

Aplikace pro identifikaci a řízení systému v prostředí LabView

Application Created in the LabView Environment for Identification
and Control of a System

Bc. Jan Antoš

Diplomová práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan Antoš**
Osobní číslo: **A11393**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **prezenční**

Téma práce: **Aplikace pro identifikaci a řízení systému v prostředí LabView**

Zásady pro vypracování:

1. Vytvořte aplikaci v prostředí LabView pro identifikaci a řízení systému, která bude splňovat konkrétní potřeby experimentu na spolupracujícím pracovišti FT UTB.
2. V teoretické části zpracujte popis základních principů měření a regulace se zřetelem na prostředí LabView.
3. Vytvořte aplikaci pro měření elektrických veličin, voltampérových charakteristik např. LED pomocí programovatelného zdroje HP6038A a multimetru HP34401A.
4. Vytvořte aplikaci pro identifikaci a řízení reálného systému – pece (rovněž s využitím zdroje HP6038A a multimetru HP34401A).
5. Zdroj a multimetr budou komunikovat pomocí sběrnice GPIB a k počítači budou připojeny pomocí USB.
6. Aplikace bude umožňovat ukládání dat do souboru, součástí uživatelského rozhraní budou grafy pro zobrazování dat v reálném čase.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. Šulc B., Vítěčková M.: **Teorie a praxe návrhu regulačních obvodů**, vydavatelství ČVUT, 2004.
2. Vlach J., Havlíček J., Vlach M.: **Začínáme s LabView**, BEN-technická literatura, 2008.
3. Balátě J.: **Automatické řízení**, BEN-technická literatura, 2003.
4. **LabView: Real-time user manual**. Austin: National Instruments Corporation, 2004.
5. Franklin, G.F., Powell J.D., Emami-Naeini A.: **Feedback Control of Dynamic Systems**, Prentice Hall, 2006.

Vedoucí diplomové práce:

doc. Ing. Marek Kubalčík, Ph.D.

Ústav řízení procesů

Datum zadání diplomové práce:

22. února 2013

Termín odevzdání diplomové práce:

22. května 2013

Ve Zlíně dne 22. února 2013

prof. Ing. Vladimír Vašek, CSc.

děkan



doc. Mgr. Roman Jašek, Ph.D.

ředitel ústavu

ABSTRAKT

Práce se zabývá tvorbou aplikací pro měření, identifikaci a řízení systémů v prostředí LabView. Aplikace pro identifikaci umožňuje zejména měření voltampérových charakteristik a dále proměření soustav pomocí různých zdrojů signálu a jejich následnou identifikaci. Aplikace pro řízení umožňují provádět řízení s pevnými parametry (PID regulátor) i adaptivní řízení (1dof regulátor navržený polynomiální metodou a průběžná identifikace pomocí ARX modelu a průběžné metody nejmenších čtverců). Všechny 3 aplikace splňují konkrétní potřeby experimentu na spolupracujícím pracovišti FT UTB. Pro měření jsou použity 2 přístroje: programovatelný zdroj HP6038A a multimetr HP34401A, které komunikují s aplikacemi pomocí GPIB sběrnice a k počítači jsou připojeny pomocí USB rozhraní. Součástí práce jsou i ukázky naměřených výstupů. Aplikace umožňují ukládání dat do souboru a součástí uživatelského rozhraní jsou i grafy, které zobrazují data v reálném čase.

Klíčová slova: LabView, měření, identifikace, řízení, regulace

ABSTRACT

This thesis deals with creating applications in the LabView environment. The applications perform measurement, identification and control of systems. The application for identification enables to measure especially UI characteristics. Further it enables to activate a system by a range of signal sources and consequently identify the system. The application for control enables both control with fixed parameters (PID controller) and adaptive control (1dof regulator based on polynomial methods and identification based on ARX model and recursive least squares method). All three applications meet the specific requirements of the collaborating laboratory of FT UTB. There are two instruments which are used for measurement: HP6038A programmable source and HP34401A multimeter. The instruments communicate with the applications using the GPIB bus and are connected via USB interface. The thesis also contains examples of measured outputs. The applications enable to store data and display it in real time.

Keywords: LabView, measurement, identification, controlling, regulation

Poděkování

Chtěl bych poděkovat vedoucímu práce panu doc. Ing. Markovi Kubalčíkovi, Ph.D. za jeho spolupráci a cenné rady při vypracování diplomové práce. Dále bych rád poděkoval panu doc. Ing. et Ing. Ivo Kuřítkovi, Ph.D. et Ph.D. za spolupráci, připomínky a poskytnutí potřebného zázemí a prostředků potřebných pro vývoj aplikací. Také bych rád poděkoval panu Ing. Pavlu Urbánkovi za zpětnou vazbu při nasazení aplikací na koncovou stanici.

Motto

„Kdo ví proč, dokáže jakékoli jak.“

Nietzsche Friedrich

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 PROSTŘEDÍ LABVIEW	12
1.1 VIRTUAL INSTRUMENTS	12
1.2 PROJEKT	12
1.3 GPIB.....	13
1.3.1 Základní parametry	13
2 MĚŘENÍ	14
2.1 MĚŘENÍ VA CHARAKTERISTIK	14
3 IDENTIFIKACE SOUSTAVY	16
3.1 STREJCOVA METODA	16
3.1.1 a) Proporcionální soustava 2.řádu s rozdílnými časovými konstantami	17
3.1.2 b) Proporcionální soustava n-tého řádu se stejnými časovými konstantami	18
3.2 ITERAČNÍ NUMERICKÁ METODA URČENÍ PŘENOSU.....	19
3.3 IDENTIFIKACE SOUSTAVY PSEUDONÁHODNÝM SIGNÁLEM.....	20
3.4 IDENTIFIKACE SOUSTAVY POMOCÍ PNBS	20
3.5 METODA NEJMENŠÍCH ČTVERCŮ.....	21
4 REGULACE SOUSTAVY	22
4.1 PID REGULÁTOR	22
4.1.1 Proporcionální složka.....	22
4.1.2 Integrační složka	23
4.1.3 Derivační složka.....	23
4.1.4 Windup efekt.....	23
4.1.5 Ideální PID regulátor.....	24
4.2 URČENÍ PŘENOSU PID REGULÁTORU NASLINOVOU METODOU	24
4.3 POLYNOMIÁLNÍ METODA (1DOF KONFIGURACE).....	26
II PRAKTICKÁ ČÁST	29
5 APLIKACE IDENTIFIKACE	30
5.1 INSTALACE APLIKACE IDENTIFIKACE	30
5.2 OVLÁDÁNÍ APLIKACE IDENTIFIKACE	32
5.2.1 Popis programu	32
5.2.2 Chování programu.....	33
5.2.3 Front panel	34
Grafy	35
Buttons	36
Outputs.....	36
Menu	37
5.2.4 Zobrazení.....	45
5.2.5 Fronta	45
5.3.1 Generování akční veličiny $u(t)$	47
5.3.2 Integration time	47

5.4	VÝVOJ APLIKACE	48
5.4.1	Ovládací prvky	48
5.4.2	Převod výstupní veličiny	48
5.4.3	Generování signálu	51
5.4.4	Ukládání dat do souboru	51
5.4.5	Komunikace	52
5.4.6	Zobrazení časového průběhu	52
5.4.7	Data flow a chybové linky	53
	Ukončení běhu 2 paralelních cyklů	54
5.4.8	Ošetření race condition a deadlock	54
5.4.9	Ošetření chyb	54
5.4.10	Shift registr a lokální funkční proměnná	55
5.4.11	Ukládací smyčka – stavový automat	55
5.4.12	Měření a ukládání času	57
5.4.13	Property Nodes a Invoke Nodes	58
5.4.14	Moduly (subVIs)	58
5.4.15	Tvorba spustitelného souboru	58
5.4.16	Tvorba instalačního souboru	60
5.5	MĚŘENÍ VA CHARAKTERISTIK	61
5.6	IDENTIFIKACE SOUSTAVY	63
5.6.1	Popis tepelné soustavy (pece)	63
5.6.2	Měření přechodové charakteristiky soustavy	65
5.6.3	Určení přenosu soustavy	66
5.6.4	Simulace přechodové funkce	67
5.7	MĚŘENÍ STATICKÉ CHARAKTERISTIKY SOUSTAVY	68
5.7.1	Simulace měření statické charakteristiky	68
5.7.2	Reálné měření statické charakteristiky	68
5.7.3	Identifikace soustavy pomocí generátoru pseudonáhodného a PNBS signálu	70
6	APLIKACE REGULACE	71
6.1	INSTALACE APLIKACE REGULACE	71
6.2	OVLÁDÁNÍ APLIKACE REGULACE	71
6.2.1	Popis programu	71
	Graf	73
	Buttons	74
	Outputs	74
	Menu	75
6.4.1	Zobrazení	79
6.4.2	Fronta	79
6.4.3	Soubor	79
6.4.4	Generování žádané veličiny $w(t)$	80
6.5	VÝVOJ APLIKACE REGULACE	80
6.5.1	Tvorba spustitelného a instalačního souboru	81
6.6	REGULACE TEPELNÉ SOUSTAVY	81
6.6.1	Výpočet regulátoru PID	81
6.6.2	Simulace regulace	82
6.6.3	Regulace – reálný experiment	83

7	ADAPTIVNÍ REGULACE SOUSTAVY	85
7.1	OVLÁDÁNÍ APLIKACE REGULACE	85
7.1.1	Popis programu	85
	Graf	86
	Menu	86
7.3.1	Soubor	88
7.4	VÝVOJ APLIKACE REGULACE	88
7.5	REGULACE REÁLNÉ SOUSTAVY	89
7.5.1	Simulace regulace	89
7.5.2	Regulace – reálný experiment	90
	ZÁVĚR	91
	ZÁVĚR V ANGLIČTINĚ	92
	SEZNAM POUŽITÉ A PŘEHLEDOVÉ LITERATURY	93
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	94
	SEZNAM OBRÁZKŮ	95
	SEZNAM TABULEK	98
	SEZNAM PŘÍLOH	99

ÚVOD

Cílem práce je vytvořit aplikace pro měření elektrických veličin, identifikaci a řízení systému v prostředí LabView. Aplikace mají pracovat se dvěma reálnými přístroji programovatelným zdrojem HP6038A a multimetrem HP34401A, které jsou připojeny pomocí GPIB sběrnice.

Měření a identifikace pracují v tomto případě na stejném principu, tj. na zjišťování odezvy soustavy na určitý vstupní signál. U řízení s pevnými parametry lze řídit časově invariantní soustavy a pro řízení složitějších soustav lze použít řízení adaptivní.

Práce je rozdělena na 2 části – teoretickou a praktickou. Teoretická část stručně popisuje prostředí LabView a vybrané metody měření, identifikace a řízení systému. Praktická část se zabývá popisem jednotlivých aplikací, jejich možnostem, vývojem a řešeným problémům, a zahrnuje také některé výsledky měření/řízení a simulací.

I. TEORETICKÁ ČÁST

1 PROSTŘEDÍ LABVIEW

LabView je objektově orientované grafické vývojové prostředí, které umožňuje vytvoření a nasazení měřicích a řídicích systémů [8], [10]. Jednotlivé objekty jsou ve formě modulů (VI). Tyto objekty spolu interagují pomocí linek [1].

1.1 Virtual instruments

Programy v LabView se nazývají virtual instruments nebo VIs. Cílem těchto programů je vytvořit virtuální prostředí imitující měřicí a řídicí přístroje jako třeba multimetry a osciloskopy. Lze také vytvářet virtuální přístroje, které nemají reálný ekvivalent. Programy se tvoří pomocí dvou panelů – front panel a block panel.

Front panel je uživatelské rozhraní, které obsahuje ovládací prvky (controls) a indikátory (indicators). Ovládací prvky jsou vstupní mechanismy, pomocí kterých lze s aplikací interagovat. Jsou to například tlačítka, spínače, formuláře pro zadávání číselných či textových hodnot, rolovací menu s výčtovými typy a další. Indikátory jsou pak výstupní prvky jako např. grafy, LED, výstupní formuláře a další.

Block panel pak obsahuje zdrojový kód ve formě diagramu. Tento diagram pak obsahuje jednotlivé moduly (VIs) propojené mezi sebou pomocí linek. Jednotlivá data poté tečou tímto diagramem a tím uvádí program v činnost. Kromě předdefinovaných VIs je možné tvořit i vlastní VIs, které mají určitou funkci a s okolními VI komunikují pomocí vstupů a výstupů. Jednotlivé VI lze vnořovat a tím vytvářet přehlednou stromovou strukturu podle logické funkce jednotlivých VI.

Aplikace v LabView lze využít ke komunikaci s hardwarem a tak je možné efektivně řídit systémy, sbírat a vizualizovat data. Komunikace je možná přes různá rozhraní např. GPIB, PXI, VXI, RS232 a RS485.

1.2 Projekt

Prostředí LabView umožňuje tvorbu tzv. projektů, které sdružují všechny potřebné komponenty pro činnost aplikace. Můžou se zde nacházet jednotlivá VIs, vlastní ovládací prvky nebo také soubory pro tvorbu spustitelných a instalačních souborů. Projekt obvykle obsahuje jedno hlavní VI a několik dalších modulů (subVIs). Projekt může být členěn jinak než na disku.

1.3 GPIB

GPIB je velmi často používaná sběrnice laboratorních měřicích systémů navržena firmou Hewlett-Packard definována standardem IEEE 488.

1.3.1 Základní parametry

Jedná se o paralelní sběrnici, která umožňuje propojení až 14 přístrojů a jednoho řídicího členu (např. PC). Maximální přenosová rychlost je 1 MB/s. Má 24 vodičů (8 datových, 3 pro „handshaking“, 5 pro vysílání a 8 zemních). Maximální délka sběrnice může být až 20 m a maximální vzdálenost mezi dvěma jednotkami až 2 m. Přenos dat je asynchronní a je řízen hardwarově. Elektrické úrovně signálu odpovídají negativní TTL logice [2].

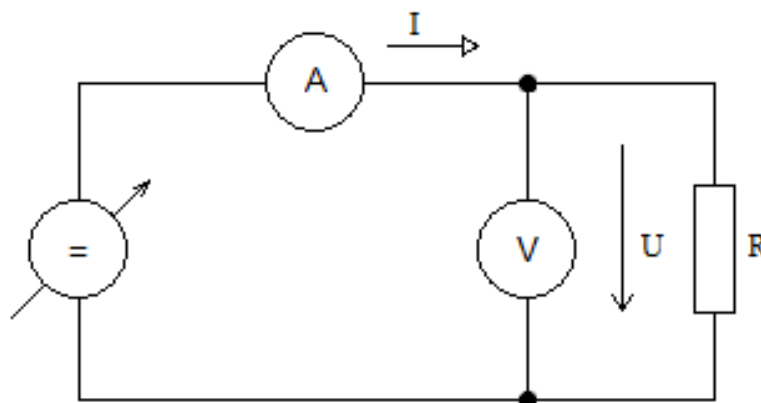
2 MĚŘENÍ

Měření je proces, při kterém zkoumáme kvantifikovatelné vlastnosti určitého systému. Tímto systémem může být např. elektronická součástka, u které můžeme měřit elektrické veličiny (odpor, napětí, proud, výkon, ...). Měřená vlastnost je v našem případě fyzikální veličina. Výsledkem měření je tedy odhad hodnoty veličiny a její chyby v určitých fyzikálních jednotkách (působí zde náhodné vlivy).

2.1 Měření VA charakteristik

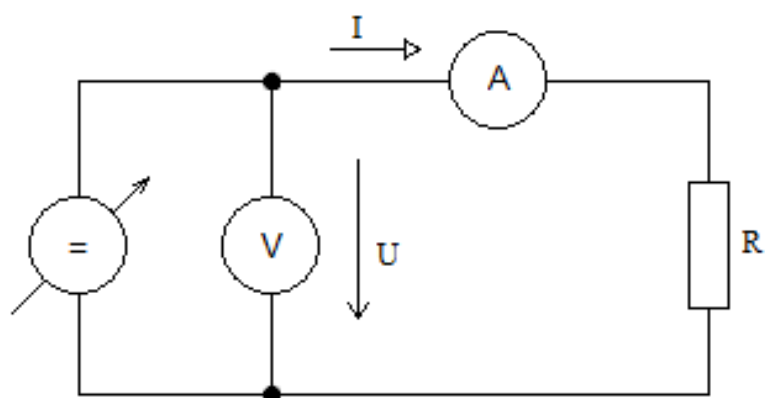
VA charakteristika udává závislost velikosti proudu na velikosti napětí. K měření VA charakteristiky je tedy zapotřebí měřit 2 elektrické veličiny – proud $I(t)$ a napětí $U(t)$. Pro měření velikosti proudu se používá ampérmetr a pro měření velikosti napětí voltmetr. Požadovaný výkon dodává do obvodu napěťový zdroj. Ideální napěťový zdroj a ampérmetr by měly mít nulový sériový vnitřní odpor a ideální voltmetr by měl mít vnitřní odpor nekonečný. Všechny reálné přístroje se však vyznačují parazitními vlastnosti, které vnášejí do měření chybu. Při měření zátěže je tedy zapotřebí zvolit správné rozmístění elektrických prvků tak, aby chyba byla co nejmenší.

Existují 2 varianty zapojení ampérmetru a voltmetru.



Obrázek 1 – Měření VA charakteristiky – varianta 1

Při použití varianty 1 (Obrázek 1) se měří správně velikost napětí na zátěži, avšak ampérmetr měří velikost proudu, který je složen z proudu protékajícího voltmetrem a zátěží.



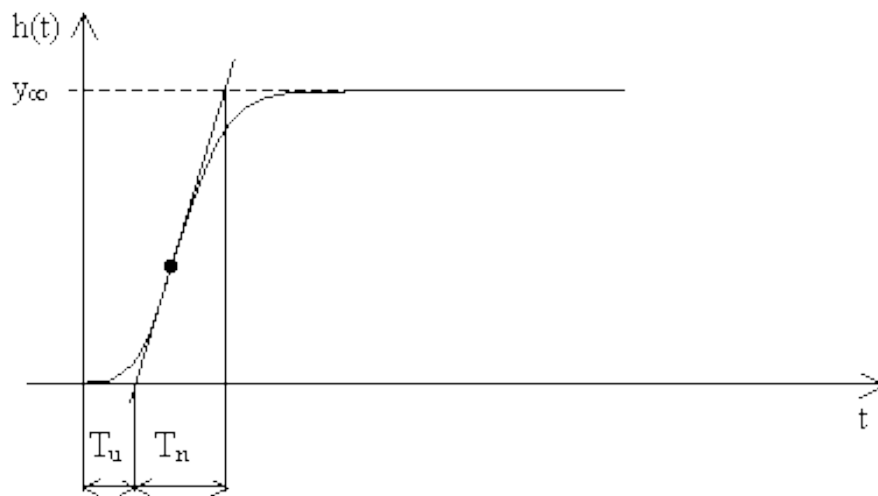
Obrázek 2 – Měření VA charakteristiky – varianta 2

V zapojení podle varianty 2 (Obrázek 2) je správně měřen proud procházející zátěží, avšak voltmetr měří součet úbytků napětí na ampérmetru a zátěži. V aplikaci Identifikace se místo voltmetru nastavuje napětí na zdroji, takže ve výsledku je použito toto zapojení.

3 IDENTIFIKACE SOUSTAVY

Cílem identifikace soustavy je zejména získání matematického modelu soustavy. V práci je využita jedna z deterministických metod identifikace spočívající ve vyhodnocování přechodových charakteristik. Konkrétně se jedná o aproximaci nekmitavé soustavy vyššího řádu Strejcovou metodou [3].

3.1 Strejcová metoda



Obrázek 3 - Přechodová charakteristika s aperiodickým průběhem a s vyznačenou dobou náběhu a dobou průtahu

Vykazuje-li odezva systému aperiodický průběh, lze ji aproximovat pomocí proporcionální soustavy 2. řádu s rozdílnými časovými konstantami nebo proporcionální soustavou n -tého řádu se stejnými časovými konstantami. Volba soustavy závisí na hodnotě parametru τ , který se vypočte:

$$\tau = \frac{T_u}{T_n} \quad (1)$$

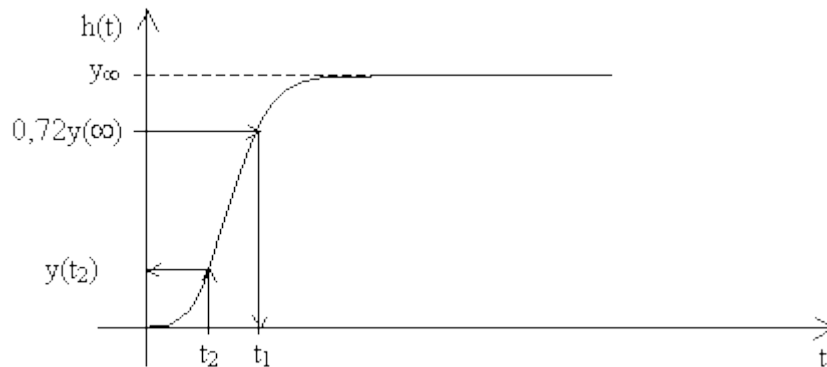
kde T_u - doba průtahu, T_n - doba náběhu.

