechodová charakteristika vypadat po zapojení inteligentního regulátoru na soustavu, si ukážeme na obr. 13. Na obrázku můžeme vidět výrazné zmenšení prvotního překmitu, oproti soustavě, v které jsme jej neregulovali.



Obr. 13 - přechodová charakteristika s inteligentním regulátorem

Na obr. 13 vidíme průběhy teplot soustavy pro časové konstanty *případu 2*. Hodnoty jsou uvedeny v tabulka 3. Jdou vidět dva průběhy teplot, světlemodrý a tmavě modrý. Světlemodrý nám udává průběh teploty bez regulace prvotního překmitu a tmavě modrá barva ukazuje průběh teploty s jeho regulací. Výrazné zmenšení velikosti prvotního překmitu má za následek zlepšení regulace následujících kmitů. Nedochozí k velkému podkmitnutí horní hrany hystereze, tak jak to bylo u soustavy s neregulovaným prvním kmitem. Adaptibilní vypínací úroveň prvotního překmitu je na obrázku znázorněna růžovou barvou. Její hodnota je zhruba 19,75 °C. Což značí výraznou změnu vypínací úrovně prvotního překmitu. Následná regulace soustavy probíhá principem popsáným v kapitole 4.2.

6.1.2 Popis způsobu regulace prvotního překmitu způsobem 2

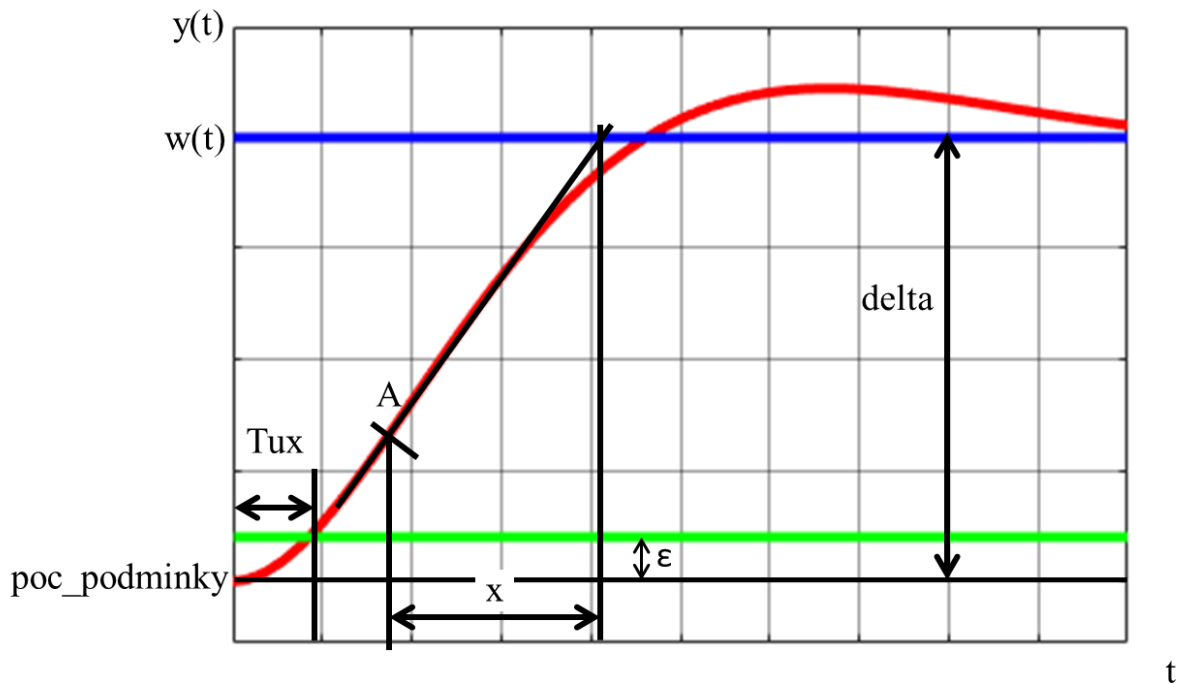
Tento způsob regulace byl založen na jiném principu zjišťování hodnoty Tux . Následné porovnávání Tux s hodnotou x zůstává stejné jako v prvním případě. Na rozdíl od prvního způsobu, kde se hodnota Tux s časem měnila, ji v tomto případě nastavujeme na pevno. Princip nastavení lze vidět na obr. 14. Hodnotu x získáváme stejným způsobem jako v případě 1. Velikost hodnoty Tux získáme v okamžiku, kdy nám regulovaná veličiny $y(t)$ protne předem nastavenou hranici (ϵ - zelená barva na obr. 14). Hranici, kterou má regulovaná veličina protnout získáme dle rovnice (14).

$$\text{delta} = w(t) - \text{poc_podminky} \quad (13)$$

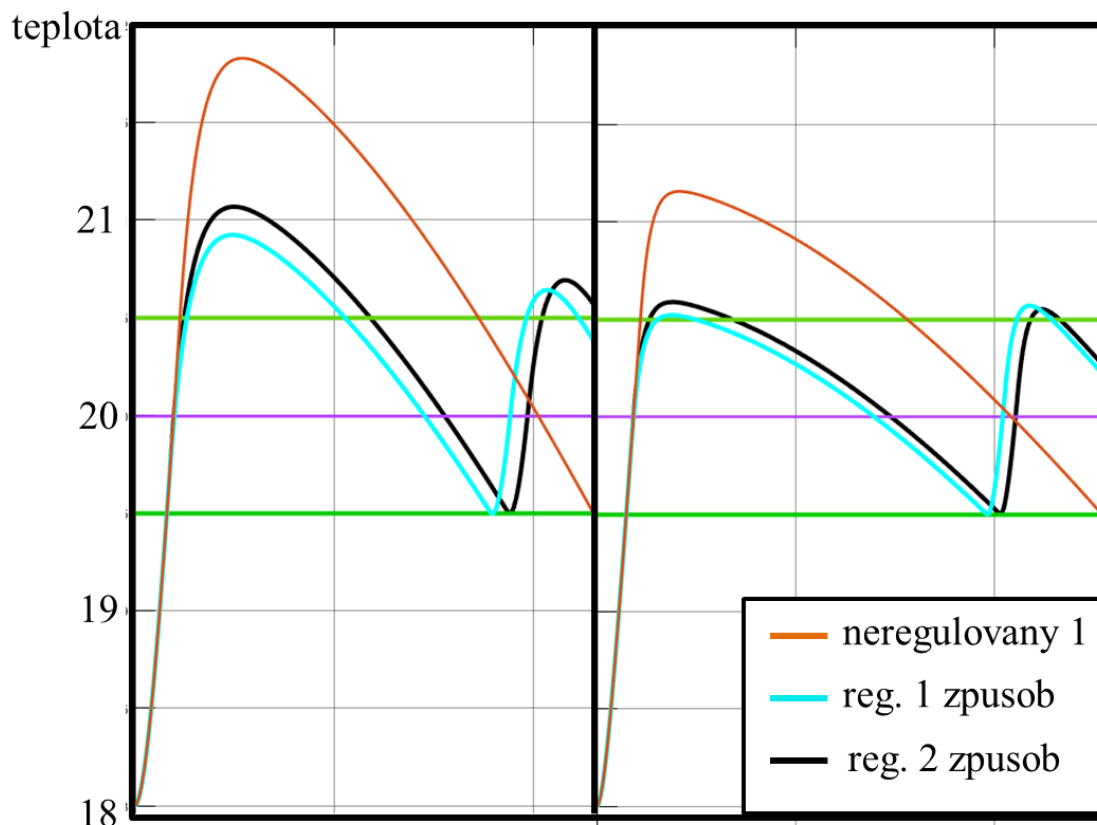
$$\varepsilon = \frac{\text{delta}}{10} \quad (14)$$

Regulaci s inteligentním regulátorem se získáváním parametrů oběma způsoby jsme znázornili na obr. 15. Vidíme porovnání regulace s oběma způsoby získávání parametru T_{ux} . Vidíme také srovnání mezi regulací prvotního překmitu a bez něj. Toto srovnání je uděláno pro dvě soustavy s rozdílnými časovými konstantami, časové konstanty soustav pro případ 1 a případ dva lze vyčíst z tabulka 3.

