

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**Fakulta strojní – Ústav přístrojové a řídicí techniky**



**DIPLOMOVÁ PRÁCE**

**Anisochronní řízení využitím regulačního  
systému Tecomat Foxtrot**

**Martin Hunčovský**

**2016**

**Vysoká škola: ČVUT v Praze**

**Pracoviště: Ústav přístrojové a řídicí techniky**

**Fakulta: strojní**

**Akademický rok: 2015/2016**

## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

*pro:*

**Bc. Martina HUNČOVSKÉHO**

*obor:* Přístrojová a řídicí technika

*Název tématu:* **Anisochronní řízení využitím regulačního systému Tecomat Foxtrot**

*Název anglicky:* Anisochronic Control by Control System Tecomat Foxtrot

### *Zásady pro zpracování*

- a) Seznamte se s metodou "požadovaného modelu" a s regulačním systémem Tecomat Foxtrot.
- b) Pro řízení soustavy popsané daným anisochronním modelem vytvořte programový modul pro systémy Tecomat s využitím metody "požadovaného modelu".
- c) Na simulačních příkladech a zvolené reálné soustavě ověřte vytvořené řízení a porovnejte s řízením PID regulátorem seřazeného dle metody Ziegler&Nichols.

**Rozsah grafických prací:** dle pokynů vedoucího DP

**Rozsah průvodní zprávy:** min. 50 stran + přílohy

*Seznam odborné literatury:*

[1] Vítěček, A.: – Vítěčková, M.: Zpětnovazební řízení mechatronických systémů. 1. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2013.

[2] Åström, K. J. – Wittenmark, B.: Adaptive Control. 2nd ed., Addison-Wesley Publishing Company, 1995.

[3] Zítek, P. – Vítěček, A.: Návrh řízení podsystémů se zpožděními a nelinearitami. Vydavatelství ČVUT, Praha, 1999.

[4] Příručky SW. Teco Advanced Automation: Industrial Automation, Intelligent Buildings, Smart Grid [online]. Kolín: Teco, 2009 [cit. 2016-03-23]. Dostupné z: <http://www.tecomat.com/index.php?a=cat.303>

*Vedoucí diplomové práce:* prof. Ing. M. Hofreiter, CSc.

*Datum zadání diplomové práce:* 12. 4. 2016

*Termín odevzdání diplomové práce:* 15. 6. 2016

*J. Chyský*

doc. Ing. Jan Chyský, CSc.  
vedoucí ústavu



*M. Valášek*

prof. Ing. Michael Valášek, DrSc.  
děkan fakulty

**V Praze dne: 5. 4. 2016**

## Prohlášení

*Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně s tím, že její výsledky mohou být dále použity podle uvážení vedoucího diplomové práce jako jejího spoluautora. Souhlasím také s případnou publikací výsledků diplomové práce nebo její podstatné části, pokud budu uveden jako její spoluautor.*

Dne .....

Podpis .....

## Poděkování

Rád bych touto cestou vyjádřil poděkování vedoucímu své diplomové práce, prof. Ing. Milanu Hofreiterovi, CSc. za jeho připomínky a čas, který mi věnoval. Děkuji také Ing. Pavlu Trnkovi za jeho ochotu a technickou podporu v laboratoři 111.

Název práce: Anisochronní řízení využitím regulačního systému Tecomat Foxtrot

## Abstrakt

Tato práce se zabývá návrhem anisochronního řízení pomocí metody „požadovaného modelu“ a vytvořením programového modulu anisochronního regulátoru pro systém Tecomat Foxtrot. Dále práce porovnává řízení navrženým anisochronním regulátorem s řízením PID regulátorem seřazeným metodou Ziegler-Nichols na simulačních příkladech a na reálné úloze. V této práci navržený anisochronní regulátor se ukázal v porovnání s PID regulátorem seřazeným metodou Ziegler-Nichols na simulačních příkladech v programu Matlab Simulink jako lepší, s kratší dobou regulace a menším relativním překmitem. V porovnání na vybrané reálné soustavě měla přechodová charakteristika uzavřeného regulačního obvodu s anisochronním regulátorem menší relativní překmit regulované veličiny než s PID regulátorem, doba regulace byla u obou regulátorů přibližně stejná. Při porovnání reakce na skokovou změnu poruchové veličiny se ukázalo řízení navrženým anisochronním regulátorem jako horší jak v simulačních příkladech, tak na reálné soustavě.

**Klíčová slova:** anisochronní řízení, PID regulátor, Ziegler-Nichols, metoda „požadovaného modelu“, Tecomat Foxtrot, PLC

Title: Anisochronic Control by Control System Tecomat Foxtrot

## Abstract

This paper describes design of an anisochronic controller using the Desired Model Method and design of a program module for anisochronic control for Tecomat Foxtrot control system. This paper also compares the proposed anisochronic controller with a PID controller (tuned using the Ziegler Nichols method) on simulation examples and a real system. The proposed anisochronic controller showed, compared with the PID controller (on simulation examples in Matlab Simulink), better regulation time and smaller relative overshoot. Compared on the real system, the regulation time was almost the same for both controllers, but the step response of the closed control loop with anisochronic controller had smaller relative overshoot than the loop with the PID controller. The disturbance rejection of the anisochronic controller, showed by both simulation and experiment, was worse than the disturbance rejection of the PID controller tuned using the Ziegler-Nichols method.

**Keywords:** Anisochronic Control, PID Controller, Ziegler-Nichols, Desired Model Method, Tecomat Foxtrot, PLC

# Obsah

<b>1.</b>	<b>ÚVOD</b> .....	<b>1</b>
<b>2.</b>	<b>POUŽITÉ METODY A SOFTWARE</b> .....	<b>2</b>
2.1.	METODA POŽADOVANÉHO MODELU .....	2
2.2.	APROXIMACE INTEGRACE LICHOBĚŽNÍKOVOU NÁHRADOU (TUSTINOVA METODA).....	4
2.3.	METODA VOLBY PERIODY VZORKOVÁNÍ.....	5
2.4.	METODA NASTAVENÍ PID REGULÁTORU .....	6
2.4.1.	<i>Ziegler-Nichols</i> .....	6
2.4.2.	<i>Åström-Hägglund</i> .....	6
2.5.	PROGRAMOVÁNÍ PODLE NORMY IEC 61131-3.....	7
2.5.1.	<i>Typy dat</i> .....	7
2.5.2.	<i>Proměnné</i> .....	7
2.5.3.	<i>Konfigurace, zdroje a úlohy</i> .....	8
2.5.4.	<i>Programové organizační jednotky</i> .....	8
2.5.5.	<i>Programovací jazyky</i> .....	9
2.6.	PROGRAMOVACÍ PROSTŘEDÍ MOSAIC .....	10
2.6.1.	<i>Dodávka programu</i> .....	11
2.6.2.	<i>Programování PLC TECOMAT, FOXTROT, TECOREG</i> .....	11
<b>3.</b>	<b>SYSTÉM TECOMAT FOXTROT</b> .....	<b>13</b>
3.1.	ZÁKLADNÍ PARAMETRY SYSTÉMU TECOMAT FOXTROT.....	13
3.2.	ZÁKLADNÍ MODUL TECOMAT FOXTROT CP-1015 .....	14
<b>4.</b>	<b>NÁVRH ANISOCHRONNÍHO REGULÁTORU</b> .....	<b>16</b>
4.1.	NÁVRH REGULÁTORU METODOU POŽADOVANÉHO MODELU.....	16
4.2.	DISKRETIZACE PŘENOSU REGULÁTORU .....	17
4.3.	VOLBA PERIODY VZORKOVÁNÍ .....	19
4.4.	VOLBA ČASOVÉ KONSTANTY FILTRU .....	20
4.5.	VLIV PERIODY VZORKOVÁNÍ A ČASOVÉ KONSTANTY FILTRU NA PRŮBĚH REGULOVANÉ VELIČINY .....	20
4.6.	DALŠÍ ÚPRAVA NAVRHOVANÉHO REGULÁTORU .....	24
<b>5.</b>	<b>PROGRAMOVÝ MODUL PRO PLC TECOMAT FOXTROT</b> .....	<b>26</b>
5.1.	PRVNÍ VERZE REGULÁTORU.....	26
5.1.1.	<i>Funkční blok regulátoru</i> .....	26
5.1.2.	<i>Volání funkčního bloku regulátoru z hlavního programu</i> .....	30
5.2.	DRUHÁ VERZE REGULÁTORU .....	30
5.2.1.	<i>Funkční blok regulátoru</i> .....	31
5.2.2.	<i>Volání funkčního bloku regulátoru z hlavního programu</i> .....	33
5.3.	OVLÁDÁNÍ - WEBMAKER.....	33



5.4.	TEST NAVRŽENÉHO REGULÁTORU NA REÁLNÉ SOUSTAVĚ .....	34
<b>6.</b>	<b>SROVNÁNÍ ANISOCHRONNÍHO ŘÍZENÍ A ŘÍZENÍ S PID REGULÁTOREM .....</b>	<b>35</b>
6.1.	POUŽITÝ PID REGULÁTOR .....	35
6.2.	TEPLOVZDUŠNÝ MODEL (TVM) .....	36
6.3.	SIMULACE – MATLAB SIMULINK .....	39
6.3.1.	<i>Simulace na soustavě šestého řádu .....</i>	<i>39</i>
6.3.2.	<i>Simulace na anisochronním modelu soustavy TVM.....</i>	<i>44</i>
6.4.	SROVNÁNÍ NA REÁLNÉ SOUSTAVĚ .....	46
6.4.1.	<i>Parametry PID regulátoru.....</i>	<i>46</i>
6.4.2.	<i>Parametry anisochronního regulátoru .....</i>	<i>48</i>
6.4.3.	<i>Srovnání obou regulátorů .....</i>	<i>48</i>
<b>7.</b>	<b>ZÁVĚR.....</b>	<b>53</b>
7.1.	ÚKOLY PRÁCE .....	53
7.2.	VÝHLED DO BUDOUCNA .....	54
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>56</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>57</b>
	<b>SEZNAM TABULEK.....</b>	<b>59</b>
	<b>SEZNAM ROVNIC .....</b>	<b>60</b>
	<b>SEZNAM ZKRATEK A SYMBOLŮ .....</b>	<b>62</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>64</b>

# 1. Úvod

Cílem této práce je navrhnout anisochronní řízení, realizovat ho na programovatelném automatu Tecomat Foxtrot, odzkoušet a následně porovnat s řízením pomocí klasického PID regulátoru seřizovaného metodou Ziegler & Nichols.

Přenosová funkce regulátoru je navrhována pro model soustavy s anisochronním přenosem (1.1).

$$G_a(s) = \frac{K \cdot e^{-s\tau_u}}{(\tau_1 s + 1)(\tau_2 s + e^{-s\tau_y})} \quad (1.1)$$

Anisochronní model soustavy byl zvolen, protože je velmi universální a vhodný pro modelování systémů s dopravním zpožděním [1]. Další výhodou anisochronního modelu je možnost odhadu jeho parametrů pomocí rozšířené reléové identifikace, popsané v [2] a [3].

Pro návrh přenosové funkce regulátoru byla zvolena metoda „požadovaného modelu“, protože je velmi jednoduchá a umožňuje nám pomocí jednoho z volených parametrů ovlivnit tvar výsledné přechodové charakteristiky uzavřeného regulačního obvodu.

Pro realizaci anisochronního řízení byl vybrán programovatelný automat Tecomat Foxtrot CP-1015 vyráběný českou firmou Teco. Tento automat se programuje pomocí programovacích jazyků normy IEC-61131-3, což případně umožní jednoduše přenést kód programu i na jiné průmyslové automaty.

## 2. Použité metody a software

### 2.1. Metoda požadovaného modelu

Metoda požadovaného modelu (MPM) – převzato z [4], dříve nazývaná také metoda inverze dynamiky, byla rozpracována na Fakultě strojní VŠB – Technické univerzitě Ostrava. Metoda požadovaného modelu vychází ze vztahu

$$G_C(s) = \frac{1}{G_P(s)} \frac{G_{wy}(s)}{1 - G_{wy}(s)} \quad (2.1)$$

pro přímou syntézu přenosu regulátoru, kde

$$G_P(s) = G'_P(s) \cdot e^{-s\tau_u} \quad (2.2)$$

je přenos soustavy,

$$G_{wy}(s) = \frac{k_0}{s + k_0 e^{-\tau_u s}} \cdot e^{-s\tau_u} \quad (2.3)$$

je požadovaný přenos řízení a  $k_0$  je zesílení otevřeného regulačního obvodu.

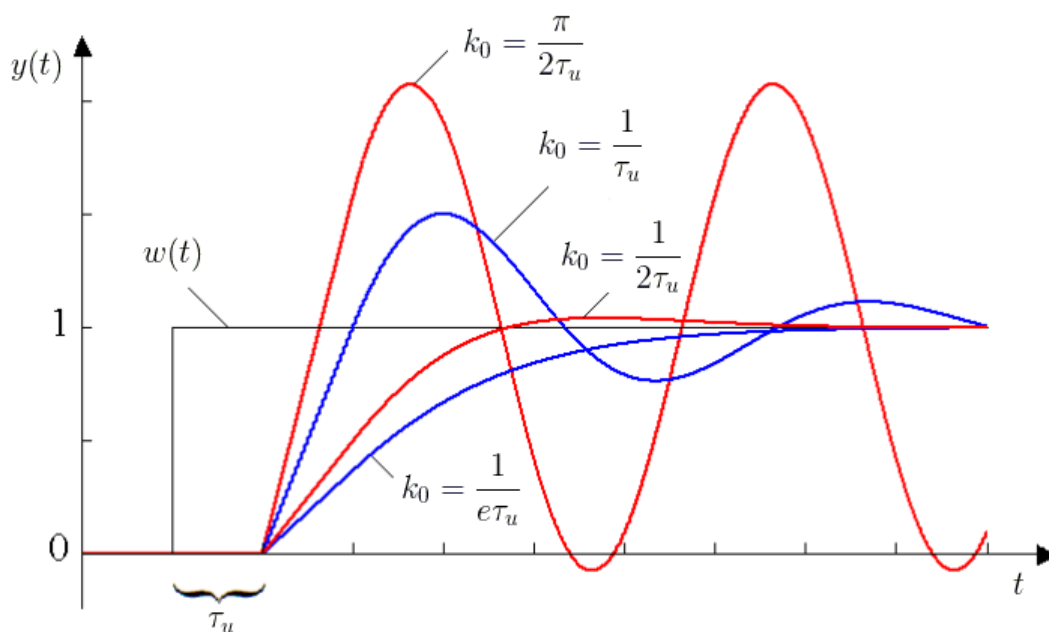
Požadovanému přenosu řízení (2.3) odpovídá velmi jednoduchý přenos otevřeného regulačního obvodu

$$G_0(s) = G_C(s)G_P(s) = \frac{k_0}{s} \cdot e^{-s\tau_u}. \quad (2.4)$$

Po dosazení (2.2) a (2.3) do (2.1) se dostane přenos navrhovaného regulátoru

$$G_C(s) = \frac{k_0}{sG'_P(s)}. \quad (2.5)$$

Výhoda požadovaného přenosu řízení (2.3) ve tvaru anisochronního matematického modelu spočívá v tom, že změnou zesílení otevřeného regulačního obvodu  $k_0$  lze snadno dosáhnout různého průběhu odezvy na skokovou změnu žádané veličiny  $w(t)$  od nekmitavého až po kmitavý s různým překmitem viz Obr. 2.1.



Obr. 2.1: Vliv zesílení otevřeného regulačního obvodu  $k_0$  na průběh přechodové charakteristiky regulačního obvodu [4]

Podle [4] lze zesílení otevřeného regulačního obvodu  $k_0$  vyjádřit ve tvaru

$$k_0 = \frac{1}{\beta\tau_u}, \quad (2.6)$$

kde  $\beta$  je koeficient závislý na průběhu přechodové charakteristiky regulačního obvodu (Obr. 2.1). Hodnoty koeficientu  $\beta$  lze volit podle požadovaného relativního překmitu  $\kappa$  podle tabulky 2.1, nebo podle vztahu

$$\beta(\kappa) = 2,718 - 0,4547\kappa^{0,3432}, \quad (2.7)$$

kde  $\kappa$  je relativní překmit v procentech.

Tabulka 2.1: Hodnoty koeficientu  $\beta$  pro požadovaný relativní překmit  $\kappa$  [4]

$\kappa$	0	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
$\beta$	2,718	1,944	1,720	1,561	1,437	1,337	1,248	1,172	1,104	1,045	0,992

## 2.2. Aproximace integrace lichoběžníkovou náhradou (Tustinova metoda)

Převzato z [5]. Necht'  $y$  značí výstup a  $x$  vstup integrátoru, tj.

$$y(t) = \int_0^t x(\tau) d\tau. \quad (2.8)$$

Pro  $t = k \cdot T$  pomocí lichoběžníkové náhrady vyjádříme

$$\begin{aligned} y(t) = y(kT) = y[k] &= \int_0^{(k-1)T} x(\tau) d\tau + \int_{(k-1)T}^{kT} x(\tau) d\tau \\ &\approx y[k-1] + \frac{x[k] + x[k-1]}{2} \cdot T \end{aligned} \quad (2.9)$$

a tedy

$$y[k] - y[k-1] \approx \frac{x[k] + x[k-1]}{2} \cdot T. \quad (2.10)$$

Využitím operátoru posunu  $q$  získáme

$$y[k] - q^{-1}y[k] \approx \frac{x[k] + q^{-1}x[k]}{2} \cdot T, \quad (2.11)$$

neboli

$$y[k] \approx \frac{1 + q^{-1}}{1 - q^{-1}} \cdot \frac{T}{2} \cdot x[k]. \quad (2.12)$$

Z (2.8) plyne pro  $t = k \cdot T$

$$\frac{dy(t)}{dt} = p \cdot y(t) = x(t) = x(k \cdot T). \quad (2.13)$$

Při zavedeném označení  $y(k \cdot T) = y[k]$  a  $x(k \cdot T) = x[k]$  porovnáním vztahů (7.2) a (2.12) získáme

$$p \approx \frac{1 + q^{-1}}{1 - q^{-1}} \cdot \frac{T}{2} = \frac{2}{T} \cdot \frac{q - 1}{q + 1}. \quad (2.14)$$

Stejný vztah jako mezi operátory  $p$  a  $q$  platí i pro argumenty Laplaceovy a Z-transformace tj. pro komplexní proměnné  $s$  a  $z$ , tj.

$$s \approx \frac{2}{T} \cdot \frac{z - 1}{z + 1} \quad (2.15)$$

Tustinovou aproximací ze stabilního spojitého přenosu získáme stabilní diskrétní přenos.

### 2.3. Metoda volby periody vzorkování

Převzato z [5]. Pro statické a nekmitavé systémy (včetně systémů s dopravním zpožděním), které lze vyjádřit přenosem

$$G_P(s) = \frac{K \cdot (\tau_{1N} \cdot s + 1) \cdot \dots \cdot (\tau_{mN} \cdot s + 1)}{(\tau_1 \cdot s + 1) \cdot \dots \cdot (\tau_n \cdot s + 1)} \cdot e^{-\tau_u \cdot s}, \quad (2.16)$$

lze z přechodové charakteristiky odhadnout souhrnnou časovou konstantu  $T_\Sigma$  určenou vztahem

$$\tau_\Sigma = \tau_1 + \tau_2 + \dots + \tau_n - \tau_{1N} - \tau_{2N} - \dots - \tau_{mN} + \tau_u \quad (2.17)$$

a periodu vzorkování  $T$  zvolit tak, aby bylo splněno

$$T < 0,1 \cdot \tau_\Sigma. \quad (2.18)$$

## 2.4. Metoda nastavení PID regulátoru

### 2.4.1. Ziegler-Nichols

Převzato z [4]. Zieglerova-Nicholsova metoda kritických parametrů (metoda uzavřeného regulačního obvodu) vychází ze skutečného regulačního obvodu, který se při vyřazené integrační činnosti ( $T_i \rightarrow \infty$ ) a derivační činnosti ( $T_d \rightarrow 0$ ) regulátoru zvyšováním jeho zesílení  $K_P$  přivede na kmitavou mez stability.

Pak se z periodického průběhu libovolné veličiny regulačního obvodu odečte kritická perioda  $T_k$  a z odpovídajícího nastavení analogového regulátoru – kritické zesílení  $r_k$ .

Hodnoty stavitelných parametrů zvoleného analogového regulátoru se vypočtou na základě tabulky:

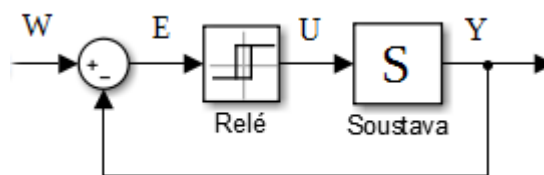
Tabulka 2.2: Stavitelné parametry analogových regulátorů pro Zieglerovu-Nicholsovu metodu kritických parametrů [4]

Regulátor	$r_0$	$T_i$	$T_d$
<i>P</i>	$0,50 \cdot r_k$	–	–
<i>PI</i>	$0,45 \cdot r_k$	$0,83 \cdot T_k$	–
<i>PID</i>	$0,60 \cdot r_k$	$0,50 \cdot T_k$	$0,125 \cdot T_k$

### 2.4.2. Åström-Hägglund

Pro zjištění kritického zesílení  $r_k$  a kritické periody kmitů  $T_k$  lze použít metoda Åströma a Hägglunda, která spočívá v připojení dvoupolohového regulátoru – relé k regulované soustavě a odečtení potřebných hodnot (amplitudy akčního zásahu  $u_a$ , amplitudy regulované veličiny  $y_a$  a kritické periody  $T_k$ ) z průběhů akčního zásahu a regulované veličiny. Z hodnot  $u_a$  a  $y_a$  pak lze vypočítat hodnotu kritického zesílení  $r_k$  podle vztahu

$$r_k = \frac{4 \cdot u_a}{\pi \cdot y_a} \quad (2.19)$$



Obr. 2.2: Schéma uzavřeného regulačního obvodu s relé

## 2.5. Programování podle normy IEC 61131-3

Norma IEC 61131-3 standardizuje programovací jazyky pro průmyslovou automatizaci. Specifikuje syntaxi a sémantiku unifikovaného souboru programovacích jazyků, včetně obecného softwarového modelu a strukturujícího jazyka. Tato norma byla přijata jako směrnice u většiny významných výrobců PLC. Převzato z [6].