Pokud parametr τ je menší než 0,1, volíme pro aproximaci obrazový přenos s rozdílnými časovými konstantami. Pokud je τ větší než 0,1, volíme obrazový přenos se stejnými časovými konstantami. Tedy:

$$\tau < 0,1 \Rightarrow G(s) = \frac{K}{(T_1s + 1)(T_2s + 1)} \quad (2)$$

$$\tau > 0,1 \Rightarrow G(s) = \frac{K}{(Ts + 1)^n}$$

3.1.1 a) Proporcionální soustava 2.řádu s rozdílnými časovými konstantami



Obrázek 4 - Aproximace proporcionální soustavou 2.řádu s rozdílnými časovými konstantami

Nejdříve vypočítáme parametry přenosu.

$$G(s) = \frac{K}{(T_1s + 1)(T_2s + 1)} \quad (3)$$

Při výpočtu parametrů postupujeme následujícím způsobem.

1. Vypočítáme zesílení

$$K = \frac{y(\infty)}{u(\infty)} \quad (4)$$

2. Pro hodnotu $0,72y(\infty)$ odečteme z přechodové charakteristiky časový okamžik t_1 a vypočteme součet časových konstant T_1 a T_2 podle vztahu:

$$T_1 + T_2 = \frac{t_1}{1,2564} \quad (5)$$

3. Vypočteme časový okamžik t_2 podle vzorce:

$$t_2 = 0,3574(T_1 + T_2) \quad (6)$$

4. Z grafu přechodové charakteristiky odečteme hodnotu $y(t_2)$.
5. Podle tabulky 1 (Tabulka 1) určíme poměr časových konstant τ_2 , kde

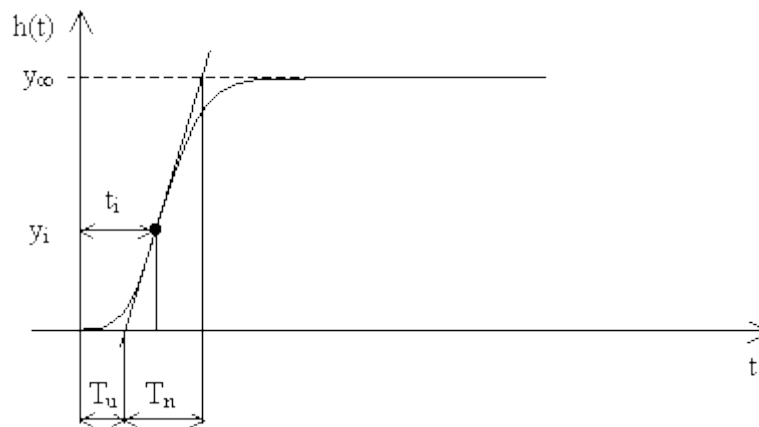
$$t_2 = \frac{T_1}{T_2} \quad (7)$$

6. Ze známého součtu a poměru časových konstant vypočteme T_1 a T_2

Tabulka 1 – Určení poměru časových konstant

$y(t_2)$	τ_2	$y(t_2)$	τ_2
0,3	0	0,22	0,183
0,29	0,023	0,21	0,219
0,28	0,043	0,2	0,264
0,27	0,063	0,19	0,322
0,26	0,084	0,18	0,403
0,25	0,105	0,17	0,538
0,24	0,128	0,16	1
0,23	0,154		

3.1.2 b) Proporcionální soustava n-tého řádu se stejnými časovými konstantami



Obrázek 5 - Aproximace proporcionální soustavou 2.řádu se stejnými časovými konstantami

Nejdříve vypočítáme parametry přenosu.

$$G(s) = \frac{K}{(Ts + 1)^n} \quad (8)$$

1. Vypočítáme zesílení

$$K = \frac{y(\infty)}{u(\infty)} \quad (9)$$

2. Přechodovou charakteristiku normujeme vzhledem k ustálené hodnotě $y(\infty)$.
3. Sestrojíme tečnu v inflexním bodě a určíme τ podle vztahu:

$$\tau = \frac{T_u}{T_n} \quad (10)$$

4. Podle hodnoty τ určíme z tab. 2 nejbližší vyšší řád n aproximační soustavy a souřadnici inflexního bodu y_i .
5. Pomocí y_i určíme v grafu přechodové charakteristiky inflexní bod a odečteme souřadnici t_i .
6. Hodnotu časové konstanty T určíme ze vztahu:

$$T = \frac{t_i}{n-1}, \text{ resp. } T = \frac{t_i - T_d}{n-1} \text{ pro } T_d > 0 \quad (11)$$

kde t_i - souřadnice času v inflexním bodě přechodové charakteristiky, T_d - časová konstanta dopravního zpoždění, n - řád aproximační soustavy.

Tabulka 2 – Stanovení řádu n aproximační soustavy a zpřesnění polohy inflexního bodu

n	2	3	4	5	6	7	8	9	10
τ	0,104	0,218	0,319	0,41	0,493	0,57	0,642	0,709	0,773
y_i	0,264	0,327	0,359	0,371	0				

3.2 Iterační numerická metoda určení přenosu

Tato metoda je založena na postupném nastavování jednotlivých konstant přenosu soustavy tak, aby vypočtená přechodová charakteristika co nejvíce odpovídala charakteristice naměřené.

1. Určíme obecný přenos soustavy.

$$G(s) = \frac{k}{(T_1s + 1)(T_2s + 1)} = \frac{k}{T_1T_2s^2 + (T_1 + T_2)s + 1} \quad (12)$$

2. Pomocí zpětné Laplaceovy transformace určíme přechodovou funkci, přičemž na vstupu soustavy uvažujeme Heavisideův skok.

$$h(t) = L^{-1}\left\{\frac{G(s)}{s}\right\} = k\left(1 - \frac{1}{T_1 - T_2}\right)\left(T_1e^{-\frac{t}{T_1}} - T_2e^{-\frac{t}{T_2}}\right) \quad (13)$$

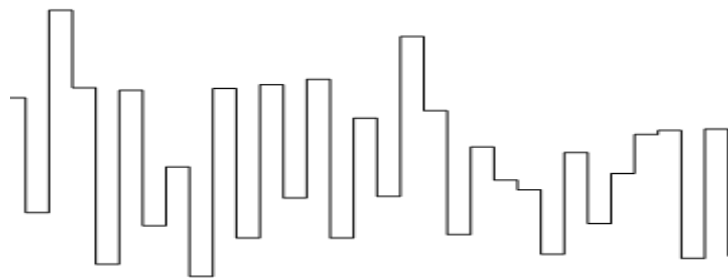
3. Nastavujeme konstanty k , T_1 a T_2 tak dlouho, dokud není suma čtverců odchylek naměřené a vypočtené přenosové funkce minimální.

Toto je optimalizační problém, kde hledáme nejmenší hodnotu účelové funkce (suma odchylek) pro správnou kombinaci argumentů (k , T_1 a T_2). Prostor možných řešení je tedy hledán na trojrozměrné hyperploše ve čtyřrozměrném prostoru. Iterační numerická metoda je tedy pouze obecným popisem, který definuje problém a tento problém lze optimalizovat mnoha různými způsoby např. evolučními algoritmy, analytickými metodami, enumerativně apod. Hodnota účelové funkce má pak následující tvar.

$$CV = \sum_{k=1}^n (h_{modelu}(k) - h_{změřená}(k))^2 \quad (14)$$

3.3 Identifikace soustavy pseudonáhodným signálem

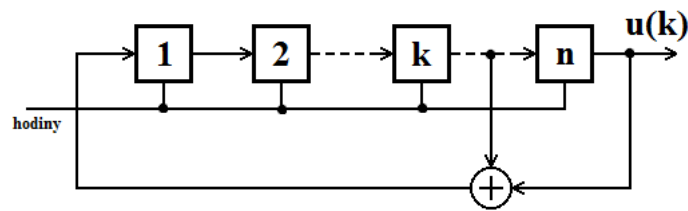
U systému, kde je použití skoků nevhodné, lze využít identifikaci soustavy pomocí pseudonáhodného signálu. Perioda změny signálu však musí být nastavena tak, aby byla zohledněna dynamika soustavy. Impulsní i přechodovou charakteristiku lze získat korelační analýzou dat, popř. určením přenosu soustavy (např. metodou nejmenších čtverců) a dosazením Diracova impulsu, popř. Heavisideova skoku na vstup této soustavy. Model soustavy lze potom získat regresními metodami popřípadě pomocí korelační analýzy.



Obrázek 6 – Časový průběh pseudonáhodného signálu

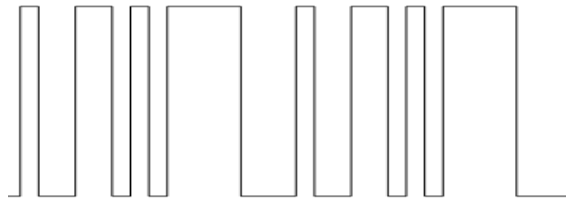
3.4 Identifikace soustavy pomocí PNBS

Pro identifikaci soustavy lze využít také pseudonáhodný binární signál. Ten je generován pomocí rotačního registru s vnitřní zpětnou vazbou.



Obrázek 7 – Generátor PNBS

Jelikož je velikost rotačního registru omezená, dojde při dostatečném počtu rotací k opakování signálu. Tento signál je vhodný k identifikaci, neboť autokorelační funkce se blíží autokorelační funkci bílého šumu. Typický časový průběh PNBS je na obrázku.



Obrázek 8 – Časový průběh PNBS

3.5 Metoda nejmenších čtverců

Metoda nejmenších čtverců vychází z obecné rovnice

$$\mathbf{y} = \mathbf{F}\hat{\boldsymbol{\theta}} + \hat{\mathbf{e}} \quad (15)$$

kde \mathbf{y} je výstupní vektor, \mathbf{F} je matice dat, $\hat{\boldsymbol{\theta}}$ je vektor neznámých koeficientů a $\hat{\mathbf{e}}$ je vektor odchylek.

Dále je definováno kritérium

$$J = \hat{\mathbf{e}}^T \hat{\mathbf{e}} = (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\theta}})^T \cdot (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\theta}}) \quad (16)$$

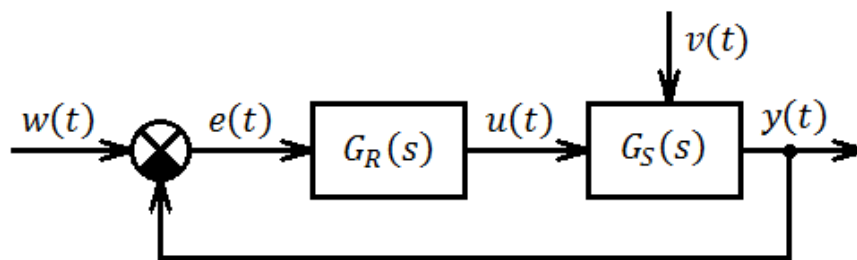
Pro dosažení nejmenších odchylek se kritérium zderivuje podle koeficientů a položí rovno nule.

$$\begin{aligned} \frac{\partial J}{\partial \hat{\boldsymbol{\theta}}} &= \frac{\partial (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\theta}})^T}{\partial \hat{\boldsymbol{\theta}}} (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\theta}}) + \frac{\partial (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\theta}})^T}{\partial \hat{\boldsymbol{\theta}}} (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\theta}}) \\ \frac{\partial J}{\partial \hat{\boldsymbol{\theta}}} &= -\mathbf{F}^T (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\theta}}) - \mathbf{F}^T (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\theta}}) = -2\mathbf{F}^T (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\theta}}) \\ \frac{\partial J}{\partial \hat{\boldsymbol{\theta}}} &= \mathbf{0} \Rightarrow 2\mathbf{F}^T \mathbf{y} = 2\mathbf{F}^T \mathbf{F}\hat{\boldsymbol{\theta}} \Rightarrow \hat{\boldsymbol{\theta}} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{y} \end{aligned} \quad (17)$$

Tvar matice \mathbf{F} je pak určen použitým modelem.

4 REGULACE SOUSTAVY

Regulace je proces řízení se zpětnou vazbou [7], [9]. Při regulaci dochází k udržování fyzikální veličiny na požadované hodnotě při působení vnějších poruch. Základním principem regulace je porovnávání výstupní veličiny $y(t)$ s žádanou hodnotou $w(t)$. Rozdíl mezi těmito veličinami se nazývá regulační odchylka $e(t)$ a na základě této hodnoty se určuje velikost akční veličiny $u(t)$. Základní schéma regulačního obvodu je na obrázku.



Obrázek 9 – Základní schéma regulačního obvodu

4.1 PID regulátor

PID regulátor [4] patří mezi spojité regulátory. Skládá se z proporcionální, integrační a derivační části. Řadí se před regulovanou soustavu, na kterou působí akčním zásahem. Vstupem do regulátoru je tedy regulační odchylka $e(t)$ a výstupem je akční veličina $u(t)$.

Přenos soustavy je určen poměrem Laplaceových obrazů výstupní a vstupní veličiny za nulových počátečních podmínek.

$$R(s) = \frac{U(s)}{E(s)} \quad (18)$$

4.1.1 Proporcionální složka

Proporcionální složka P je pouhým zesilovačem. Pro P regulátor platí, že akční veličina $u(t)$ je přímo úměrná regulační odchylce $e(t)$.

$$u(t) = r_0 e(t) \quad (19)$$

kde r_0 je činitel zesílení.

Přenos P regulátoru je pak

$$R(s) = r_0 \quad (20)$$

Samotný P regulátor nedokáže trvale odstranit regulační odchylku $e(t)$.

4.1.2 Integrační složka

I regulátor je takový, kde je akční veličina $u(t)$ přímo úměrná integrálu regulační odchylky $e(t)$.

$$u(t) = r_i \int_0^t e(t) + u(0) \quad (21)$$

r_i je pak zesílení I regulátoru.

Přenos je vyjádřen jako

$$R(s) = \frac{r_i}{s} \quad (22)$$

Výhodou I regulátoru je, že dokáže úplně eliminovat regulační odchylku $e(t)$, avšak zpomaluje regulovaný proces a může zhoršovat stabilitu.

4.1.3 Derivační složka

D regulátor je takový, kde akční veličina $u(t)$ je přímo úměrná derivaci regulační odchylky $e(t)$.

$$u(t) = r_d \frac{de(t)}{dt} \quad (23)$$

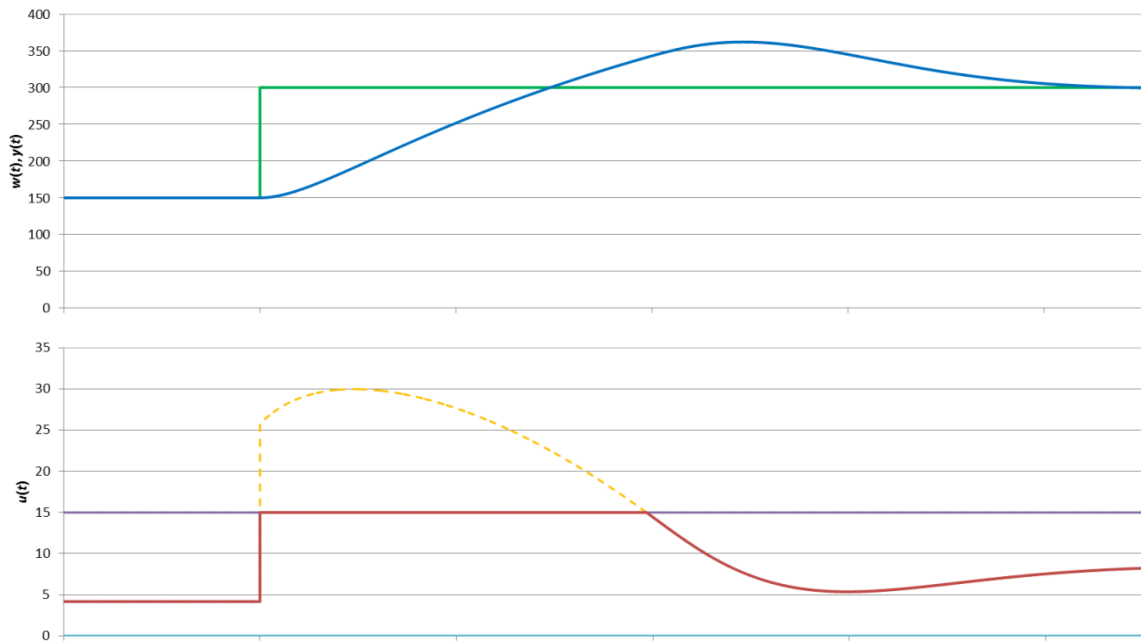
Přenos je pak vyjádřen jako

$$R(s) = r_d s \quad (24)$$

Derivační regulátor urychluje regulaci, avšak zesiluje šum. D regulátor se nikdy nevyskytuje samostatně, protože se nejedná o kauzální systém.

4.1.4 Windup efekt

Omezíme-li akční veličinu, může u I regulátoru docházet k tzv. wind-up efektu, kdy dochází k vnitřnímu přeregulování/podregulování akční veličiny. Vnější akční veličina je po dobu windup konstantní, avšak vnitřně stále roste/klesá. Po dosažení svého maxima/minima začne opět klesat/stoupat, avšak navenek se projeví až po překročení horního/spodního limitu. Pod dobu windup efektu je tedy regulátor necitlivý.



Obrázek 10 – Windup efekt a jeho vliv na výstupní veličinu

4.1.5 Ideální PID regulátor

Ideální PID regulátor vznikne sumací jednotlivých složek.

$$u(t) = r_0 e(t) + r_d \frac{de(t)}{dt} + r_i \int_0^t e(t) + u(0) \quad (25)$$

Přenos pak odpovídá následujícímu vztahu

$$R(s) = r_0 + r_d s + \frac{r_i}{s} = K_R \left(1 + T_D s + \frac{1}{T_I s} \right) \quad (26)$$

kde K_R je zesílení, T_D je derivační časová konstanta a T_I je integrační časová konstanta. V praxi se pak používají právě tyto 3 konstanty.

Ostatní regulátory vzniknou jinou kombinací jednotlivých složek. V praxi se tedy můžeme setkat s regulátory P, I, PI, ID, PD a PID.

4.2 Určení přenosu PID regulátoru Naslinovou metodou

Naslinova metoda [5] patří do třídy *analytických* metod, která předpokládá znalost přenosu regulované soustavy. Vychází se ze vztahů mezi koeficienty charakteristického polynomu přenosu uzavřeného systému.

$$A(s) = A_n s^n + A_{n-1} s^{n-1} + \dots + A_1 s + A_0 \quad (27)$$

Vztahy mezi koeficienty jsou vázány rovnicemi.

$$A_i^2 = \alpha A_{i-1} A_{i+1} \quad \text{pro } i = 1, 2, \dots, n - 1 \quad (28)$$

Maximální přeregulování σ [%] pak závisí na koeficientu α podle následující tabulky.

Tabulka 3 – Závislost maximálního přeregulování na koeficientu α podle Naslina

α	1,75	1,8	1,9	2,0	2,2	2,4
σ [%]	16	12	8	5	3	1

Jelikož charakteristický polynom n -tého stupně poskytuje pouze $n - 1$ rovnic typu (28), můžeme určit pouze $n - 1$ nastavitelných parametrů. Při použití regulátoru s více nastavitelnými parametry je tedy nutné přebývajících parametry zvolit a teprve zbývajících $n - 1$ dopočítat. Naopak při použití regulátoru s méně parametry nelze podmínky splnit.

Metoda je použita při odvození vztahů pro výpočet parametrů ideálního PID regulátoru

$$R(s) = \frac{r(s)}{p(s)} = \frac{r_1 s^2 + r_0 s + r_{-1}}{s} \quad (29)$$

a pro řízení výstupu proporcionálního systému 2. řádu na konstantní hodnotu.

$$G(s) = \frac{b(s)}{a(s)} = \frac{b_0}{s^2 + a_1 s + a_0} \quad (30)$$

Charakteristický polynom přenosu uzavřeného regulačního obvodu je ve tvaru

$$\begin{aligned} A(s) &= a(s)p(s) + b(s)r(s) \\ A(s) &= s^3 + (a_1 + b_0 r_1) s^2 + (a_0 + b_0 r_0) s + b_0 r_{-1} \\ A(s) &= A_3 s^3 + A_2 s^2 + A_1 s + A_0 \end{aligned} \quad (31)$$

Pro koeficienty polynomu $A(s)$ platí podle (28) vztah

$$A_1^2 = \alpha A_0 A_2, \quad A_2^2 = \alpha A_1 A_3 \quad (\text{kde } A_3 = 1) \quad (32)$$

a po dosazení

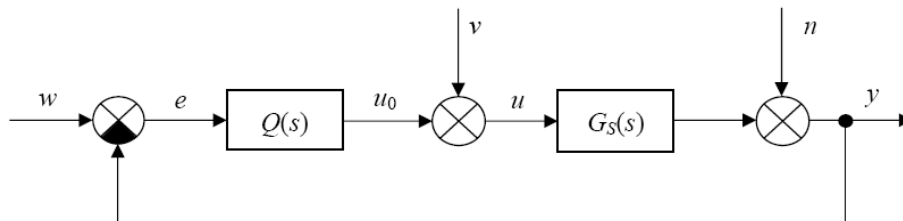
$$\begin{aligned} (a_1 + b_0 r_1)^2 &= \alpha (a_0 + b_0 r_0), \\ (a_0 + b_0 r_0)^2 &= \alpha b_0 r_{-1} (a_1 + b_0 r_1) \end{aligned} \quad (33)$$

Jelikož jsou k dispozici dvě rovnice o třech neznámých, je nutné jeden parametr regulátoru zvolit a ostatní dopočítat. Pokud je potřeba použít PI regulátor, volíme r_1 blízký nule.