Regulace oběma způsoby jsme otestovali na reálných soustavách.



Obr. 14 - znázornění parametrů potřebných k regulaci prvotního překmitu dle druhého způsobu



Obr. 15 - porovnání regulace způsobem 1 a 2 pro případ 1(vlevo) a 2(vpravo) z tabulky 3

Algoritmus pracující s parametry T_{ux} a x dle tohoto druhého způsobu je uvedený v příloze 6.

7 Reálný experiment

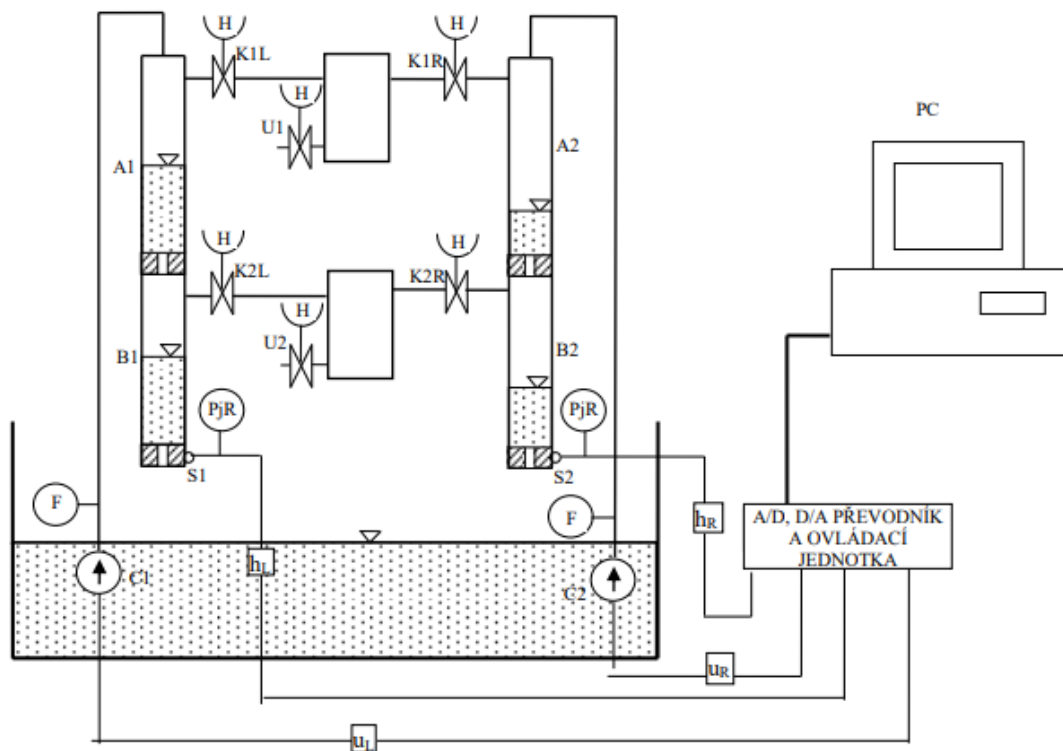
Reálný experiment byl prováděn v laboratoři pro *Přístrojovou a řídicí techniku* fakulty Strojní, Českého vysokého učení technického (místnost 111).

Pro zkoušení inteligentního regulátoru, jsme vzhledem k velikosti časových konstant neprováděli na místnosti jako takové, ale zvolili jsme soustavu s rychlejším průběhem přenosové funkce. V odborných časopisech se nám nabízí dvě analogie, které můžeme použít jako analogii k měření teploty v místnosti. Jednou variantou je elektrický RC obvod, v kterém zapojujeme přídavného kondenzátory a rezistory, dle toho jak vysoký řád diferenciální rovnice chceme získat. Tento modle, ale není vhodný, protože do něj nejsme schopni zakomponovat nesymetrickou dynamiku soustavy.

Další analogickou soustavu si můžeme představit jako kaskádové nádoby spojené za sebou. Počtem spojených nádob určíme výslednou velikost řádu soustavy. Vzhledem k výskytu podobné soustavy v laboratoři 111 jsme se rozhodli otestovat námi navrhovaný regulátor na této variantě.

7.1 Realizace experimentu na kaskádových nádobách

Schématický náčrt kaskádových nádob je zobrazen na obr. 16. Na obrázku vidíme čtyři válcové nádoby. Experiment byl prováděn na nádobách v levé části schématu, nádoby *A1* a *B1*. Výška hladiny v nádobě *B1* je snímána tlakovým snímačem v místě *S1*. Voda se do horní nádrže *A1* dostane pomocí čerpadla *C1*, které je umístěné v dolní nádrži s destilovanou vodou. Sériově spojené nádoby *A1* a *B1* tvoří soustavu druhého řádu. Hladina v nádrži *B1* je zpožděná oproti nádrži *A1*. Otevření ventilů *K2L* a *K1L* určuje, zda nad hladinami nádob bude barometrický tlak, či nikoli. Tyto ventily byly v průběhu experimentu zavřeny, proto nádoby můžeme považovat za sériové zapojení. Hodnota akčního zásahu čerpadla u_L , byla nadefinována tak, že je čerpadlo buď úplně vypnuté, nebo je zapnuté na jeho maximální výkon. To stimuluje zapnutí nebo vypnutí tepelné soustavy místnosti. Dále na obrázku vidíme ovládací jednotku s převodníkem, které sbírají signály ze soustavy, konkrétně hodnotu h_L , což je výstup ze soustavy, tedy regulovaná veličina. Do počítače se nahrává námi vytvořený algoritmus, který následně reguluje chování čerpadla *C1*.



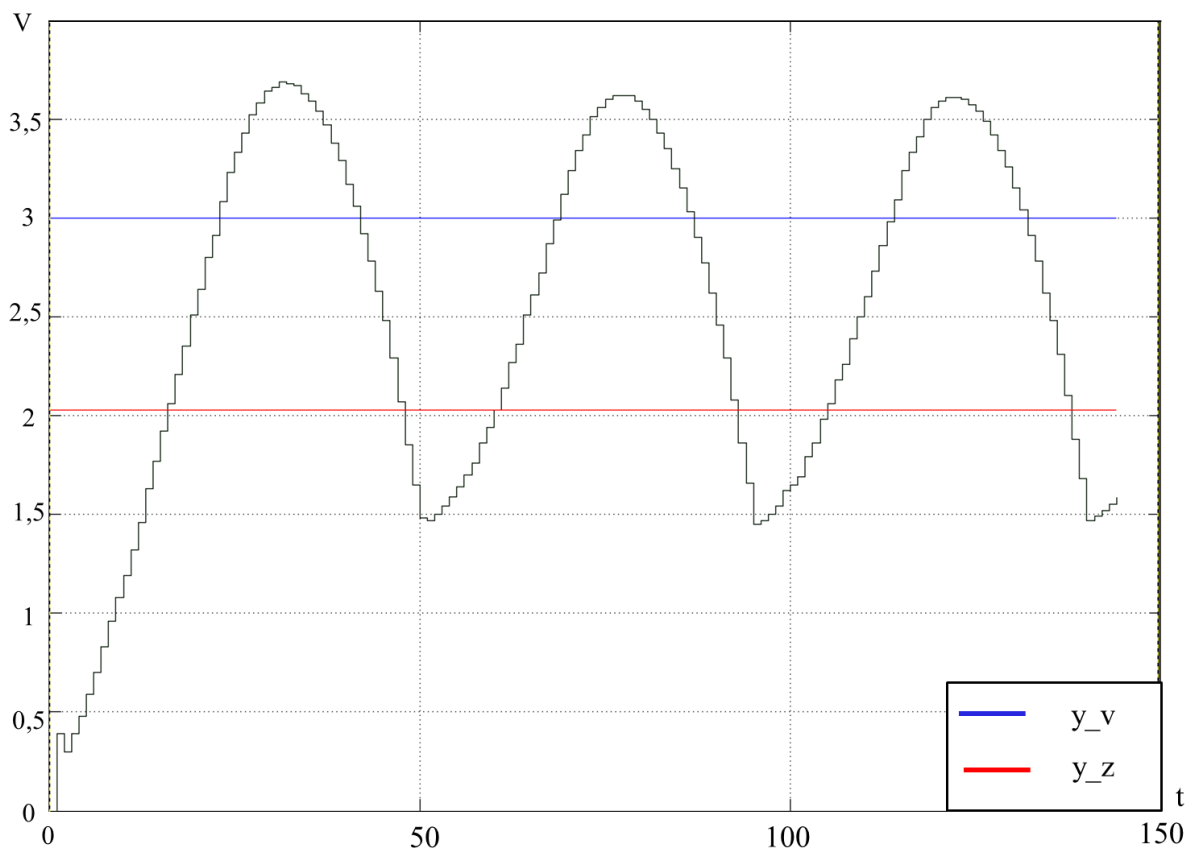
Obr. 16 - schéma kaskádových nádob (převzato z [10])