### 2.5.1. Typy dat

Běžné datové typy jsou **BOOL**, **BYTE**, **WORD**, **INT** (Integer), **REAL**, **DATE**, **TIME**, **STRING** atd. Z těchto základních datových typů je pak možné odvozovat vlastní uživatelské datové typy, tzv. odvozené datové typy. Tímto způsobem můžeme např. definovat jako samostatný datový typ analogový vstupní kanál a opakovaně ho používat pod definovaným jménem.

### 2.5.2. Proměnné

Proměnné mohou být přiřazeny explicitně k hardwarovým adresám (např. vstupům, výstupům) pouze v konfiguracích, zdrojích nebo programech. Tímto způsobem je dosaženo vysokého stupně hardwarové nezávislosti a možnosti opakovaného využití softwaru na různých hardwarových platformách. Oblast působnosti proměnných je běžně omezena pouze na tu programovou organizační jednotku, ve které byly deklarovány (proměnné jsou v ní lokální). To znamená, že jejich jména mohou být používána v jiných částech bez omezení. Pokud mají mít proměnné globální působnost, např. v rámci celého projektu, pak musí být jako globální deklarovány (**VAR\_GLOBAL**). Aby bylo možné správně nastavit počáteční stav procesu nebo stroje, může být parametrům přiřazena počáteční hodnota při startu nebo studeném restartu.



### 2.5.3. Konfigurace, zdroje a úlohy

Na nejvyšší úrovni je celé softwarové řešení určitého problému řízení formulováno jako tzv. konfigurace (Configuration). Konfigurace je závislá na konkrétním řídicím systému, včetně uspořádání hardwaru, jako jsou například typy procesorových jednotek, paměťové oblasti přiřazené vstupním a výstupním kanálům a charakteristiky systémového programového vybavení (operačního systému).

V rámci konfigurace můžeme pak definovat jeden nebo více tzv. zdrojů (Resource). Na zdroj se můžeme dívat jako na nějaké zařízení, které je schopno vykonávat IEC programy.

Uvnitř zdroje můžeme definovat jednu nebo více tzv. úloh (Task). Úlohy řídí provádění souboru programů a/nebo funkčních bloků. Tyto jednotky mohou být prováděny buď periodicky, nebo po vzniku speciální spouštěcí události, což může být např. změna proměnné.

Programy jsou vystavěny z řady různých softwarových prvků, které jsou zapsány v některém z jazyků definovaném v normě. Často je program složen ze sítě funkcí a funkčních bloků, které jsou schopny si vyměňovat data. Funkce a funkční bloky jsou základní stavební kameny, které obsahují datové struktury a algoritmus.

### 2.5.4. Programové organizační jednotky

Funkce, funkční bloky a programy jsou v rámci normy IEC 61 131 nazývány společně programové organizační jednotky (Program Organization Units, zkratka POUs).

#### *Funkce*

IEC 61 131-3 definuje standardní funkce a uživatelem definované funkce. Standardní funkce je např. **ADD** pro sčítání. Jednou definované uživatelské funkce, mohou být používány opakovaně.

#### *Funkční bloky*

Na funkční bloky se můžeme dívat jako na integrované obvody, které reprezentují hardwarové řešení specializované řídicí funkce. Obsahují algoritmy i data,

takže mohou zachovávat informaci o minulosti. Mají jasně definované rozhraní a skryté vnitřní proměnné. Umožňují tím jednoznačně oddělit různé úrovně programátorů nebo obslužného personálu. Klasickými příklady funkčního bloku jsou např. regulační smyčka pro teplotu nebo PID regulátor.

Jakmile je jednou funkční blok definován, může být používán opakovaně v daném programu, nebo v jiném programu, nebo dokonce i v jiném projektu. Je tedy univerzální a mnohonásobně použitelný. Funkční bloky mohou být zapsány v libovolném z jazyků definovaném v normě. Mohou být tedy plně definovány uživatelem. Odvozené funkční bloky jsou založeny na standardních funkčních blocích, ale v rámci pravidel normy je možno vytvářet i zcela nové zákaznické funkční bloky.

Interface funkcí a funkčních bloků je popsán stejným způsobem: Mezi deklarací označující název bloku a deklarací pro konec bloku je uveden soupis deklarací vstupních proměnných, výstupních proměnných a vlastní kód v tzv. těle bloku.

## *Programy*

Na základě výše uvedených definic lze říci, že program je vlastně sítí funkcí a funkčních bloků. Program může být zapsán v libovolném z jazyků definovaných v normě.

### *2.5.5. Programovací jazyky*

V rámci standardu jsou definovány čtyři programovací jazyky. Dělí se do dvou základních kategorií:

#### a) Textové jazyky

- IL – Instruction List – jazyk seznamu instrukcí
- ST – Structured Text – jazyk strukturovaného textu

#### b) Grafické jazyky

- LD – Ladder Diagram – jazyk kontaktních schémat
- FBD – Function Block Diagram – jazyk funkčních blokových schémat

```

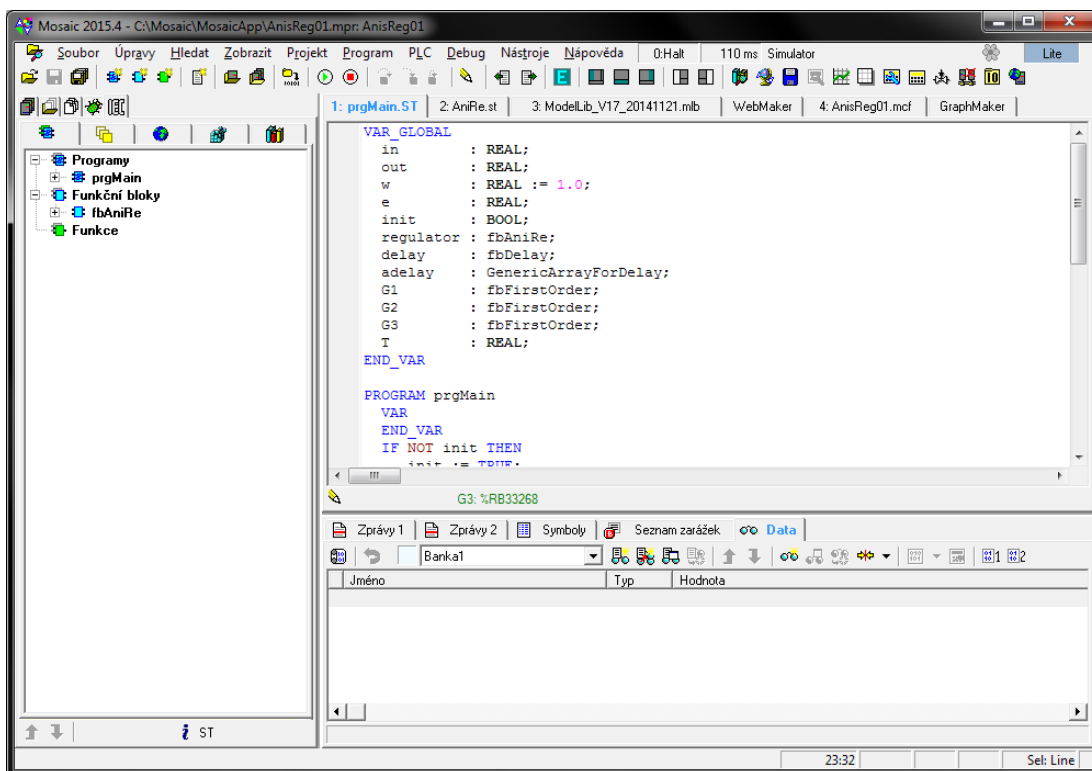
FUNCTION_BLOCK PříkladDeklaraceProm
VAR_INPUT
    logPodminka      : BOOL; (* binární hodnota *)
END_VAR
VAR_OUTPUT
    vysledek         : INT;  (* celočíselná hodnota se
                             znaménkem *)
END_VAR
VAR
    kontrolniSoucet : UINT; (* celočíselná hodnota *)
    mezivysledek    : REAL; (* reálná hodnota *)
END_VAR
END_FUNCTION_BLOCK

```

Obr. 2.3: Příklad deklarace proměnných POU, převzato z [6]

## 2.6. Programovací prostředí Mosaic

Převzato z [7]. Mosaic je vývojové prostředí pro tvorbu a ladění programů pro programovatelné logické systémy (PLC, Programmable Logic Controller) TECOMAT® a TECOREG® z produkce firmy Teco a.s. Kolín. Program Mosaic je dodáván od roku 2000. Prostředí je vyvíjeno ve shodě s mezinárodní normou IEC EN-61131-3, která definuje strukturu programů a programovací jazyky pro PLC.



Obr. 2.4: Ukázka programovacího prostředí Mosaic

### 2.6.1. Dodávka programu

Program Mosaic je dodáván stylem „all in one“, což znamená, že instalace obsahuje všechny nástroje, které jsou v dané chvíli k dispozici.

Pokud po nainstalování programu není přítomný HW klíč, Mosaic funguje ve verzi Lite, která je plně dostačující pro výuku a prověření všech vlastností včetně plné simulace. Kromě toho umožňuje bez omezení programovat nejmenší PLC z řady PLC TECOMAT®. Pro větší typy PLC je potřebný HW klíč, který umožní deklarace většího počtu I/O modulů.

### 2.6.2. Programování PLC TECOMAT, FOXTROT, TECOREG

Mosaic umožňuje programovat všechny PLC dodávané firmou Teco. U nové generace systémů TECOMAT® TC700 a TC650 a Foxtrot® se obvykle používá programování podle normy IEC EN 61131-3 v textových jazycích IL a ST a grafických jazycích LD a FBD.

Jak již bylo řečeno, programovat lze v grafických i textových jazycích. Programování v grafických jazycích je jednoduché a intuitivní. Jednotlivé kontakty

nebo bloky se vybírají z nástrojové lišty okna editoru a umísťují se na ploše. Prostředí samo nabízí dialogová okna pro přiřazení proměnné nebo výběr požadované POU ve chvíli vložení kontaktu, resp. bloku. Jak proměnné, tak POU lze definovat předem prostřednictvím IEC manažeru nebo až v okamžiku prvního použití.

Podobnou podporu nabízí prostředí Mosaic i v textových jazycích. V jazyku strukturovaného textu ST lze při psaní programu využít IEC asistenta, který nabízí dokončení rozepsaných konstrukcí, umožňuje vkládat dostupné proměnné, jejich definování apod. Vše je k dispozici prostřednictvím horkých kláves nebo pravého tlačítka myši. Při psaní programu je možné jednotlivé jazyky kombinovat. Jakmile však je zvolen pro zápis POU určitý jazyk, není možné ho již měnit. Další POU však může být zapsána opět v jiném jazyku. To umožňuje rozčlenit program a například část řídicí logiky psát v jazyku LD a část, kde se provádí matematické výpočty a objevují se složité výrazy, napsat v jazyku ST.

Deklarační část programu je všem jazykům společná. Podporovány jsou všechny datové typy definované výše uvedenou normou včetně datových typů pro práci s časem, datem nebo řetězci. Deklarace vlastních datových typů včetně struktur a polí je podporována stejně jako deklarace všech typů POU.

Prostředí Mosaic má integrovanou možnost používat knihovny bloků a vytvářet vlastní uživatelské knihovny POU.

## 3. Systém Tecomat Foxtrot

### 3.1. Základní parametry systému Tecomat Foxtrot

Systém Tecomat Foxtrot je kompaktní řídicí a regulační systém s možností modulárního rozšíření. Je určen pro řízení technologií v nejrůznějších oblastech průmyslu i v jiných odvětvích. Převzato z [8].

Systém Tecomat Foxtrot je konstrukčně řešen pro montáž do skříní a stojanů, všechny moduly Foxtrot jsou opatřeny plastovým ochranným pouzdem a držákem pro usazení modulu na U lištu. Základní parametry systému Tecomat Foxtrot a jeho provozní podmínky jsou uvedeny v tabulkách 3.1 a 3.2.

Tabulka 3.1: Základní parametry systému Tecomat Foxtrot [8]

Norma výrobku	ČSN EN 61131-2:2008
Třída ochrany elektrického předmětu ČSN EN 61140:2003 (IEC 61140:2001)	II
Druh zařízení	vestavné
Stupeň krytí ČSN EN 60529:1993 (IEC 529:1989)	IP20
Životnost	10 let

Tabulka 3.2: Provozní podmínky systému Tecomat Foxtrot [8]

Prostory ČSN 33 2000-3:1995 (IEC 364-3:1993)	normální
Rozsah provozních teplot	-20 °C až + 55 °C
Povolená teplota při přepravě	-25 °C až +70 °C
Relativní vlhkost vzduchu	10 % až 95 % bez kondenzace
Atmosférický tlak	min. 70 kPa (< 3000 m n. m.)
Stupeň znečištění ČSN EN 60664-1:2004 (IEC 60664-1:1992)	1
Přepětíová kategorie instalace ČSN EN 60664-1:2004 (IEC 60664-1:1992)	II
Pracovní poloha	svislá
Druh provozu	trvalý
Odolnost vůči vibracím (sinusovým)	10 až 57 Hz - amplituda 0,075 mm 57 až 150 Hz - zrychlení 1G
Elektromagnetická kompatibilita:	
Emise EN 55022:1999 (CISPR22:1997)	třída A
Imunita	min. dle požadavku ČSN EN 61131-2:2008

## 3.2. Základní modul Tecomat Foxtrot CP-1015

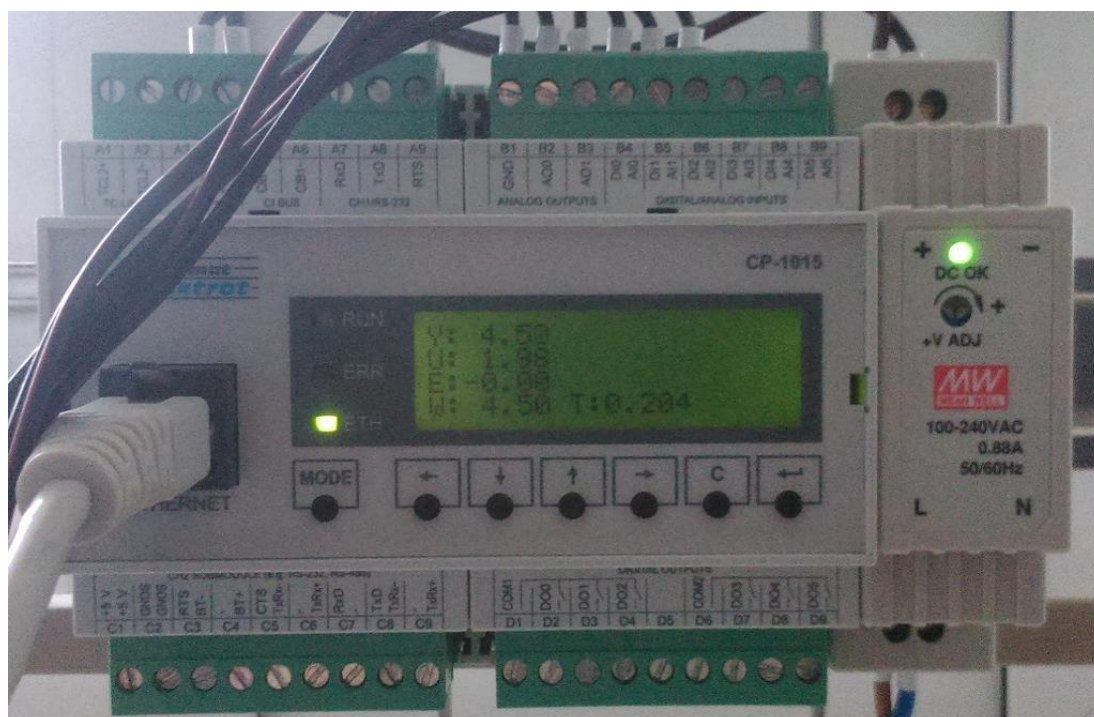
Základní modul Tecomat Foxtrot CP-1015 se skládá z několika částí [8]. První část tvoří centrální jednotka s hlavním procesorem systému, dvěma sériovými kanály, rozhraním Ethernet a systémovou sběrnicí TCL2 pro komunikaci s moduly.

Druhou část tvoří procesor zajišťující komunikaci na sběrnici CIB s moduly rodiny CFox. Na systémové sběrnici se hlásí pod jménem MI2-01M nebo CF-1140.

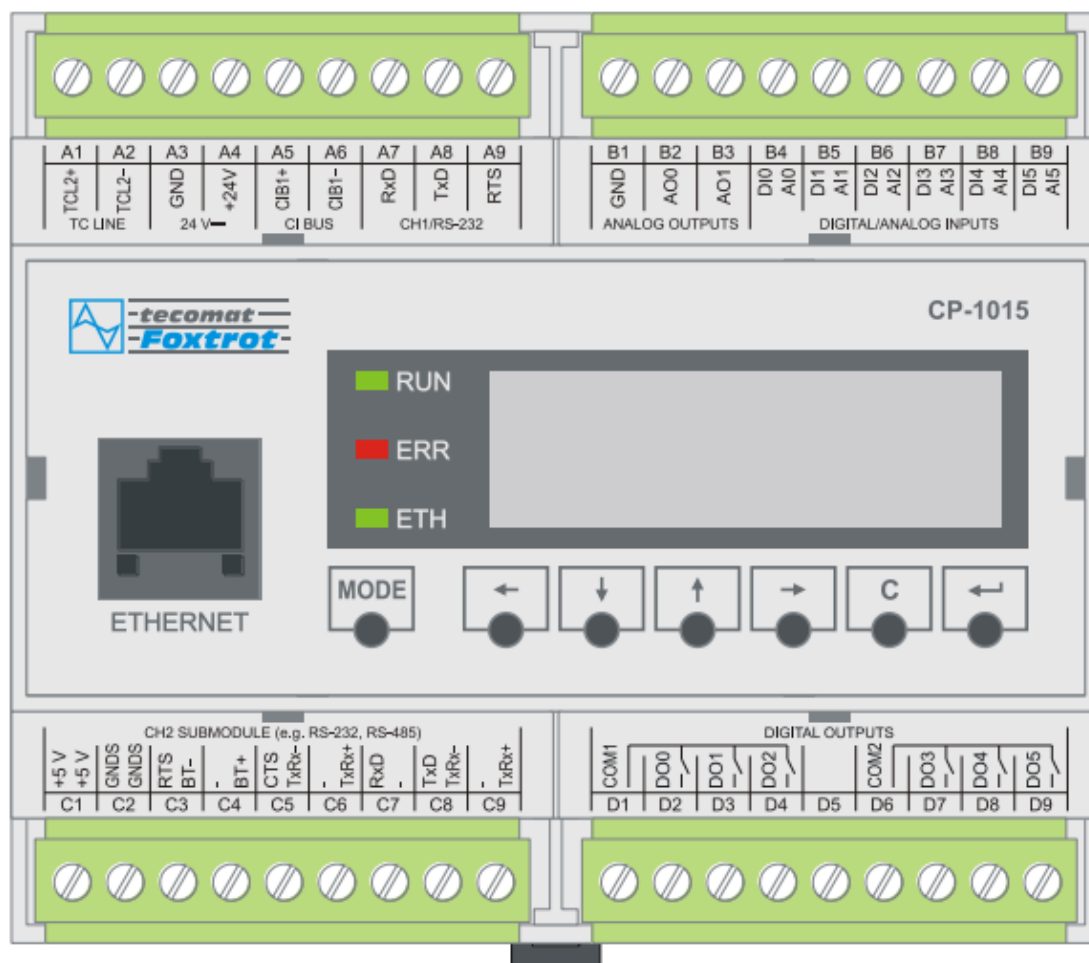
Třetí část je periferní a tvoří ji deska IR-1064 obsluhující vstupy a výstupy (starší provedení periferní desky má označení IR-1056).

Modul CP-1015 navíc obsahuje integrovaný LCD display 4 x 20 znaků a 6 uživatelských tlačítek. Procesor, který tuto část obsluhuje, se hlásí na systémové sběrnici pod jménem OI-1083 (starší provedení má označení OI-1073).

Základní parametry základního modulu Tecomat Foxtrot CP-1015 jsou uvedeny v tabulce 3.3 a provedení na Obr. 3.2.



Obr. 3.1: Použitý modul Tecomat Foxtrot CP-1015 s napájecím zdrojem (vpravo)



Obr. 3.2: Základní modul Tecomat Foxtrot CP-1015 [8]

Tabulka 3.3: Základní parametry modulu CP-1015 [8]

<b>Napájení modulu</b>	
Napájecí napětí	24 V DC, +25%, -15%
Interní jištění	vratná pojistka
Maximální příkon	10 W
<b>Připojení vodičů k modulu</b>	
Typ svorek	vyjímatelné svorkovnice
Průřez vodiče	max. 2,5 mm <sup>2</sup>
Rozhraní Ethernet	konektor RJ-45
<b>Mechanické řešení modulu</b>	
Rozměry modulu	105 x 92 x 63 mm
Šířka modulu v násobcích M (17,5 mm)	6M
Držák na U lištu	ano
<b>Vstupy a výstupy</b>	
Galvanické oddělení napájení od vnitřních obvodů	ne
Počet vstupů (volitelně binárních / analogových)	6
Počet reléových výstupů	6
Počet analogových výstupů	2
<b>Uživatelský display</b>	
Uživatelský LCD display (počet znaků)	4 x 20
Počet uživatelských tlačítek	6



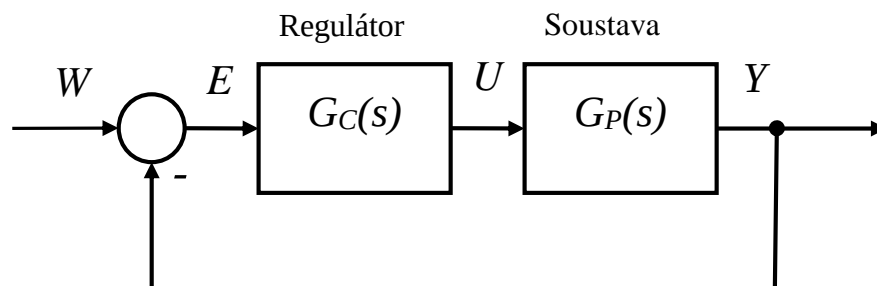
## 4. Návrh anisochronního regulátoru

### 4.1. Návrh regulátoru metodou požadovaného modelu

Přenosová funkce regulátoru byla navrhována pro model soustavy s anisochronním přenosem (4.1).

$$G_P(s) = \frac{K \cdot e^{-s\tau_u}}{(\tau_1 s + 1)(\tau_2 s + e^{-s\tau_y})} \quad (4.1)$$

Anisochronní model soustavy byl zvolen, protože je velmi universální a vhodný pro modelování systémů s dopravním zpožděním [1]. Další výhodou anisochronního modelu je možnost snadného odhadu jeho parametrů pomocí rozšířené reléové identifikace, popsané v [2] a [3].