Pozn.: Při praktickém řízení je nežádoucí, aby parametry konvenčních regulátorů pro řízení systémů s kladným zesílením, byly záporné. Proto volíme jeden z parametrů tak, aby ostatní dopočítané byly nezáporné.

4.3 Polynomiální metoda (1DOF konfigurace)

Polynomiální metoda [6] umožňuje získat nejen strukturu regulátoru, ale také jeho parametry. Uvažujeme-li jednoduchý regulační obvod



Obrázek 11 – Systém řízení s jedním stupněm volnosti

pak přenos soustavy je uvažován ve tvaru:

$$G_S(s) = \frac{b(s)}{a(s)} \quad (34)$$

kde $a(s)$ a $b(s)$ jsou nesoudělné polynomy a přenos je ryzí $\deg b \leq \deg a$.

Přenos regulátoru je pak

$$G_R(s) = \frac{q(s)}{p(s)} \quad (35)$$

kde $q(s)$ a $p(s)$ jsou nesoudělné polynomy a přenos je ryzí $\deg p \leq \deg q$.

Přenos uzavřeného obvodu je pak ve tvaru (pro přehlednost již nebude uváděna proměnná s):

$$G_{wy}(s) = \frac{G_S G_R}{1 + G_S G_R} = \frac{bq}{ap + bq} = \frac{bq}{d} \quad (36)$$

Referenční signál a poruchy jsou uvažovány jako deterministické funkce, jejichž obraz lze zapsat jako:

$$w = \frac{h_w}{f_w}, \quad v = \frac{h_v}{f_v}, \quad n = \frac{h_n}{f_n} \quad (37)$$

Z obrázku můžeme vypsát následující rovnice:

$$y = n + \frac{b}{a}u = n + \frac{b}{a}(u_0 + v)$$

$$u = u_0 + v \quad (38)$$

$$u_0 = \frac{q}{p}e = \frac{q}{p}(w - y)$$

Po dosazení a využití diofantické rovnice (polynom d) dostaneme:

$$y = \left(1 - \frac{ap}{d}\right)w + \frac{p}{d}(an + bv)$$

$$e = \frac{p}{d}[aw - bv - an] = \frac{p}{d}[a(w - n) - bv] \quad (39)$$

$$u_0 = \frac{q}{d}[a(w - n) - bv]$$

Z požadavků na stabilitu, vnitřní ryzost systému a asymptotické sledování referenčního systému a kompenzace poruch vyplývají následující vztahy:

$$d = ap + bq$$

$$\deg q \leq \deg p$$

$$\lim_{t \rightarrow \infty} [e(s)] = \lim_{s \rightarrow 0} [s e(s)] \quad (40)$$

$$p = f\tilde{p} \Rightarrow af\tilde{p} + bq = d$$

Výsledný přenos regulátoru je:

$$G_R = \frac{q}{f\tilde{p}} \quad (41)$$

kde polynom f je současně dělitelný všemi jmenovateli přenosů $w(s)$, $v(s)$ a $n(s)$.

Ze vztahů pro určení stupně polynomu d , počtu rovnic, počtu neznámých a z požadavku na ryzost vyplývají následující vztahy:

$$\deg q \leq \deg a + \deg f - 1$$

$$\deg \tilde{p} \geq \deg a - 1 \quad (42)$$

$$\deg d \geq 2 \deg a + \deg f - 1$$

Parametry regulátoru se pak vypočítají metodou neurčitých koeficientů z polynomiální diofantické rovnice. Metodu lze použít jak ve spojitě, tak v diskrétní oblasti.

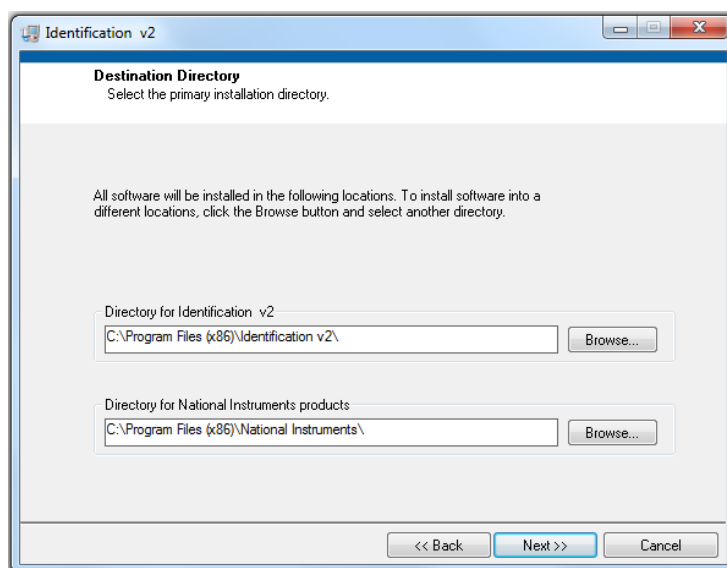
II. PRAKTICKÁ ČÁST

5 APLIKACE IDENTIFIKACE

5.1 Instalace aplikace Identifikace

Instalace aplikace je velmi jednoduchá. Aplikace potřebuje ke svému běhu LabView RunTime Engine, ovladač GPIB a podporu VISA rozhraní. Tyto moduly byly přidány při buildování¹ jako additional installers. Aplikaci lze tedy jednoduše nainstalovat pomocí průvodce instalací. Aplikaci lze nainstalovat na operační systémy Windows XP a pozdější.

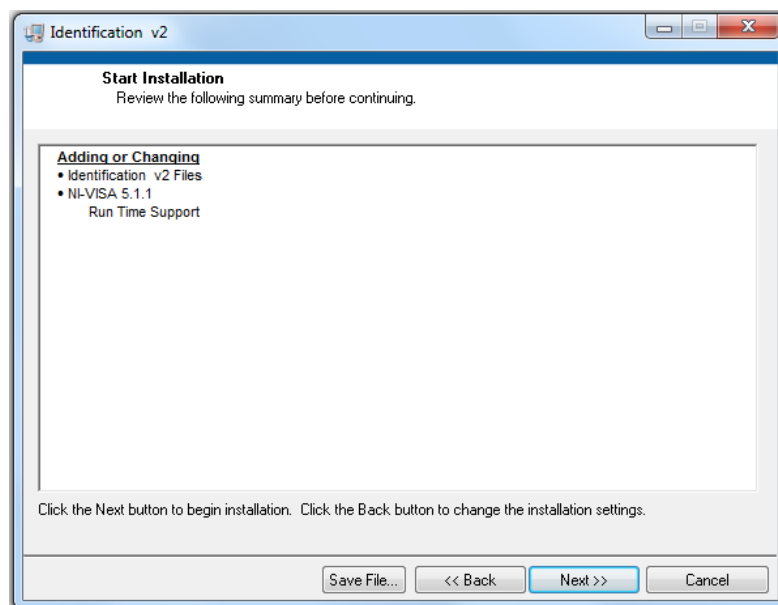
Nejprve spustíme soubor setup.exe a zvolíme umístění instalace (Obrázek 12).



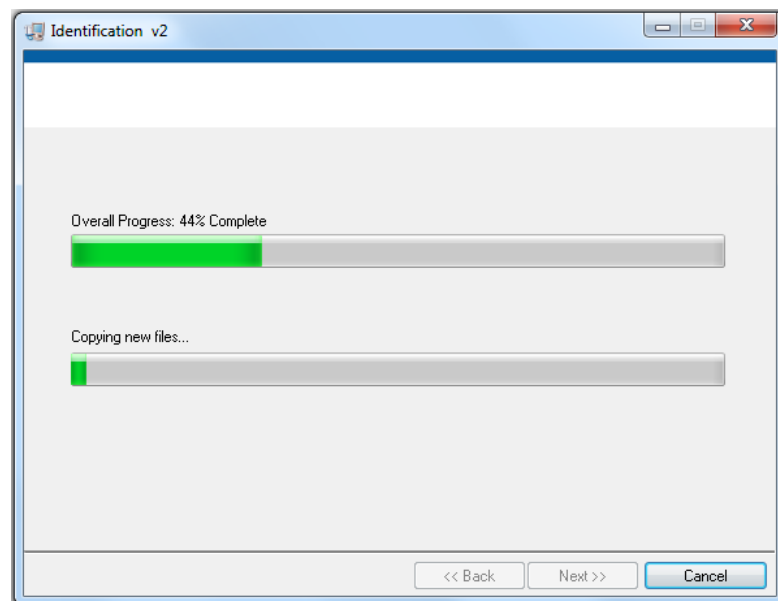
Obrázek 12 – Instalace aplikace Identifikace – krok 1

¹ Buildování je proces pro překlad kódu do spustitelné podoby.

V dalším okně se nám zobrazí seznam přidáných souborů. Zahájíme instalaci stisknutím tlačítka Next a vyčkáme na dokončení instalace.

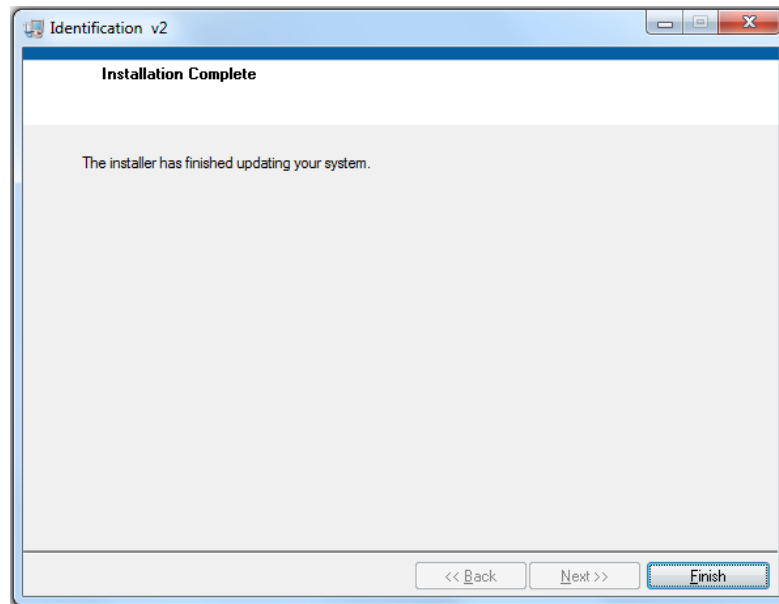


Obrázek 13 – Instalace aplikace Identifikace – krok 2



Obrázek 14 – Instalace aplikace Identifikace – krok 3

Instalaci dokončíme stiknutím tlačítka Finish.



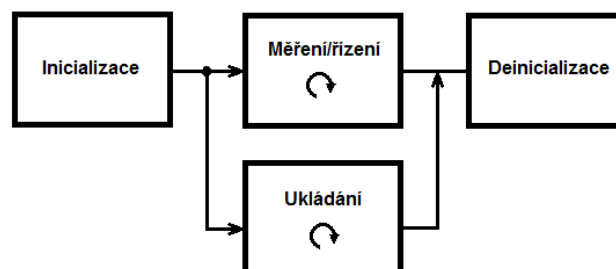
Obrázek 15 – Instalace aplikace Identifikace – krok 4

Pozn.: Pokud se před zahájením instalace objeví dialog pro potvrzení spuštění (ochrana UAC), potvrdíme jej a pokračujeme v instalaci. Postup instalace může být odlišný od výše popsaného postupu, při instalaci se tedy řídíme průvodcem.

5.2 Ovládání aplikace Identifikace

5.2.1 Popis programu

Všechny 3 aplikace mají základní strukturu stejnou (Obrázek 16).

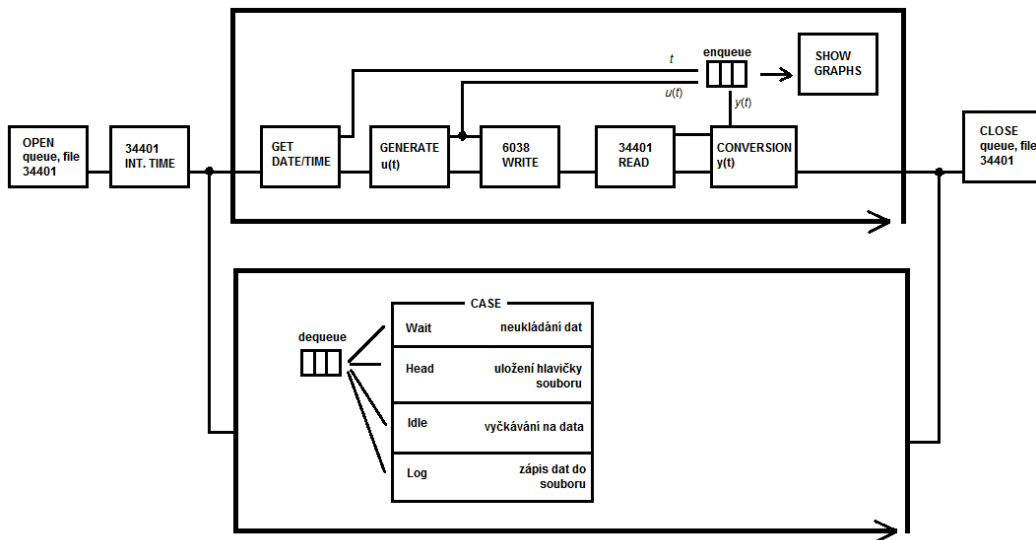


Obrázek 16 – Obecné blokové schéma aplikací

Blok inicializace obsahuje všechny operace, které se musí vykonat před začátkem měření. Smyčka Měření/řízení pak slouží k ovládání přístrojů, sběru dat, ale také k zobrazení jednotlivých průběhů. Smyčka ukládání slouží k ukládání dat do souboru, podle aktuální

konfigurace ovládacích prvků aplikace. Blok deinitializace obsahuje veškeré operace, které je potřeba provést po ukončení měření.

Konkrétní blokové schéma je pak vidět na následujícím obrázku.



Obrázek 17 – Blokové schéma aplikace Identifikace

Blok OPEN obsahuje bloky pro vytvoření/otevření fronty, souboru a inicializační VI pro multimetr HP34401. V bloku INT. TIME je nastaven Integration time pro multimetr. V měřicí smyčce (na obrázku ta vrchní) je nejprve sejmuto aktuální časové razítko a následuje vygenerování signálu $u(t)$ (podle zvoleného zdroje dat). Hodnota tohoto signálu je zapsána na výstup zdroje HP6038. Po vyčtení výstupní veličiny $y(t)$ (34401) je tato převedena pomocí tabulky nebo aproximačního polynomu. Důležité hodnoty jsou pak uloženy do fronty, která posílá data ukládací smyčce (na obrázku ta spodní). Průběh signálů je pak zobrazen ve dvou grafech.

V ukládací smyčce je přepínáno mezi jednotlivými stavy podle zvoleného módu (nejsou zobrazeny všechny stavy). Po ukončení měření je v bloku CLOSE uzavřena fronta, soubor a komunikace s multimetrem.

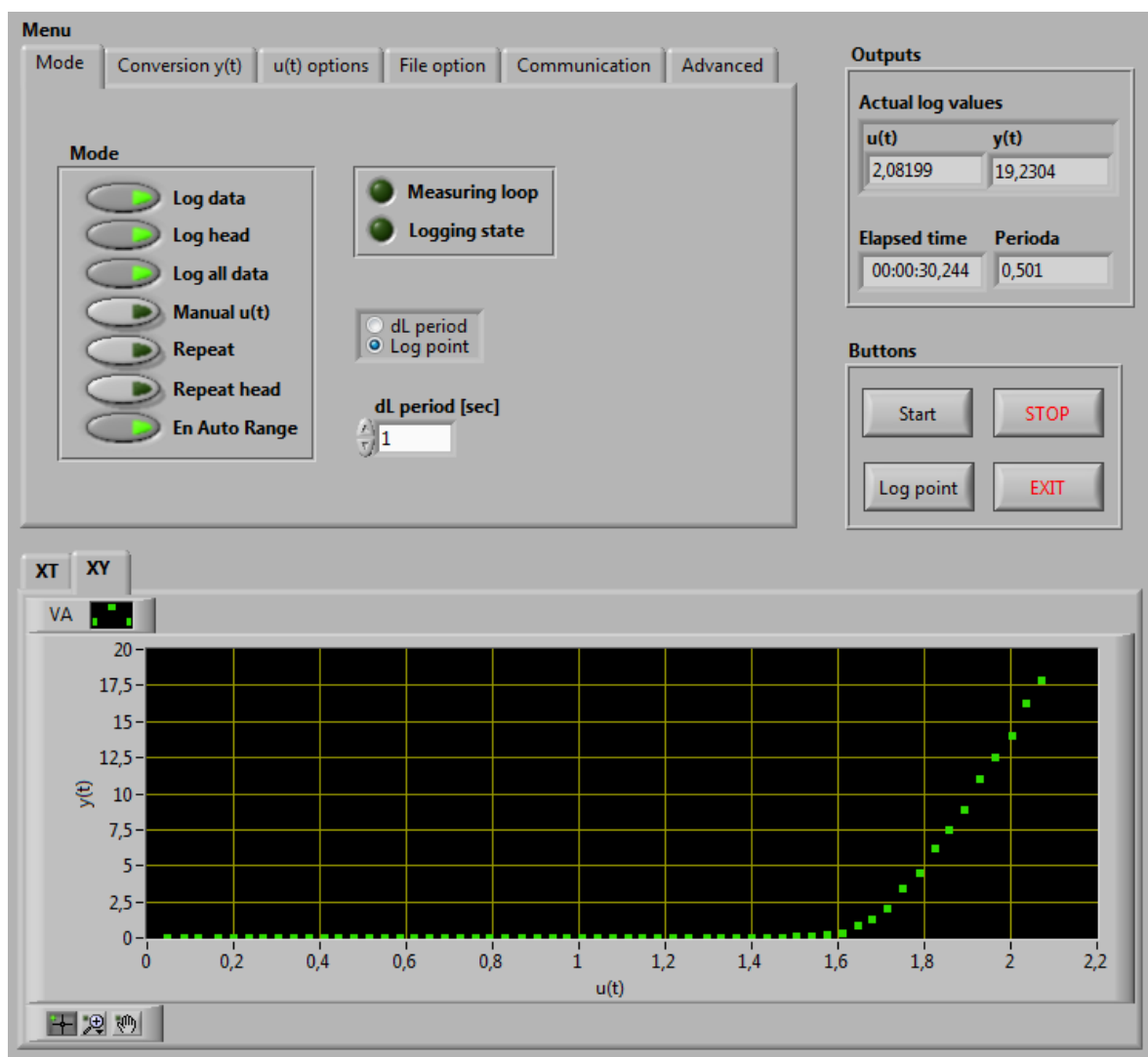
5.2.2 Chování programu

Po spuštění programu je aplikace v nastavovacím módu, kdy je možné nastavit hodnoty prvků Front panelu. Po stisknutí tlačítka **Start** se otevře OpenFileDialog (je-li to nastaveno), kde je možné zvolit umístění souboru pro ukládání dat. Po výběru cesty a jména souboru se spustí samotné měření.

Aplikace pracuje vždy nad určitým souborem, i když nejsou data ukládána. Při pokusu o zápis do existujícího souboru je vygenerován dialog, pomocí kterého je možné soubor přepsat, nebo ukončit měření stisknutím tlačítka **Cancel**. Měření je také ukončeno při výskytu jakékoliv chyby nebo při stisknutí tlačítka **Stop**. Po ukončení měření je aplikace opět v nastavovacím módu a je možné spustit nové měření. Pro ukončení aplikace slouží tlačítko **Exit**.

5.2.3 Front panel

Front panel (Obrázek 18) slouží pro ovládání aplikace a vizualizaci dat v reálném čase. Skládá se z několika bloků. Blok **Menu** slouží pro ovládání aplikace a nastavení parametrů, blok **Outputs** ukazuje aktuální výstupy a blok **Buttons** obsahuje tlačítka. Ve spodní části jsou k dispozici 2 grafy, které slouží k vizualizaci průběhu měření.