Na reálném modelu soustavy dle obr. 16, jsou ztráty realizované průtokem vody z nádrže B1 přes šterbinu v místě S1 do sběrné nádrže.

Na reálnou soustavu jsme postupně připojili tři varianty regulátorů. Obvyčné relé, regulátor regulující překmity následující první překmit a na regulátor s regulací prvotního překmitu.

7.2 Obyčejné relé

Na obr. 17 vidíme přechodovou charakteristiku soustavy dvou válcových nádob. Na soustavu je připojený jednoduchý relé přepínač. Svislá osa je udávána ve voltech a značí nám měřenou hladinu spodního válce tlakovým snímačem S1. Pásmo hystereze hladiny bylo nastaveno od 2,04 až do 3V. S tím, že vazba mezi volty a centimetry je zhruba 3 centimetry na jeden volt.



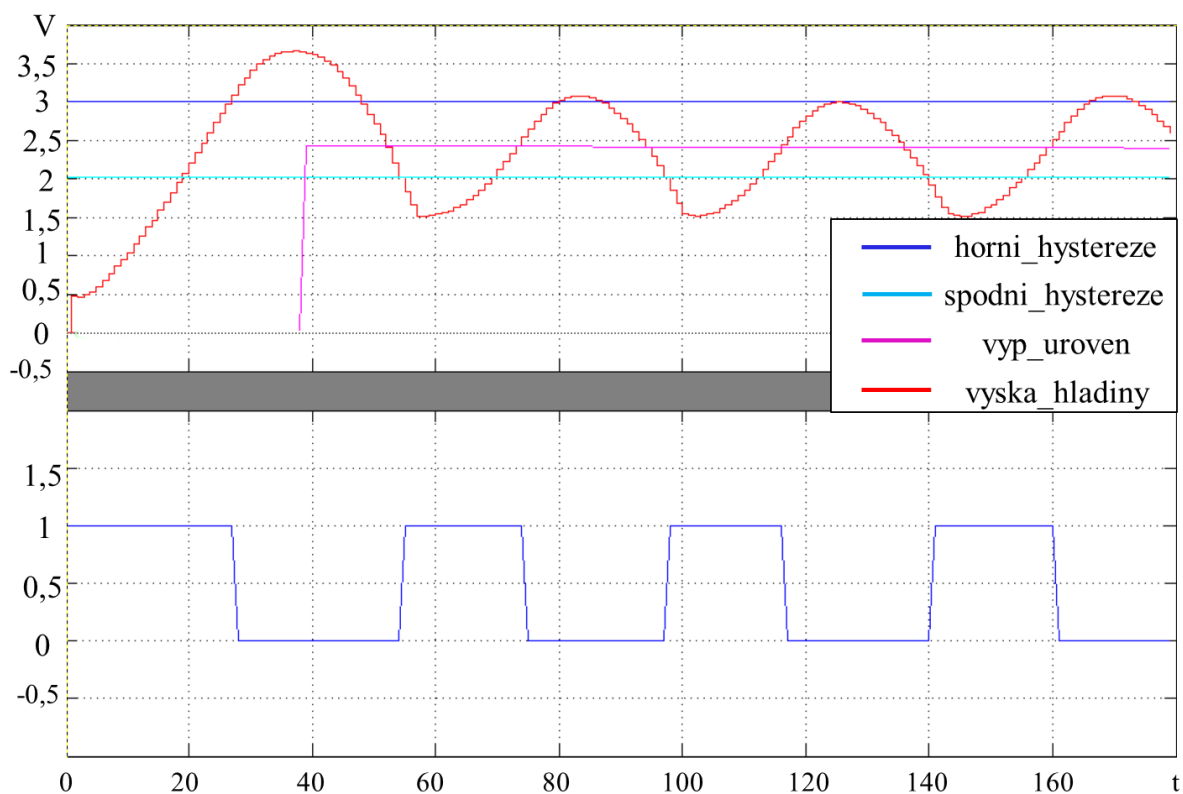
Obr. 17- reálná úloha s relé

Na obrázku můžeme vidět, že má soustava velké překmity nad i pod vypínací úroveň. To je způsobeno tím, že je soustava při zapnutí i při vypnutí čerpadla symetrická.

7.3 Regulace překmitů následující první překmit

Horní graf na obr. 18 zobrazuje průběh hladiny s regulací kmitů následujících první překmit. Vidíme, že hladina po adaptaci kmitá jenom těsně kolem horní hrany hystereze. K změně vypínací úrovně, která je uvedena růžovou barvou, dochází pouze nepatrně, protože na soustavu nepůsobí proměnlivé poruchové veličiny. Její úroveň po prvním překmitu je nastavena podle rovnice (7). A dále je prováděna její změna pomocí adaptivní regulace popsané rovnicí (6).

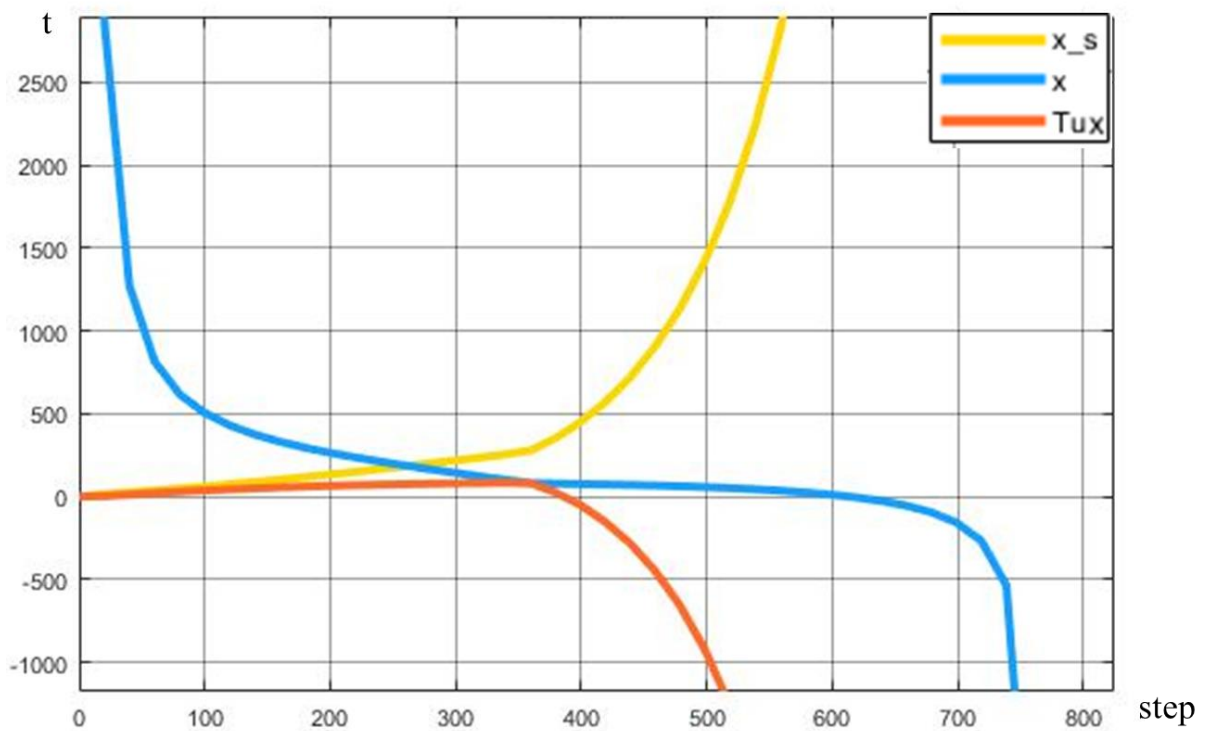
Na spodním grafu vidíme akční zásah čerpadla u . Je-li $u = 1$, čerpadlo pracuje na jeho maximální výkon a v případě, že je $u = 0$, čerpadlo nečerpá.