Obr. 4.1: Schéma uzavřeného regulačního obvodu

Pro návrh spojitě přenosové funkce regulátoru  $G_C(s)$  byla použita metoda požadovaného modelu. Dosazením přenosové funkce modelu soustavy (4.1) do vztahu pro přenos otevřeného regulačního obvodu

$$G_0(s) = G_C(s)G_P(s) = \frac{k_0}{s} \cdot e^{-s\tau_u}, \quad (4.2)$$

získáme

$$G_0(s) = G_C(s) \frac{K \cdot e^{-s\tau_u}}{(\tau_1 s + 1)(\tau_2 s + e^{-s\tau_y})} = \frac{k_0}{s} \cdot e^{-s\tau_u}, \quad (4.3)$$

z čehož plyne

$$G_C(s) = \frac{k_0 (\tau_1 s + 1)(\tau_2 s + e^{-s\tau_y})}{K s}. \quad (4.4)$$

Aby bylo možné takovouto přenosovou funkci fyzicky realizovat, je třeba ji doplnit o low-pass filtr (dolní propust') tak, aby řád jmenovatele byl alespoň tak velký jako řád čitatele. Přenosová funkce regulátoru je potom

$$G_C(s) = \frac{k_0 (\tau_1 s + 1)(\tau_2 s + e^{-s\tau_y})}{K (\tau_f s + 1)^r s}, \quad (4.5)$$

kde  $\tau_f$  je časová konstanta filtru a  $r$  řád filtru. Časová konstanta  $\tau_f$  nám také částečně umožní omezit průběh akční veličiny  $u$ .

## 4.2. Diskretizace přenosu regulátoru

Pro použití navržené přenosové funkce regulátoru v programovatelném automatu je třeba ji nejprve převést do diskrétního tvaru. K diskretizaci navrženého spojitého přenosu regulátoru (4.5) byla použita náhrada Tustinovou aproximací

$$s = \frac{2}{T} \frac{z - 1}{z + 1}, \quad (4.6)$$

která zachovává stabilitu, a z části přesná náhrada Laplaceova operátoru (4.8) pro náhradu časového zpoždění v jednom ze členů přenosu.

$$z = e^{sT}, \quad (4.7)$$

$$s = \frac{\ln z}{T}, \quad (4.8)$$

kde  $T$  je vzorkovací perioda.

Po dosazení náhrad (4.6) a (4.8) do vztahu (4.4) dostaneme diskrétní přenos následujícího tvaru

$$G_C(z) = \frac{k_0}{K} \frac{\left(\tau_1 \frac{2}{T} \frac{z-1}{z+1} + 1\right) \left(\tau_2 \frac{2}{T} \frac{z-1}{z+1} + e^{-\frac{\tau_y}{T} \ln z}\right)}{\left(\tau_f \frac{2}{T} \frac{z-1}{z+1} + 1\right) \frac{2}{T} \frac{z-1}{z+1}}. \quad (4.9)$$

Vztah (4.9) je možno pomocí několika matematických operací a substitucí převést na tvar

$$G_C(z) = \frac{k_0}{K} \cdot \frac{b_0 + b_1 z + b_2 z^2 + b_3 z^{-D+2} + b_4 z^{-D+1} + b_5 z^{-D}}{a_0 + a_1 z + a_2 z^2}, \quad (4.10)$$

jehož koeficienty  $a_0$  až  $a_2$ ,  $b_0$  až  $b_5$  a  $D$  získáme z rovnic (4.11) až (4.20).

$$a_0 = 4 \cdot \frac{\tau_f}{T^2} - \frac{2}{T} \quad (4.11)$$

$$a_1 = -8 \cdot \frac{\tau_f}{T^2} \quad (4.12)$$

$$a_2 = 4 \cdot \frac{\tau_f}{T^2} + \frac{2}{T} \quad (4.13)$$

$$b_0 = 4 \cdot \frac{\tau_1 \tau_2}{T^2} - 2 \cdot \frac{\tau_2}{T} \quad (4.14)$$

$$b_1 = -8 \cdot \frac{\tau_1 \tau_2}{T^2} \quad (4.15)$$

$$b_2 = 4 \cdot \frac{\tau_1 \tau_2}{T^2} + 2 \cdot \frac{\tau_2}{T} \quad (4.16)$$

$$b_3 = 1 + 2 \cdot \frac{\tau_1}{T} \quad (4.17)$$

$$b_4 = 2 \quad (4.18)$$

$$b_5 = 1 - 2 \cdot \frac{\tau_1}{T} \quad (4.19)$$

$$D = \frac{\tau_y}{T} \quad (4.20)$$

Aby bylo možno regulátor naprogramovat, je třeba převést rovnici přenosu (4.10) na diferenční rovnici, čehož docílíme inverzní z-transformací. Diferenční rovnice pro přenos (4.10) potom vypadá takto:

$$\begin{aligned} u[k] = & \frac{k_0}{K \cdot a_2} \cdot (b_0 e[k-2] + b_1 e[k-1] + b_2 e[k] \\ & + b_3 e[k-D] + b_4 e[k-D-1] + b_5 e[k-D-2]) \\ & - \frac{1}{a_2} \cdot (a_0 u[k-2] + a_1 u[k-1]), \end{aligned} \quad (4.21)$$

kde  $k$  je diskrétní čas  $t = kT$ ,  $u[k]$  je akční veličina v čase  $kT$  a  $e[k]$  je regulační odchylka v čase  $kT$ .

### 4.3. Volba periody vzorkování

Pro správnou funkci regulátoru je třeba zvolit dostatečně krátkou periodu vzorkování, aby dostatečně rychle reagoval na dynamiku řízené soustavy a byl stabilní. Pro volbu periody vzorkování byla zvolena volba ze souhrnné časové konstanty

$$T < \frac{(\tau_1 + \tau_2 - \tau_y)}{v_T}, \quad (4.22)$$

$$v_T > 10. \quad (4.23)$$

Do souhrnné časové konstanty nebylo zahrnuto dopravní zpoždění  $\tau_u$ , protože nemá vliv na tvar a rychlost přechodové charakteristiky po jeho uplynutí.

Dále je třeba volit vzorkovací periodu  $T$  s ohledem na koeficient  $D$  diskrétního přenosu regulátoru (4.10) tak, aby koeficient  $D$  byl kladné celé číslo. Pokud by byl zvolen jako číslo reálné, bylo by nutné získat vzorky regulační odchylky  $e[k-D]$ ,  $e[k-D-1]$  a  $e[k-D-2]$  interpolací mezi dvěma naměřenými vzorky.

#### 4.4. Volba časové konstanty filtru

Filtr alespoň prvního řádu je nutný pro realizovatelnost regulátoru, ale zároveň chceme, aby ho příliš neovlivňoval. Také je možné ho částečně použít k omezení počátečního skoku akční veličiny  $u$  při skokové změně žádané hodnoty nebo poruchových veličin.

Pro implementaci v programovatelném automatu byl zvolen filtr prvního řádu a časová konstanta filtru byla zvolena přibližně o řád menší než souhrnná časová konstanta soustavy.

$$\tau_f = \frac{(\tau_1 + \tau_2 - \tau_y)}{v_{\tau_f}} \quad (4.24)$$

$$v_{\tau_f} < 10 \quad (4.25)$$

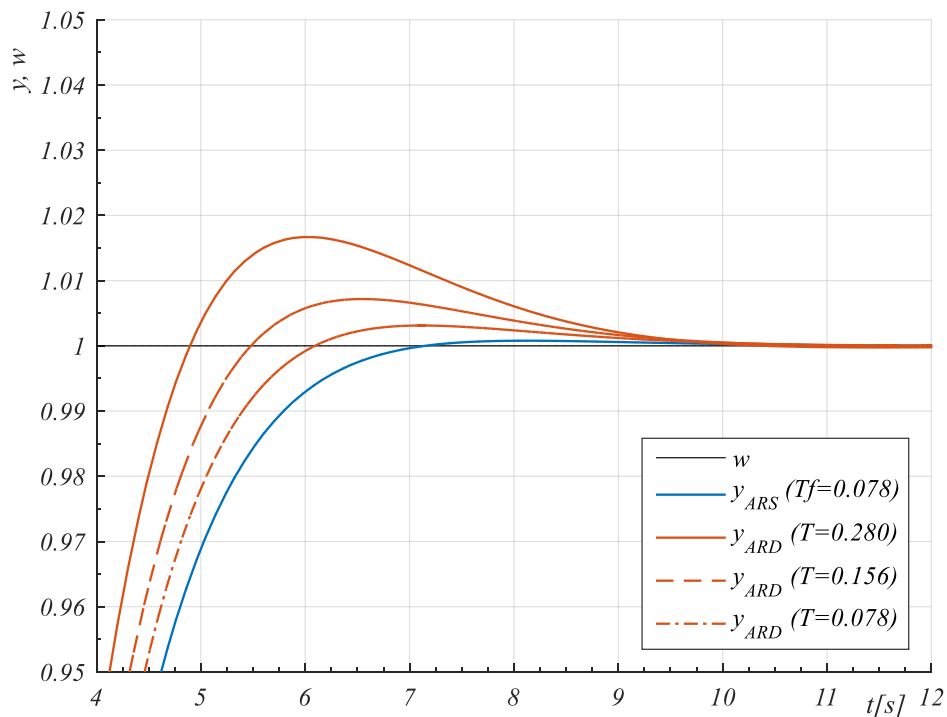
#### 4.5. Vliv periody vzorkování a časové konstanty filtru na průběh regulované veličiny

Vzorkovací perioda diskrétního regulátoru i časová konstanta filtru mohou mít na průběh regulované veličiny výrazné nežádoucí účinky, proto byly provedeny simulace v programu MATLAB Simulink, aby bylo možné posoudit zvolené metody jejich volby.

První simulace na posouzení vlivů volby vzorkovací periody a časové konstanty filtru byla provedena na uzavřeném regulačním obvodu, ve kterém byl použit model relativně rychlé soustavy s anisochronním přenosem ve tvaru

$$G_P(s) = \frac{1,6 \cdot e^{-0,9s}}{(0,7s + 1)(8,5s + e^{-1,4s})}. \quad (4.26)$$

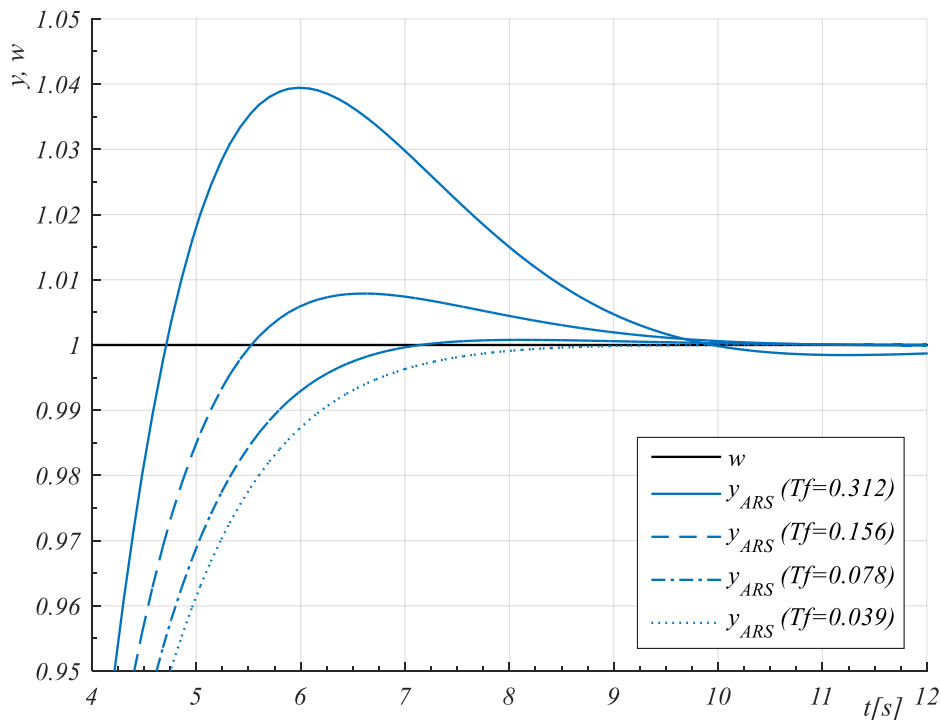
Regulátor byl nastaven na požadovaný relativní překmit regulované veličiny  $\kappa=0\%$ .



Obr. 4.2: Vliv vzorkovací periody  $T$  na průběh přechodové charakteristiky uzavřeného regulačního obvodu – „rychlá“ soustava

Na Obr. 4.2 je vidět vliv vzorkovací periody diskrétního regulátoru na rychlost regulace a překmit regulované veličiny při pevně zvolené časové konstantě filtru  $\tau_f = 0,078s$  ( $v_{\tau_f} = 100$ ). Vzorkovací perioda byla volena podle vztahů (4.22) a (4.23), kde  $v_T = 25, 50, 100$ . Pro porovnání je zobrazen i průběh simulace spojitěho regulátoru. Se zvyšující se vzorkovací periodou se průběh přechodové charakteristiky zhoršuje.

Na Obr. 4.3 je potom vidět vliv časové konstanty filtru  $\tau_f$  volené podle vztahů (4.24) a (4.25), kde  $\nu_{\tau_f} = 25, 50, 100, 200$  při pevně zvolené periodě vzorkování  $T=0,28s$  ( $\nu_T = 25$ ). Vykresleny jsou přechodové charakteristiky uzavřeného regulačního obvodu se spojitou simulací anisochronního regulátoru. Z grafu je zřejmé, že se zvyšující se časovou konstantou  $\tau_f$  se překmit zvětšuje.



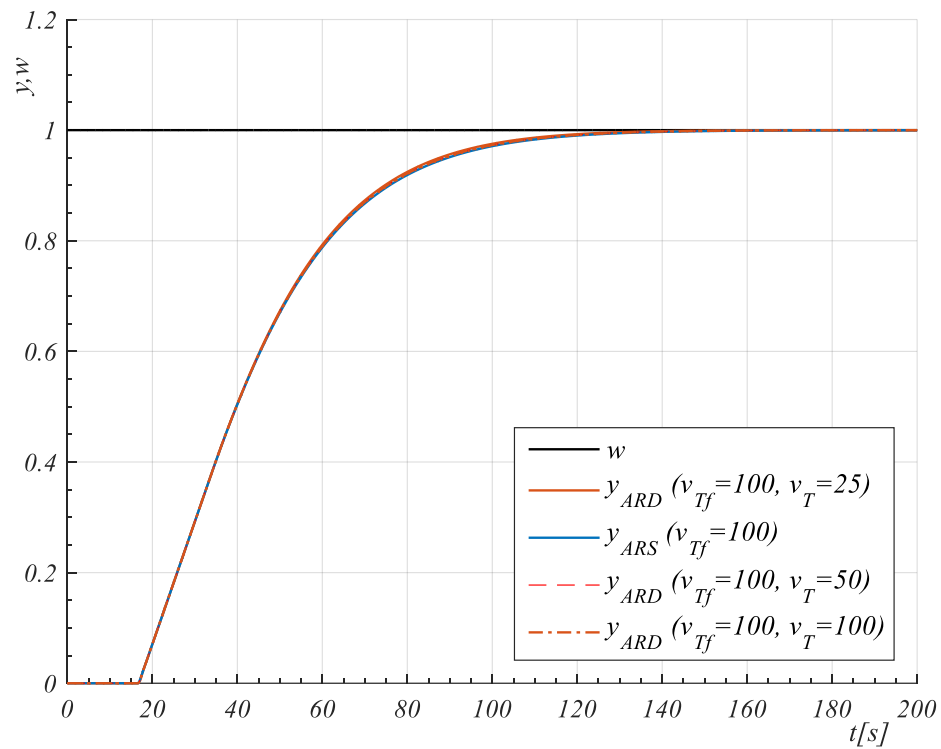
Obr. 4.3: Vliv časové konstanty filtru  $\tau_f$  na průběh přechodové charakteristiky uzavřeného regulačního obvodu – „rychlá“ soustava

Druhá simulace k posouzení vlivů vzorkovací periody a časové konstanty filtru byla provedena na uzavřeném regulačním obvodu, ve kterém byl použit model pomalejší soustavy s anisochronním přenosem ve tvaru

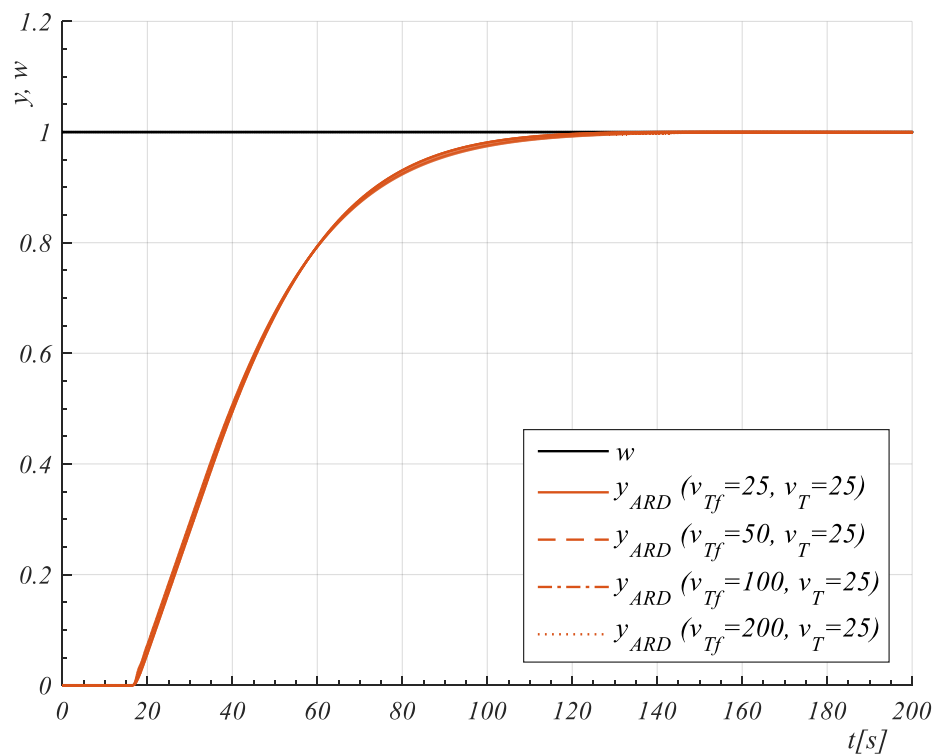
$$G_P(s) = \frac{1,6 \cdot e^{-16,6s}}{(7,4s + 1)(24s + e^{-10s})}. \quad (4.27)$$

Regulátor byl opět nastaven na požadovaný relativní překmit regulované veličiny  $\kappa=0\%$ .

Z grafů zobrazujících přechodové charakteristiky uzavřeného regulačního obvodu na Obr. 4.4 a Obr. 4.5 je patrné, že vzorkovací perioda a časová konstanta filtru nemají tak výrazný vliv na jejich průběh jako u soustavy s rychlejší dynamikou.



Obr. 4.4: Vliv vzorkovací periody  $T$  na průběh přechodové charakteristiky uzavřeného regulačního obvodu – „pomalá“ soustava



Obr. 4.5: Vliv časové konstanty filtru  $\tau_f$  na průběh přechodové charakteristiky uzavřeného regulačního obvodu – „pomalá“ soustava



## 4.6. Další úprava navrhovaného regulátoru

Po posouzení simulačních příkladů s navrženým regulátorem, hlavně vlivu filtru na přechodovou charakteristiku uzavřeného regulačního obvodu, byl regulátor dále upraven.

Úpravou přenosové funkce regulátoru získané metodou požadovaného modelu (4.4) na (4.28) zjistíme, že jedinou částí regulátoru, která nespĺňuje podmínku realizovatelnosti, je jeho derivační část  $\tau_1\tau_2s$ , podobně jako u PID regulátoru. Proto byl filtr přesunut pouze k této části regulátoru, jak je vidět na rovnici (4.29).

$$G_C(s) = \frac{k_0}{K} \cdot \left( \tau_1\tau_2s + \tau_2 + \tau_1e^{-s\tau_y} + \frac{e^{-s\tau_y}}{s} \right) \quad (4.28)$$

$$G_C(s) = \frac{k_0}{K} \cdot \left( \frac{\tau_1\tau_2s}{\tau_f s + 1} + \tau_2 + \tau_1e^{-s\tau_y} + \frac{e^{-s\tau_y}}{s} \right) \quad (4.29)$$

Přenos regulátoru (4.29) by následně rozdělen na čtyři části a diskretizován pomocí náhrad (4.6) a (4.8).

$$U(z) = \frac{k_0}{K} \cdot (G_1(z) + G_2(z) + G_3(z) + G_4(z)) \cdot E(z) \quad (4.30)$$

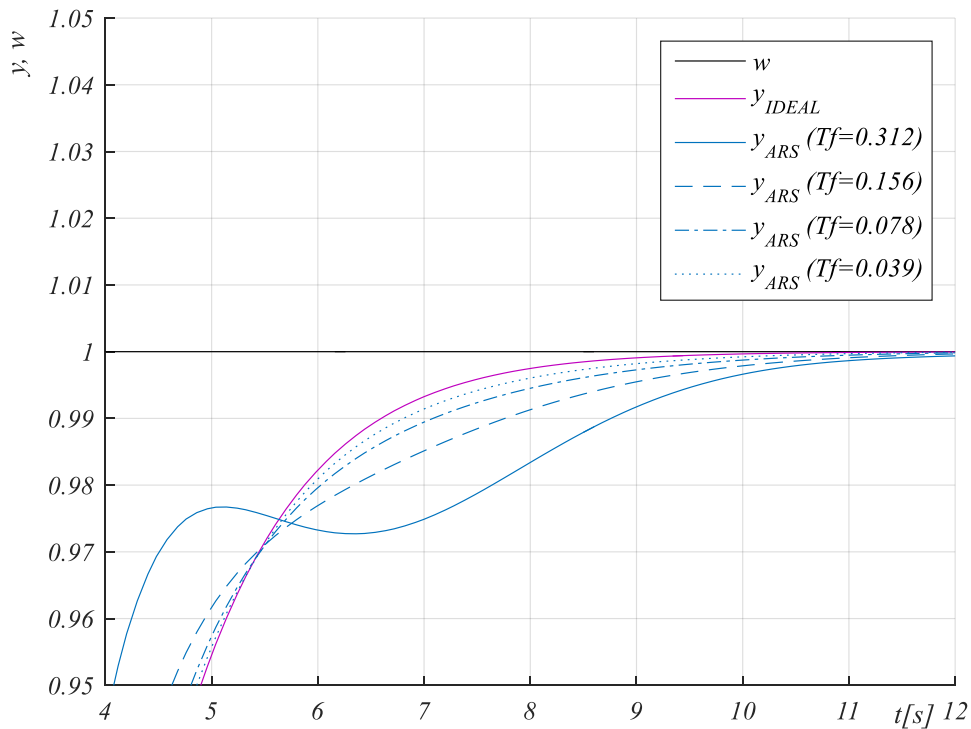
$$G_1(z) = \frac{\frac{2\tau_1\tau_2}{T} - \frac{2\tau_1\tau_2}{T} \cdot z^{-1}}{1 + \frac{2\tau_f}{T} + \left(1 - \frac{2\tau_f}{T}\right) \cdot z^{-1}} \quad (4.31)$$

$$G_2(z) = \tau_2 \quad (4.32)$$

$$G_3(s) = \tau_1 \cdot z^{-\frac{\tau_y}{T}} \quad (4.33)$$

$$G_4(s) = \frac{T}{2} \cdot \frac{1 + z^{-1}}{1 - z^{-1}} \cdot z^{-\frac{\tau_y}{T}} \quad (4.34)$$

Na Obr. 4.6 je vidět vliv časové konstanty filtru  $\tau_f$  na průběh přechodové charakteristiky uzavřeného regulačního obvodu s upraveným regulátorem. Fialovou barvou je znázorněn průběh, který bychom dostali, pokud by bylo možno použít regulátor bez filtru podle metody požadovaného modelu. V porovnání s Obr. 4.3 je vidět, že se lépe podařilo dosáhnout požadovaného překmitu  $\kappa=0\%$  než u první verze regulátoru.