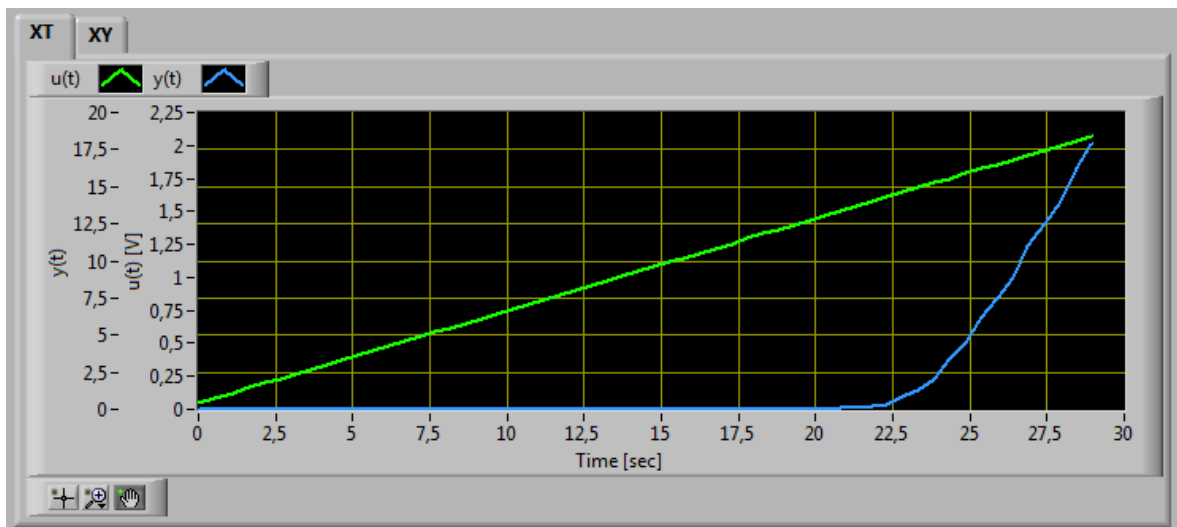


Obrázek 18 – Front panel aplikace Identifikace

Grafy

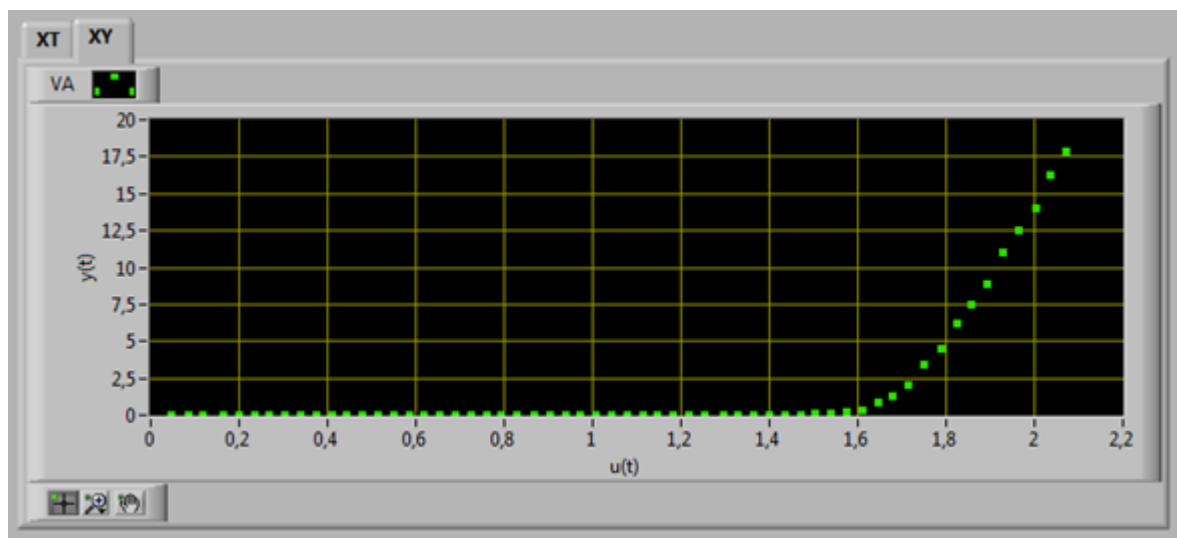
V aplikaci jsou k dispozici 2 druhy grafů. Všechny osy mění rozsah automaticky podle potřeby. V XY grafu je průběh zobrazen jako jednotlivé body, což zabraňuje špatnému zobrazení dat (spojení prvního a posledního bodu při opakování signálu definicí). U XT grafu se předpokládá, že čas je vždy rostoucí, a proto jsou jednotlivé body spojeny úsečkou.

Graf **XT** (Obrázek 19) zobrazuje průběh akční a výstupní veličiny v závislosti na čase.



Obrázek 19 – Vizualizace dat (Identifikace) – graf XT

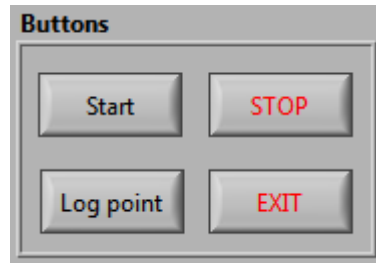
Graf **XY** (Obrázek 20) zobrazuje průběh výstupní veličiny $y(t)$ v závislosti na akční veličině $u(t)$.



Obrázek 20 – Vizualizace dat – graf XY

Buttons

Blok Buttons (Obrázek 21) obsahuje tlačítka pro ovládání aplikace.



Obrázek 21 – Tlačítka

Význam jednotlivých tlačítek je následující.

Start – spouští měření. (Dialog pro výběr souboru)

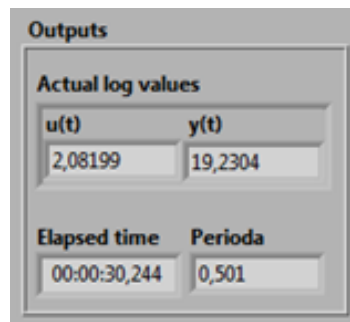
STOP – ukončuje měření.

Log point – při stisku uloží aktuální (následující) data (jeden bod). Je nutné mít vypnuté ukládání všech dat a přepnuto na *Log point* v sekci Menu -> Mode.

EXIT – Ukončí aplikaci a zavře Front Panel.

Outputs

Blok Outputs (Obrázek 22) slouží k zobrazení aktuálních výstupů aplikace.



Obrázek 22 – Výstupy

Význam jednotlivých výstupů je následující.

u(t) – akční veličina.

y(t) – výstupní veličina.

Elapsed time – čas ve formátu (dd HH:mm:ss.SSS) změřený od začátku měření.

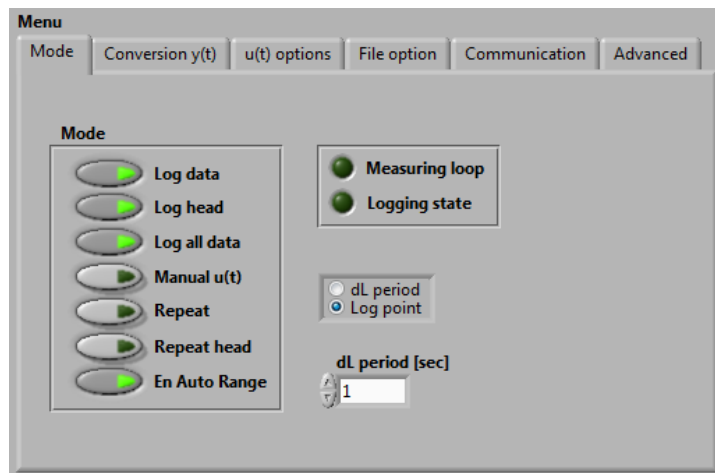
Perioda – aktuální doba mezi dvěma měřenými body.

Menu

Slouží k nastavení parametrů aplikace. Skládá se z několika dalších subbloků.

Mode

Tento subblok (Obrázek 23) obsahuje aktuální nastavení konfigurace ukládání a indikaci stavu měření a ukládání.



Obrázek 23 – Mód aplikace

Význam jednotlivých prvků je následující.

Log data – zapíná/vypíná ukládání dat do souboru.

Log head – zapíná/vypíná ukládání hlavičky do souboru.

Log all data – zapíná/vypíná ukládání všech dat do souboru. Pokud je tento přepínač v pozici vypnuto, je možné ukládat data ve dvou následujících režimech.

dL period – pokud je zaškrtnutý tento radiobutton, data jsou ukládána s periodou dL. Perioda je pouze přibližná a musí být nejméně dvojnásobek předpokládané periody měření.

Log point – umožňuje ukládat data po jednotlivých bodech.

Manual u(t) – přepíná mezi manuálním ovládáním a generováním signálu $u(t)$ podle vybraného zdroje (viz dále).

Repeat – je-li v pozici zapnuto, měření běží do doby, než uživatel stiskne tlačítko STOP nebo Exit. V opačném případě měření probíhá po dobu definovanou v $u(t)$ definiton.

Repeat head – zapíná/vypíná ukládání hlavičky při opakování akčního signálů $u(t)$.

En Auto Range – zapíná/vypíná automatický rozsah multimetru.

Význam jednotlivých LED je následující

Measuring loop – datový tok signálu se nachází v aktivní části měřicí smyčky.

Logging state – datový tok signálu se nachází v některém ze stavů, v němž se ukládají data (Head, Log, Log init).

Příklad 1:

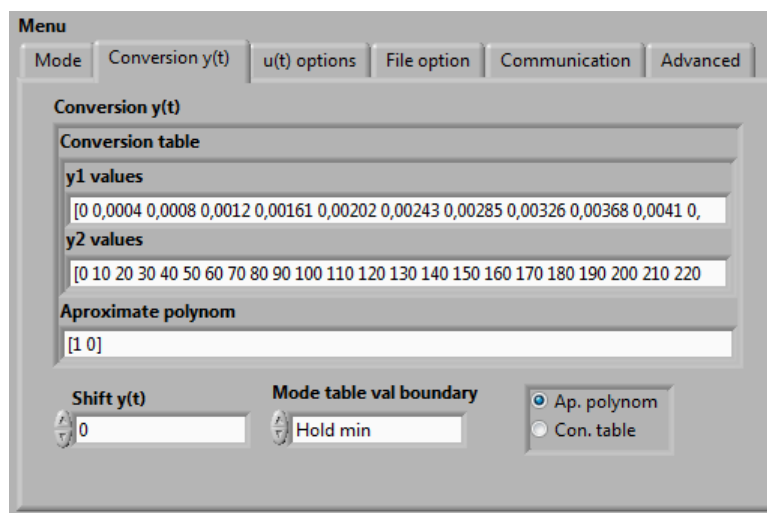
Chceme-li měřit VA charakteristiku, nadefinujeme v *u(t)definition* pilovitý signál (viz *u(t) options*). Zapneme *Log all data* a vypneme *Manual u(t)* a *Repeat*.

Příklad 2:

Chceme-li ukládat jednotlivé body, vypneme *Log all data* a zapneme *Repeat*. Zaškrtneme radiobutton *Log Point* a tlačítkem *Log Point* můžeme ukládat jednotlivé body.

Conversion y(t)

Toto submenu (Obrázek 24) slouží k definování parametrů převodu výstupní veličiny $y(t)$. Výstupní veličina projde transformací vždy. Výchozí nastavení převede výstupní veličinu aproximačním polynomem $y_2(t) = 1 y_1(t) + 0$ (zapsáno jako [1 0]). Dochází tedy k násobení výstupní veličiny číslem 1



Obrázek 24 – Transformace výstupní veličiny $y(t)$

Na výběr máme ze dvou druhů převodů, které je možné přepínat pomocí Radiobuttonů.

Aproximate polynom – udává hodnoty koeficientů transformačního polynomu tak, že vlevo je koeficient nejvyšší mocniny a vpravo je absolutní člen.

Conversion table – udává převodní tabulku mezi vstupem a výstupem. V poli *y1 values* jsou hodnoty vstupní veličiny $y_1(t)$ a v poli *y2 values* jsou hodnoty transformované veličiny $y_2(t)$. Funkce je mezi body spojena úsečkami. Pokud přijde na vstup hodnota mimo rozsah *y1 values*, chová se aplikace podle zvoleného **Mode table val boundary**.

Shift y(t) – Přičte zadanou hodnotu k transformované hodnotě $y_2(t)$.

Mode table val boundary

Slouží pro popis chování aplikace v případě, že vstupní hodnota $y_1(t)$ bude mimo rozsah *y1 values*. Na výběr je celkem 6 módů.

Base – bude-li vstupní hodnota mimo rozsah, je generována chyba „*Input is out of range.*“ a měření je ukončeno.

Hold min – bude-li vstupní hodnota menší než spodní hranice povoleného intervalu (první hodnota *y1 values*), aplikace podrží na výstupu první hodnotu *y2 values*. V případě hodnoty větší než maximální (poslední hodnota *y1 values*) je generována chyba a měření je ukončeno.

Hold max – bude-li vstupní hodnota větší než horní hranice povoleného intervalu, aplikace podrží na výstupu poslední hodnotu *y2 values*. V případě hodnoty menší než minimální je generována chyba a měření je ukončeno.

Hold min&max – kombinace předchozích 2 módů.

Mod – provede se vstupní hodnotou modulo velikosti intervalu *y1 values*.

Abs input – dá vstupní hodnotu do absolutní hodnoty a provede transformaci.

Příklad 1:

Budeme-li chtít měřit VA charakteristiku kde proud chceme ukládat v mA, zaškrtneme radiobutton *Ap. polynom* a do pole *Aproximate polynom* zapíšeme výraz [1e3 0] (závorky nejsou povinné).

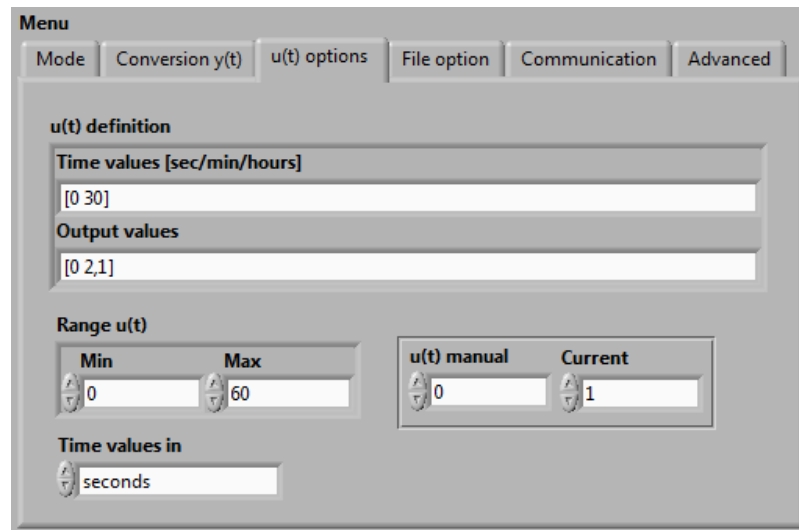
Příklad 2:

Budeme-li chtít měřit teplotu pomocí termočládku. Přepneme na radiobutton *Con. Table* a do polí *Conversion table* vložíme převodní tabulku čidla.

Pozn.: Conversion table podporuje i definici skokových signálů

u(t) options

Slouží k nastavení parametrů signálu akční veličiny $u(t)$.



Obrázek 25 – Nastavení parametrů akční veličiny $u(t)$

Význam jednotlivých prvků je následující.

u(t) definition – určuje tvar generovaného signálu. Ve výchozím stavu je nastaven na pilový průběh napětí: 0 - 2,1 V a 0 - 30 s (připraveno pro měření VA charakteristiky červené LED).

Time values [sec/min/hours] – časový vektor signálu. Vstup se zadává ve tvaru $[t_0 \ t_1 \dots \ t_n]$ (závorky nejsou povinné).

Output values – vektor hodnot (napětí) signálu. Hodnoty se zadávají ve tvaru $[u_0 \ u_1 \dots \ u_n]$.

Range u(t) – minimální a maximální hodnota, která jde na výstup zdroje. Výchozí hodnoty jsou nastaveny na rozsah napěťového zdroje 0 - 60 V.

u(t) manual – slouží k ručnímu zadávání napěťových hodnot na výstupu zdroje (musí být povoleno v záložce **Mode**).

Current – proudové omezení zdroje.

Time values in – umožňuje určit jednotku časového vektoru signálu v sekundách, minutách nebo hodinách.

Pozn.: Před začátkem měření je nutné nastavit dostatečně velkou hodnotu Current, aby mohl být dosažen požadovaný výkon. Aplikace pouze nastavuje hodnotu napětí, která má být na výstupu zdroje. Nekontroluje však, zda zadaná hodnota odpovídá skutečnosti.

Příklad 1:

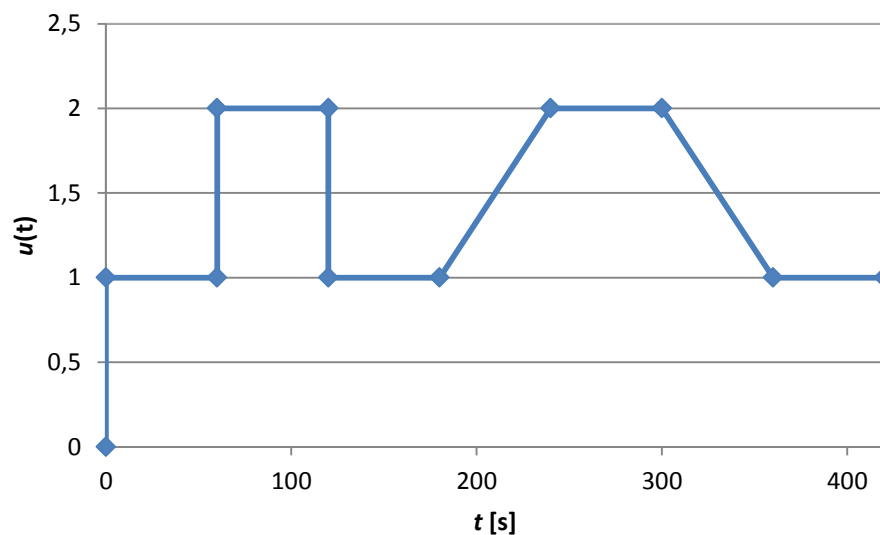
Chceme-li ovládat zdroj ručně, přepneme v záložce **Mode** přepínač *Manual u(t)* do polohy zapnuto. Hodnota zadaná v *u(t) manual* pak bude na výstupu zdroje (změna se projeví s další periodou měření). Je také dobré zapnout přepínač *Repeat*, jinak by měření skončilo v čase určeném definicí signálu.

Příklad 2:

Chceme-li namodelovat na výstup zdroje určitý tvar signálu, přepneme v záložce **Mode** přepínač *Manual u(t)* do polohy vypnuto. Pokud necháme zapnutý přepínač *Repeat*, bude se průběh signálu opakovat (jinak měření skončí po jednom průběhu signálu). Ponecháme-li zapnutou i volbu *Repeat head* (v záložce **Mode**), bude na začátku každého opakování vložena do souboru hlavička. Pro namodelování signálu na obrázku (Obrázek 26) budou hodnoty *u(t) definition* následující:

Tabulka 4 – Definice signálu

Time values [min]	[0 0 1 1 2 2 3 4 5 6 7]
Output values	[0 1 1 2 2 1 1 2 2 1 1]

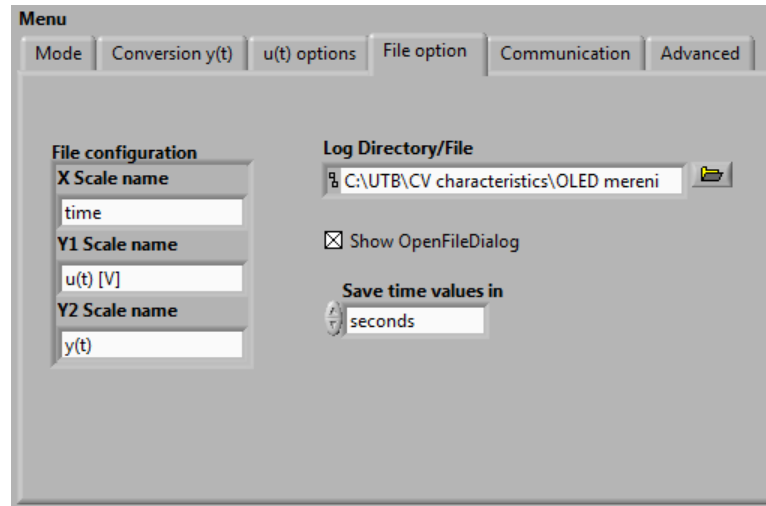


Obrázek 26 – Příklad modelovaného signálu $u(t)$

Pozn.: Při definici delšího časového úseku je vhodné využít volbu Time values in. Tato volba nemá vliv na jednotku času ukládaného do souboru. Aplikace vnitřně pracuje v sekundách a proto všechny objekty, které využívají časový vstup, kontrolují vybranou jednotku v popup menu.

File option

Slouží k nastavení vlastností ukládaného souboru.



Obrázek 27 – Nastavení vlastností souboru

Význam jednotlivých prvků je následující.

Log Directory/File – výchozí adresář nebo soubor.

Show OpenFileDialog – v případě hodnoty *true* je hodnota předchozího prvku výchozím adresářem a soubor je vybrán v OpenFileDialogu po stisknutí tlačítka **Start**.

V případě hodnoty *false* je hodnota předchozího prvku cestou k souboru, kam se mají ukládat data a OpenFileDialog se nezobrazí.

Save time values in – určuje časovou jednotku ukládaných dat (sekundy, minuty, hodiny).

File configuration – určuje, jaký tvar bude mít hlavička souboru (popis sloupců). Řetězce, které jsou zadány, budou uloženy do hlavičky souboru jako názvy jednotlivých měřených veličin (viz kapitola Soubor).

Příklad:

Budeme-li chtít měřit VA charakteristiku, přepneme funkci multimetru (viz Communication) na měření stejnosměrného proudu. Do File configuration vyplníme následující údaje a do zdroje napětí pustíme pilovitý signál.