Obr. 18 - regulace překmitů následující první překmit

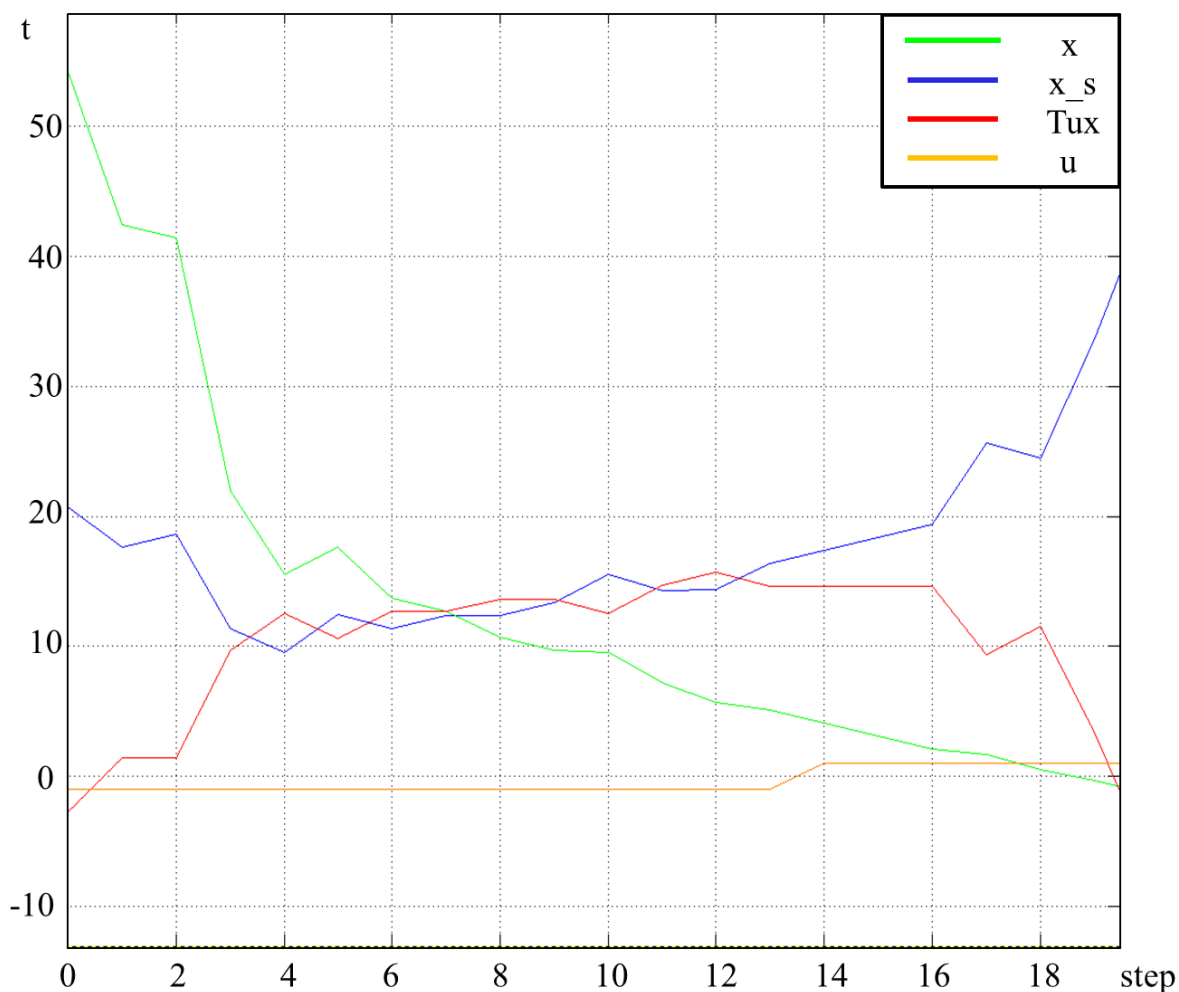
7.4 Soustava s inteligentním regulátorem (1 způsob)

Po připojení inteligentního regulátoru na reálnou soustavu neprobíhala regulace prvotního překmitu úplně stejným způsobem, jak tomu bylo v teoretické části. Při regulaci se vyskytly nepatrné chyby. V soustavě docházelo k častému přepínání akční hodnoty u , tedy se nám čerpadlo často zapínalo a vypínalo. Důvodem bylo špatné vyhodnocení podmínek, které určovaly, jak se bude soustava regulovat. Algoritmus řídící regulaci je řízen základními logickými příkazy *if, then, else*.



Obr. 19 - hodnoty x_s , x a Tux teoretické soustavy

V realitě jsme se setkali s problémem kmitání vstupů vstupujících do regulátoru. Na obr. 21 vidíme výňatek ze zdrojového kódu realizující část regulace. Význam hodnot x a Tux byly vysvětleny v kapitole 6.1.1. Hodnota $pTux$ se rovná hodnotě Tux předchozího kroku. Tato podmínka nám dělala potíže tím, že při simulaci na reálné úloze nebylo Tux v každém kroku vyšší než $pTux$, což by mělo platit, až do doby, než dojde k vypnutí čerpadla v rámci prvotního překmitu a regulátor přejde do fáze regulování kmitů následujících první překmit. Tuto skutečnost můžeme vidět, porovnáme-li přiložené obrázky (obr. 19 a obr. 20). Na obr. 19 vidíme, že parametry mají hladký průběh, proto nám podmínky nastavené v podmínce *if* na obr. 21 nedělají problém. Podíváme-li se na obr. 20 vidíme, že tyto podmínky neplatí vždy, vzhledem k tomu, jak nám signál kmitá.