Obr. 4.6: Vliv časové konstanty filtru  $\tau_f$  na průběh přechodové charakteristiky uzavřeného regulačního obvodu s upraveným regulátorem

Dále byl upraven způsob volby časové konstanty filtru, která je volena přibližně stejná jako vzorkovací perioda.

$$\tau_f = \frac{T}{v_{\tau_f}} \quad (4.35)$$

$$v_{\tau_f} \approx 1 \quad (4.36)$$

## 5. Programový modul pro PLC Tecomat Foxtrot

K naprogramování anisochronního regulátoru byl zvolen jako nejvhodnější jazyk ST (Structured Text) podle normy IEC 61 131-3. Textové jazyky LD a FBD nejsou pro programování algoritmů příliš vhodné a jazyk IL je příliš složitý pro intuitivní a rychlé programování.

### 5.1. První verze regulátoru

První verze regulátoru byla naprogramována podle přenosové funkce (4.10). Celý kód je v příloze Příloha A.

#### 5.1.1. Funkční blok regulátoru

Vlastní regulátor i s nastavením parametrů byl naprogramován jako jeden funkční blok realizující řídicí algoritmus. Jak pro reprezentaci vstupujících a vystupujících veličin, tak i pro realizaci regulačního algoritmu byl zvolen datový typ **REAL**, protože jeho použití je pro implementaci řídicího algoritmu jednodušší než použití datového typu **INT**. Použití celočíselné aritmetiky by bylo výhodou s ohledem na rychlost vykonávání algoritmu.

#### Vstupy funkčního bloku regulátoru

Jako vstupy do funkčního bloku regulátoru vstupují předem zjištěné nebo odhadnuté parametry anisochronního modelu (4.1), ze kterých jsou později nastaveny parametry diskrétního modelu regulátoru (4.10), regulovaná veličina  $y$ , žádaná hodnota regulované veličiny  $w$ , požadovaný relativní překmit  $\kappa$  v procentech a parametry pro úpravu volby vzorkovací periody  $v_T$  a časové konstanty filtru  $v_{tf}$ . Dále je zde vstup pro resetování nastavení regulátoru.

```
VAR_INPUT
  // Z identifikace
  K          : REAL;
  T1         : REAL;
  T2         : REAL;
  Ty         : REAL;
  Tu         : REAL;
  // Vstupy
```

```

w          : REAL;
Y          : REAL;
// Pozadovany relativni prekmit k v %
kappa     : SINT := 10;
// pro volbu casove konst. filtru Tf, vTf ~ min 10,
// Tf = (T1+T2-Ty)/vTf
vTf       : REAL := 35;
// pro volbu periody vzorkovani T a zpozdeni D,
// vT ~ min 10, T < (T1+T2-Ty)/vT
vT        : REAL := 35;
// Reset inicializace
reset     : BOOL := FALSE;
END_VAR

```

### Výstupy funkčního bloku regulátoru

Funkční blok regulátoru má dvě vystupující veličiny, akční zásah  $u$  a regulační odchylku  $e$ .

```

VAR_OUTPUT
// Vystup - akcni velicina
u          : REAL := 0.0;
// Vystup - regulacni odchylka
e          : REAL;
END_VAR

```

### Vnitřní proměnné funkčního bloku regulátoru

Vnitřními proměnnými funkčního bloku regulátoru jsou všechny parametry diskrétního modelu regulátoru, jsou to zejména vzorkovací perioda  $T$ , parametry  $a_0$  až  $a_2$ ,  $b_0$  až  $b_5$ ,  $D$  a  $k_0$ . Dále jsou to zpožďovací bloky `fbDelay` z knihovny `ModelLib` pro implementaci zpoždění regulační odchylky a akční veličiny.

```

VAR
// init flag
init      : BOOL := FALSE;
// Vzorkovaci perioda
T         : REAL := 0.1;
Tmax     : REAL := 5.0;
// Casova konstanta filtru
Tf       : REAL := 0.25;
// Pole predchozich hodnot pro bloky zpozdeni

```

```

aEkm2      : GenericArrayForDelay;
aEkm1      : GenericArrayForDelay;
aEkmD      : GenericArrayForDelay;
aEkmDm1    : GenericArrayForDelay;
aEkmDm2    : GenericArrayForDelay;
aUkm2      : GenericArrayForDelay;
aUkm1      : GenericArrayForDelay;
// Bloky zpozdeni
Ekm2       : fbDelay;
Ekm1       : fbDelay;
EkmD       : fbDelay;
EkmDm1     : fbDelay;
EkmDm2     : fbDelay;
Ukm2       : fbDelay;
Ukm1       : fbDelay;
// Koef. diskretniho modelu
b0         : REAL;
b1         : REAL;
b2         : REAL;
b3         : REAL;
b4         : REAL;
b5         : REAL;
a0         : REAL;
a1         : REAL;
a2         : REAL;
D          : INT;
k0         : REAL;
// Cas posledniho behu
lastT      : TIME;
END_VAR

```

### Výkonná část funkčního bloku regulátoru

Výkonná část funkčního bloku regulátoru se skládá ze dvou hlavních částí. Z části inicializační, která proběhne pouze při prvním spuštění funkčního bloku a vypočítá a nastaví všechny potřebné proměnné diskretního regulátoru.

```

// Volba k0 podle pozadovaneho rel. prekmitu kappa
k0 := 1.0/Tu/(2.718-0.4547*EXPT(SINT_TO_REAL(kappa),
                               0.3432));
// Volba casove konstanty filtru, default: vTf=35
Tf := (T1+T2-Ty)/vTf;
// Volba periody vzorkovani a zpozdeni D, default:
vT=35
Tmax := (T1+T2-Ty)/vT;
D := 1;

```

```

T := Ty/INT_TO_REAL(D);
WHILE T > Tmax DO
    D := D+1;
    T := Ty/INT_TO_REAL(D);
END_WHILE;
// Vypocet koef. diskretniho modelu
b0 := 4.0*T1*T2/T/T - 2.0*T2/T;
b1 := -2.0*4.0*T1*T2/T/T;
b2 := 4.0*T1*T2/T/T + 2.0*T2/T;
b3 := 2.0*T1/T + 1.0;
b4 := 2.0;
b5 := 1.0 - 2.0*T1/T;
a0 := 4.0*Tf/T/T - 2.0/T;
a1 := -2.0*4.0*Tf/T/T;
a2 := 4.0*Tf/T/T + 2.0/T;
// Aktualizace regulacni odchyly
e := w-y;
// Inicializace zpozdocich bloku
Ekm2( In := e, K := 2, T := T, Buffer := aEkm2 );
Ekm1( In := e, K := 1, T := T, Buffer := aEkm1 );
EkmD( In := e, K := D, T := T, Buffer := aEkmD );
EkmDm1(In := e, K := D+1, T := T, Buffer := aEkmDm1);
EkmDm2(In := e, K := D+2, T := T, Buffer := aEkmDm2);
Ukm2( In := u, K := 2, T := T, Buffer := aUkm2 );
Ukm1( In := u, K := 1, T := T, Buffer := aUkm1 );
// Vypocet nove hodnoty akcni veliciny u
u := k0/K/a2*(b0*Ekm2.Out + b1*Ekm1.Out + b2*e
    + b3*EkmD.Out + b4*EkmDm1.Out + b5*EkmDm2.Out)
    - (a0*Ukm2.Out + a1*Ukm1.Out)/a2;
// Nastaveni init flag
init := TRUE;

```

A z části, která aktualizuje hodnoty regulované veličiny, regulační odchyly a akční veličiny na základě aktuálních hodnot vstupů. Aktualizace se provádí každou vzorkovací periodu  $T$ .

```

// Aktualizace vstupu zpozdocich bloku
Ekm2(In := e, Buffer := aEkm2 );
Ekm1(In := e, Buffer := aEkm1 );
EkmD(In := e, Buffer := aEkmD );
EkmDm1(In := e, Buffer := aEkmDm1);
EkmDm2(In := e, Buffer := aEkmDm2);
Ukm2(In := u, Buffer := aUkm2 );
Ukm1(In := u, Buffer := aUkm1 );

// Vzorkovani
IF (gettime()-lastT) > REAL_TO_TIME(T*1000.0) THEN

```

```

// Aktualizace regulacni odchylky
e := w-y;
// Vypocet nove hodnoty akcni veliciny u
u := k0/K/a2*(b0*Ekm2.Out + b1*Ekm1.Out + b2*e
      + b3*EkmD.Out + b4*EkmDm1.Out + b5*EkmDm2.Out)
      - (a0*Ukm2.Out + a1*Ukm1.Out)/a2;
lastT := gettime();
END_IF;

```

### 5.1.2. Volání funkčního bloku regulátoru z hlavního programu

K použití funkčního bloku regulátoru je třeba ho zavolat z hlavního programu. Při prvním volání se musí inicializovat zadáním parametrů anisochronního modelu řízené soustavy. Příklad části hlavního programu:

```

PROGRAM prgMain
VAR
    regulator          : fbAniRe;
END_VAR
// Posun do pracovniho bodu.
y := prutokomer.ENG - y0;

// Inicializace regulatoru - nastaveni parametru
IF NOT init THEN
    init := TRUE;
    regulator(K:=1.61, T1:=0.6855, T2:=8.5185,
Ty:=1.4283, Tu:=0.9209, w:=w-y0, y:=y, vT:=35.0,
vTf:=35.0, kappa:=0, u=>u, e=>e);

// Volani bloku regulatoru v kazdem dalsim cyklu
ELSE
    regulator(w:=w-y0, y:=y, u=>u, e=>e);
END_IF;

// Vypocet akcniho zasahu
ventilator.ENG := u + u0;

END_PROGRAM

```

## 5.2. Druhá verze regulátoru

Druhá verze regulátoru byla naprogramována podle přenosové funkce (4.30), kde je filtrována pouze derivační složka přenosu. Celý kód je v příloze Příloha B.

### 5.2.1. Funkční blok regulátoru

Vlastní regulátor i s nastavením parametrů byl opět naprogramován jako jeden funkční blok realizující řídicí algoritmus. Oproti první verzi regulátoru byly změněny hlavně některé vnitřní proměnné a vlastní algoritmus řízení.

#### Vstupy funkčního bloku regulátoru

Oproti první verzi regulátoru přibyly dva vstupy pro omezení maximální a minimální hodnoty výstupu akční veličiny  $u$ .

```
// Max, Min hodnoty u
umax      : REAL := 100;
umin      : REAL := -100;
```

#### Výstupy funkčního bloku regulátoru

Funkční blok regulátoru má dvě vystupující veličiny, akční zásah  $u$  a regulační odchylku  $e$ , které byly oproti předchozí verzi přejmenovány.

```
VAR_OUTPUT
  // Vystup - akcni velicina
  out_u : REAL := 0.0;
  // Vystup - regulacni odchylka
  out_e : REAL;
END_VAR
```

#### Vnitřní proměnné funkčního bloku regulátoru

Oproti původnímu regulátoru byl zredukován počet zpožďovacích bloků ze sedmi na dva a byly zavedeny náhradní proměnné. Dále byly zavedeny proměnné pro výpočet jednotlivých částí akční veličiny  $u$ .

```
// Bloky zpozdeni
ekmD      : fbDelay;
ekmDm1    : fbDelay;
// Promenne pro zpozdeni a pomocne promenne pro vypocet
u
u1        : REAL;
u2        : REAL;
u3        : REAL;
u4        : REAL;
ekm1      : REAL := 0.0; // e[k-1]
```



```

u1km1      : REAL := 0.0; // u1[k-1]
u4km1      : REAL := 0.0; // u4[k-1]

```

### Výkonná část funkčního bloku regulátoru

Ve výkonné části funkčního bloku regulátoru byl upraven algoritmus pro výpočet akčního zásahu  $u$  a pro volbu časové konstanty filtru  $\tau_f$ . Dále bylo přidáno omezení integrace při překročení maximální nebo minimální hodnoty akčního zásahu  $u$ , aby zbytečně nedocházelo k „přintegrování“ při překročení daných mezí.

```

// Volba casove konstanty filtru, default: vTf=1
Tf := T/vTf;

// Vypocet koef. diskretniho modelu
b0 := 2.0*T1*T2/T;
b1 := -2.0*T1*T2/T;
a0 := 1.0+2.0*Tf/T;
a1 := 1.0-2.0*Tf/T;

// Kontrola integrace
IF (u > umax) OR (u < umin) THEN
    stop_int := TRUE;
ELSE
    stop_int := FALSE;
END_IF;

// Vypocet nove hodnoty akcni veliciny u
u1 := b0/a0*e + b1/a0*ekm1 - a1/a0*u1km1;
u2 := T2*e;
u3 := T1*ekmD.out;
IF NOT stop_int THEN
    u4 := T/2.0*(ekmD.out + ekmDm1.out) + u4km1;
END_IF;
u1km1 := u1;
u4km1 := u4;
ekm1 := e;

u := k0/K*(u1 + u2 + u3 + u4);

// Vystupni veliciny a jejich omezeni
out_u := u;
IF u > umax THEN out_u := umax;
END_IF;
IF u < umin THEN out_u := umin;
END_IF;

out_e := e;

```

### 5.2.2. Volání funkčního bloku regulátoru z hlavního programu

Volání upraveného regulátoru je stejné jako u regulátoru původního, až na změnu názvů výstupů, změněný význam parametru  $v_{Tf}$  a parametry pro omezení velikosti akčního zásahu. Příklad volání regulátoru z hlavního programu:

```
// Inicializace regulatoru - nastaveni parametru
IF NOT init THEN
  init := TRUE;
  regulator(K:=1.61, T1:=0.6855, T2:=8.5185,
Ty:=1.4283, Tu:=0.9209, w:=w-y0, y:=y, vT:=35.0,
vTf:=1.0, kappa:=0, umin:=-u0, umax:=10.0-u0, out_u=>u,
out_e=>e);

// Volani bloku regulatoru v kazdem dalsim cyklu
ELSE
  regulator(w:=w-y0, y:=y, out_u=>u, out_e=>e);
END_IF;
```

### 5.3. Ovládání - WebMaker

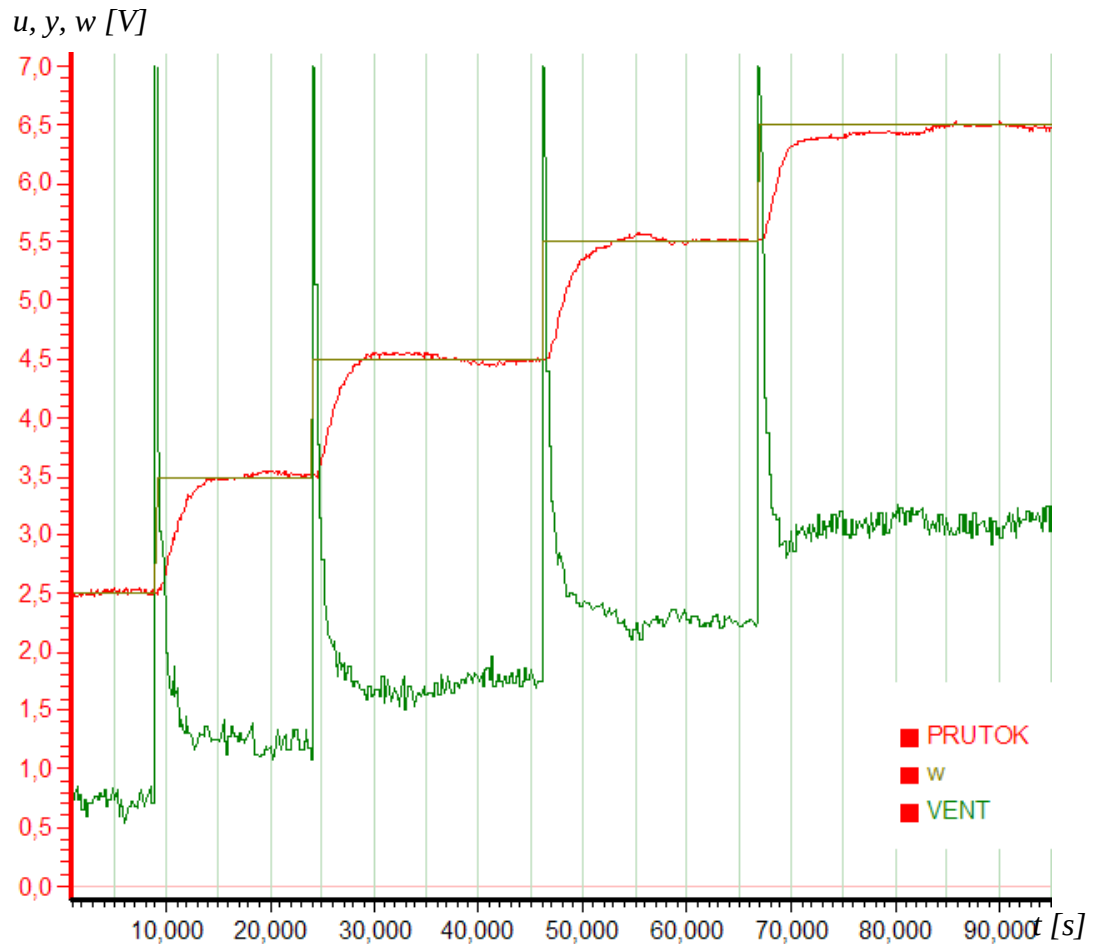
Pro ovládání regulátoru bylo vytvořeno jednoduché webové rozhraní pro zadávání požadované hodnoty regulované veličiny a sledování dalších hodnot. K vytvoření byl využit nástroj WebMaker, který je součástí vývojového prostředí Mosaic.

<b>Index</b>	
<i>Anisochronní regulátor</i>	Žádaná hodnota w [V]: <input type="text" value="0"/>
<b>Index</b>	Ventilátor u [V]: <input type="text" value="0"/>
<b>Logout</b>	Průtokoměr y [V]: <input type="text" value="0"/>
	Regulační odchylka e [V]: <input type="text" value="0"/>
	Stop: <input type="text" value="0"/>

Obr. 5.1: Jednoduché rozhraní pro zadávání žádané hodnoty regulované veličiny

## 5.4. Test navrženého regulátoru na reálné soustavě

Navržený programový modul anisochronního regulátoru byl otestován na teplovzdušném modelu, který je popsán v kapitole 6. Průběh jednoho z testů je vidět na Obr. 5.2.