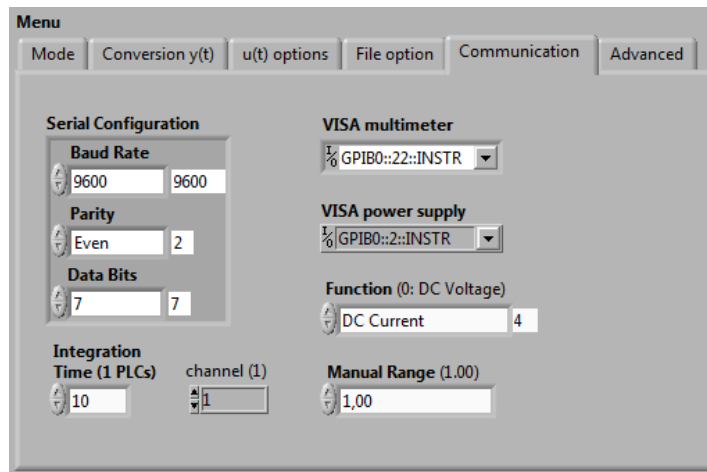
Tabulka 5 – Hlavička – popis sloupců

X Scale name	Čas [s]
Y1 Scale name	Napětí zdroje [V]
Y2 Scale name	Proud [mA]

Pozn.: Hodnota Save time values in nemá vliv na časovou jednotku se kterou se pracuje v aplikaci. Lze tak například definovat průběh signálu v minutách, ale ukládat jej v sekundách.

Communication

Slouží k nastavení parametrů komunikace se zdrojem a multimetrem.



Obrázek 28 – Nastavení parametrů komunikace se zdrojem a multimetrem

Význam jednotlivých prvků je následující.

Serial Configuration – parametry komunikace v případě připojení multimetru sériovou linkou.

VISA multimeter – adresa zařízení multimetru na sběrnici GPIB.

VISA power supply – adresa zařízení zdroje na sběrnici GPIB.

Function – funkce multimetru (měřená elektrická veličina).

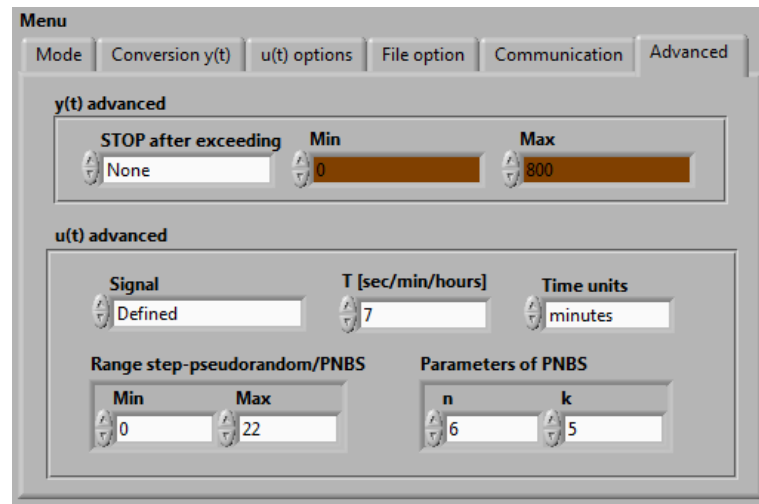
Integration Time – integrační doba multimetru. Souvisí s přesností měření.

channel – kanál zdroje.

Manual Range – manuální nastavení rozsahu multimetru.

Advanced

Slouží k pokročilejšímu nastavení aplikace.



Obrázek 29 – Nastavení pokročilých parametrů aplikace

Význam jednotlivých prvků je následující.

y(t) advanced – pokročilé nastavení výstupní veličiny.

STOP after exceeding – nastavení módu ukončení měření při hodnotě výstupní veličiny (převedené) mimo zadané meze. Na výběr jsou celkem 4 módy.

None – hodnota výstupní veličiny může být libovolná.

Min – hodnota výstupní veličiny nesmí být menší než hodnota nastavená v ovládacím prvku Min. V opačném případě je měření ukončeno a aplikace přechází do nastavovacího módu.

Max – hodnota výstupní veličiny nesmí překročit hodnotu Max.

Min/Max – hodnota výstupní veličiny se nesmí nalézat mimo interval zadaný ovládacími prvky Min a Max.

Min – určuje minimální hodnotu výstupní veličiny, při níž může probíhat měření. Pokud je pozadí prvku hnědé barvy, není na hodnotu tohoto prvku brán ohled, ale je možné tuto hodnotu změnit.

Max – určuje maximální možnou hodnotu výstupní veličiny, při níž může probíhat měření.

u(t) advanced – pokročilé nastavení akční veličiny.

Signal – určuje zdroj signálu akční veličiny, pokud není zvolen manuální režim.

Defined – zdrojem signálu je signál vytvořený podle **u(t) definition**.

Step-pseudorandom – zdrojem signálu je schodovitý pseudonáhodný signál s rovnoměrným rozdělením v zadaném intervalu.

PNBS – zdrojem signálu je generátor PNBS.

T [sec/min/hours] – určuje hodnotu periody změn pseudonáhodného nebo PNBS signálu.

Time units – určuje časovou jednotku periody (sekundy, minuty, hodiny).

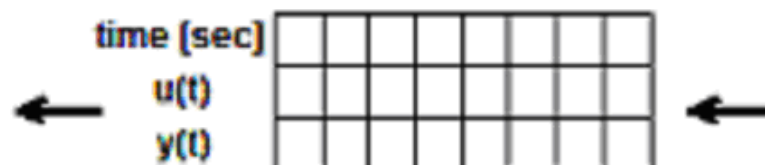
Range step-pseudorandom/PNBS – určuje rozsah pseudonáhodného signálu nebo krajní hodnoty PNBS (amplituda).

Parameters of PNBS – parametry PNBS generátoru (n - počet buněk posuvného registru, k – stupeň vnitřní zpětné vazby).

Pozn.: Změna generovaného pseudonáhodného nebo PNBS signálu podle periody se určuje porovnáním celočíselného podílu času a periody s jeho předchozí hodnotou. Při změně periody v průběhu měření může tedy dojít k nečekaným změnám signálu. Doporučuje se tedy periodu změny signálu během měření neměnit.

5.2.4 Zobrazení

Veličiny $time$, $u(t)$ a $y(t)$ se zobrazují v XT a XY grafu. Tyto hodnoty se vkládají do vyrovnávací paměti (Obrázek 30), která udržuje tolik hodnot, kolik odpovídá jednomu modelovanému průběhu $u(t)$. Necháme-li tedy *Time values* ve výchozí hodnotě [0 30], bude v XT grafu zobrazeno 30 s. Paměť je typu FIFO, takže po dosažení času 30 s budou následující hodnoty „vytlačovat“ ty původní z fronty.



Obrázek 30 – Vyrovnávací paměť pro hodnoty zobrazené v grafech XT a XY

5.2.5 Fronta

Z blokového schématu (Obrázek 16) je patrné, že se aplikace skládá ze dvou hlavních smyček: měřicí a ukládací. Aby se mohlo s daty pracovat, je potřeba, aby spolu tyto 2 smyčky komunikovaly. Data se tedy předávají pomocí fronty, což umožňuje potlačit

problémy způsobené paralelním během dvou vláken. Fronta² je typu cluster³ polí⁴, což výrazně ulehčuje budoucí vývoj aplikace z hlediska počtu předávaných parametrů frontou.

Jako první parametr se předává **čas** (časové razítko), který byl změřen na začátku měřicí smyčky. V poli typu double se předávají veličiny $u(t)$ a $y(t)$ (převedená). Dále se pak předávají další pomocné informace, které pomáhají zajistit správný běh aplikace.

5.3 Soubor

Do souboru se ukládají naměřená data, je-li to povoleno. O ukládání dat se stará ukládací smyčka, která funguje jako stavový automat. Nejdříve je do souboru uložena hlavička (je-li to povoleno) s prvním řádkem naměřených dat (u stavu Log Point se první data neukládají). Poté se již přejde do stavu *Log*, kde jsou ukládána samotná data. Je-li zakázáno ukládání všech dat, jsou některé stavy *Log* vynechány a stav *Log* se střídá se stavem *Idle* (viz kapitola 5.4.11). Soubor je ukládán ve tvaru

```
*****
Integration Time [NPLC]
0,02
-----
time [sec]          u(t)          y(t)
0          1,3371      0,004732
0,095006  1,532659     -0,012922
...
```

Pozn.: Do souboru se ukládá čas od časového razítka doručeného frontou při přechodu do stavu Head (tj. čas_naměřený – čas_ohlavičkování). Pokud je ukládání hlavičky zakázané, ukládá se čas od prvního doručeného razítka.

² Fronta je zásobník typu FIFO (First In First Out), který obsahuje synchronizační mechanismy pro bezpečné předávání jejího obsahu.

³ Cluster je datový objekt, který umožňuje sdružit objekty různých datových typů.

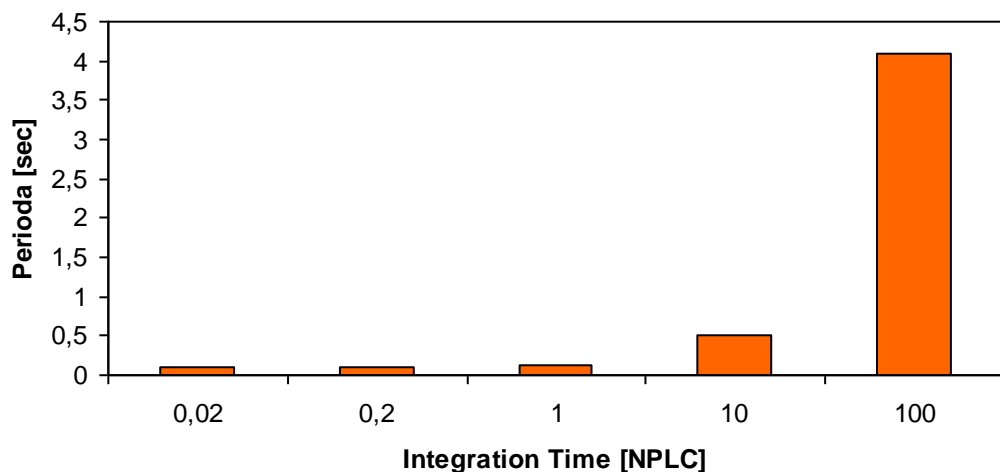
⁴ Pole je datový objekt, který umožňuje sdružit objekty stejných datových typů. Každý prvek pole má svůj index (pořadí), což umožňuje snadnou práci s prvky pole (na rozdíl od zásobníku není nutné pro vyvolání některého prvku uprostřed pole všechny předchozí prvky vyjmout).

5.3.1 Generování akční veličiny $u(t)$

Pro generování signálu se používá stejné subVI, jako pro převod výstupní veličiny $y(t)$ pomocí tabulky. SubVI je přepnuto do módu *Mod* což umožňuje periodické opakování signálu. Aktuální hodnota akční veličiny $u(t)$ je spočítána za běhu jako funkce času t (viz $u(t)$ options).

5.3.2 Integration time

Integration time přímo ovlivňuje dobu periody měření multimetru a limituje rychlost měřící smyčky. Při testu rychlosti měření (bez okolního kódu a připojené zátěže) byly naměřeny následující hodnoty periody:



Obrázek 31 – Závislost Periody měření na nastavení Integration Time

Při zvýšení rychlosti měření (zmenšení periody) se ovšem snižuje přesnost měřených dat. Odpovídající hodnoty rozlišení jsou vidět v následující tabulce:

Tabulka 6 – Závislost Resolution na Integration Time

Integration Time	Resolution
0,02 NPLC	0,0001 × Full-Scale
0,2 NPLC	0,00001 × Full-Scale
1 NPLC	0,000003 × Full-Scale
10 NPLC	0,000001 × Full-Scale
100 NPLC	0,0000003 × Full-Scale

Pozn.: Velikost periody v závislosti na Integration time byla změřena v testovací aplikaci pro měření s multimetrem. Skutečná velikost periody tedy nemusí odpovídat naměřeným hodnotám.

5.4 Vývoj aplikace

Aplikace byla naprogramována v prostředí NI LabView 2011. Jednotlivé funkce aplikace byly přidávány postupně (přírůstkově) podle jednotlivých požadavků, které jsou vyžadovány pro účely aplikace. Při tvorbě aplikace byly základem projektu examples „Agilent 34401 Read Single Measurement.vi“ a „HP6xxxA Set Voltage/Current.vi“. Tyto programy byly modifikovány a použity v programu pro měření dat. Aplikace splňuje všechny požadavky.

5.4.1 Ovládací prvky

Kromě výchozích ovládacích prvků byly vytvořeny i vlastní ovládací prvky. Jedná se zejména o PopupMenu⁵ s výčtovými typy⁶ a také jeden cluster.

5.4.2 Převod výstupní veličiny

Na vstup multimetru je možné umístit měřicí člen, který měří různé fyzikální veličiny. Často však tuto veličinu neměří přímo, ale pomocí jiné fyzikální veličiny (např. napětí). Proto je nutné výstupní veličinu převést do stejných jednotek, jako je měřená fyzikální veličina. Aplikace umožňuje tento převod provést dvěma způsoby.

Prvním způsobem je převod pomocí aproximačního polynomu. To se hodí v případě, že známe průběh funkce nebo máme k dispozici tabulku hodnot. Tento způsob je výhodný proto, že je převedena výstupní veličiny v celém rozsahu reálných čísel (rozsahu double). Tento způsob může být však nepřesný a mimo předpokládaný interval může dokonce vykazovat zcela chybné hodnoty. Do aplikace zadáváme pouze hodnoty koeficientů v pořadí od nejvyšší mocniny k nejnižší.

Zápisu $[a_n a_{n-1} \dots a_1 a_0]$ odpovídá převod

⁵ PopupMenu je rozbalovací menu.

⁶ Výčtový typ je dvojice hodnot, kdy jedna je typu integer (celočíselná) a druhá je typu řetězec. Výčtový typ je výhodný pro lepší srozumitelnost uživateli. Jedná se vlastně o slovník, kdy k danému klíči řetězci je přiřazena číselná hodnota, která je lépe zpracovatelná počítačovým algoritmem.

(43)

$$y_2 = a_n y_1^n + a_{n-1} y_1^{n-1} + \dots + a_1 y_1 + a_0$$

kde y_1 je pomocná fyzikální veličina (např. napětí) a y_2 je měřená výstupní veličina (např. teplota).

Druhým způsobem je převod tabulkou. Aplikace obsahuje 2 formuláře, kam se zadávají hodnoty pomocné veličiny (y_1 values) a hodnoty měřené veličiny (y_2 values). Jelikož jsou tyto hodnoty zadány diskrétně, je potřeba mezilehlé body dopočítat. Mezilehlý bod pak leží na úsečce spojující dva sousední body veličiny. Při dostatečné hustotě bodů je tento způsob docela přesný, avšak zadané hodnoty leží v určitém intervalu, takže je nutné řešit hraniční případy. Hraniční případy se řeší podle zvoleného módu (viz Mode table val boundary). Převod tabulkou podporuje také tvorbu skokových signálů. Algoritmus převodu je následující.

1. Zkontroluj, zda je počet prvků y_1 values a y_2 values totožný, a zda je počet prvků větší nebo rovno dvěma.
2. Zkontroluj, zda se bod nachází v celém intervalu y_1 values, jinak se zachovej podle zvoleného módu.
3. Zkontroluj správné pořadí y_1 values ($y_1(n) \geq y_1(n-1)$), jinak vygeneruj chybu „Next "y1 value" element is smaller then previous.“.
4. Vyhledej subinterval, do kterého bod patří. Pokud bod leží na hranici, patří tento bod do pravého subintervalu. (Pozn.: subinterval se hledá při kontrole pořadí hodnot y_1 values)
5. Pokud v daném intervalu platí $y_1(n) = y_1(n-1)$ nastav na výstup hodnotu $y_2(n)$.
6. Pokud platí $y_1(n) \neq y_1(n-1)$ vypočítej hodnotu y_2 podle následujícího vzorce.

$$y_2 = ay_1 + b$$

$$a = \frac{y_2(n) - y_2(n-1)}{y_1(n) - y_1(n-1)} \quad (44)$$

$$b = y_2(n-1) - ay_1(n-1)$$

kde

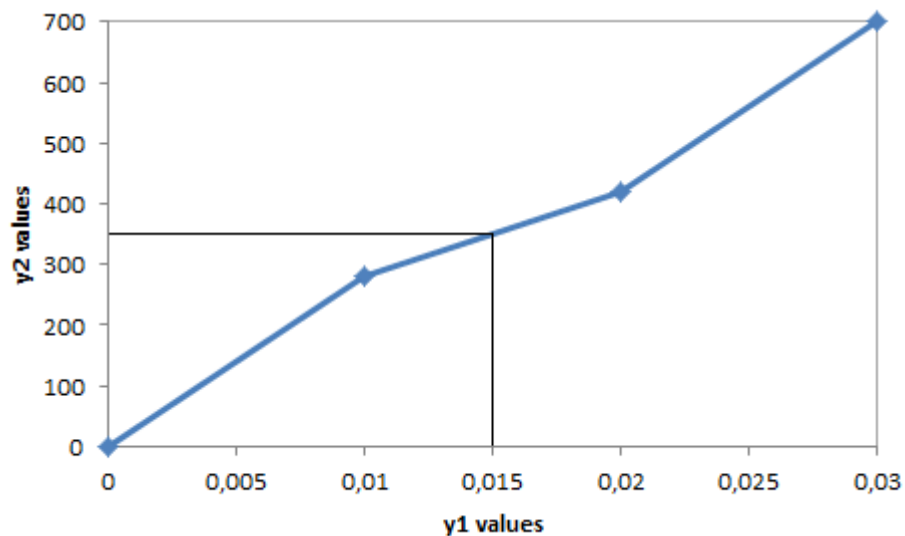
y_1	naměřená pomocná veličina
y_2	převedené výstupní veličina

$y_1(n-1)$	y_1 values předcházející naměřené y_1
$y_1(n)$	y_1 values následující naměřené y_1
$y_2(n-1)$	y_2 values odpovídající bodu $y_1(n-1)$
$y_2(n)$	y_2 values odpovídající bodu $y_1(n)$
a	směrnice přímky (úsečky)
b	absolutní člen přímky (úsečky)

Na obrázku níže máme příklad převodu výstupní veličiny tabulkou. Pro tento příklad je nutné zadat následující hodnoty.

Tabulka 7 – Převod výstupní veličiny

y1 values	[0 0,01 0,02 0,03]
y2 values	[0 280 420 700]



Obrázek 32 – Příklad převodu výstupní veličiny tabulkou

K převedené (měřené) výstupní veličině je pak přičten offset „y(t) shift“, neboť např. termoelektrický článek měří relativně vzhledem k referenční hodnotě. Můžeme zde tedy zahrnout např. teplotu v místnosti apod. Pro výslednou převedenou veličinu tedy platí

$$y' = y_2 + y_{shift} \quad (45)$$

5.4.3 Generování signálu

Aplikace umožňuje generovat signál ručně nebo podle vybraného zdroje. Výstup signálu může být omezen minimální a maximální hodnotou. Při generování signálu je také třeba dát pozor, aby proudové omezení „Current“ bylo dostatečné, abychom dosáhli požadovaného výkonu.

Mezi ručním ovládáním a ovládáním podle definice (případně náhodným či PNBS signálem) lze přepínat pomocí přepínače **Manual u(t)**. Pokud však ovládáme signál ručně, jsou dále „vnitřně“ generovány ostatní typy signálů, z čehož vyplývá, že hodnoty zadané v definici signálu ovlivňují dobu měření (pokud je **Repeat** vypnuto) a velikost zobrazovaného intervalu v grafu.

Pro generování signálu definicí je použito stejné subVI, jako v případě převodu výstupní veličiny tabulkou. Toto subVI je však přepnuto do módu Mod, takže v případě opakovaného (Repeat) signálu je signál generován periodicky. Jakmile je dosaženo konce signálu, je vyslán impuls pro ukončení měření. Tento impuls je také odeslán ukládací smyčce jako řídicí signál v případě potřeby opakovat hlavičku.

5.4.4 Ukládání dat do souboru

Měření musí probíhat vždy nad určitým souborem a to i přesto, že data nejsou ukládána (vytvoří se prázdný soubor). V inicializační části dochází k otevření daného souboru, v ukládací smyčce jsou pak data ukládána a po skončení měření je opět soubor zavřen. V případě výskytu chyby je ukončeno měření, avšak aplikace se dostane do nastavovacího módu a je možné spustit nové měření. Do souboru lze ukládat buď všechna data, vybraná data dle zvolené periody nebo jednotlivé body. Uživatelská perioda dL je pouze přibližná, neboť perioda měření závisí hlavně na Integration time. Perioda dL musí být nejméně dvojnásobná oproti předpokládané periodě měření, neboť ukládací smyčka funguje jako stavový automat a přechod mezi těmito stavy trvá jednu měřicí periodu.

Data se ukládají buď s hlavičkou, nebo bez hlavičky. Hlavička obsahuje hodnotu Integration Time a popis jednotlivých sloupců lze zadat v clusteru **File configuration**.

V případě opakování signálu lze hlavičku uložit na začátek každého opakování, ale jelikož je do souboru uložen čas od doby ohlavičkování, bude každé toto opakování začínat s nulovým ukládacím časem.

V aplikaci ve verzi 2.4 je možné zvolit výchozí adresář nebo soubor. Soubor je zadán přímo, nebo se vybere pomocí OpenFileDialogu, který se objeví až po stisku tlačítka **Start**. Již existující soubor je možné přepsat, popř. zrušit volbu a při startu zvolit soubor nový. Pokud není do jména souboru zadána jeho přípona, je soubor ukládán ve formátu tsv (Tab Separated Values). V opačném případě je soubor uložen se zvolenou příponou, avšak struktura dat je stále ve formátu tsv.