Obr. 20 - hodnoty x_s , x a Tux reálné soustavy

Problém jsme vyřešili úpravou podmínky $Tux > pTux$ tak, že jsme z grafu hodnoty Tux zjistili, o jakou hodnotu zhruba $pTux$ podkmitává Tux . Poté jsme upravili podmínku do podoby $Tux > (pTux - podkmit)$. Tato podmínka samotné regulaci nevedí, protože funguje pouze jako pojistka. Tato realizace je možná i z toho důvodu, že při vypnutí čerpadla dojde vlivem dobíhání regulované veličiny do svého extrému, k zmenšení hodnoty Tux do záporných hodnot, a to je logicky menší než původní kladné hodnoty Tux .

```

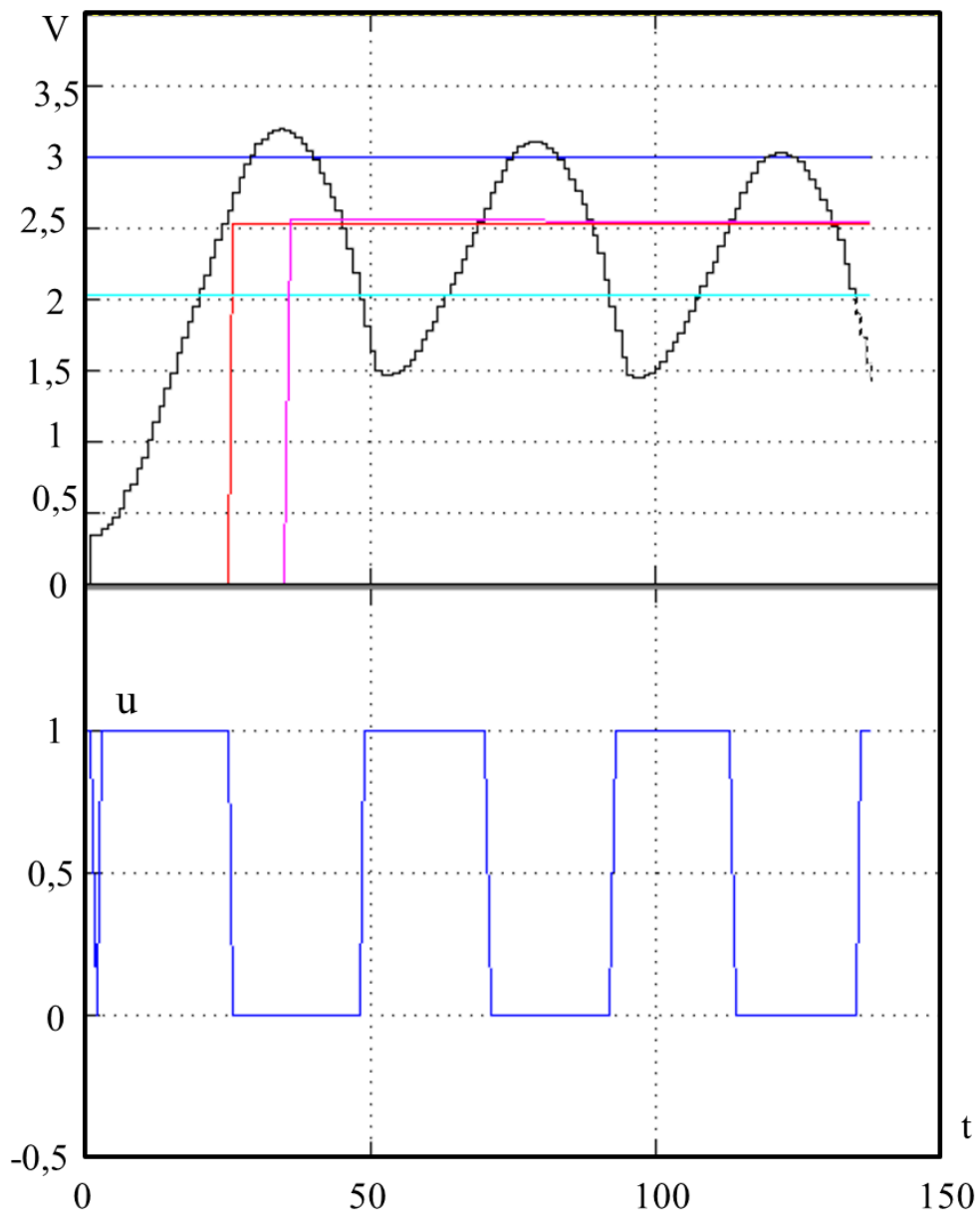
if x>Tux && Tux> pTux
    block.OutputPort(1).Data=-1;
    block.OutputPort(3).Data=teplota;
else
    block.OutputPort(1).Data=1;
    block.OutputPort(3).Data=pTeplota;
    block.OutputPort(5).Data=pTeplota;
end

```

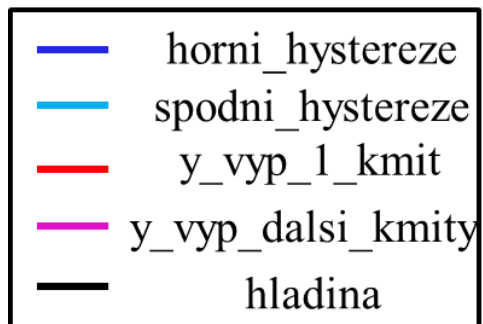
Obr. 21 - výňatek zdrojového kódu (přílohaX)

Další vážnější problém nastal při podmínce $x > Tux$. Po úpravě druhé podmínky nám začal regulátor regulovat správným způsobem, ale vzhledem k dynamice celého systému nám podmínka $x > Tux$ vypínala čerpání vody do nádrže ještě předtím, než hladina vody dosáhla spodní hranice hystereze. To způsobilo nesprávné fungování algoritmu realizujícího regulaci a regulátor se nikdy nedostal do stavu, že by adaptoval vypínací úroveň pro následující překmity. Opět jsme museli zakročit upravením této podmínky. Na grafu s vynesenyými hodnotami x a Tux jsme museli odhadem zjistit, o jakou velikost (v) ponížít hodnotu Tux tak, aby mohla regulace správně fungovat. Výsledná podmínka tedy vypadá $x > (Tux - v)$. S takto upravenými podmínkami regulace systému fungovala. Docílili jsme ponížení prvotního překmitu a dostali jsme se do fáze, kdy regulátor adaptabilně upravoval vypínací úroveň následujících kmitů.

Výslednou přechodovou charakteristiku s připojeným inteligentním regulátorem můžeme vidět na obr. 22. Černou barvou je znázorněná hladina spodní nádrže. Modrá a světlemodrá barva určují spodní a horní hranici hystereze. Červená barva značí, v jaké úrovni hladiny bylo vypnuté čerpadlo při regulování prvotního překmitu a růžovou barvou je znázorněná adaptabilní vypínací úroveň pro následující kmity. Na spodním grafu je modrou barvou vyznačena hodnota akční veličiny u , určující zda bylo čerpadlo zapnuto nebo vypnuto.



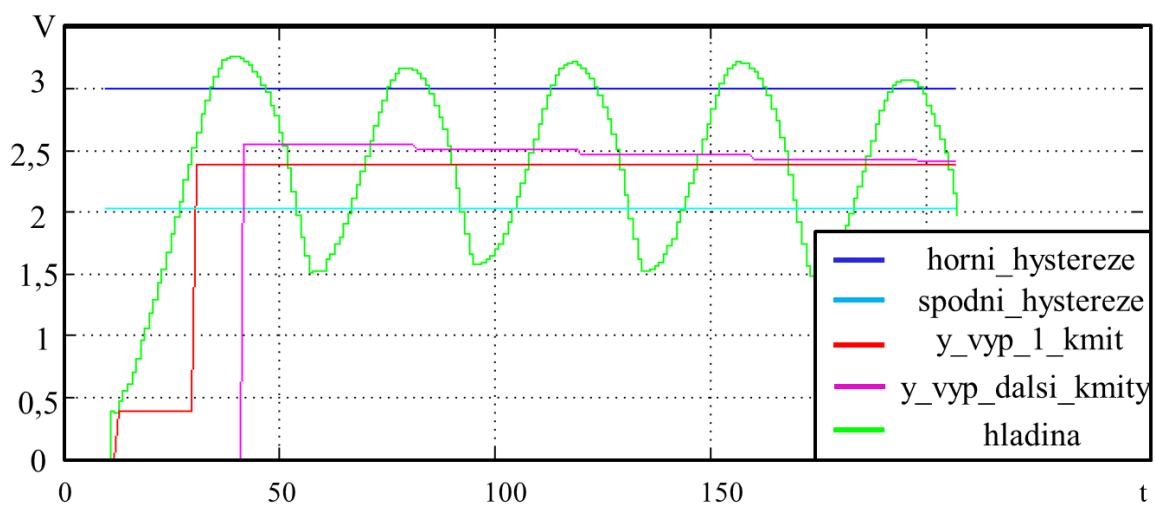
Obr. 22 - přechodová charakteristika s inteligentním regulátorem(1 způsob)



Obrázek 23 - legenda k obr. 22

7.5 Soustava s inteligentním regulátorem (2 způsob)

Nyní jsme testovali připojení inteligentního regulátoru s parametrem Tux získaným způsobem 2 uvedeném v kapitole 6.1.2. Při aplikaci regulátoru s tímto algoritmem se nám při regulaci nevyskytly chyby, protože zvolená hranice ϵ sama o sobě eliminovala šum signálu. Šum signálu nám při regulaci prvním způsob dělala největší potíže, kvůli kterým jsme museli zasahovat do zdrojového kódu algoritmu. Průběh regulace lze vidět na obr. 24. Při porovnání s obr. 18 vidíme, že došlo ke zmenšení prvotního překmitu. Následně se soustava dostala do fáze regulování překmitů následující první překmit, kde postupně dochází k regulaci vypínací úrovně.