Obr. 5.2: Test anisochronního regulátoru na teplovzdušném modelu, graf z programovacího prostředí Mosaic

## 6. Srovnání anisochronního řízení a řízení s PID regulátorem

### 6.1. Použitý PID regulátor

Pro porovnání PID a anisochronního řízení byl použit funkční blok z knihovny ModelLib pro Mosaic [9], který zajišťuje funkci PID regulátoru s filtrovanou derivační složkou a také ošetřuje omezení integrační složky při saturaci výstupu (antiwind-up). Regulační algoritmus mimo saturaci aproximuje chování popsané rovnicí

$$u(t) = r_0 \cdot \left( e(t) + \frac{1}{T_i} \cdot \int_{t_0}^t e(t) dt - T_d \cdot \frac{dy_f(t)}{dt} \right), \quad (6.1)$$

kde  $y_f$  je filtrovaný vstup derivační složky. Filtraci zajišťuje filtr prvního řádu dle rovnice

$$T_f \cdot \frac{dy_f(t)}{dt} + y_f(t) = y(t). \quad (6.2)$$

Parametry regulačního algoritmu PID jsou  $r_0$  – zesílení,  $T_i$  – integrační časová konstanta,  $T_d$  – derivační časová konstanta,  $T$  – vzorkovací perioda regulátoru a  $T_f$  – časová konstanta filtrace derivační složky. Výstupem algoritmu je součet složek P, I a D, jejichž aproximace jsou popsány rovnicemi (6.3), (6.4) a (6.5).

$$P = r_0 \cdot e \quad (6.3)$$

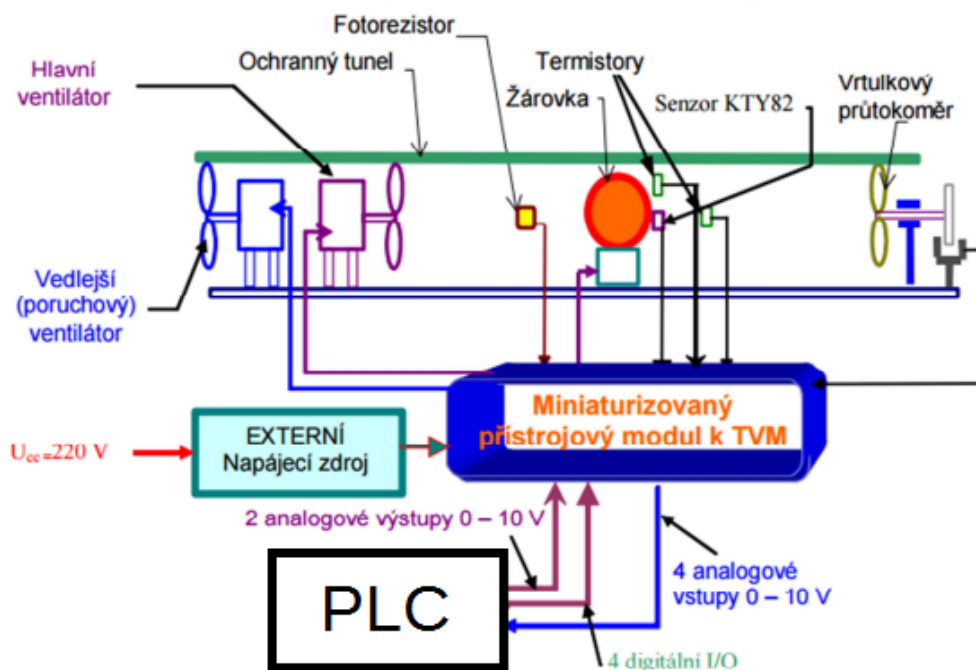
$$I = \frac{r_0 \cdot T}{T_i} \cdot \sum_{i=0}^k \left( \frac{e[i] + e[i-1]}{2} \right) \quad (6.4)$$

$$D = r_0 \cdot T_d \cdot \frac{y_f[k] - y_f[k-1]}{T} \quad (6.5)$$

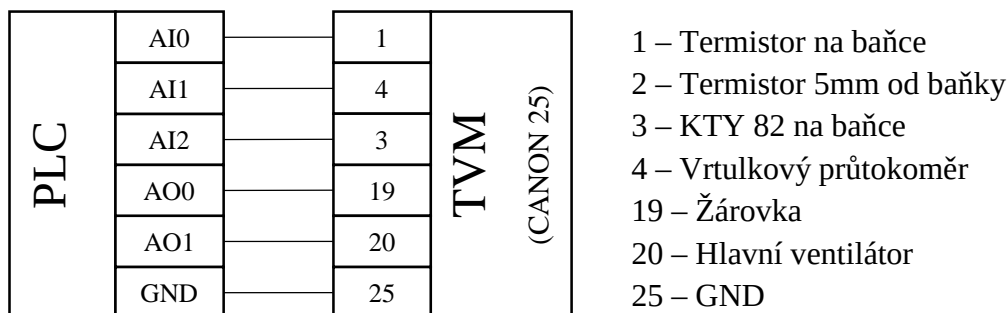
## 6.2. Teplovzdušný model (TVM)

Teplovzdušný model se skládá z krytého tunelu, do kterého je nasáván nebo z něho vytlačován vzduch pomocí dvou ventilátorů – hlavního a vedlejšího. Uprostřed tunelu je žárovka napájená pulzně šířkově modulovaným napětím z ovládaného zdroje, která slouží jako zdroj tepla. Průtok vzduchu tunelem je měřen vrtulkovým průtokoměrem na konci tunelu a teplota je měřena dvěma termistory a křemíkovým teplotním senzorem KTY-82.

Teplovzdušný model je řízen pomocí programovatelného automatu, ke kterému může být připojen pomocí čtyř analogových výstupů s rozsahem 0 – 10V pro snímání hodnot ze tří snímačů teploty a jednoho průtokoměru, dvou analogových vstupů s rozsahem 0 – 10V k ovládání žárovky a hlavního ventilátoru a čtyřmi digitálními vstupy, pomocí kterých je ovládán vedlejší ventilátor. Schéma modelu je na Obr. 6.1.

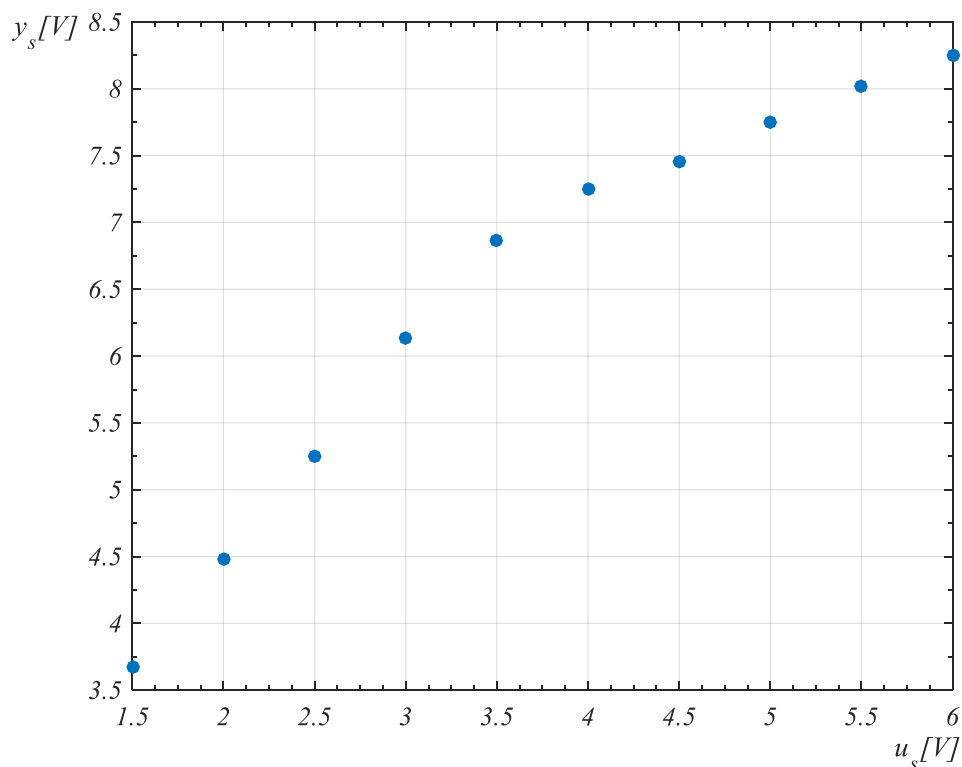


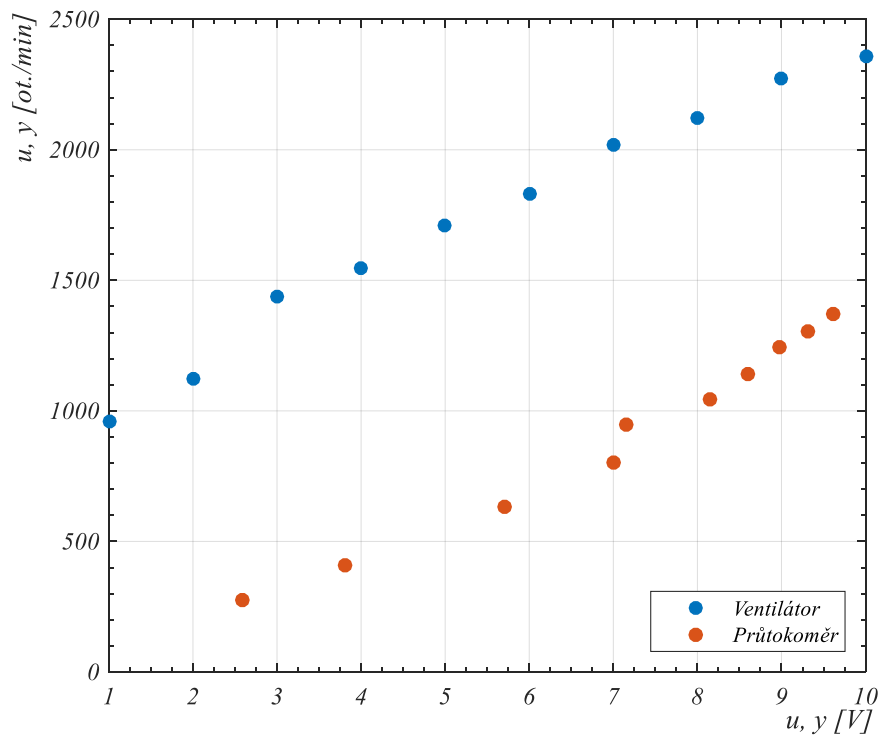
Obr. 6.1: Schéma Teplovzdušného modelu (vstupy a výstupy z pohledu PLC)



Obr. 6.2: Schéma zapojení konektoru k PLC (nebyly využity všechny vstupy/výstupy)

Aby mohl být zvolen pracovní bod pro testování a porovnání regulátorů, byla na teplovzdušném modelu změřena statická charakteristika. Jako akční zásah do soustavy TVM bylo zvoleno napětí na vstupu ventilátoru, jako výstupní regulovaná veličina bylo zvoleno napětí na výstupu vrtulkového průtokoměru. Naměřená statická charakteristika je na obrázku 6.3.

Obr. 6.3: Naměřené body statické charakteristiky TVM (ventilátor  $u_s$ , průtokoměr  $y_s$ )



Obr. 6.4: Naměřené otáčky odpovídající napětí na ventilátoru a průtokoměru



Obr. 6.5: Fotografie teplovzdušného modelu

## 6.3. Simulace – MATLAB Simulink

### 6.3.1. Simulace na soustavě šestého řádu

Pro porovnání regulátorů v programu MATLAB Simulink byla zvolena soustava šestého řádu s dopravním zpožděním s přenosem

$$G_P(s) = \frac{1}{(5s + 1)^6} \cdot e^{-8s}, \quad (6.6)$$

kteřá byla následně identifikována rozšířenou reléovou identifikací. Z identifikace byly získány parametry anisochronního modelu (6.7) v tabulce 6.1.

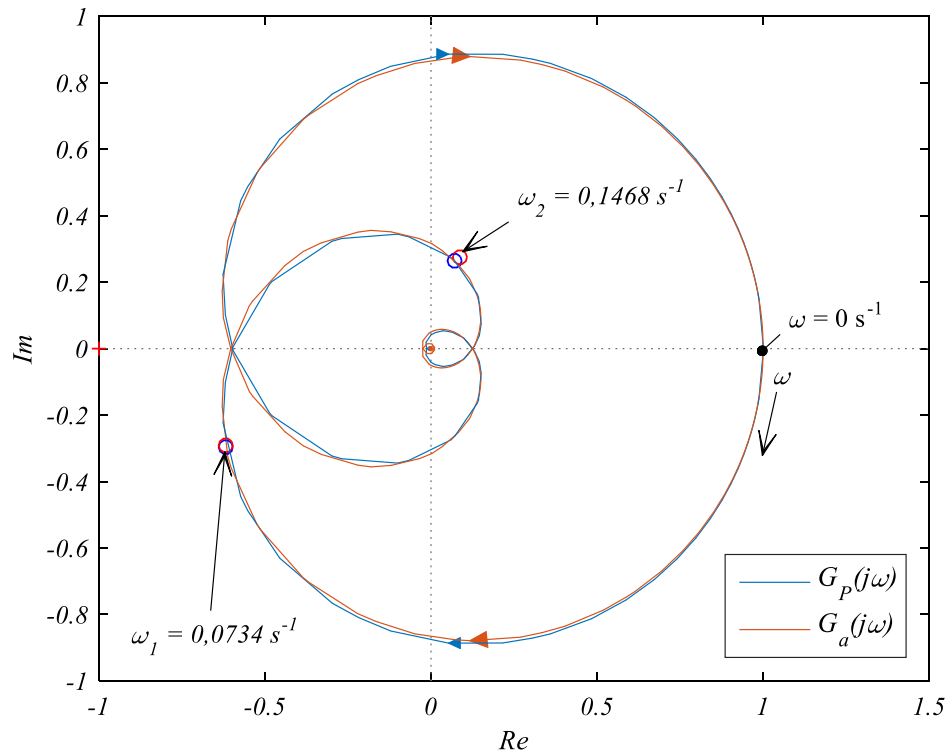
$$G_a(s) = \frac{K \cdot e^{-s\tau_u}}{(\tau_1 s + 1)(\tau_2 s + e^{-s\tau_y})} \quad (6.7)$$

Tabulka 6.1: Parametry anisochronního modelu soustavy – simulace

$K[-]$	$\tau_u[s]$	$\tau_1[s]$	$\tau_2[s]$	$\tau_y[s]$
1	16,4258	14,6313	17,0399	9,2660

Pro porovnání přenosu původní simulované soustavy a přenosu anisochronního modelu soustavy byla také vykreslena Nyquistova křivka na Obr. 6.6.





Obr. 6.6: Nyquistova frekvenční charakteristika – simulace

Dále bylo pro nastavení anisochronního regulátoru potřeba stanovit zesílení otevřeného regulačního obvodu  $k_0$  podle požadovaného relativního překmitu regulované veličiny  $y$ , což bylo provedeno pro zvolený relativní překmit  $\kappa=5\%$  podle rovnic (6.8) a (6.9).

$$\beta(\kappa = 5) = 2,718 - 0,4547\kappa^{0,3432} = 1,928 \quad (6.8)$$

$$k_0 = \frac{1}{\beta\tau_u} = 0,0316 \quad (6.9)$$

Pro diskrétní anisochronní regulátor bylo dále potřeba stanovit periodu vzorkování  $T$ , která byla zvolena tak, aby odpovídala omezením (6.10) a (6.11). Časová konstanta filtru  $\tau_f$  byla zvolena podle rovnice (6.12).

$$T < \frac{\tau_1 + \tau_2 - \tau_y}{20} = 1,12 \quad (6.10)$$

$$T = \frac{\tau_y}{D} = \frac{9,27}{9} = 1,03, \quad D \in \mathbb{N} \quad (6.11)$$

$$T_f = \frac{\tau_1 + \tau_2 - \tau_y}{35} = 0,64 \quad (6.12)$$

Parametry  $a_0$  až  $a_2$  a  $b_0$  až  $b_5$  byly stanoveny podle rovnic (4.11) až (4.19). Všechny parametry jsou shrnuty v tabulce 6.2.

Tabulka 6.2: Parametry anisochronního regulátoru – simulace

$\tau_u$	16,43	$K$	1,00	$a_0$	0,47	$b_1$	-1881,60	$b_5$	-27,42
$\tau_1$	14,63	$\tau_f$	0,64	$a_1$	-4,83	$b_2$	973,93	$k_0$	0,0316
$\tau_2$	17,04	$T$	1,03	$a_2$	4,36	$b_3$	29,42		
$\tau_y$	9,27	$D$	9,00	$b_0$	907,72	$b_4$	2,00		

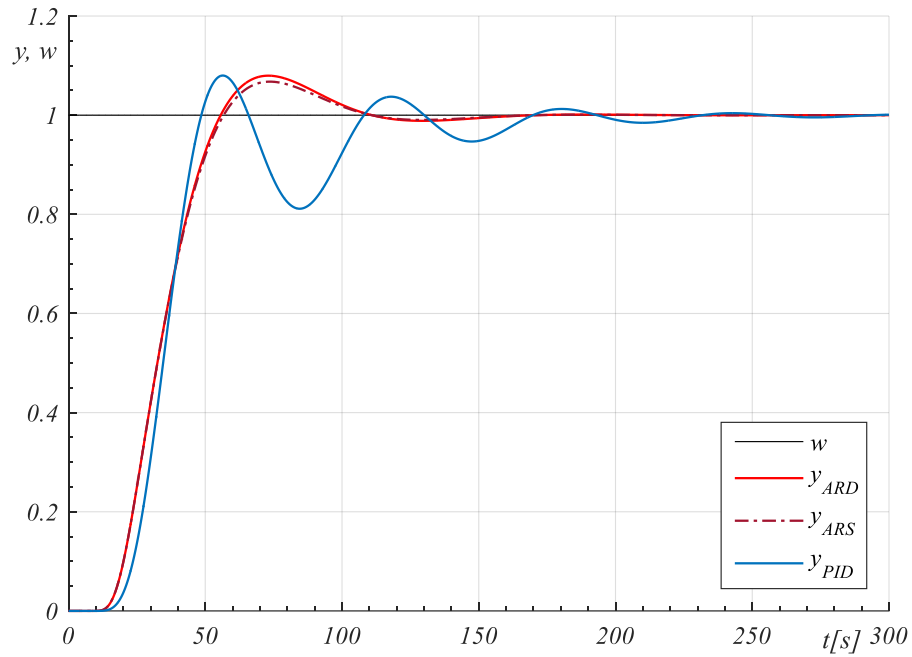
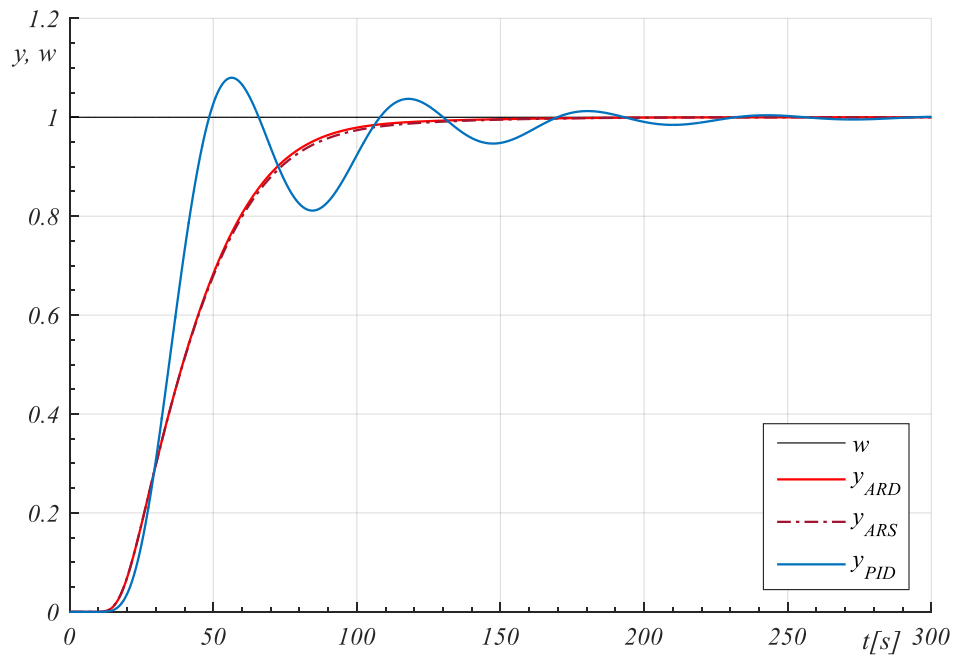
Dále byla použita metoda Ziegler-Nichols pro nastavení parametrů PID regulátoru. Pro zjištění kritického zesílení  $r_k$  a kritické periody kmitů  $T_k$  byla použita reléová metoda Åström-Hägglund. Zjištěné parametry PID regulátoru, hodnota kritického zesílení a hodnota kritické periody jsou v tabulce 6.3.

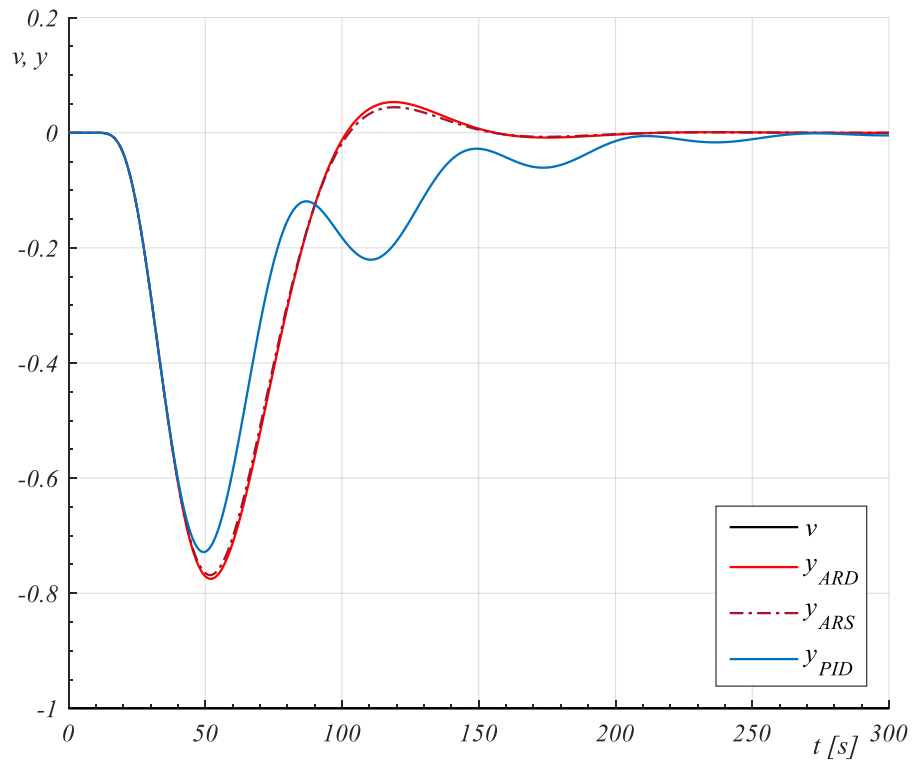
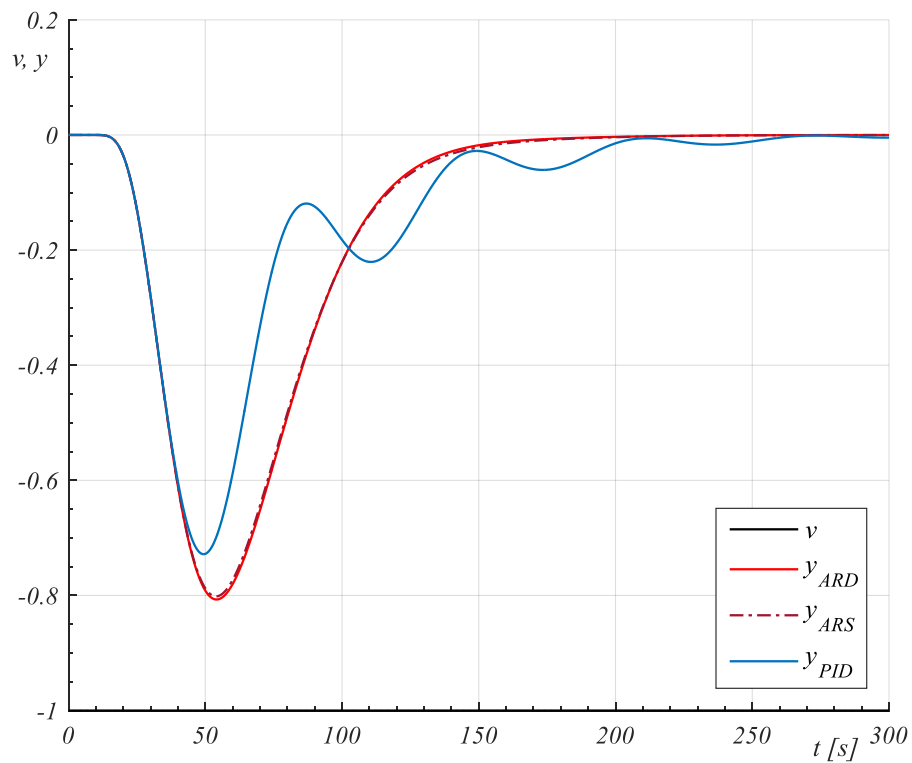
Tabulka 6.3: Parametry PID regulátoru – metoda Ziegler-Nichols – simulace

$r_k [-]$	$T_k [s]$	$r_0 [-]$	$T_i [s]$	$T_d [s]$
1,60	75,20	0,96	37,60	9,40

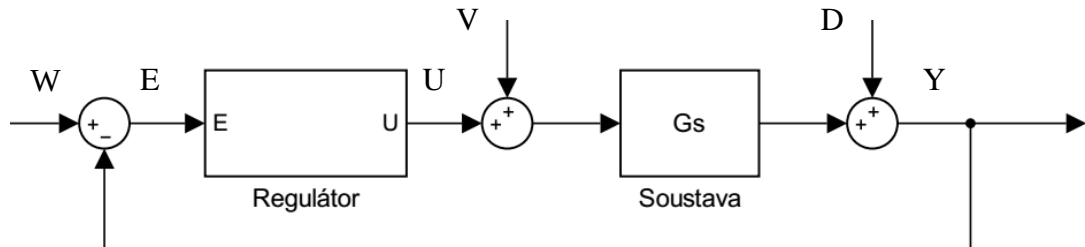
Na Obr. 6.7 je vidět srovnání přechodových charakteristik simulovaného uzavřeného regulačního obvodu s PID regulátorem, se spojitým anisochronním regulátorem a s diskretizovaným anisochronním regulátorem. Z průběhů přechodových charakteristik je vidět, že soustava s anisochronním regulátorem se ustálí rychleji na žádané hodnotě a průběh regulované veličiny je méně kmitavý než s PID regulátorem seřízeným metodou kritického zesílení Zieglera a Nicholse.