5.4.5 Komunikace

Multimetr i zdroj jsou připojeny na GPIB sběrnici. Do počítače jsou připojeny pomocí GPIB/USB redukce. Program s přístroji komunikuje pomocí VISA rozhraní a každý přístroj má přidělenou svoji GPIB adresu.

Aby mohla komunikace fungovat, je nutné mít na cílovém počítači nainstalovaný GPIB ovladač a podporu VISA rozhraní. Tyto installery byly přidány již do instalačního souboru v sekci Additional Installers (viz kapitolu Tvorba instalačního souboru).

5.4.6 Zobrazení časového průběhu

Jedním z požadavků bylo zobrazení časového průběhu v reálném čase. Při použití klasického waveformu⁷ jsou však data zobrazena tak, že na ose x je pořadí dané hodnoty (případně je možné zvolit velikost násobku). Aby bylo možné zobrazit data v závislosti na čase, je nutné použít XY graf. Tento graf očekává na svém vstupu cluster dvou polí, kde první řádek hodnot je nezávislá veličina (osa x) a druhý řádek hodnot závislá veličina (osa y). Aby bylo možné zobrazit více průběhů v jednom grafu, musí se tyto clustery sjednotit do jednoho pole a teprve toto je přivedeno na vstup XY grafu.

Hodnoty, které se mají zobrazit, se musí někde uchovávat, a proto byl vytvořen FIFO zásobník datového typu cluster polí. Jednotlivé prvky pole $(t, u(t), y(t))$ se přidávají do jednotlivých polí až do dosažení jednoho průběhu signálu $u(t)$ (podle definice). Po přijetí dalších prvků dochází k posunu všech polí tak, že první prvky jsou z paměti vytlačovány a počet prvků je již konstantní. Toto zamezuje nekonečnému růstu paměťové oblasti. Tato vlastnost však také způsobuje, že v grafu je vždy zobrazen časový úsek odpovídající velikosti jednoho průběhu $u(t)$ a to i v případě ovládní napěťového zdroje manuálně. Nezobrazená data však zůstávají uložena v souboru.

⁷ Waveform je typ grafu v prostředí LabView (indikátor).

5.4.7 Data flow⁸ a chybové linky

Datové linky mezi jednotlivými moduly (subVI) určují tok dat a tím je zajištěna jednoznačná posloupnost operací. V případě více nezávislých toků je každý tento tok dat samostatným vláknem. V případě více vláken není tedy jednoznačně určeno, která část se vykoná jako první. Pro správné fungování aplikace je tedy velmi důležité dodržovat Data flow.

Jednotlivé moduly se vykonají až v momentě, kdy mají všechna vstupní data (pokud není vstup připojen, má výchozí hodnotu). Poté provedou svoji činnost a aktivují výstup. Data flow mezi dvěma moduly lze tedy jednoduše vytvořit jejich spojením linkou. K určení data flow je výhodné využívat chybových linek.

Používání proměnných přerušuje data flow, a proto by jejich výskyt měl být co nejmenší. V případě paralelního běhu dvou vláken mohou spolu tato vlákna „komunikovat“ pomocí synchronizačních prostředků, které zabraňují výskytu problémů paralelního programování, jako jsou např. race condition⁹ či deadlock¹⁰. Takovým komunikačním prostředkem může být například notifikátor¹¹, fronta¹² či semafor¹³.

K určení data flow se v měřicí aplikaci využívá právě chybových linek, které začínají jako jedno vlákno. V průběhu programu se rozvětví do měřicí a ukládací smyčky a na konci se opět spojí pomocí prvku Merge.

Měřicí aplikace obsahuje 2 paralelní smyčky (měřicí, ukládací). Data si předávají pomocí fronty, což zajišťuje jejich vzájemnou synchronizaci. Aplikace obsahuje i další smyčky jako je například inicializační smyčka, která udrží aplikaci v nastavovacím módu a vyčká na stisk tlačítka **Start** nebo **Exit**. Po stisku tlačítka proběhne inicializační část a následují právě měřicí a ukládací smyčka. Po ukončení měření proběhne CLOSE blok, který se postará o uvolnění potřebných zdrojů. Aby bylo možné měření opakovat, je celý kód aplikace umístěn v další velké smyčce.

⁸ Data flow je způsob určení posloupností operací tak, aby měly přesně definované pořadí.

⁹ Race condition (česky souběh) je situace, kdy dvě nebo více vláken pracují s daty v kritické oblasti.

¹⁰ Deadlock je situace, kdy si dvě vlákna (nebo více) vzájemně drží prostředky a jejich běh je tak blokován.

¹¹ Notifikátor je prvek podobný frontě, avšak uchovává pouze jednu hodnotu.

¹² FIFO (First In First Out) zásobník pro předávání dat.

¹³ Semafor je synchronizační prvek pro přidělování zdrojů.

Ukončení běhu 2 paralelních cyklů

Jedním z požadavků na aplikaci je ukončení aplikace při výskytu jakékoliv chyby popř. na žádost uživatele. Jelikož aplikace obsahuje dvě paralelní smyčky, je nutné je správně ukončit. Pro ukončení obou smyček je využito dvou lokálních funkčních proměnných (viz dále), které komunikují mezi smyčkami do kříže. Použití dvou proměnných zamezuje výskytu souběhu typu zápis-zápis, který by mohl nastat v případě sdílené proměnné. V případě ukončení jedné smyčky je druhá smyčka okamžitě informována a naopak. Oba cykly jsou ukončeny při výskytu jakékoliv chyby nebo při stisknutí tlačítka **STOP** nebo **EXIT**.

5.4.8 Ošetření race condition a deadlock

Jak už bylo zmíněno, je potřeba synchronizovat paralelně běžící vlákna. Jelikož zápis a čtení není obecně atomická operace, mohlo by v případě použití proměnné dojít k tomu, že bude čtení a zápis dat probíhat současně. Výsledkem by mohly být chybně předané hodnoty. V případě nestejného počtu cyklů smyček by některá data mohla být ztracena či duplikována. Proto je pro komunikaci mezi smyčkami použit synchronizační prostředek fronta. Ta zajistí bezztrátovost dat a také ošetření race condition.

Fronta má navíc další funkci, která umožňuje ošetřit deadlock. V případě stisknutí tlačítka STOP nebo EXIT by totiž mohlo dojít k situaci, že se ukončí měřicí smyčka, zatímco ukládací smyčka bude čekat na data, která nikdy nepřijdou. Jelikož prvek Merge, který spojuje obě vlákna, nemůže pokračovat dokud nejsou k dispozici všechny vstupy, nastane deadlock. Řešením je nastavit frontě minimální čas, po který se na data čeká. V případě, že data nepřijdou, je přeskočena celá ukládací smyčka a je vyhodnocena podmínka ukončení cyklu.

5.4.9 Ošetření chyb

Moduly, ve kterých by se mohla objevit chyba, obsahují chybový vstup a výstup. Při výskytu chyby je obvykle přeskočeno celé tělo funkce.

Každá chyba má svůj status, code a source. Status je datového typu boolean¹⁴, který v případě hodnoty *true* indikuje výskyt chyby, code určuje kód chyby a source obsahuje textový popis. V případě že je definován code, ale status je na hodnotě *false*, je toto bráno jako varování.

¹⁴ Boolean je datový typ který nabývá dvou hodnot označovaných většinou jako *true* a *false*.

Kromě předdefinovaných chyb je možné vygenerovat si chyby vlastní. Při převodu výstupní veličiny může být například vygenerována vlastní chyba „Input is out of range“, která signalizuje, že je pomocná veličina mimo rozsah y1 values.

Chybové linky zároveň definují data flow. Na konci aplikace je umístěn dialog, který v případě chyby zobrazí informaci o chybě.

5.4.10 Shift registr a lokální funkční proměnná

Přivedeme-li data na vstup smyčky, vytvoří se datový tunel, který obsahuje stejná (počáteční) vstupní data přes všechny iterace smyčky. Často je však potřeba předávat data mezi jednotlivými iteracemi. Ukládací smyčka funguje jako stavový automat. Jednotlivé stavy se mezi iteracemi předávají právě pomocí shift registrů. Shift registr je tedy paměť, která má v sobě uloženou hodnotu z minulé iterace. Shift registr může být na začátku inicializovaný (je na něj přivedena linka) nebo neinicializovaný (obsahuje na začátku výchozí hodnotu).

Lokální funkční proměnná je paměťová oblast, kde si lze uchovat data. Kromě samotného uchování dat je však možné provést nad daty určitou funkci (proto funkční proměnná). Tato proměnná je vlastně tvořena smyčkou obsahující shift registr, který však nemá počáteční inicializaci. V případě, že obsahuje ještě case¹⁵ strukturu, je možné externě volit funkci podle aktuálních požadavků (např. inicializuj, zapiš a čti).

5.4.11 Ukládací smyčka – stavový automat

Ukládací smyčka funguje jako stavový automat. Jednotlivé stavy se přepínají podle aktuální konfigurace nastavení ukládání a podle aktuálního stavu. Tělo jednotlivých stavů je pak implementováno uvnitř case struktury (funkce stavu a přechodová logika je oddělená).

Jelikož zakreslení stavového automatu by bylo nepřehledné, je zde uvedena pouze tabulka přechodů mezi stavy.

¹⁵ Case je struktura pro větvení programu na základě hodnoty testovaného vstupu.

Tabulka 8 – Stavový automat – tabulka přechodů stavů

Hodnoty v poli jsou zapisovány v pořadí: LD, LH, LAD, CHL, DL_sig, LP. Zápis <i>t</i> vyjadřuje hodnotu <i>true</i> a zápis <i>f</i> hodnotu <i>false</i> . Na hodnotách <i>x</i> nebo chybějících nezáleží.							
aktuální stav	nový stav						
	Init	Init_first_ts	Wait	Head	Log	Idle	Log_init
Init	-	else	-	t, t	-	-	t, f
Init_first_ts	-	-	f	t, t, t	else	(t, x, f, 0, f), (t, x, f, 1, x, f)	-
Wait	-	-	f	t, t, t	else	(t, x, f, 0, f), (t, x, f, 1, x, f)	-
Head	-	-	f	-	else	(t, x, f, 0, f), (t, x, f, 1, x, f)	-
Log	-	-	f	-	else	(t, x, f, 0, f), (t, x, f, 1, x, f)	-
Idle	-	-	f	-	(t, x, f, 0, t), (t, x, f, 1, x, t), (t, x, t)	else	-
Log_init	-	-	f	-	else	(t, x, f, 0, f), (t, x, f, 1, x, f)	-

kde

- LD: Log data
- LH: Log head
- LAD: Log all data
- CHL: Change log
- DL_sig: DL signal
- LP: Log point

Pro jednoduchost není v tabulce uvedena reakce na prepínač Repeat Head. Je-li tento prepínač na hodnotě *true* a nastane-li konec průběhu jednoho signálu definicí, navrací se automat do stavu Head.

Význam jednotlivých stavů je následující.

Init – tento pseudostav se nachází v inicializační části, takže není součástí stavového automatu.

Init_first_ts – tento stav slouží k inicializaci výchozího časového razítka.

Wait – v tomto stavu nejsou ukládána data. Pro přechod do jiného stavu je nutné povolit ukládání dat.

Head – tento stav slouží k uložení hlavičky a prvního řádku měřených dat. Tento stav je stavem průchozím, a proto je-li povoleno ukládání všech dat, přechází se do stavu *Log*, v opačném případě se přechází do stavu *Idle*.

Log – tento stav slouží k ukládání dat. Tento stav se může střídát se stavem *Idle* v případě, že jsou některá data vynechána (souvisí s ukládáním vybraných dat).

Idle –stav, který vyčkává na příchod dalších užitečných dat. Jakmile mají být data uložena, přechází se do stavu *Log*.

Log_init – pomocný stav, který ukládá data a zároveň inicializuje výchozí časové razítko.

5.4.12 Měření a ukládání času

Doba jednoho cyklu měřicí smyčky se skládá z doby potřebné pro komunikaci, doby pro měření veličiny multimetrem (ovlivněno hodnotou Integration Time), doby nastavení napětí na zdroji a doby potřebné k vykonání dalších operací aplikace.

Měření času probíhá pomocí časových razítek datového typu time/date¹⁶. Hodnota času se získá rozdílem časových razítek. Výsledkem je čas v sekundách.

První časové razítko se zaznamená za inicializační částí po stisknutí tlačítka Start. Další časové razítko se pravidelně zaznamenává na začátku aktivní části měřicí smyčky. Tato značka se posílá ukládací smyčce k dalšímu zpracování. Rozdílem těchto dvou značek dostaneme čas od začátku měření.

$$t_{\text{měření}} = td_{\text{aktuální}} - td_{\text{start}} \quad (46)$$

kde td vyjadřuje časovou značku typu time/date a t vyjadřuje čas v sekundách.

Jedním z požadavků bylo, aby se data ukládala od času 0 s. Při ukládání hlavičky dochází k inicializaci aktuální časové značky jako výchozí. Od této značky je měřen čas ukládání.

$$t_{\text{ukládání}} = td_{\text{aktuální}} - td_{\text{hlavička}} \quad (47)$$

V aplikaci tedy existují 2 různé časy. Je-li nastaveno opakování hlavičky, pak každé nové opakování začíná časem 0 s. Tento čas je uložen do souboru.

Jelikož pro zobrazení časového průběhu je používán XY graf, nelze na vstup XY grafu poslat $t_{\text{ukládání}}$, neboť by docházelo ke grafickému přepisu dat. Pro zobrazování se tedy používá $t_{\text{měření}}$. K tomuto času se také váže tvorba signálu podle definice.

¹⁶ Time/date je datový typ, který v sobě uchovává informace o datu a času. Je to počet sekund od 1. 1. 1904 12:00 am.

V aplikaci nelze změřit přesný čas přiřazený k hodnotám $u(t)$ a $y(t)$. Při měření času v pomalých soustavách jsou však chyby zanedbatelné.

5.4.13 Property Nodes a Invoke Nodes

Pomocí Property Nodes můžeme pracovat s vlastnostmi jednotlivých prvků. V aplikaci se využívají Property Nodes hlavně na určení viditelnosti, zamčení/odemčení editace ovládacích prvků, popř. skrytí jednotlivých toolbarů¹⁷ ve frontpanelu uživatele (je nežádoucí, aby měl uživatel k dispozici tlačítko abort). Objekty se předávají pomocí referencí¹⁸.

Pomocí Invoke Nodes můžeme vykonat funkce nad jednotlivými objekty.

5.4.14 Moduly (subVIs)

Modul se skládá ze vstupů, těla a výstupů. Vstupy lze dělit na povinné a nepovinné, u kterých je definována výchozí hodnota. Kromě předdefinovaných modulů obsahuje program také vlastní moduly. Modul by měl plnit určitou logickou funkci.

Každý modul má svoji ikonu, která by měla co nejlépe vystihovat logickou funkci daného modulu. Je možné si vybrat ze sady předdefinovaných ikon nebo si nakreslit vlastní. Pro přehlednost kódu je tedy důležité volit správné ikony modulů. Např. pro modul, který měří teplotu, je vhodné zvolit ikonu teploměru. Důležité je také správně pojmenovat vstupy a výstupy popř. přidat k modulu popis jeho činnosti.

Moduly lze vnořovat, čímž se vytváří hierarchická stromová struktura. Jednotlivé moduly lze v LabView jednoduše vytvořit tak, že označíme část kódu, která tvoří logický celek, a zvolíme v menu možnost *create subVI*.

5.4.15 Tvorba spustitelného souboru

Před tvorbou instalačního souboru je nutné vytvořit spustitelný soubor. Instalační soubor lze vytvořit pomocí Build Specification. Při jeho vytváření je potřeba nastavit několik důležitých parametrů.

- Information
 - Build specification name: „Identification v 2.4“

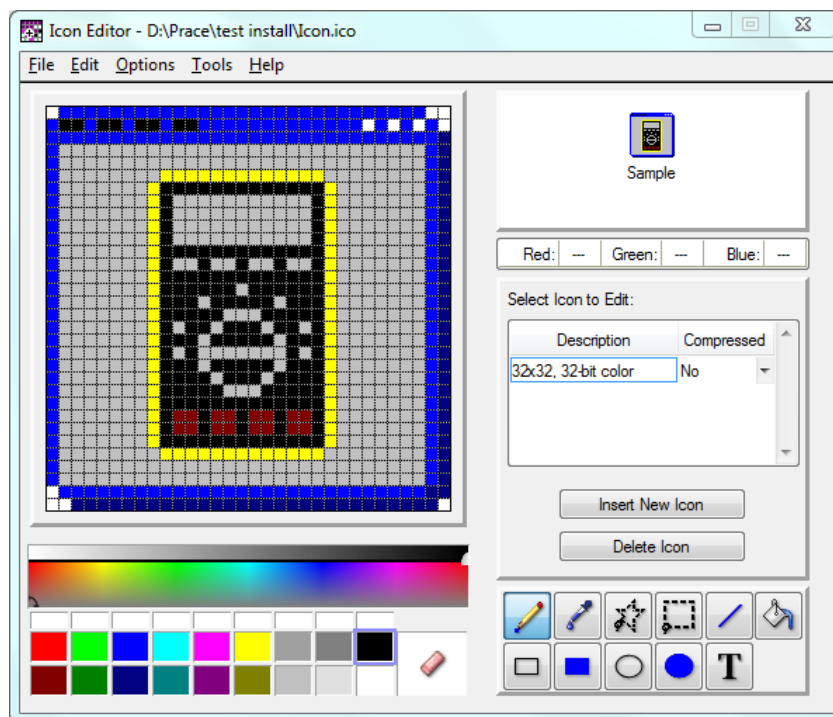
¹⁷ Toolbar je lišta s ikonkami, přičemž kliknutí na ikonku vyvolá událost, která je obsloužena příslušnou metodou. Obsahuje tedy sadu nástrojů pro práci s aplikací.

¹⁸ Reference je odkaz na příslušný objekt.

- Target filename: „Identification v2.4.exe“
- Destination directory: cílový adresář
- Source Files
 - Startup vIs: „identifikace v2.4.vi“
 - Always Included: ostatní VIs
- Destinations
 - Destination path: cílová cesta pro vytvoření spust. souboru
- Icon
 - Icon Image: 32 x 32, 32-bit color
- Version Information
 - Product name: „Identification v 2.4“
 - Company name: „UTB ve Zline“
 - Internal name: „Identification v 2.4“
 - Description: „Identification v 2.4“

Ve vlastnostech spustitelného souboru lze samozřejmě nastavit více parametrů, než je zde uvedeno.

Ikona aplikace byla vytvořena tak, aby co nejlépe vystihovala účel aplikace. Ikona byla vytvořena pomocí Icon Editoru.



Obrázek 33 – Tvorba ikony aplikace Identifikace v Icon Editoru

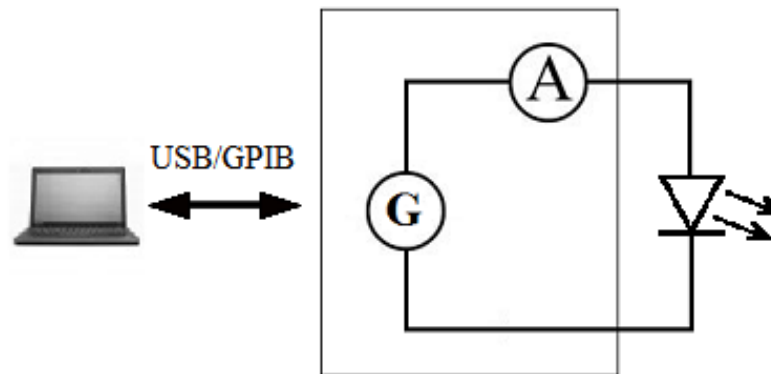
5.4.16 Tvorba instalačního souboru

Poté co je vytvořen spustitelný soubor, je možné vytvořit instalační soubor. Také instalační soubor lze vytvořit pomocí Build Specification. Při jeho tvorbě je nutné nastavit několik důležitých parametrů.

- Product Information
 - Build specification name: „Identification v2“
 - Product name: „Identification v2“
 - Installer destination: cílové umístění instalátoru
- Destinations
 - cílové umístění nainstalované aplikace, např. Program Files
- Source Files
 - zdrojové soubory jsou tři: Identification v2.4.exe, Identification v2.4.aliases a Identification v2.4.ini
- Shortcuts
 - Name: Identification v2.4
 - Target File: „ProgramFilesFolder\Identification v2\Identification v2.4.exe“
 - Directory: Program Menu
 - Subdirectory: Identification v2
- Additional Installers
 - National Instruments Installers to Include
 - NI LabVIEW Run_time Engine 2011 SP1
 - NI-488.2.3.0
 - NI-VISA Runtime 5.1.1
 - Install type: Full
- Version Information
 - Company name: UTB ve Zline
 - Company URL: <http://utb.cz>
- Advanced
 - Systém requirements: Windows XP or later

5.5 Měření VA charakteristik

Při měření VA charakteristik je akční veličinou napětí zdroje a výstupní veličinou stejnosměrný proud naměřený multimetrem. Pro měření VA charakteristik zapojíme oba přístroje sériově a zařadíme před (nebo za) měřenou LED (Obrázek 34).