Obr. 24 - přechodová charakteristika s inteligentním regulátorem(2 způsob)

8 Závěr

Výsledné simulace chování teploty teoretických soustav s napojeným inteligentním regulátorem jsou příznivé. Regulátor reguluje podle očekávání a dochází k zmenšení překmitů, způsobující energetické ztráty celé soustavy. Byl vymyšlen algoritmus s dvěma způsoby úpravy prvotního překmitnutí, což nebylo v regulaci dvoupolohovým regulátorem v tepelné technice budov dříve řešeno.

Algoritmus regulátoru s porovnáváním hodnot T_{ux} a x dle kapitoly 6.1.1 jsme připojili na reálnou soustavu. Bohužel jsme při tvorbě algoritmu na reálné soustavě nepočítali s problémy, které můžou u reálné soustavy nastat, proto u regulace na reálné soustavě nepracoval algoritmus bezchybně. Museli se udělat dodatečné úpravy algoritmu, aby mohla regulace správně probíhat. Při nastavování hodnoty T_{ux} a x dle druhého způsobu uvedeném v kapitole 6.1.2, je algoritmus jednodušší a použitelnější na reálných soustavách, protože se nastavení hodnoty T_{ux} zůstává neměnné. Funkci regulátoru s druhým způsobem regulace jsme také ověřili na reálné soustavě a výsledky jsou velmi příznivé. Porovnáme-li regulaci oběma způsoby, jeví se nám regulace druhým způsobem spolehlivější než prvním. Nemusíme zasahovat do zdrojového kódu a volba hranice ϵ sama osobě eliminuje vliv šumu signálu.

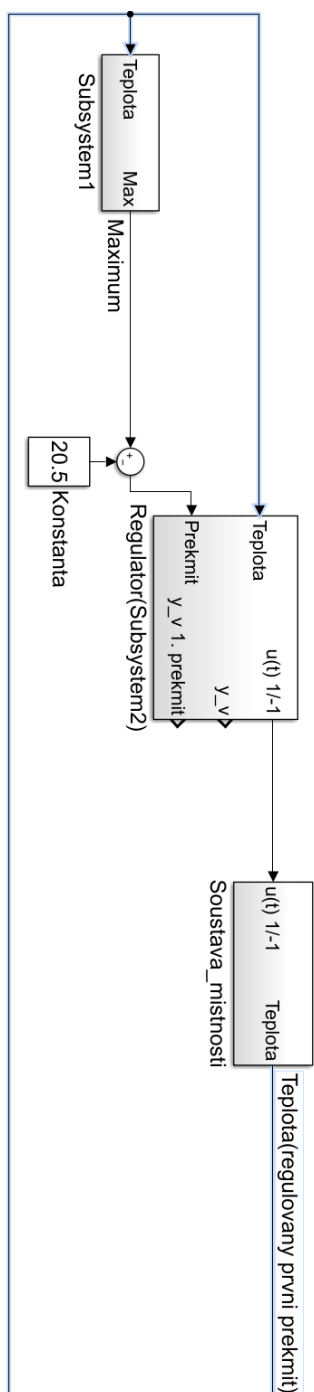
Nabízí se další vylepšování algoritmu. Základní logika regulace zůstává stejná, ale je potřeba připojení regulátoru na více reálných soustav a prozkoumat jeho chování ve více případech. S tak nabytými zkušenostmi bude programátor schopen zakomponovat všechny nástrahy reálného světa do algoritmu a tím vytvořit jeho finální podobu, možná taky vytvořit kombinaci obou uvedených způsobů regulace.

Za přínosy bakalářské práce považuji realizaci algoritmu pomocí S-funkce, navrhované způsoby regulace prvotního překmitu a jejich ověření na reálné soustavě. Nastavení směru pro další bádání při řešení problémů spojené s reálným světem.

9 Přílohy

9.1 Příloha 1

Celkový model soustavy



9.2 Příloha 2

Zdrojový kód nahrazující funkci regulátoru pro 1 způsob regulace uvedeném v 6.1.1, nahrán do *subsystému 2*

```
function nazev S-Funkce(block)

setup(block);

function setup(block)

    % Register number of ports
    block.NumInputPorts = 10;
    block.NumOutputPorts = 12;

    block.NumDialogPrms = 0;
    % Register sample times
    % [0 offset] : Continuous sample time

    block.SampleTimes = [0 0];

    block.SimStateCompliance = 'DefaultSimState';

    block.RegBlockMethod('Outputs', @Outputs); % Required
    block.RegBlockMethod('SetInputPortSamplingMode', @SetInpPortFrameData);
    block.RegBlockMethod('Terminate', @Terminate); % Required
    %end setup

function Outputs(block)
    teplota = block.InputPort(1).Data; % teplota soustavy
    dT = block.InputPort(2).Data; % derivace teploty
    prekmit = block.InputPort(3).Data; % prekmit soustavy nad horni mez
    hystereze
    p_vyp=block.InputPort(4).Data; % znarornuje predchozi vypinaci
    uroven soustavy
    pTeplota=block.InputPort(5).Data; % predchozi teplota, porovnanim
    predchozi teploty a stavajici teploty soustavy se urci extrem systemu
    time=block.InputPort(6).Data; % cas iniciace soustavy
    pdelta=block.InputPort(8).Data; % nastavuje velikost predchoziho
    prekmitu soustavy nad dovolenou uroven, prepisuje se kazdym dalsim vrcholem
    pTux=block.InputPort(10).Data; % hodnota Tu v predchozim kroku

    if prekmit <-1 % nutna podminka pro spusteni
    soustavy

        x=(20.5-teplota)/(dT);
        x_s = (teplota-18)/dT;
        Tux = time-x_s;
        block.OutputPort(7).Data=x_s;
        block.OutputPort(9).Data=x;
        block.OutputPort(10).Data=Tux;
        if dT == 0
            block.OutputPort(1).Data=-1;
        else
            if x>Tux && Tux> pTux % podminky realizujici
            regulaci prvnioho prekmitu
                block.OutputPort(1).Data=-1;
```