Na Obr. 6.8 je potom srovnání přechodových charakteristik stejného uzavřeného regulačního obvodu při použití anisochronního regulátoru s relativním překmitem nastaveným na  $\kappa=0\%$ , tzn. bez překmitu.

Obr. 6.7: Srovnání přechodových charakteristik – simulace ( $\kappa=5\%$ )Obr. 6.8: Srovnání přechodových charakteristik – simulace ( $\kappa=0\%$ )

Obr. 6.9: Srovnání reakce na jednotkový skok poruchové veličiny  $v$  – simulace ( $\kappa=5\%$ )Obr. 6.10: Srovnání reakce na jednotkový skok poruchové veličiny  $v$  – simulace ( $\kappa=0\%$ )

Na obrázcích Obr. 6.9 a Obr. 6.10 je vidět srovnání reakcí uzavřených regulačních obvodů s PID regulátorem a s anisochronním regulátorem (Obr. 6.11) na jednotkový skok poruchové veličiny  $v$ . U anisochronního regulátoru je vidět větší vychýlení regulované veličiny než u PID regulátoru, ale anisochronní regulátor vyreguluje poruchu rychleji než PID regulátor.



Obr. 6.11: Uzavřený regulační obvod – simulace

### 6.3.2. Simulace na anisochronním modelu soustavy TVM

Další simulace byla provedena na anisochronním modelu získaném identifikací soustavy TVM. Parametry PID regulátoru byly nastaveny podle simulované dvupolohové regulace metodou Ziegler-Nichols.

Tabulka 6.4: Parametry anisochronního modelu TVM

$K[-]$	$\tau_u[s]$	$\tau_1[s]$	$\tau_2[s]$	$\tau_y[s]$
1,6	0,9	0,7	8,5	1,4

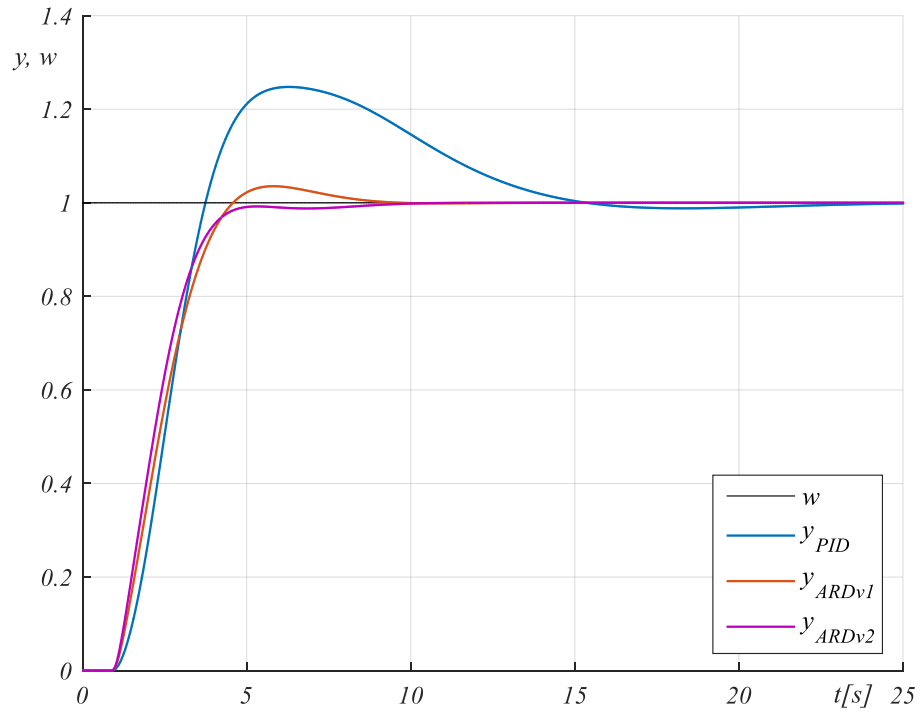
Tabulka 6.5: Parametry PID regulátoru získané metodou Z-N ze simulovaného modelu TVM

$r_k[-]$	$T_k[s]$	$r_0[-]$	$T_i[s]$	$T_d[s]$
4,05	7,76	2,43	3,88	0,97

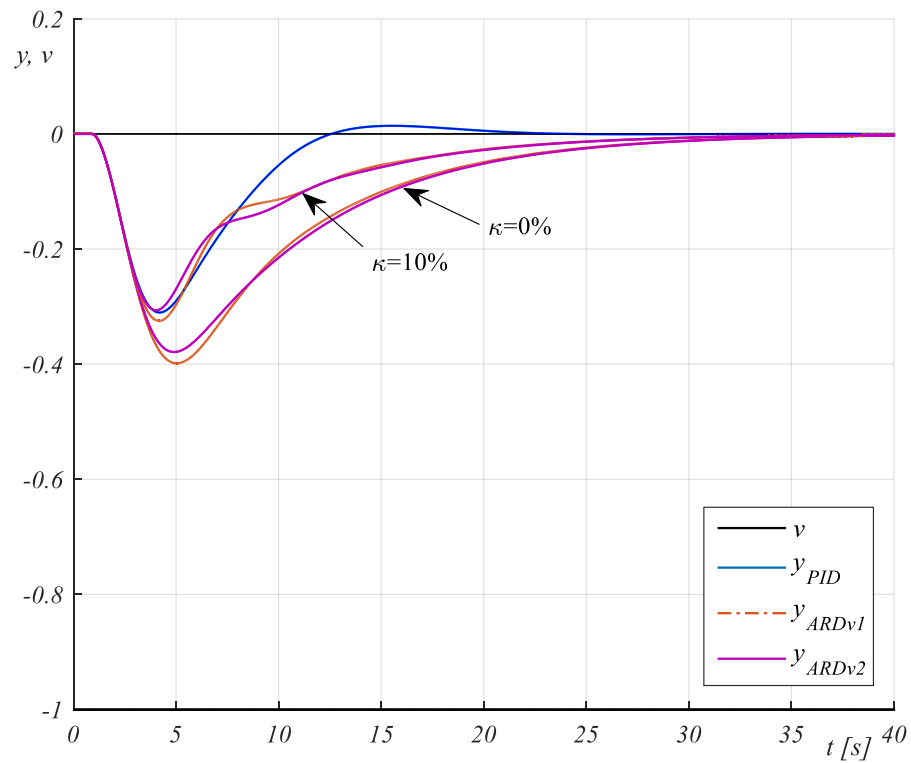
Tabulka 6.6: Parametry anisochronního regulátoru pro simulovaný model TVM

$T[s]$	$T_f[s]$	$k_0[-]$	$\kappa[\%]$
0,20	0,20	0,4088	0,00

Přechodová charakteristika uzavřeného regulačního obvodu je vidět na Obr. 6.12. Z obrázku je vidět, že v simulaci vychází průběh přechodové charakteristiky URO s anisochronním regulátorem lépe, než s PID regulátorem seřízeným metodou Ziegler-Nichols.



Obr. 6.12: Srovnání přechodových charakteristik simulovaného uzavřeného regulačního obvodu s modelem TVM



Obr. 6.13: Srovnání reakce na jednotkový skok poruchové veličiny  $v$  (model TVM)

Na Obr. 6.13 je vidět srovnání reakcí uzavřených regulačních obvodů s PID regulátorem a s anisochronním regulátorem na jednotkový skok poruchové veličiny  $v$ . U anisochronního regulátoru je vidět větší vychýlení regulované veličiny  $y$  a delší doba regulování. U anisochronního regulátoru s nastaveným požadovaným relativním překmitem  $\kappa=10\%$  je vidět menší vychýlení regulované veličiny, způsobené poruchou  $v$ , než u anisochronního regulátoru nastaveného na  $\kappa=0\%$ .

## 6.4. Srovnání na reálné soustavě

Jako reálná soustava pro porovnání obou regulátorů (PID regulátoru z knihovny ModelLib pro Mosaic a navrženého anisochronního regulátoru) byl vybrán teplovzdušný model z laboratoře 111 Fakulty Strojní ČVUT v Praze, který je popsán v podkapitole 6.2.

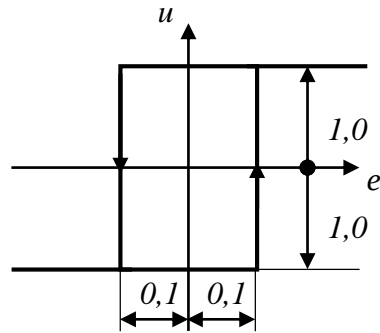
Za regulovanou veličinu byl zvolen průtok vzduchu tunelem teplovzdušného modelu, který je měřen vrtulkovým průtokoměrem na konci tunelu. Výstupem průtokoměru je napětí v rozsahu 0-10V. Jako akční člen slouží ventilátor, který nasává okolní vzduch do tunelu. Ventilátor je ovládán napětím v rozsahu 0-10V.

Pracovní bod byl zvolen v části statické charakteristiky teplovzdušného modelu, ve které je tento model přibližně lineární:

$$\begin{aligned} u_0 &= 2,0 \text{ V} , \\ y_0 &= 4,5 \text{ V} . \end{aligned} \tag{6.13}$$

### 6.4.1. Parametry PID regulátoru

Parametry pro nastavení použitého PID regulátoru byly zjištěny pomocí metody Ziegler-Nichols. Hodnoty kritického zesílení  $r_k$  a kritické periody  $T_k$  byly stanoveny z průběhu dvoupolohové regulace symetrickým relé se symetrickou hysterezí metodou Åströma a Hägglunda.

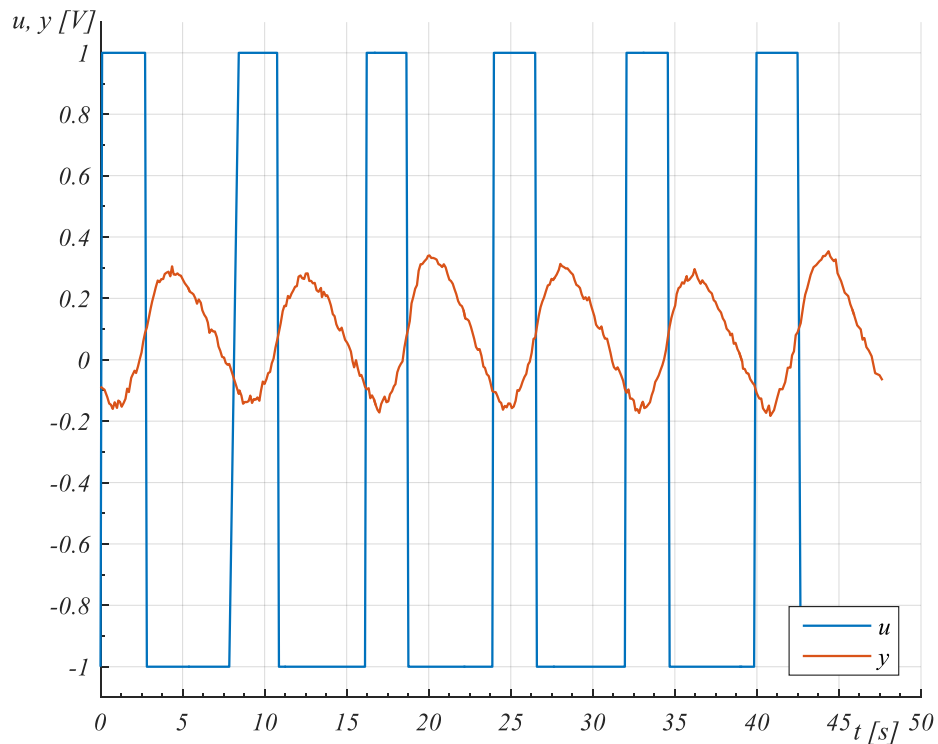


Obr. 6.14: Charakteristika použité dvupolohové regulace

Průběh dvupolohové regulace použité k určení hodnot potřebných ke stanovení parametrů je na Obr. 6.15. Použité parametry PID regulátoru jsou v tabulce 6.7.

Tabulka 6.7: Parametry PID regulátoru pro řízení průtoku vzduchu TVM

$r_0 [-]$	$T_i [s]$	$T_d [s]$	$T [s]$	$T_f [s]$
3,14	4,00	1,00	0,10	0,25



Obr. 6.15: Průběh dvupolohové regulace pro identifikaci parametrů PID regulátoru pro řízení průtoku vzduchu TVM



### 6.4.2. Parametry anisochronního regulátoru

Parametry navrženého anisochronního regulátoru byly určeny ze dvou bodů frekvenční charakteristiky a statické citlivosti změřených na soustavě TVM.

$$\begin{aligned}
 \omega_1 &= 1,164 \\
 \omega_2 &= 0,582 \\
 K &= 1,61 \\
 G(j\omega_1) &= 0,35 \cdot e^{-j2,328} \\
 G(j\omega_2) &= 0,13 \cdot e^{-j3,492}
 \end{aligned} \tag{6.14}$$

Parametry anisochronního modelu byly vypočítány pomocí skriptu pro Matlab, který využívá optimalizační funkci `fmincon`. Skript poskytl prof. Milan Hofreiter.

Tabulka 6.8: Parametry anisochronního modelu získané z frekvenční charakteristiky TVM

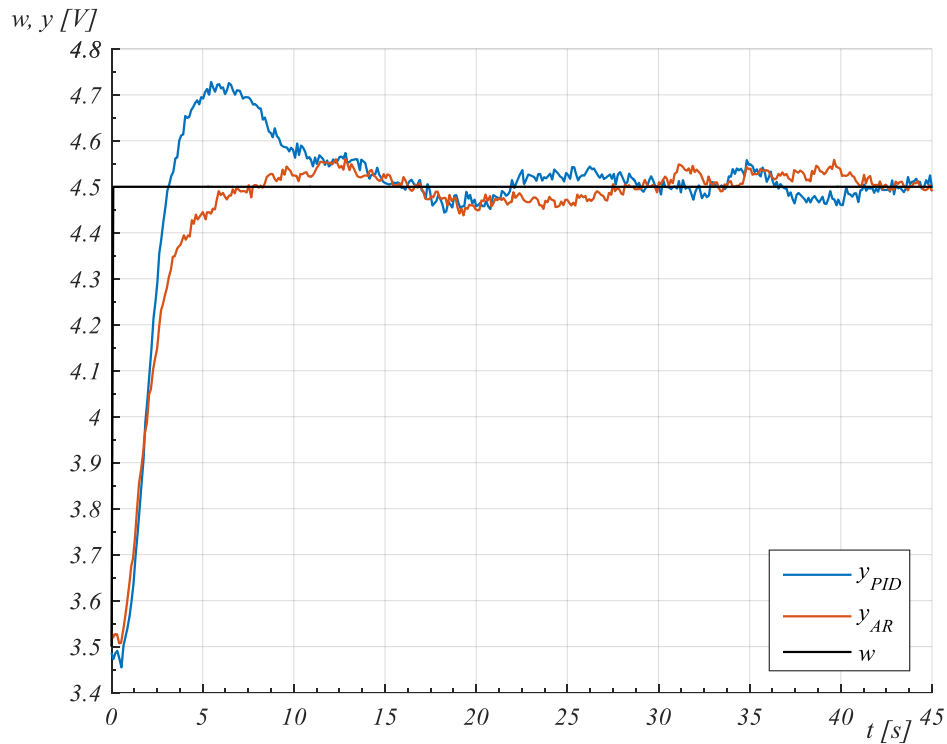
$K[-]$	$\tau_u[s]$	$\tau_1[s]$	$\tau_2[s]$	$\tau_y[s]$
1,61	0,9209	0,6855	8,5185	1,4283

Pro nastavení regulátoru byly dále zvoleny parametry:

$$\begin{aligned}
 v_T &= 35 \\
 v_{\tau_f} &= 1 \\
 \kappa &= 0\%
 \end{aligned} \tag{6.15}$$

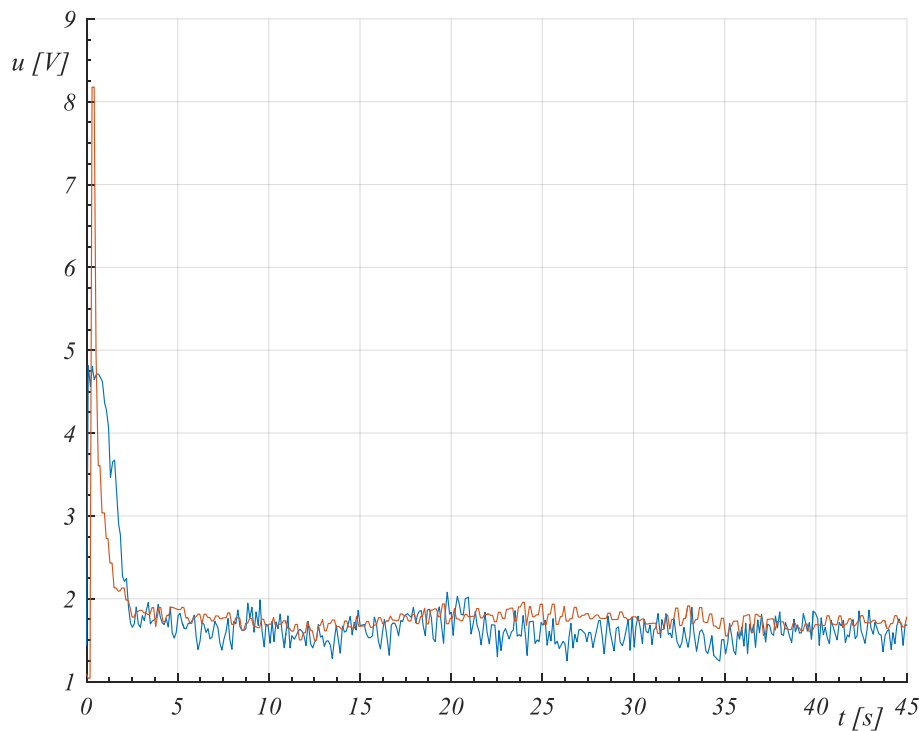
### 6.4.3. Srovnání obou regulátorů

Srovnání přechodových charakteristik uzavřeného regulačního obvodu je vidět na Obr. 6.16. Jednotkový skok pro porovnání byl zvolen z žádané hodnoty regulované veličiny z  $w=3,5$  V na  $w=4,5$  V. Z průběhů charakteristik je vidět, že navržený regulátor nemá takový překmit jako PID regulátor seřízený metodou Ziegler-Nichols a na požadovanou hodnotu zreguluje soustavu o něco rychleji.



Obr. 6.16: Přejchodová charakteristika uzavřeného regulačního obvodu – srovnání navrženého anisochronního regulátoru a PID regulátoru seřízeného metodou Ziegler-Nichols na TVM

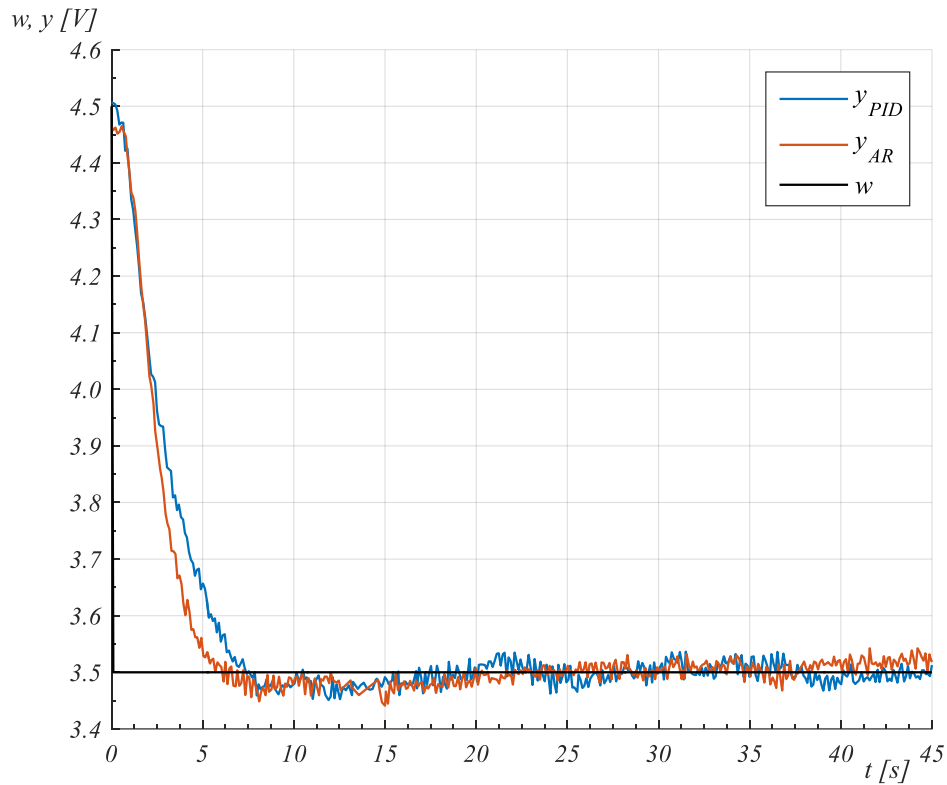
Na Obr. 6.17 je potom vidět srovnání akčních zásahů obou regulátorů vyvolaných jednotkovým skokem žádané hodnoty regulované veličiny. Z obrázku je vidět, že počáteční skok hodnoty akční veličiny u navrženého anisochronního regulátoru je přibližně dvakrát větší než skok vyvolaný použitým PID regulátorem.