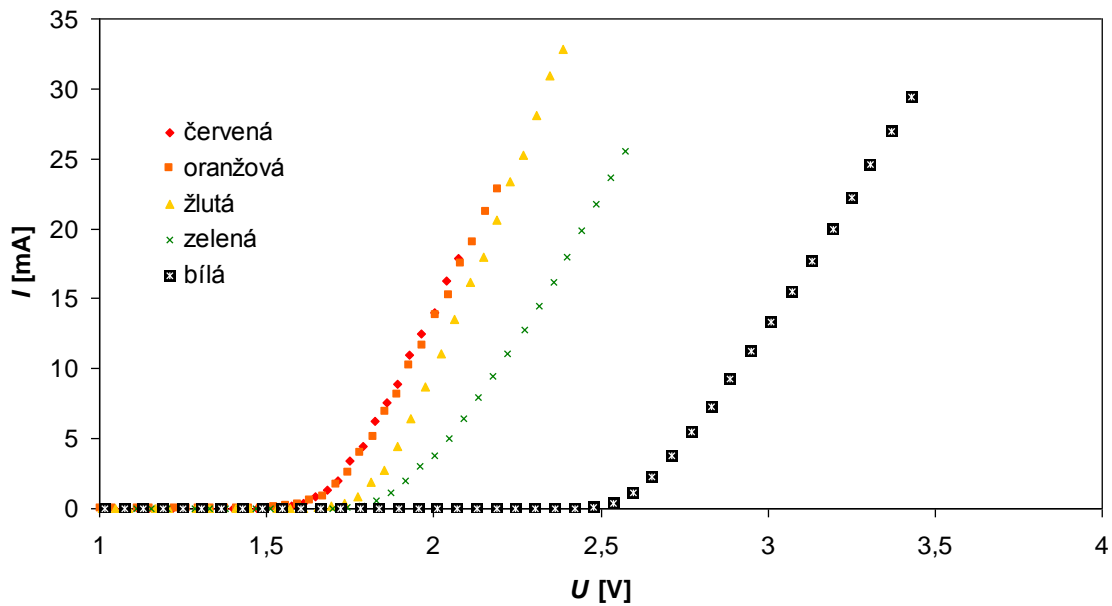
Obrázek 34 – Schéma zapojení pro měření VA charakteristik

Dále je potřeba nastavit na zdroji pilovitý signál. Pro měření klasické červené LED jsou parametry napěťového signálu následující.

- Time values: [0 30]
- Output values: [0 2,1]

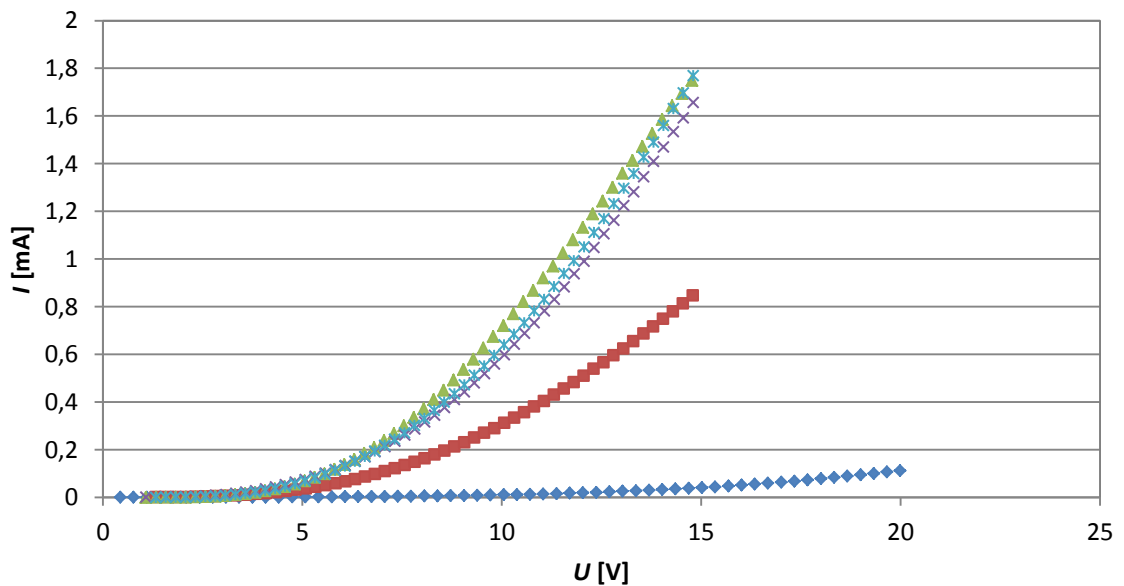
Za dobu 30 s se napětí zvýší lineárně z 0 V na 2,1 V. Přepneme-li na XY graf, uvidíme VA charakteristiku červené LED.

Na obrázku jsou VA charakteristiky několika komerčních LED.



Obrázek 35 – VA charakteristiky komerčních LED

Kromě komerčních LED byly změřeny také OLED vyrobené z polymerů. Napětí těchto OLED se pohybuje řádově v desítkách voltů, avšak proud se pohybuje již od řádu jednotek mA.



Obrázek 36 – VA charakteristiky OLED

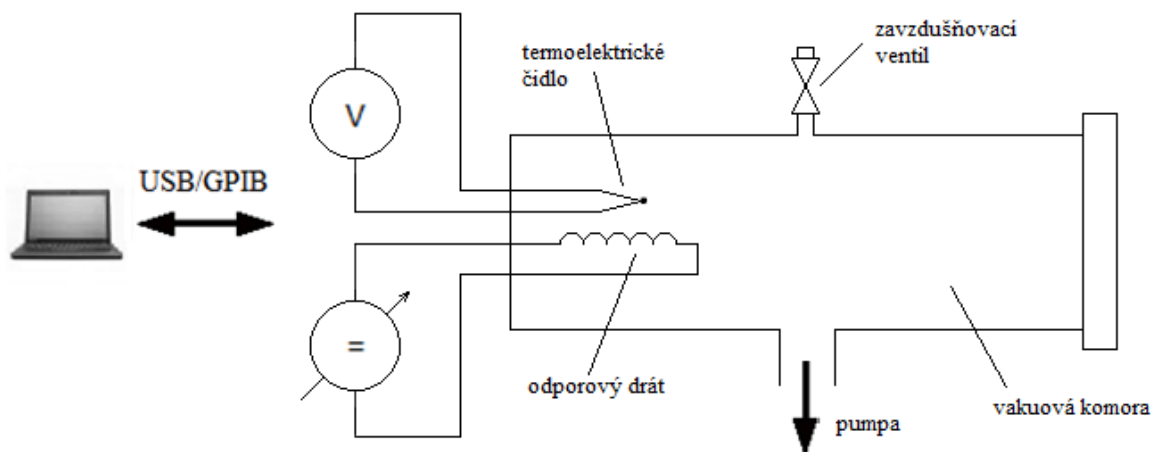
Jak je patrné z obrázku u vybraných OLED byl naměřen celkem velký rozptyl v proudové oblasti. Tvar jednotlivých VA charakteristik totiž závisí na mnoha faktorech, jako je například tloušťka jednotlivých vrstev, homogenita materiálu či velikost styčných ploch elektrod.

5.6 Identifikace soustavy

Pro identifikaci soustavy lze využít stejné aplikace, jako pro měření VA charakteristik. Obě měření jsou založena na stejném principu, tj. nastavení akční veličiny (napětí) a měření odezvy (proud, napětí, teplota, ...).

5.6.1 Popis tepelné soustavy (pece)

Aplikace Identifikace byla také využita pro identifikaci tepelné soustavy. Jedná se o pec, která se skládá z odporového drátu a termoelektrického článku. Kolem pece je umístěno teplotní zrcadlo, které zabraňuje přenosu tepla mimo soustavu. Celá pec je umístěna ve vakuové komoře, takže je možno dosahovat vyšších teplot aniž by hrozilo nebezpečí shoření odporového drátu.



Obrázek 37 – Schéma zapojení pece

Pec nemá aktivní chlazení, takže doba potřebná pro chlazení soustavy je různá od doby ohřívání soustavy.



Obrázek 38 – Tepelná soustava - pec

Odporový drát je napájen z řízeného napěťového zdroje, takže akční veličina $u(t)$ je výstupní napětí zdroje (respektive hodnota vstupující do modulu zdroje). Termoelektrické čidlo dává na výstupu napětí ve voltech, které je potřeba převést na požadovanou fyzikální veličinu – teplotu. Jelikož termoelektrický člen nemá referenční teplotu $0\text{ }^{\circ}\text{C}$, je nutné k výsledné hodnotě přičíst teplotu v místnosti (v našem případě $24\text{ }^{\circ}\text{C}$). K převodu byla využita tabulka pro NiCr-Ni termoelektrický člen.

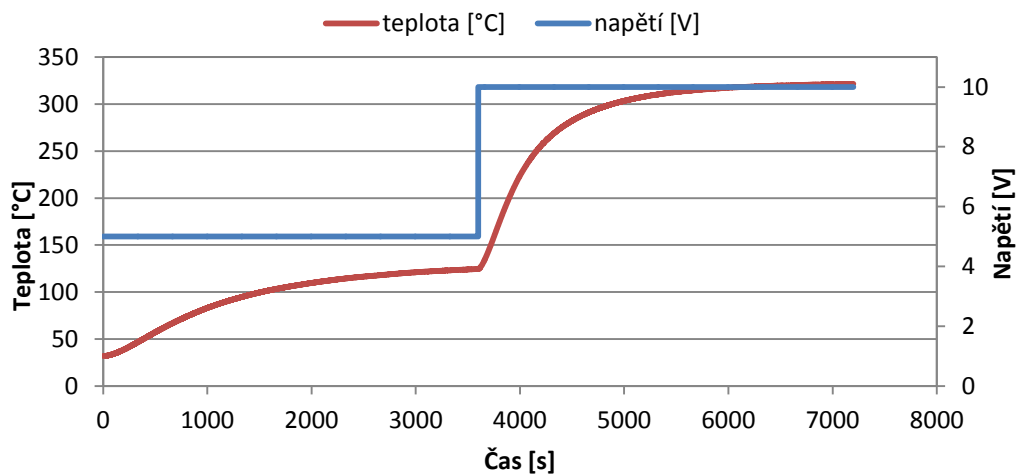
Tabulka 9 – Převodní tabulka termoelektrického čidla

$^{\circ}\text{C}$	0	10	20	30	40	50	60	70	80	90	100	mV/ $^{\circ}\text{C}$	Toll. in $^{\circ}\text{C}$
0	0,00	0,40	0,80	1,20	1,61	2,02	2,43	2,85	3,26	3,68	4,10	0,041	±3
100	4,10	4,51	4,92	5,33	5,73	6,13	6,53	6,93	7,33	7,73	8,13	0,040	
200	8,13	8,54	8,94	9,34	9,75	10,16	10,57	10,98	11,39	11,80	12,21	0,041	
300	12,21	12,63	13,04	13,46	13,88	14,29	14,71	15,13	15,55	15,98	16,40	0,042	±4,5
400	16,40	16,82	17,24	17,67	18,09	18,51	18,94	19,36	19,79	20,22	20,65	0,042	
500	20,65	21,07	21,50	21,92	22,35	22,78	23,20	23,63	24,06	24,49	24,91	0,043	
600	24,91	25,34	25,76	26,19	26,61	27,03	27,45	27,87	28,29	28,72	29,14	0,042	±6
700	29,14	29,56	29,97	30,39	30,81	31,23	31,65	32,06	32,48	32,89	33,30	0,042	
800	33,30	33,71	34,12	34,53	34,93	35,34	35,75	36,15	36,55	36,96	37,36	0,041	
900	37,36	37,76	38,16	38,56	38,95	39,35	39,75	40,14	40,53	40,92	41,31	0,040	±8
1000	41,31	41,70	42,09	42,48	42,87	43,25	43,63	44,02	44,40	44,78	45,16	0,039	
1100	45,16	45,54	45,92	46,29	46,67	47,04	47,41	47,78	48,15	48,52	48,89	0,037	
1200	48,89	49,25	49,62	49,98	50,34	50,69	51,05	51,41	51,76	52,11	52,46	0,036	

Při měření byl použit hraniční mód `hold_min`, a proto při hodnotě akční veličiny menší než 0 V byla hodnota výstupní veličiny $y(t)$ (teplota) shodná s nastavenou teplotou v místnosti. Při překročení horní hranice by však měření skončilo chybou. Velikostí rozsahu tabulky lze také ochránit akční člen tak, aby nepřekročil stanovenou hodnotu.

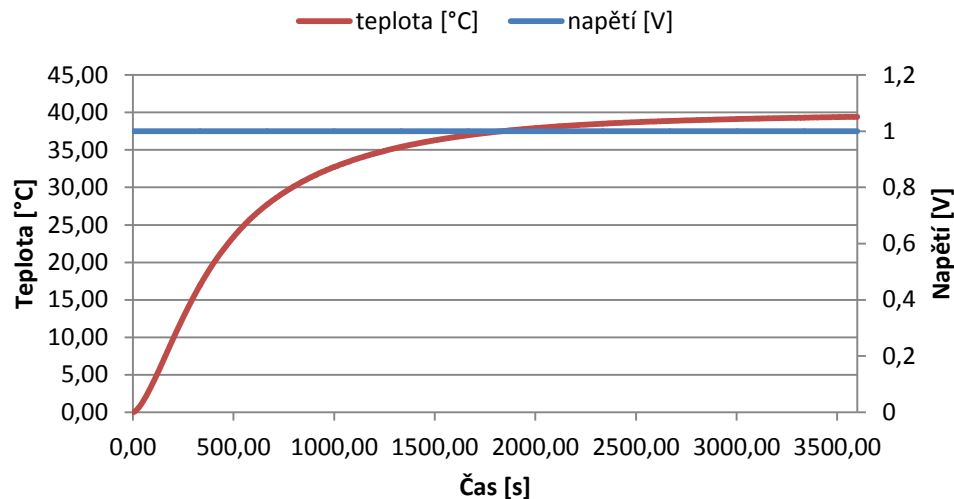
5.6.2 Měření přechodové charakteristiky soustavy

Při měření přechodové charakteristiky se měří reakce výstupní veličiny na napěťový skok. Aby soustava byla na počátku v ustáleném stavu, je akční veličina generována jako dvojice skoků. První skok uvede tepelnou soustavu do ustáleného stavu a pomocí druhého skoku se určí přenos soustavy (přechodová charakteristika).



Obrázek 39 – Měření přechodové charakteristiky soustavy – původní signál

Signál je samozřejmě nutné před výpočtem upravit posunutím závislé veličiny (teplota a napětí) i nezávislé veličiny (čas). Teprve z tohoto průběhu lze určit přenos soustavy některou z metod identifikace.



Obrázek 40 – Přejchodová charakteristika soustavy

5.6.3 Určení přenosu soustavy

Přenos soustavy lze určit pomocí některé z metod pro identifikaci. U měřené soustavy (pece) byly využity 2 metody pro identifikaci. Pomocí Strejcovy metody byl určen odhad parametrů přenosu (zesílení, časové konstanty) a následně byly tyto parametry zadány jako proměnné hodnoty do sešitu v aplikaci MS Excel. Pomocí doplňku řešitel byly parametry přenosu postupně nastavovány tak, aby suma čtverců odchylek vypočítané a naměřené přechodové charakteristiky byla minimální (iterační metoda, viz kapitolu 3.2). Pomocí této metody byly určeny časové konstanty a zesílení přenosu soustavy.

$$k = 39,04 \text{ } ^\circ\text{C V}^{-1}$$

$$T_1 = 519,29 \text{ s} \quad (48)$$

$$T_2 = 40,80 \text{ s}$$

Přenos je pak ve tvaru

$$G(s) = \frac{k}{(T_1s + 1)(T_2s + 1)} = \frac{39,04}{(519,29s + 1)(40,80s + 1)} \quad (49)$$

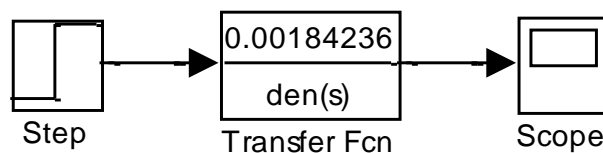
Koeficient zesílení byl dále upraven podle směrnice aproximované přímky na statickou charakteristiku (viz kapitola 5.7.2) a přenos byl převeden do tvaru vhodného pro určení PID parametrů Naslinovou metodou.

$$G(s) = \frac{35,88}{(519,29s + 1)(40,80s + 1)} \quad (50)$$

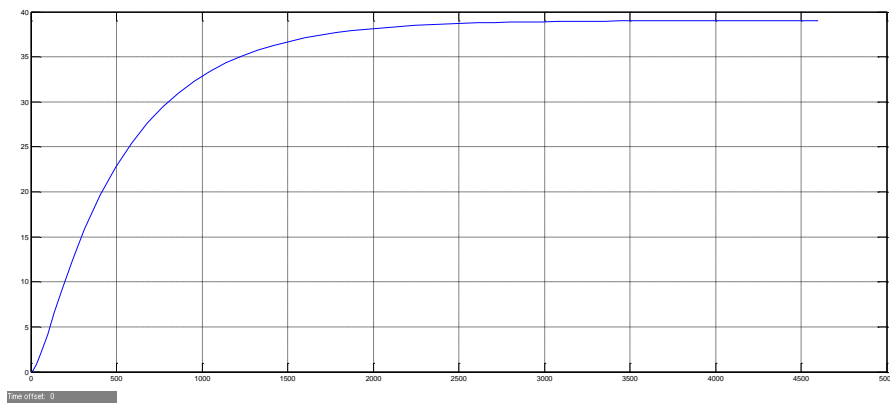
$$G(s) = \frac{0,0016933}{s^2 + 0.026433s + 0.000047}$$

5.6.4 Simulace přechodové funkce

Pro ověření přechodové funkce byl vytvořen model v prostředí Simulink, kde na vstupu přechodové funkce byl umístěn jednotkový skok a výstup byl zobrazen do grafu.



Obrázek 41 – Simulace přechodové charakteristiky
- model



Obrázek 42 – Simulace přechodové charakteristiky – graf

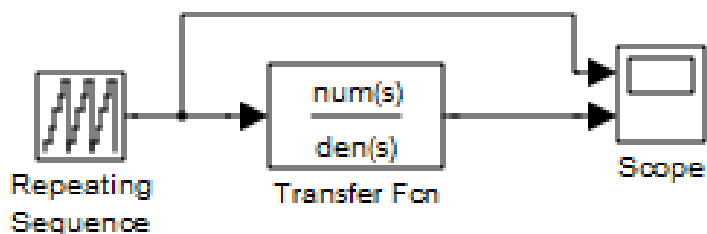
Z grafu je patrná celkem dobrá shoda mezi modelem a reálnou soustavou.

5.7 Měření statické charakteristiky soustavy

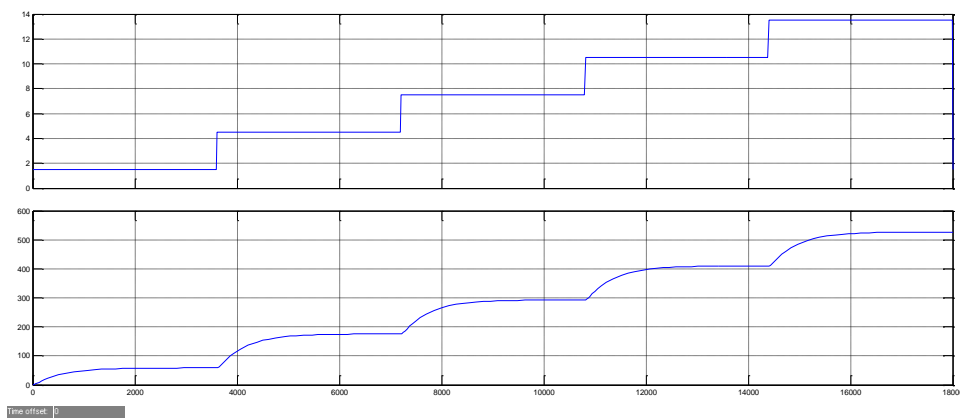
Pro řízení s pevně nastavenými parametry je nutné, aby statická charakteristika byla lineární. Statické zesílení pro různou oblast teplot pak bude stejné a parametry přenosu budou konstantní.

5.7.1 Simulace měření statické charakteristiky

Při simulaci statické charakteristiky bylo použito pěti skoků o velikosti 3 V. Simulační čas každého skoku byl nastaven na 1 hodinu. Jelikož počáteční podmínky reálné soustavy nejsou nulové, byla od všech skoků odečtena hodnota 1,5 V.