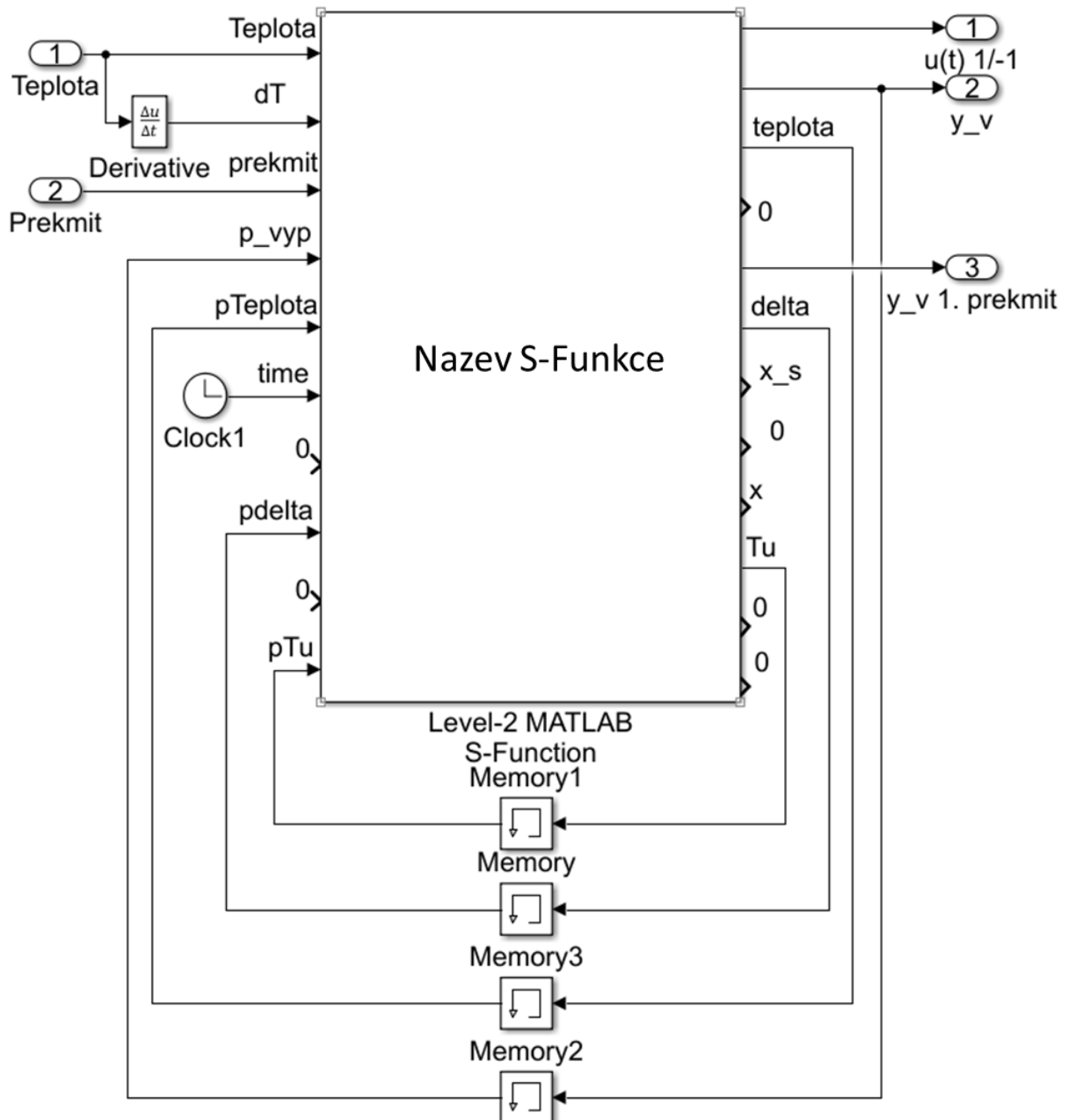
```

        block.OutputPort(3).Data=teplota;
    else
        block.OutputPort(1).Data=1;
        block.OutputPort(3).Data=pTeplota;
        block.OutputPort(5).Data=pTeplota;
    end
end
else %podminky realizujici
regulaci dalsich prekmitu
    delta=prekmit;
    block.OutputPort(3).Data=teplota;
    block.OutputPort(6).Data=delta;
    if p_vyp ==0
        p_vyp = 20.15;
    end
    if delta<pdelta||delta>pdelta
        y=p_vyp-delta/5;
        if teplota <(y)  && dT>=0
            block.OutputPort(1).Data=-1;
            block.OutputPort(2).Data=y;
        elseif teplota>(y)
            block.OutputPort(1).Data = 1;
            block.OutputPort(2).Data=y;
        elseif teplota>19.5 && dT<0
            block.OutputPort(1).Data =1;
            block.OutputPort(2).Data=y;
        else
            block.OutputPort(1).Data=-1;
            block.OutputPort(2).Data=y;
        end
    else
        y = p_vyp;
        if teplota <(y)  && dT>=0
            block.OutputPort(1).Data=-1;
            block.OutputPort(2).Data=y;
        elseif teplota>(y)
            block.OutputPort(1).Data = 1;
            block.OutputPort(2).Data=y;
        elseif teplota>19.5 && dT<0
            block.OutputPort(1).Data =1;
            block.OutputPort(2).Data=y;
        else
            block.OutputPort(1).Data=-1;
            block.OutputPort(2).Data=y;
        end
    end
end
end
function SetInpPortFrameData(block, idx, fd)
    block.InputPort(idx).SamplingMode = fd;
    for i =1:block.NumOutputPorts
        block.OutputPort(i).SamplingMode = fd;
    end
end
function Terminate(block)

```


9.3 Příloha 3

Model subsystému 2



9.4 Příloha 4

Zdrojový kód hledající maxima každého překmitu

```
function Nazev S-Funkce(block)

setup(block);

function setup(block)

block.NumInputPorts = 4;
block.NumOutputPorts = 3;

block.NumDialogPrms = 0;

block.SampleTimes = [0 0];

block.SimStateCompliance = 'DefaultSimState';

block.RegBlockMethod('Outputs', @Outputs); % Required
block.RegBlockMethod('SetInputPortSamplingMode', @SetInpPortFrameData);
block.RegBlockMethod('Terminate', @Terminate); % Required
function Outputs(block)

teplota = block.InputPort(1).Data;
pTeplota = block.InputPort(2).Data;
pdT = block.InputPort(4).Data;
dT = block.InputPort(3).Data;

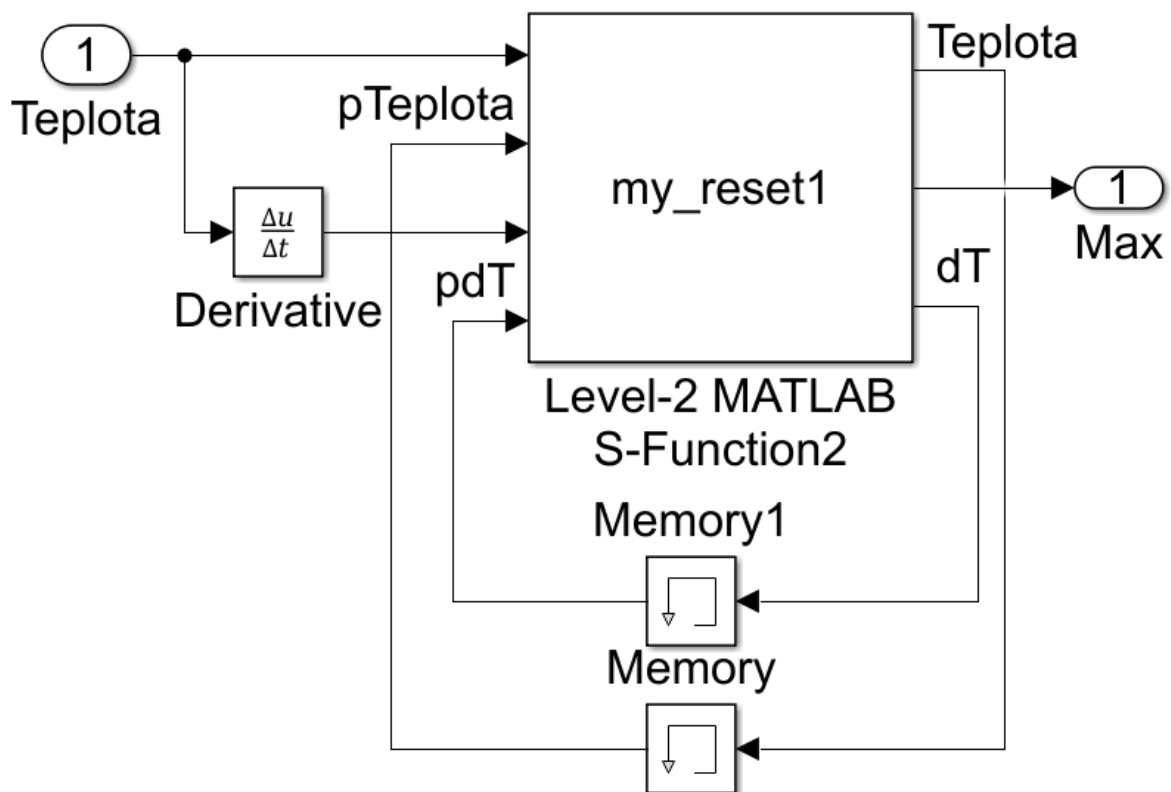
%nize uvedene podminky vyvadi vystupem 2. maximalni hodnotu prekmitu
if dT<pdT
    if teplota > pTeplota
        block.OutputPort(1).Data = teplota;
        block.OutputPort(2).Data = teplota;
    elseif teplota<pTeplota
        block.OutputPort(2).Data = pTeplota;
    end
else
    block.OutputPort(1).Data =teplota;
end

function SetInpPortFrameData(block, idx, fd)
    block.InputPort(idx).SamplingMode = fd;
    for i =1:block.NumOutputPorts
        block.OutputPort(i).SamplingMode = fd;
    end

function Terminate(block)
```