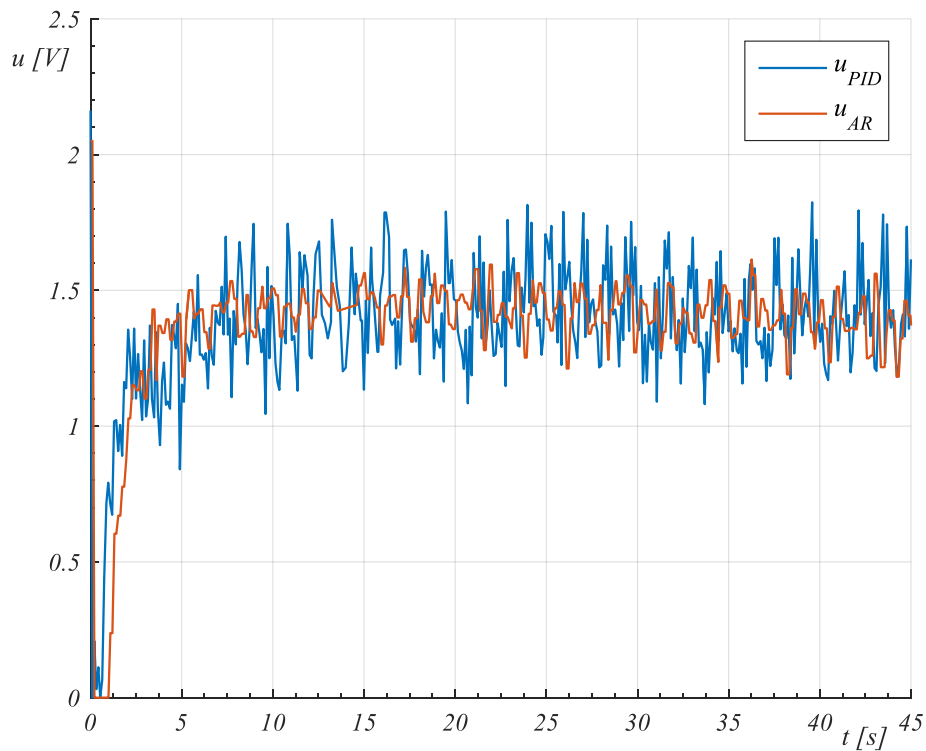
Obr. 6.17: Akční zásahy regulátorů v reakci na jednotkový skok žádané hodnoty – srovnání navrženého anisochronního regulátoru a PID regulátoru seřízeného metodou Ziegler-Nichols na TVM

Na dalších obrázcích (Obr. 6.18 a Obr. 6.19) je vidět porovnání přechodových charakteristik a akčních zásahů obou regulátorů pro jednotkový skok žádané hodnoty regulované veličiny z  $w=4,5$  V na  $w=3,5$  V.

Z grafů je vidět, že oba regulátory zregulovaly regulovanou veličinu na požadovanou hodnotu přibližně stejně rychle a jejich průběhy jsou téměř identické. Z Obr. 6.19 je vidět, že anisochronní regulátor narazil na omezení akční veličiny  $u=0$  V.



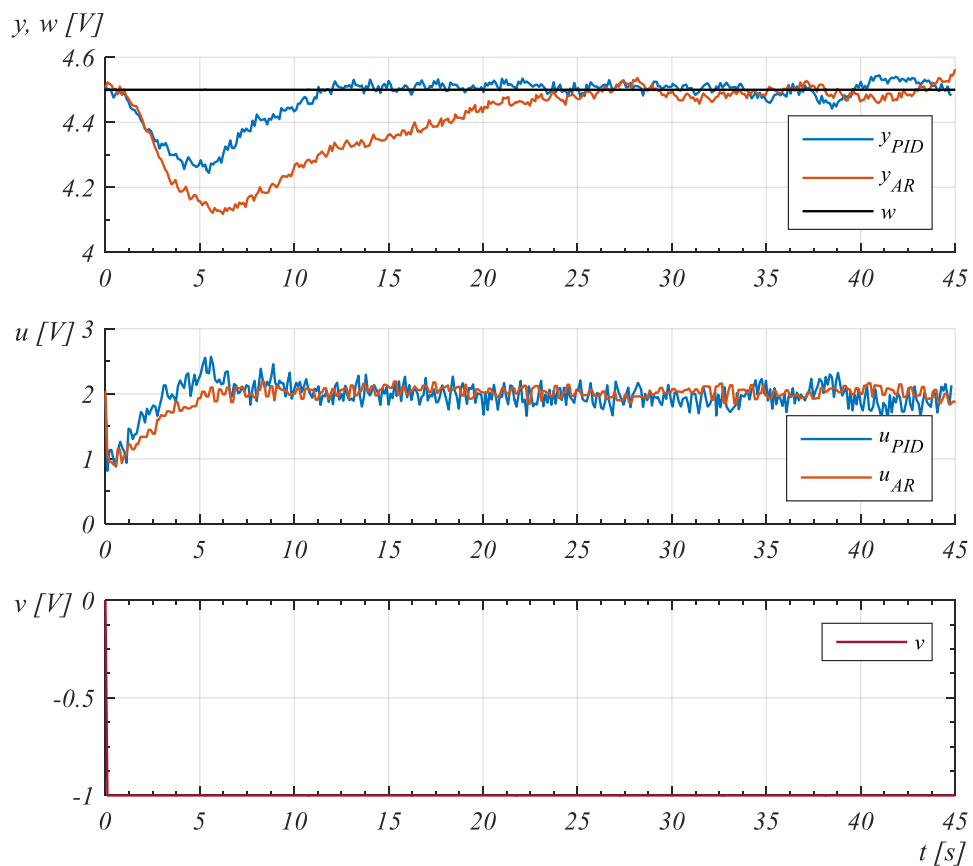
Obr. 6.18: Přechodová charakteristika uzavřeného regulačního obvodu – srovnání navrženého anisochronního regulátoru a PID regulátoru seřízeného metodou Ziegler-Nichols na TVM



Obr. 6.19: Akční zásahy regulátorů v reakci na jednotkový skok žádané hodnoty – srovnání navrženého anisochronního regulátoru a PID regulátoru seřízeného metodou Ziegler-Nichols na TVM

Na Obr. 6.20 je vidět reakce uzavřeného regulačního obvodu na skokovou změnu poruchy  $v$ . Ze srovnání průběhů regulovaných veličin u obou regulátorů je vidět, že anisochronní regulátor poruchu vyreguloval později a s větším vychýlením regulované veličiny od žádané hodnoty než PID regulátor seřízený metodou Ziegler-Nichols.

Pravděpodobnou příčinou tak velké reakce anisochronního regulátoru na skokovou změnu poruchové veličiny je absence integračního členu v přenosu regulátoru, který by reagoval na změny regulované veličiny bez zpoždění.



Obr. 6.20: Vliv jednotkového skoku poruchové veličiny  $v$  na uzavřený regulační obvod s PID a anisochronním regulátorem

## 7. Závěr

### 7.1. Úkoly práce

Úkolem této práce bylo

- seznámit se s metodou „požadovaného modelu“ a regulačním systémem Tecomat Foxtrot,
- vytvořit programový modul pro systém Tecomat s využitím metody „požadovaného modelu“ pro řízení soustavy popsané daným anisochronním modelem,
- na simulačních příkladech a na zvolené reálné soustavě ověřit vytvořené řízení a porovnat jej s řízením PID regulátorem seřizovaným dle metody Ziegler-Nichols.

V této práci byl s využitím metody „požadovaného modelu“ navržen anisochronní regulátor pro řízení soustavy popsané spojitým anisochronním modelem

$$G_a(s) = \frac{K \cdot e^{-s\tau_u}}{(\tau_1 s + 1)(\tau_2 s + e^{-s\tau_y})}. \quad (7.1)$$

Navržený regulátor s přenosem

$$G_C(s) = \frac{k_0 (\tau_1 s + 1)(\tau_2 s + e^{-s\tau_y})}{K s}, \quad (7.2)$$

byl dále diskretizován za použití Tustinovy aproximace, která zachovává stabilitu, a doplněn o filtr prvního řádu. Po otestování na simulacích a reálné úloze byla navržena druhá verze anisochronního regulátoru, u které se filtruje pouze derivační složka přenosu, aby se snížil nepříznivý vliv filtru na relativní překmit přechodové charakteristiky uzavřeného regulačního obvodu.

Regulátor byl poté naprogramován jako funkční blok jazyka ST podle normy IEC EN 61131-3 pro programování PLC a dále doplněn o funkci omezení integrace při překročení dovolených mezí velikosti akčního zásahu.

Ověření funkce navrženého anisochronního regulátoru bylo provedeno jak na simulačních příkladech, tak na reálné soustavě.

Z porovnání přechodových charakteristik URO s PID regulátorem seřízeným metodou Ziegler-Nichols, na zvolené simulované soustavě šestého řádu s dopravním zpožděním i na simulovaném anisochronním modelu TVM, vyšel anisochronní regulátor jako lepší (zreguloval regulovanou veličinu rychleji na žádanou hodnotu s menším překmitem).

Při srovnání reakcí simulovaného URO na skokovou změnu poruchové veličiny  $v$  měla regulovaná veličina při regulaci anisochronním regulátorem větší výchylku než URO s PID regulátorem. Doba vyregulování poruchy byla na simulované soustavě šestého řádu kratší s anisochronním regulátorem než při použití PID regulátoru. U simulovaného anisochronního modelu soustavy TVM trvalo vyregulování poruchy déle anisochronnímu regulátoru.

Při porovnání přechodových charakteristik uzavřených regulačních obvodů s oběma regulátory na zvolené reálné soustavě (TVM) nebyly rozdíly mezi anisochronním řízením a řízením PID regulátorem seřízeným metodou Ziegler-Nichols tak velké jako v simulačních příkladech. Anisochronní regulátor s nastaveným požadovaným relativním překmitem  $\kappa=0\%$  ale dosáhl požadované hodnoty regulované veličiny s výrazně menším překmitem než použitý PID regulátor.

Při srovnání reakcí regulačních obvodů s oběma regulátory na skokovou změnu poruchové veličiny  $v$  na soustavě TVM trvalo vyregulování poruchy anisochronním regulátorem přibližně dvakrát déle než PID regulátorem, stejně tak vychýlení regulované veličiny od žádané hodnoty bylo u obvodu s anisochronním regulátorem přibližně dvakrát větší. Pravděpodobnou příčinou horší reakce na skokovou změnu poruchové veličiny při použití anisochronního regulátoru je absence integračního členu uvnitř jeho přenosu, který by reagoval na změny hodnot regulované veličiny bez zpoždění.

## 7.2. Výhled do budoucna

Anisochronní řízení se jeví jako vhodné pro řízení soustav s dopravním zpožděním, jelikož může reagovat nejen na současné hodnoty regulační odchylky, ale i na ty minulé.

Za předpokladu automatizace identifikace anisochronního modelu, například za použití rozšířené reléové identifikace navržené v článku [2], by se anisochronní regulátory s auto-tuningem mohly uplatnit v průmyslových aplikacích.



## Seznam použité literatury

- [1] HOFREITER, Milan. Plant identification from *relay feedback and anisochronic* controller design. In: Recent Advances in Systems: Proceedings of the 19th International Conference on Systems (part of CSCC '15). Zakynthos Island, Greece, 2015, s. 271-275. ISBN 978-1-61804-321-4. ISSN 1790-5117.
- [2] HOFREITER, Milan. Extension of relay feedback identification. In: 2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). IEEE, 2015, s. 61-66. DOI: 10.1109/ICUMT.2015.7382406. ISBN 978-1-4673-9283-9. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7382406>
- [3] HOFREITER, Milan, Pavel TRNKA, Martin HUNČOVSKÝ a Petr SIEGEL. Extended Relay Feedback Identification and Anisochronic Control. In: *International Journal of Control Systems and Robotics*. 2016, s. 57-62. ISSN 2367-8917.
- [4] VÍTEČEK, Antonín a Miluše VÍTEČKOVÁ. Zpětnovazební řízení mechatronických systémů. 1. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2013. ISBN 978-80-248-3232-6.
- [5] HOFREITER, Milan. Identifikace systémů I. Vyd. 1. V Praze: České vysoké učení technické, 2009. ISBN 978-80-01-04228-1.
- [6] Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic [online]. Desáté vydání - listopad 2007. Kolín: Teco, 2007 [cit. 2016-03-15]. TXV00321. Dostupné z: [http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00321\\_01\\_Mosaic\\_ProgIEC\\_cz.pdf](http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00321_01_Mosaic_ProgIEC_cz.pdf)
- [7] ZAČÍNÁME V PROSTŘEDÍ MOSAIC [online]. 8. vydání - duben 2010. Kolín: Teco, 2010 [cit. 2016-03-15]. TXV00320. Dostupné z: [http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00320\\_01\\_Mosaic\\_ProgStar\\_cz.pdf](http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00320_01_Mosaic_ProgStar_cz.pdf)
- [8] PROGRAMOVATELNÉ AUTOMATY TECOMAT FOXTROT CP-1005, CP-1015 [online]. 7. vydání - říjen 2014. Kolín: Teco, 2014 [cit. 2016-03-15]. TXV00435. Dostupné z: [http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00435\\_01\\_Foxtrot\\_CP\\_1005.pdf](http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00435_01_Foxtrot_CP_1005.pdf)
- [9] Knihovna pro modelování procesů [online]. Čtvrté vydání. Kolín: Teco, 2012 [cit. 2016-04-24]. TXV 003 44.01. Dostupné z: [http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00344\\_01\\_Mosaic\\_ModelLib\\_cz.pdf](http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00344_01_Mosaic_ModelLib_cz.pdf)

# Seznam obrázků

OBR. 2.1: VLIV ZESÍLENÍ OTEVŘENÉHO REGULAČNÍHO OBVODU $K_0$ NA PRŮBĚH PŘECHODOVÉ CHARAKTERISTIKY REGULAČNÍHO OBVODU [4] .....	3
OBR. 2.2: SCHÉMA UZAVŘENÉHO REGULAČNÍHO OBVODU S RELÉ.....	7
OBR. 2.3: PŘÍKLAD DEKLARACE PROMĚNNÝCH POU, PŘEVZATO Z [6].....	10
OBR. 2.4: UKÁZKA PROGRAMOVACÍHO PROSTŘEDÍ MOSAIC .....	11
OBR. 3.1: POUŽITÝ MODUL TECOMAT FOXTROT CP-1015 S NAPÁJECÍM ZDROJEM (VPRAVO) .....	14
OBR. 3.2: ZÁKLADNÍ MODUL TECOMAT FOXTROT CP-1015 [8] .....	15
OBR. 4.1: SCHÉMA UZAVŘENÉHO REGULAČNÍHO OBVODU .....	16
OBR. 4.2: VLIV VZORKOVACÍ PERIODY $T$ NA PRŮBĚH PŘECHODOVÉ CHARAKTERISTIKY UZAVŘENÉHO REGULAČNÍHO OBVODU – „RYCHLÁ“ SOUSTAVA .....	21
OBR. 4.3: VLIV ČASOVÉ KONSTANTY FILTRU $T_F$ NA PRŮBĚH PŘECHODOVÉ CHARAKTERISTIKY UZAVŘENÉHO REGULAČNÍHO OBVODU – „RYCHLÁ“ SOUSTAVA .....	22
OBR. 4.4: VLIV VZORKOVACÍ PERIODY $T$ NA PRŮBĚH PŘECHODOVÉ CHARAKTERISTIKY UZAVŘENÉHO REGULAČNÍHO OBVODU – „POMALÁ“ SOUSTAVA .....	23
OBR. 4.5: VLIV ČASOVÉ KONSTANTY FILTRU $T_F$ NA PRŮBĚH PŘECHODOVÉ CHARAKTERISTIKY UZAVŘENÉHO REGULAČNÍHO OBVODU – „POMALÁ“ SOUSTAVA.....	23
OBR. 4.6: VLIV ČASOVÉ KONSTANTY FILTRU $T_F$ NA PRŮBĚH PŘECHODOVÉ CHARAKTERISTIKY UZAVŘENÉHO REGULAČNÍHO OBVODU S UPRAVENÝM REGULÁTOREM .....	25
OBR. 5.1: JEDNODUCHÉ ROZHRAŇÍ PRO ZADÁVÁNÍ ŽÁDANÉ HODNOTY REGULOVANÉ VELIČINY .....	33
OBR. 5.2: TEST ANISOCHRONNÍHO REGULÁTORU NA TEPLOVZDUŠNÉM MODELU, GRAF Z PROGRAMOVACÍHO PROSTŘEDÍ MOSAIC .....	34
OBR. 6.1: SCHÉMA TEPLOVZDUŠNÉHO MODELU (VSTUPY A VÝSTUPY Z POHLEDU PLC).....	36
OBR. 6.2: SCHÉMA ZAPOJENÍ KONEKTORU K PLC (NEBYLY VYUŽITY VŠECHNY VSTUPY/VÝSTUPY) .....	37
OBR. 6.3: NAMĚŘENÉ BODY STATICKE CHARAKTERISTIKY TVM (VENTILÁTOR $U_s$ , PRŮTOKOMĚR $Y_s$ ) ...	37
OBR. 6.4: NAMĚŘENÉ OTÁČKY ODPOVÍDAJÍCÍ NAPĚTÍ NA VENTILÁTORU A PRŮTOKOMĚRU.....	38
OBR. 6.5: FOTOGRAFIE TEPLOVZDUŠNÉHO MODELU.....	38
OBR. 6.6: NYQUISTOVA FREKVENČNÍ CHARAKTERISTIKA – SIMULACE.....	40
OBR. 6.7: SROVNÁNÍ PŘECHODOVÝCH CHARAKTERISTIK – SIMULACE ( $K=5\%$ ).....	42
OBR. 6.8: SROVNÁNÍ PŘECHODOVÝCH CHARAKTERISTIK – SIMULACE ( $K=0\%$ ).....	42
OBR. 6.9: SROVNÁNÍ REAKCE NA JEDNOTKOVÝ SKOK PORUCHOVÉ VELIČINY $v$ – SIMULACE ( $K=5\%$ ) ...	43
OBR. 6.10: SROVNÁNÍ REAKCE NA JEDNOTKOVÝ SKOK PORUCHOVÉ VELIČINY $v$ – SIMULACE ( $K=0\%$ ) .	43
OBR. 6.11: UZAVŘENÝ REGULAČNÍ OBVOD – SIMULACE .....	44
OBR. 6.12: SROVNÁNÍ PŘECHODOVÝCH CHARAKTERISTIK SIMULOVANÉHO UZAVŘENÉHO REGULAČNÍHO OBVODU S MODELEM TVM.....	45
OBR. 6.13: SROVNÁNÍ REAKCE NA JEDNOTKOVÝ SKOK PORUCHOVÉ VELIČINY $v$ (MODEL TVM) .....	45
OBR. 6.14: CHARAKTERISTIKA POUŽITÉ DVOUPOLOHOVÉ REGULACE .....	47
OBR. 6.15: PRŮBĚH DVOUPOLOHOVÉ REGULACE PRO IDENTIFIKACI PARAMETRŮ PID REGULÁTORU PRO ŘÍZENÍ PRŮTOKU VZDUCHU TVM.....	47

OBR. 6.16: PŘECHODOVÁ CHARAKTERISTIKA UZAVŘENÉHO REGULAČNÍHO OBVODU – SROVNÁNÍ NAVRŽENÉHO ANISOCHRONNÍHO REGULÁTORU A PID REGULÁTORU SEŘÍZENÉHO METODOU ZIEGLER-NICHOLS NA TVM .....	49
OBR. 6.17: AKČNÍ ZÁSADY REGULÁTORŮ V REAKCI NA JEDNOTKOVÝ SKOK ŽÁDANÉ HODNOTY – SROVNÁNÍ NAVRŽENÉHO ANISOCHRONNÍHO REGULÁTORU A PID REGULÁTORU SEŘÍZENÉHO METODOU ZIEGLER-NICHOLS NA TVM.....	50
OBR. 6.18: PŘECHODOVÁ CHARAKTERISTIKA UZAVŘENÉHO REGULAČNÍHO OBVODU – SROVNÁNÍ NAVRŽENÉHO ANISOCHRONNÍHO REGULÁTORU A PID REGULÁTORU SEŘÍZENÉHO METODOU ZIEGLER-NICHOLS NA TVM .....	51
OBR. 6.19: AKČNÍ ZÁSADY REGULÁTORŮ V REAKCI NA JEDNOTKOVÝ SKOK ŽÁDANÉ HODNOTY – SROVNÁNÍ NAVRŽENÉHO ANISOCHRONNÍHO REGULÁTORU A PID REGULÁTORU SEŘÍZENÉHO METODOU ZIEGLER-NICHOLS NA TVM.....	51
OBR. 6.20: VLIV JEDNOTKOVÉHO SKOKU PORUCHOVÉ VELIČINY V NA UZAVŘENÝ REGULAČNÍ OBVOD S PID A ANISOCHRONNÍM REGULÁTOREM.....	52