9.5 Příloha 5

Model *subsystému 1*



9.6 Příloha 6

Zdrojový kód nahrazující funkci regulátoru pro 2 způsob regulace uvedeném v 6.1.2, nahrán do *subsystému 2*

```
function nazev S-Funkce(block)

setup(block);

function setup(block)

    % Register number of ports
    block.NumInputPorts = 10;
    block.NumOutputPorts = 12;

    block.NumDialogPrms = 0;
    % Register sample times
    % [0 offset] : Continuous sample time

    block.SampleTimes = [0 0];

    block.SimStateCompliance = 'DefaultSimState';

    block.RegBlockMethod('Outputs', @Outputs); % Required
    block.RegBlockMethod('SetInputPortSamplingMode', @SetInpPortFrameData);
    block.RegBlockMethod('Terminate', @Terminate); % Required
    %end setup

function Outputs(block)
teplota = block.InputPort(1).Data; % teplota soustavy
dT = block.InputPort(2).Data; % derivace teploty
prekmit = block.InputPort(3).Data; % prekmit soustavy nad horni mez
hystereze
p_vyp=block.InputPort(4).Data; % znarornuje predchozi vypinaci
uroven soustavy
pTeplota=block.InputPort(5).Data; % predchozi teplota, porovnanim
predchozi teploty a stavajici teploty soustavy se urci extrem systemu
time=block.InputPort(6).Data; % cas iniciace soustavy
ptime=block.InputPort(7).Data;
pdelta=block.InputPort(8).Data; % nastavuje velikost predchoziho
prekmitu soustavy nad dovolenou uroven, prepisuje se kazdym dalsim vrcholem

if prekmit <-1 % nutna podminka pro spusteni
soustavy

    x=(20.5-teplota)/(dT);
    if teplota < 18.2
        block.OutputPort(4).Data=time;
    else
        block.OutputPort(4).Data=ptime;
    end
    Tux=ptime;
    block.OutputPort(10).Data=Tux;

if dT == 0
```

```

        block.OutputPort(1).Data=-1;
    else
        if x>Tux % podminky realizujici regulaci
prvniho prekmitu
            block.OutputPort(1).Data=-1;
            block.OutputPort(3).Data=teplota;
        else
            block.OutputPort(1).Data=1;
            block.OutputPort(3).Data=pTeplota;
            block.OutputPort(5).Data=pTeplota;
        end
    end
else %podminky realizujici
regulaci dalsich prekmitu
    delta=prekmit;
    block.OutputPort(3).Data=teplota;
    block.OutputPort(6).Data=delta;
    if p_vyp ==0
        p_vyp = 20.15;
    end
    if delta<pdelta||delta>pdelta
        y=p_vyp-delta/5;
        if teplota <(y) && dT>=0
            block.OutputPort(1).Data=-1;
            block.OutputPort(2).Data=y;
        elseif teplota>(y)
            block.OutputPort(1).Data = 1;
            block.OutputPort(2).Data=y;
        elseif teplota>19.5 && dT<0
            block.OutputPort(1).Data =1;
            block.OutputPort(2).Data=y;
        else
            block.OutputPort(1).Data=-1;
            block.OutputPort(2).Data=y;
        end
    else
        y = p_vyp;
        if teplota <(y) && dT>=0
            block.OutputPort(1).Data=-1;
            block.OutputPort(2).Data=y;
        elseif teplota>(y)
            block.OutputPort(1).Data = 1;
            block.OutputPort(2).Data=y;
        elseif teplota>19.5 && dT<0
            block.OutputPort(1).Data =1;
            block.OutputPort(2).Data=y;
        else
            block.OutputPort(1).Data=-1;
            block.OutputPort(2).Data=y;
        end
    end
end
end
function SetInpPortFrameData(block, idx, fd)
    block.InputPort(idx).SamplingMode = fd;
    for i =1:block.NumOutputPorts
        block.OutputPort(i).SamplingMode = fd;
    end
end
function Terminate(block)

```

10 Zdroje

[1] HLAVA, Jaroslav. *Nespojité (dvou- a třípolohové) regulatory*. TECHNICKÁ UNIVERZITA V LIBERCI. Fakulta mechatroniky, informatiky a mezioborových studií[online]. Dostupné: www.fm.tul.cz/esf0247/index.php?download=1030

[2] HOFREITER, Milan. *Základy automatického řízení*. Česká technika v Praze v roce 2012. ISBN:978-80-01-05007-1

[3] JENČÍK, Josef, VOLF Jaromír a kolektiv. *Technická měření*. Vydavatelství ČVUT – 2000 v Praze. ISBN: 9788001021385

[4] KOPAL, Martim. *Pomůcka pro názornou výuku regulace*. Praha, 2011. Diplomová práce. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ. Fakulta elektrotechnická

[5] NOVOTNÝ, Petr. *Základy řízení energetických strojů*. ZÁPADOČESKÁ UNIVERZITA V PLZNI. Fakulta strojní, Katedra energetických strojů a zařízení[online]. Dostupné: https://kke.zcu.cz/export/sites/kke/about/projekty/enazp/projekty/19_Automatizace_38-40/38_IUT/123_Zaklady-rizeni-energetickych-stroju-2---Novotny---P2.pdf

[6] STRAKA, Miroslav. *Hybridní systémy pro regulaci tepelných soustav*. Praha, 2004. Diplomová práce. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ. Fakulta dopravní

[7] ŠMEJKAL, Ladislav, ČERNÝ, Josef, KOVÁŘ, Josef. *Inspiromat pro výuku a Tecomat: co v učebnici automatizace nebylo*. AUTOMA, ročník 24 číslo 5 - 2018. Str. 16 – 19. ISSN:1210-9592

[8] ŠMEJKAL, Ladislav, MIROSLAV, Straka. *Tradiční teorie regulace versus nelineární soustavy*. 7th International Scientific - Technical Conference - PROCESS CONTROL 2006 June 13 – 16, 2006, Kouty nad Desnou, Czech Republic.

[9] *TecoInfo, Informační bulletin pro uživatele systému firmy Teco a.s.*. Teco, a.s., Kolektiv autorů pod vedením Jaromíra Klabana, Zář 2016, 38.

Dostupné:

<https://www.tecomat.cz/modules/DownloadManager/download.php?alias=tecoinfo-38-cz>

[10] *Creating your own block with continuous states* [online]. The MathWorks, Inc. [June 25, 2012]. Dostupné: https://blogs.mathworks.com/simulink/2012/06/25/creating-your-own-block-with-continuous-states-matlab-s-function/?s_tid=srchtitle

[11] The MathWorks, Inc. *MATLAB&SIMULINK Developing S-Function*[online]. 2018.

Dostupné: https://www.mathworks.com/help/pdf_doc/simulink/sfunctions.pdf

[12] TICHÝ, Martin. *Regulace teploty v budovách*. Plzeň, 2012. Bakalářská práce. ZÁPADOČESKÁ UNIVERZITA V PLZNI. Fakulta aplikovaných věd

[13] TRNKA, Pavel. *Frekvenční charakteristika hydraulicko-pneumatické soustavy* [online]. Dostupné: <http://vlab.fs.cvut.cz/navody/files/F4b.pdf>