# Seznam tabulek

TABULKA 2.1: HODNOTY KOEFICIENTU B PRO POŽADOVANÝ RELATIVNÍ PŘEKMIT K [4] .....	4
TABULKA 2.2: STAVITELNÉ PARAMETRY ANALOGOVÝCH REGULÁTORŮ PRO ZIEGLEROVU-NICHOLSOVU METODU KRITICKÝCH PARAMETRŮ [4].....	6
TABULKA 3.1: ZÁKLADNÍ PARAMETRY SYSTÉMU TECOMAT FOXTROT [8] .....	13
TABULKA 3.2: PROVOZNÍ PODMÍNKY SYSTÉMU TECOMAT FOXTROT [8].....	13
TABULKA 3.3: ZÁKLADNÍ PARAMETRY MODULU CP-1015 [8].....	15
TABULKA 6.1: PARAMETRY ANISOCHRONNÍHO MODELU SOUSTAVY – SIMULACE .....	39
TABULKA 6.2: PARAMETRY ANISOCHRONNÍHO REGULÁTORU – SIMULACE .....	41
TABULKA 6.3: PARAMETRY PID REGULÁTORU – METODA ZIEGLER-NICHOLS – SIMULACE .....	41
TABULKA 6.4: PARAMETRY ANISOCHRONNÍHO MODELU TVM .....	44
TABULKA 6.5: PARAMETRY PID REGULÁTORU ZÍSKANÉ METODOU Z-N ZE SIMULOVANÉHO MODELU TVM.....	44
TABULKA 6.6: PARAMETRY ANISOCHRONNÍHO REGULÁTORU PRO SIMULOVANÝ MODEL TVM .....	44
TABULKA 6.7: PARAMETRY PID REGULÁTORU PRO ŘÍZENÍ PRŮTOKU VZDUCHU TVM .....	47
TABULKA 6.8: PARAMETRY ANISOCHRONNÍHO MODELU ZÍSKANÉ Z FREKVENČNÍ CHARAKTERISTIKY TVM.....	48

# Seznam rovníc

(1.1) .....	1
(2.1) .....	2
(2.2) .....	2
(2.3) .....	2
(2.4) .....	2
(2.5) .....	2
(2.6) .....	3
(2.7) .....	3
(2.8) .....	4
(2.9) .....	4
(2.10).....	4
(2.11).....	4
(2.12).....	4
(2.13).....	5
(2.14).....	5
(2.15).....	5
(2.16).....	5
(2.17).....	5
(2.18).....	5
(2.19).....	6
(4.1) .....	16
(4.2) .....	16
(4.3) .....	17
(4.4) .....	17
(4.5) .....	17
(4.6) .....	17
(4.7) .....	18
(4.8) .....	18
(4.9) .....	18
(4.10).....	18
(4.11).....	18
(4.12).....	18
(4.13).....	18
(4.14).....	18
(4.15).....	18
(4.16).....	19
(4.17).....	19
(4.18).....	19

(4.19).....	19
(4.20).....	19
(4.21).....	19
(4.22).....	19
(4.23).....	19
(4.24).....	20
(4.25).....	20
(4.26).....	21
(4.27).....	22
(4.28).....	24
(4.29).....	24
(4.30).....	24
(4.31).....	24
(4.32).....	24
(4.33).....	24
(4.34).....	24
(4.35).....	25
(4.36).....	25
(6.1) .....	35
(6.2) .....	35
(6.3) .....	35
(6.4) .....	35
(6.5) .....	35
(6.6) .....	39
(6.7) .....	39
(6.8) .....	40
(6.9) .....	40
(6.10).....	41
(6.11).....	41
(6.12).....	41
(6.13).....	46
(6.14).....	48
(6.15).....	48
(7.1) .....	53
(7.2) .....	53

## Seznam zkratek a symbolů

$a_0$ až $a_2$	Parametry anisochronního regulátoru
ARD	Diskrétní anisochronní regulátor
ARS	Spojité anisochronní regulátor
$b_0$ až $b_5$	Parametry anisochronního regulátoru
$d$	Poruchová veličina
$D$	Parametr anisochronního regulátoru
$e$	Regulační odchylka
FBD	Function Block Diagram
$G_0$	Přenos otevřeného regulačního obvodu
$G_a$	Přenos anisochronního modelu
$G_C$	Přenos regulátoru
$G_P$	Přenos soustavy
$G_{wy}$	Přenos žádané hodnoty na regulovanou veličinu (přenos řízení)
HW	Hardware
IL	Instruction List
$Im$	Imaginární část komplexního čísla
$j$	Imaginární jednotka
$k$	Diskrétní čas
$K$	Statická citlivost
$k_0$	Zesílení otevřeného regulačního obvodu
LD	Ladder Diagram
MPM	Metoda Požadovaného Modelu
PID	Proporcionální, Integrační, Derivační
PLC	Programmable Logic Controller
POU	Program Organization Unit
$r$	Řád filtru anisochronního regulátoru
$r_0$	Proporcionální konstanta PID regulátoru
$Re$	Reálná část komplexního čísla
$r_k$	Kritické zesílení (Ziegler-Nichols)
$s$	Argument Laplaceova obrazu
ST	Structured Text
SW	Software
$t$	Čas
$T$	Vzorkovací perioda
$T_d$	Derivační časová konstanta PID regulátoru
$T_f$	Časová konstanta filtrace derivační složky PID regulátoru
$T_i$	Integrační časová konstanta PID regulátoru
$T_k$	Kritická perioda (Ziegler-Nichols)
TVM	Teplovzdušný model
$u$	Akční veličina
$u_0$	Hodnota akční veličiny v pracovním bodě
$u_a$	Amplituda akční veličiny při dvupolohové regulaci
URO	Uzavřený Regulační Obvod
$v$	Poruchová veličina
$v_T$	Parametr anisochronního regulátoru pro volbu T
$v_f$	Parametr anisochronního regulátoru pro volbu $\tau_f$
$w$	Žádaná hodnota regulované veličiny
$y$	Regulovaná veličina

$y_0$	Hodnota regulované veličiny v pracovním bodě
$y_a$	Amplituda regulované veličiny při dvupolohové regulaci
$y_f$	Filtrovaný vstup derivační složky PID regulátoru
$y_{IDEAL}$	Regulovaná veličina při požadovaném přenosu řízení z MPM
$z$	Argument Z-obrazu
Z-N	Ziegler-Nichols
$\beta$	Koeficient MPM pro volbu $k_0$
$\kappa$	Relativní překmit
$\tau_1$	Časová konstanta anisochronního modelu
$\tau_2$	Časová konstanta anisochronního modelu
$\tau_f$	Časová konstanta filtru anisochronního regulátoru
$\tau_u$	Dopravní zpoždění
$\tau_y$	Parametr anisochronního modelu
$\omega$	Úhlová frekvence
$x[k]$	Označuje $k$ -tý člen posloupnosti $x$ (hodnotu $x$ v diskrétním čase $k$ )



# Seznam Příloh

<b>PŘÍLOHA A FUNKČNÍ BLOK ANISOCHRONNÍHO REGULÁTORU VERZE 1 .....</b>	<b>1</b>
<b>PŘÍLOHA B FUNKČNÍ BLOK ANISOCHRONNÍHO REGULÁTORU VERZE 2 .....</b>	<b>4</b>
<b>PŘÍLOHA C HLAVNÍ PROGRAM – ANISOCHRONNÍ ŘÍZENÍ.....</b>	<b>7</b>
<b>PŘÍLOHA D HLAVNÍ PROGRAM – PID REGULACE .....</b>	<b>8</b>

# Příloha A Funkční blok anisochronního regulátoru verze 1

## AniRe.st

```
(*  
Název: Anizochronní Řízení  
Autor: Martin Hunčovský  
Datum: 13.03.2016
```

Přenos regulátoru podle spojitého modelu:

$$G(s) = \frac{k_0}{K} * \frac{(T_1*s+1)*(T_2*s+\exp(-s*T_y))}{(T_f*s+1)*s}$$

Diskretizovaný přenos regulátoru:

$$G(z) = \frac{k_0}{K} * \frac{b_0*z^{-2} + b_1*z^{-1} + b_2*z^0 + b_3*z^{-D} + b_4*z^{-D-1} + b_5*z^{-D-2}}{a_0*z^{-2} + a_1*z^{-1} + a_2*z^0}$$

$D = T_y/T$ , kde  $T$  je perioda vzorkování

\*)

```
FUNCTION_BLOCK fbAniRe  
VAR_INPUT  
// Z identifikace  
K : REAL;  
T1 : REAL;  
T2 : REAL;  
Ty : REAL;  
Tu : REAL;  
// Vstupy  
w : REAL;  
y : REAL;  
// Pozadovany relativni prekmit k v %  
kappa : SINT := 10;  
// pro volbu casove konst. filtru Tf, vTf ~ min 10, Tf = (T1+T2-  
Ty)/vTf  
vTf : REAL := 35;  
// pro volbu periody vzorkovani T a zpozdeni D, vT ~ min 10, T <  
(T1+T2-Ty)/vT  
vT : REAL := 35;  
// Reset inicializace  
reset : BOOL := FALSE;  
END_VAR  
VAR_IN_OUT  
END_VAR  
VAR  
// init flag  
init : BOOL := FALSE;  
// Vzorkovaci perioda  
T : REAL := 0.1;  
Tmax : REAL := 5.0;  
// Casova konstanta filtru  
Tf : REAL := 0.25;
```

```

// Pole predchozich hodnot pro bloky zpozdeni
aEkm2      : GenericArrayForDelay;
aEkm1      : GenericArrayForDelay;
aEkmD      : GenericArrayForDelay;
aEkmDm1    : GenericArrayForDelay;
aEkmDm2    : GenericArrayForDelay;
aUkm2      : GenericArrayForDelay;
aUkm1      : GenericArrayForDelay;
// Bloky zpozdeni
Ekm2       : fbDelay;
Ekm1       : fbDelay;
EkmD       : fbDelay;
EkmDm1     : fbDelay;
EkmDm2     : fbDelay;
Ukm2       : fbDelay;
Ukm1       : fbDelay;
// Koef. diskretniho modelu
b0         : REAL;
b1         : REAL;
b2         : REAL;
b3         : REAL;
b4         : REAL;
b5         : REAL;
a0         : REAL;
a1         : REAL;
a2         : REAL;
D          : INT;
k0         : REAL;
// Cas posledniho behu
lastT      : TIME;
END_VAR
VAR_OUTPUT
// Vystup - akcni velicina
u          : REAL := 0.0;
// Vystup - regulacni odchylka
e          : REAL;
// Vystup vzorkovaci perioda
TT         : REAL;
END_VAR

// Reset inicializace
IF reset THEN
    init := FALSE;
    reset := FALSE;
END_IF;

IF NOT init THEN
    // Volba k0 podle pozadovaneho rel. prekmitu k
    k0 := 1.0/Tu/(2.718-0.4547*EXPT(SINT_TO_REAL(kappa), 0.3432));
    // Volba casove konstanty filtru, default: vTf=15
    Tf := (T1+T2-Ty)/vTf;
    //IF Tf <= 0.1 THEN
    //    Tf := 0.1;
    //END_IF;
    // Volba periody vzorkovani a zpozdeni D, default: vT=25
    Tmax := (T1+T2-Ty)/vT;
    D := 1;
    T := Ty/INT_TO_REAL(D);
    WHILE T > Tmax DO
        D := D+1;
        T := Ty/INT_TO_REAL(D);
    
```

```

END_WHILE;
TT:=T;
// Vypocet koef. diskretniho modelu
b0 := 4.0*T1*T2/T/T - 2.0*T2/T;
b1 := -2.0*4.0*T1*T2/T/T;
b2 := 4.0*T1*T2/T/T + 2.0*T2/T;
b3 := 2.0*T1/T + 1.0;
b4 := 2.0;
b5 := 1.0 - 2.0*T1/T;
a0 := 4.0*Tf/T/T - 2.0/T;
a1 := -2.0*4.0*Tf/T/T;
a2 := 4.0*Tf/T/T + 2.0/T;
// Aktualizace regulacni odchylky
e := w-y;
// Inicializace zpozdozacich bloku
Ekm2( In := e, K := 2, T := T, Buffer := aEkm2 );
Ekm1( In := e, K := 1, T := T, Buffer := aEkm1 );
EkmD( In := e, K := D, T := T, Buffer := aEkmD );
EkmDm1(In := e, K := D+1, T := T, Buffer := aEkmDm1);
EkmDm2(In := e, K := D+2, T := T, Buffer := aEkmDm2);
Ukm2( In := u, K := 2, T := T, Buffer := aUkm2 );
Ukm1( In := u, K := 1, T := T, Buffer := aUkm1 );
// Vypocet nove hodnoty akcni veliciny u
u := k0/K/a2*(b0*Ekm2.Out + b1*Ekm1.Out + b2*e + b3*EkmD.Out +
b4*EkmDm1.Out + b5*EkmDm2.Out) - (a0*Ukm2.Out + a1*Ukm1.Out)/a2;
// Nastaveni init flag
init := TRUE;
ELSE
// Aktualizace vstupu zpozdozacich bloku
Ekm2(In := e, Buffer := aEkm2 );
Ekm1(In := e, Buffer := aEkm1 );
EkmD(In := e, Buffer := aEkmD );
EkmDm1(In := e, Buffer := aEkmDm1);
EkmDm2(In := e, Buffer := aEkmDm2);
Ukm2(In := u, Buffer := aUkm2 );
Ukm1(In := u, Buffer := aUkm1 );

// Vzorkovani
IF (gettime()-lastT) > REAL_TO_TIME(T*1000.0) THEN
// Aktualizace regulacni odchylky
e := w-y;
// Vypocet nove hodnoty akcni veliciny u
u := k0/K/a2*(b0*Ekm2.Out + b1*Ekm1.Out + b2*e + b3*EkmD.Out +
b4*EkmDm1.Out + b5*EkmDm2.Out) - (a0*Ukm2.Out + a1*Ukm1.Out)/a2;

lastT := gettime();
END_IF;
END_IF;

END_FUNCTION_BLOCK

```

## Příloha B Funkční blok anisochronního regulátoru verze 2

**anire\_ver\_2.st**

(\*  
Název: Anizochronní Řízení  
Autor: Martin Hunčovský  
Datum: 11.05.2016

Přenos regulátoru podle spojitého modelu:

$$G(s) = \frac{k_0}{K} * \left( \frac{T_1 T_2 s}{T_f s + 1} + T_2 + T_1 \exp(-s T_y) + \frac{1}{s} \exp(-s T_u) \right)$$

Diskretizováno Tustinovou aproximací  $s = \frac{2}{T} \frac{z-1}{z+1}$  a nahrazením  $\exp(-s T_u) = z^{(-T_u/T)}$

\*)

```
FUNCTION_BLOCK fbanire2
VAR_INPUT
// Z identifikace
K          : REAL;
T1         : REAL;
T2         : REAL;
Ty         : REAL;
Tu         : REAL;
// Vstupy
w          : REAL;
y          : REAL;
// Pozadovany relativni prekmit k v %
kappa     : SINT := 10;
// pro volbu casove konst. filtru Tf, vTf ~ min 1, Tf = T/vTf
vTf       : REAL := 1;
// pro volbu periody vzorkovani T a zpozdeni D, vT ~ min 25, T <
(T1+T2-Ty)/vT
vT        : REAL := 35;
// Reset inicializace
reset     : BOOL := FALSE;
// Max, Min hodnoty u
umax      : REAL := 100;
umin      : REAL := -100;
END_VAR
VAR_IN_OUT
END_VAR
VAR
// init flag
init      : BOOL := FALSE;
// Vzorkovaci perioda
T         : REAL := 0.1;
Tmax      : REAL := 1.0;
// Casova konstanta filtru
Tf        : REAL := 0.1;
// Akcni velicina
u         : REAL := 0.0;
// Regulacni odchylka
e         : REAL;
// Pole predchozich hodnot pro bloky zpozdeni
```

```

aEkmD      : GenericArrayForDelay;
aEkmDm1    : GenericArrayForDelay;
// Bloky zpozdeni
ekmD       : fbDelay;
ekmDm1     : fbDelay;
// Promenne pro zpozdeni a pomocne promenne pro vypocet u
u1         : REAL;
u2         : REAL;
u3         : REAL;
u4         : REAL;
ekm1       : REAL := 0.0;
u1km1     : REAL := 0.0;
u4km1     : REAL := 0.0;
// Koef. diskretniho modelu
b0         : REAL;
b1         : REAL;
a0         : REAL;
a1         : REAL;
D          : INT;
k0         : REAL;
// Cas posledniho behu
lastT      : TIME;
// Stop integrace
stop_int   : BOOL;
END_VAR
VAR_OUTPUT
// Vystup - akcni velicina
out_u      : REAL := 0.0;
// Vystup - regulacni odchylka
out_e      : REAL;
// Vystup vzorkovaci perioda
TT         : REAL;
END_VAR

// Reset inicializace
IF reset THEN
    init := FALSE;
    reset := FALSE;
END_IF;

IF NOT init THEN
    // Volba k0 podle pozadovaneho rel. prekmitu k
    k0 := 1.0/Tu/(2.718-0.4547*EXPT(SINT_TO_REAL(kappa), 0.3432));
    // Volba periody vzorkovani a zpozdeni D, default: vT=35
    Tmax := (T1+T2-Ty)/vT;
    D := 1;
    T := Ty/INT_TO_REAL(D);
    WHILE T > Tmax DO
        D := D+1;
        T := Ty/INT_TO_REAL(D);
    END_WHILE;
    TT:=T;
    // Volba casove konstanty filtru, default: vTf=1
    Tf := T/vTf;
    // Vypocet koef. diskretniho modelu
    b0 := 2.0*T1*T2/T;
    b1 := -2.0*T1*T2/T;
    a0 := 1.0+2.0*Tf/T;
    a1 := 1.0-2.0*Tf/T;
    // Aktualizace regulacni odchylky
    e := w-y;

```

```

// Inicializace zpozdozacich bloku
ekmD( In := e, K := D , T := T, Buffer := aEkmD );
ekmDm1(In := e, K := D+1, T := T, Buffer := aEkmDm1);
// Vypocet nove hodnoty akcni veliciny u
u1 := b0/a0*e + b1/a0*ekm1 - a1/a0*ulkm1;
u2 := T2*e;
u3 := T1*ekmD.out;
u4 := T/2.0*(ekmD.out + ekmDm1.out) + u4km1;

ulkm1 := u1;
u4km1 := u4;
ekm1 := e;

u := k0/K*(u1 + u2 + u3 + u4);
// Nastaveni init flag
init := TRUE;
ELSE
// Aktualizace vstupu zpozdozacich bloku
EkmD(In := e, Buffer := aEkmD );
EkmDm1(In := e, Buffer := aEkmDm1);
// Vzorkovani
IF (gettime()-lastT) > REAL_TO_TIME(T*1000.0) THEN
// Aktualizace regulacni odchylky
e := w-y;
// Kontrola integrace
IF (u > umax) OR (u < umin) THEN
stop_int := TRUE;
ELSE
stop_int := FALSE;
END_IF;
// Vypocet nove hodnoty akcni veliciny u
u1 := b0/a0*e + b1/a0*ekm1 - a1/a0*ulkm1;
u2 := T2*e;
u3 := T1*ekmD.out;
IF NOT stop_int THEN
u4 := T/2.0*(ekmD.out + ekmDm1.out) + u4km1;
END_IF;

ulkm1 := u1;
u4km1 := u4;
ekm1 := e;

u := k0/K*(u1 + u2 + u3 + u4);

// Vystupni veliciny a jejich omezeni
out_u := u;
IF u > umax THEN out_u := umax;
END_IF;
IF u < umin THEN out_u := umin;
END_IF;

out_e := e;

// Zapis casu posledniho behu
lastT := gettime();
END_IF;
END_IF;
END_FUNCTION_BLOCK

```

## Příloha C Hlavní program – anisochronní řízení

### prgMain.ST

```
VAR_GLOBAL
// Analogove vstupy
termistor_1 AT %X32 : TAI;
prutokomer AT %X44 : TAI;
termoclanek AT %X56 : TAI;
// Analogove vystupy
zarovka AT %Y8 : TAO_TC6;
ventilator AT %Y18 : TAO_TC6;
// Pracovni bod
u0 : REAL := 2.0;
y0 : REAL := 4.5;

w : REAL := 2.5;
u : REAL;
y : REAL;
e : REAL;

T : REAL; // VZORKOVACI PERIODA PRO VYKRESLENI
init : BOOL := FALSE;
stop : BOOL := FALSE;
END_VAR

PROGRAM prgMain
VAR
regulator : fbAniRe2;
END_VAR
// Posun do pracovniho bodu.
y := prutokomer.ENG - y0;
// Inicializace regulatoru - nastaveni parametru
IF NOT init THEN
init := TRUE;
regulator(K:=1.61, T1:=0.6855, T2:=8.5185, Ty:=1.4283,
Tu:=0.9209, w:=w-y0, y:=y, vT:=35.0, vTf:=1.0, kappa:=0, umin:=-u0,
umax:=10.0-u0, out_u=>u, out_e=>e);
// Volani bloku regulatoru v kazdem dalsim cyklu
ELSE
regulator(w:=w-y0, y:=y, TT=>T, out_u=>u, out_e=>e);
END_IF;
// Vypocet akcniho zasahu
ventilator.ENG := u + u0;
// Stop
IF stop THEN
ventilator.ENG := 0.0;
zarovka.ENG := 0.0;
END_IF;
END_PROGRAM
```



## Příloha D Hlavní program – PID regulace

### prgMain.ST

```
VAR_GLOBAL RETAIN
r0          :REAL := 3.14;
Ti          :REAL := 4.00;
Td          :REAL := 1.00;
END_VAR

VAR_GLOBAL
termistor_1 AT %X32 :TAI;
prutokomer AT %X44 :TAI;
termoclanek AT %X56 :TAI;
zarovka     AT %Y8  :TAO_TC6;
ventilator  AT %Y18 :TAO_TC6;

e          :REAL := 0.0;
w          :REAL := 3.5;
w0         :REAL := 0;
u, y       :REAL;

u0         : REAL := 2.0;
y0         : REAL := 4.5;

END_VAR

PROGRAM prgMain
  VAR_INPUT
  END_VAR
  VAR_OUTPUT
  END_VAR
  VAR
  regulator      :fbSimplePID;
  END_VAR
  VAR_TEMP
  END_VAR

  w0:=w-y0;
  y:=prutokomer.ENG-y0;

  regulator(y:=y, w:=w0, u_man:=0.0, max_u:=10.0-u0, min_u:=-u0,
Gain:=r0, Ti:=Ti, Td:=Td, u=>u, e=>e);
  ventilator.ENG:=u+u0;

END_PROGRAM
```