Obrázek 43 – Simulace měření statické charakteristiky – model



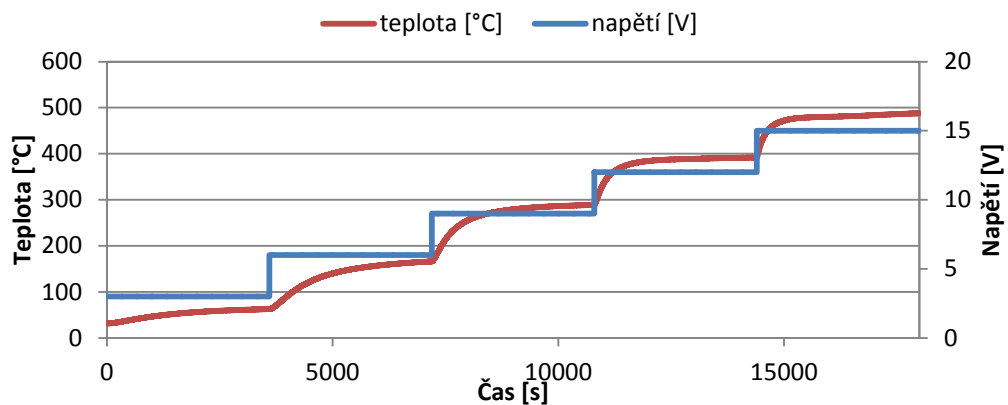
Obrázek 44 – Simulace měření statické charakteristiky – graf

Podle simulace by konečná teplota měla být menší než 600 °C.

5.7.2 Reálné měření statické charakteristiky

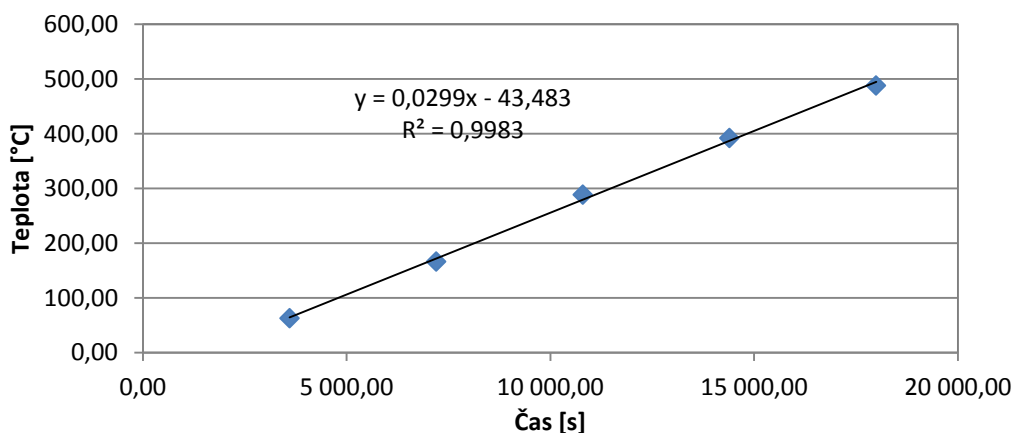
Pokud by model odpovídal skutečnosti, měla by být statická charakteristika naměřená na reálné soustavě přibližně lineární.

Vstupní veličina byla volena jako 5 skoků o velikosti 3 V.



Obrázek 45 – Měření statické charakteristiky – reálná soustava

Z grafu je patrné, že naměřená statická charakteristika zhruba odpovídá simulované statické charakteristice. Výrazně se liší doba náběhu a průtahu v jednotlivých skocích, což může být způsobeno nelinearitami systému. Z naměřené charakteristiky pak zobrazíme pouze ustálené body na koncích jednotlivých intervalů a tyto body aproximujeme přímkou.



Obrázek 46 – Statická charakteristika soustavy a lineární aproximace

Statická charakteristika soustavy je téměř lineární, neboť koeficient spolehlivosti je na hodnotě 0,9983 (99,83 %). Zesílení je tedy v různých teplotních intervalech totožné. Pomocí směrnice přímky byl také upraven koeficient zesílení přenosu soustavy. Pro výpočet je nutné znát směrnici přímky, rozdíl akční veličiny a dobu měření.

$$a = 0,0299 \text{ } ^\circ\text{C s}^{-1}$$

$$\Delta t = 5 \cdot 3600 = 18000 \text{ s} \quad (51)$$

$$\Delta u = 5 \cdot 3 = 15 \text{ V}$$

Směrnice je pak definována jako

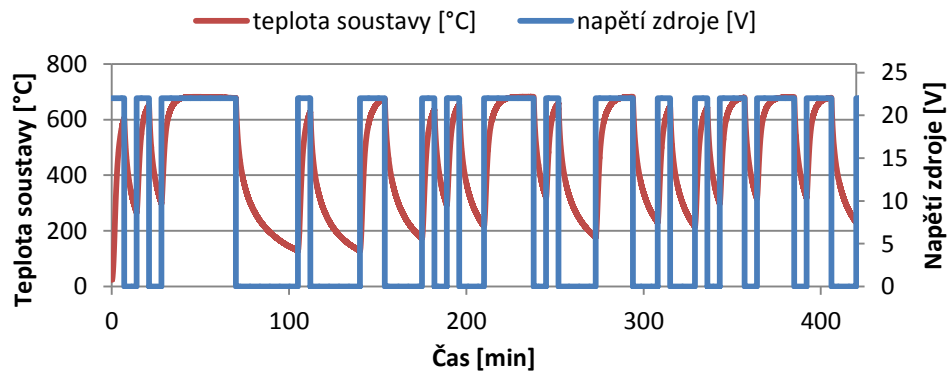
$$a = \frac{\Delta y}{\Delta t} \Rightarrow \Delta y = a \Delta t \quad (52)$$

Hodnotu zesílení lze pak určit z hodnoty směrnice následovně

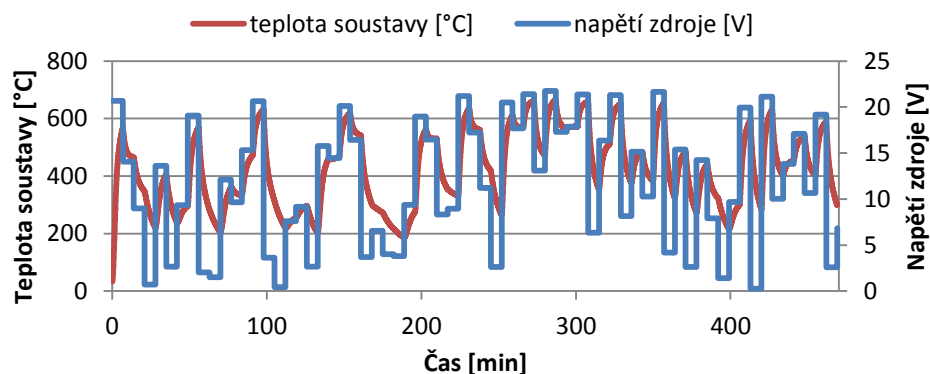
$$k = \frac{\Delta y}{\Delta u} = \frac{a \Delta t}{\Delta u} = \frac{0,0299 \cdot 18000}{15} = 35,88 \text{ } ^\circ\text{C V}^{-1} \quad (53)$$

5.7.3 Identifikace soustavy pomocí generátoru pseudonáhodného a PNBS signálu

Pro identifikaci soustavy je možné využít také generátor pseudonáhodného nebo PNBS signálu. Tyto signály jsou vhodné pro měření systémů, u nichž je nevhodné generovat skokový signál. Přenos soustavy pak můžeme určit například metodou nejmenších čtverců (aproximace určitým modelem). Z tohoto modelu lze určit impulsní i přechodovou charakteristiku (dosazením jednotkového skoku do diferenční rovnice). Tyto charakteristiky lze získat i korelační analýzou dat.



Obrázek 47 – Buzení soustavy generátorem PNBS



Obrázek 48 – Buzení soustavy pseudonáhodným signálem
s rovnoměrným rozdělením

je vygenerován akční zásah. Tato hodnota je nastavena na výstup zdroje a celý okruh se uzavírá přes regulovanou soustavu.

Hodnoty $w(t)$, $u(t)$, $y(t)$ a časové razítko jsou pak pomocí fronty poslány ukládací smyčce (na obrázku ta spodní) k dalšímu zpracování. Ta ukládá data podle zvoleného módu, kdy pomocí stavového automatu přechází mezi jednotlivými stavy (kapitola 5.4.11).

Časový průběh měřených a nastavovaných veličin je zobrazen v grafu.

Jak je vidět z blokového schématu aplikace, jedná se vlastně o rozšíření a úpravu aplikace Identifikace. Níže tedy budou popsány pouze bloky, které se od této aplikace liší.

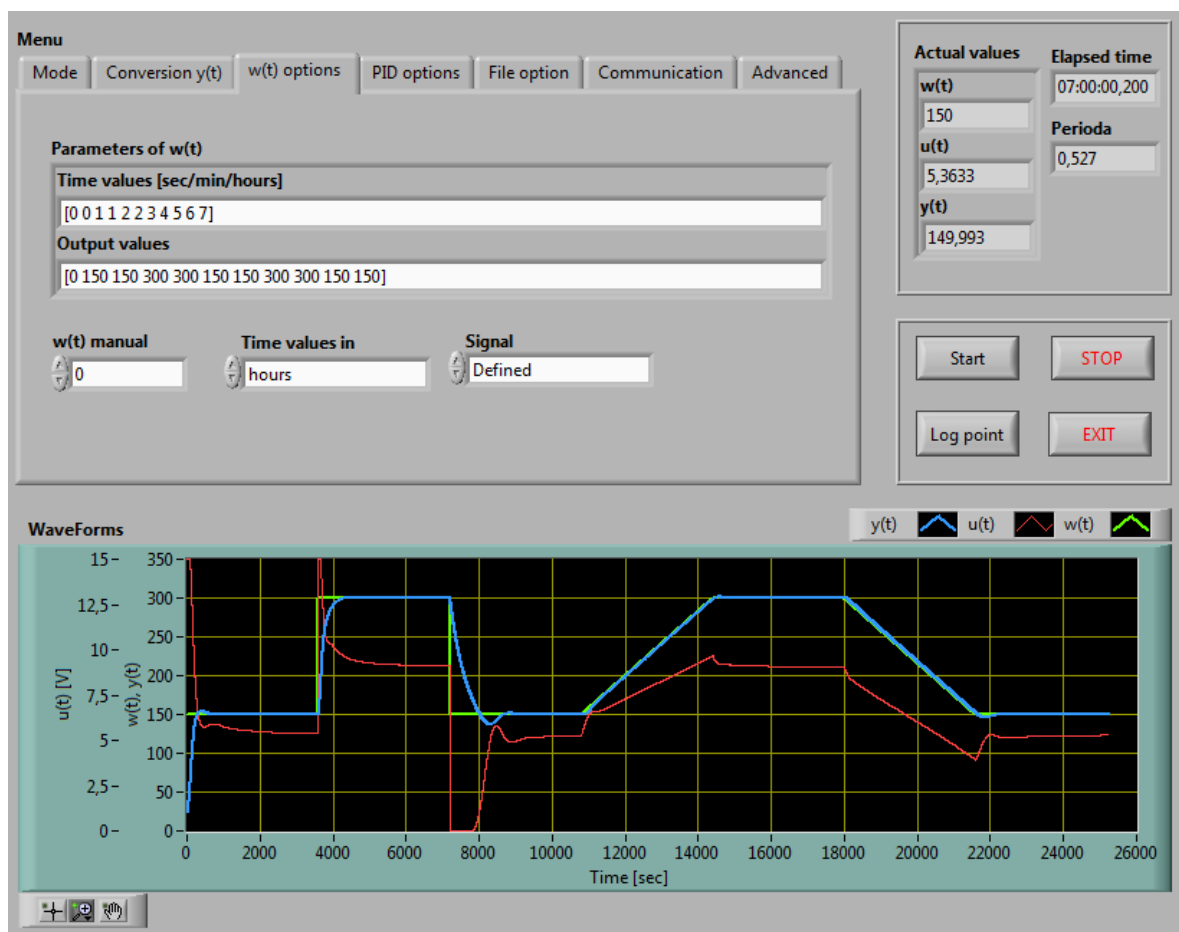
6.3 Chování programu

Po spuštění programu je aplikace v nastavovacím módu, kdy je možné nastavit hodnoty prvků Front panelu (např. umístění souboru k ukládání dat).

Po stisknutí tlačítka **Start** se spustí samotná regulace (není-li zvoleno otevření OpenFileDialogu) a ukládání dat. Aplikace pracuje vždy nad určitým souborem, i když nejsou data ukládána. Při pokusu o zápis do existujícího souboru je vygenerován dialog, kde je možné soubor přepsat, nebo ukončit měření stisknutím tlačítka **Cancel**. Měření je také ukončeno při vzniku jakékoliv chyby nebo při stisknutí tlačítka **Stop**. Po ukončení regulace je aplikace opět v nastavovacím módu a je možné spustit novou regulaci. Pro ukončení aplikace slouží tlačítko **Exit**.

6.4 Front panel

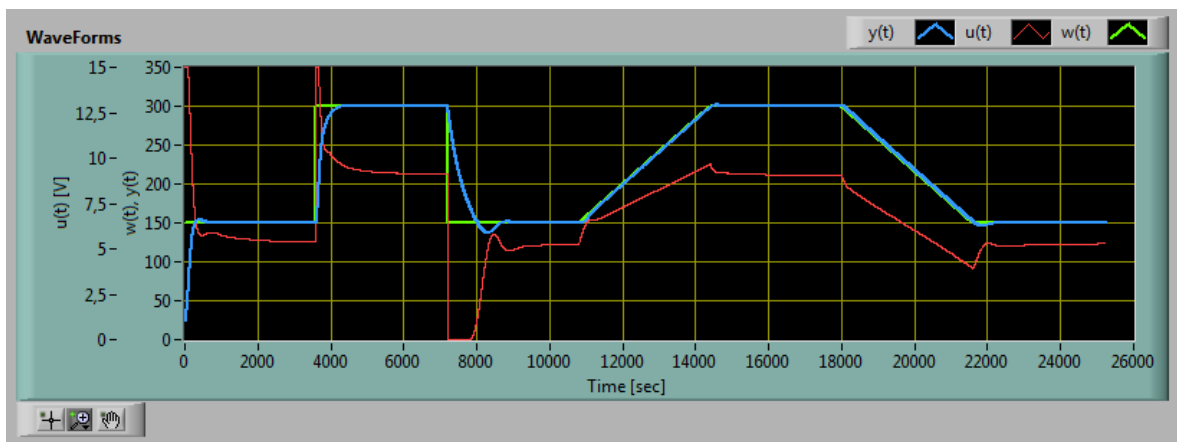
Front panel (Obrázek 50) slouží pro ovládání aplikace a vizualizaci dat v reálném čase. Skládá se z několika bloků. Blok **Menu** slouží pro ovládání aplikace a nastavení parametrů, blok **Actual values** ukazuje aktuální výstupy aplikace a blok **Buttons** obsahuje tlačítka. Ve spodní části je k dispozici graf, který zobrazuje časové průběhy veličin.



Obrázek 50 – Front Panel aplikace Regulace

Graf

V aplikaci máme k dispozici graf (Obrázek 51), který zobrazuje časové průběhy veličin. Graf obsahuje 2 vertikální osy, kde na jedné ose jsou veličiny $w(t)$ a $y(t)$ (předpokládá se, že jsou ve stejných jednotkách) a na druhé ose je akční veličina $u(t)$. Osy mají nastavenou automatickou změnu rozsahu a průběhy je navíc možné přibližovat pomocí zoomu. Aby bylo možné zobrazit časový průběh, je nutné uchovat data v mezipaměti a poté je zobrazit do XY grafu. Velikost mezipaměti je nastavena tak, aby obsahovala celý jeden průběh žádané hodnoty $w(t)$ (podle definice). Při opakování signálu jsou pak starší data „vytlačována“, aby se zbytečně neplnila paměť.



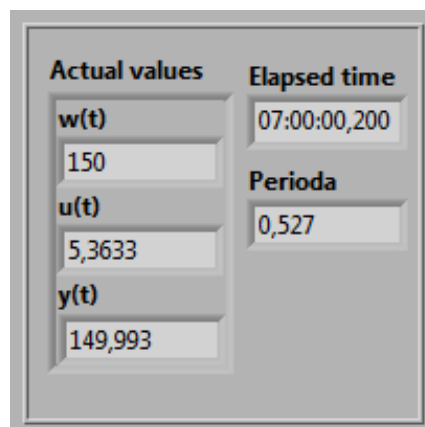
Obrázek 51 – Graf časového průběhu veličin v aplikaci Regulace

Buttons

Tlačítka a jejich funkce se shodují s funkcemi v aplikaci Identifikace.

Outputs

Blok Outputs (Obrázek 52) slouží k zobrazení aktuálních výstupů aplikace.



Obrázek 52 – Výstupy aplikace
Regulace

Význam jednotlivých výstupů je následující.

w(t) – žádaná hodnota.

u(t) – akční veličina.

y(t) – výstupní veličina.

Time – čas od začátku měření ve formátu HH:mm:ss,SSS.

Perioda – aktuální doba mezi dvěma měřenými body.

Menu

Slouží k nastavení parametrů aplikace. Skládá se z několika dalších subbloků.

Mode

Tento blok a význam jednotlivých prvků se shoduje s aplikací Identifikace. Rozdíl je pouze v tom, že u aplikace Identifikace lze ovládat či generovat akční veličinu, kdežto u aplikace Regulace lze ovládat či generovat žádanou veličinu.

Pozn.: Také generátor $w(t)$ signálu obsahuje generátor pseudonádného a PNBS signálu, které jsou vnitřně generovány neustále.

Conversion $y(t)$

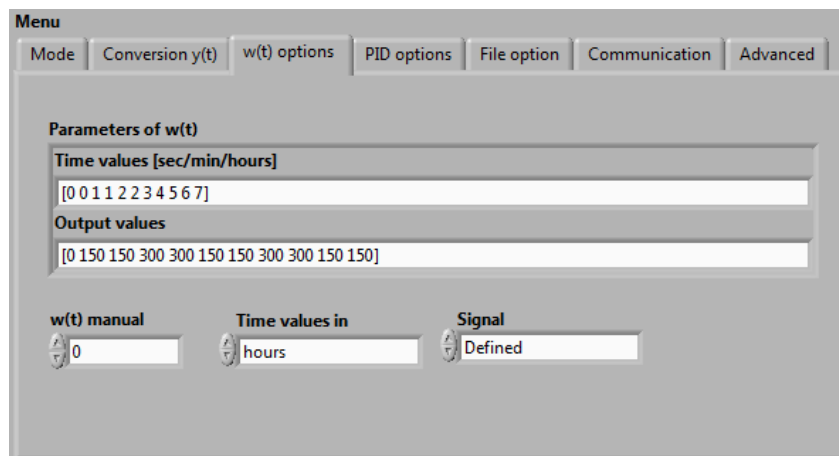
Toto submenu je shodné s aplikací Identifikace.

Příklad:

Budeme-li chtít regulovat soustavu, kde na vstupu multimetru je čidlo, které měří teplotu, přičemž toto čidlo dává na výstupu napětí úměrné této teplotě, můžeme toto napětí převést a regulovat/zobrazovat výstupní veličinu $y(t)$ ve stupních Celsia. Žádaná hodnota pak ale také musí být zadána ve stupních Celsia. K převodu výstupní veličiny můžeme využít buď aproximační polynom nebo převodní tabulku.

$w(t)$ options

Slouží k nastavení parametrů signálu žádané hodnoty $w(t)$.



Obrázek 53 – Nastavení parametrů žádané hodnoty $w(t)$

Význam jednotlivých prvků je následující.

Parameters of $w(t)$ – určuje tvar generovaného signálu.

Time values [sec/min/hours] – vektor času signálu v zadaných jednotkách. Vstup se zadává ve tvaru $[t_0 t_1 \dots t_n]$ (závorky nejsou povinné).

Output values – vektor hodnot signálu v zadaných časových bodech. Hodnoty se zadávají ve tvaru $[u_0 u_1 \dots u_n]$.

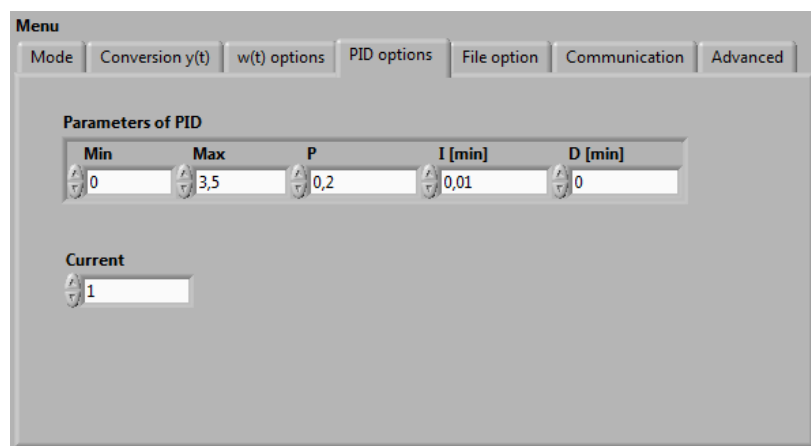
$w(t)$ manual – slouží k ručnímu zadávání žádané hodnoty $w(t)$ (musí být povoleno v záložce **Mode**).

Time values in – slouží k určení časových jednotek pro definici signálu.

Signal – slouží k určení zdroje signálu žádané hodnoty (definovaný, náhodný, PNBS).

PID options

Slouží k nastavení parametrů PID regulátoru.



Obrázek 54 – Nastavení parametrů PID regulátoru

Parameters of PID – nastavení parametrů PID regulátoru

Min/Max – minimální/maximální hodnota akční veličiny.

P – proporcionální složka regulátoru.

I – integrační konstanta v minutách.

D – derivační konstanta v minutách.

Current – proudové omezení zdroje.

Příklad:

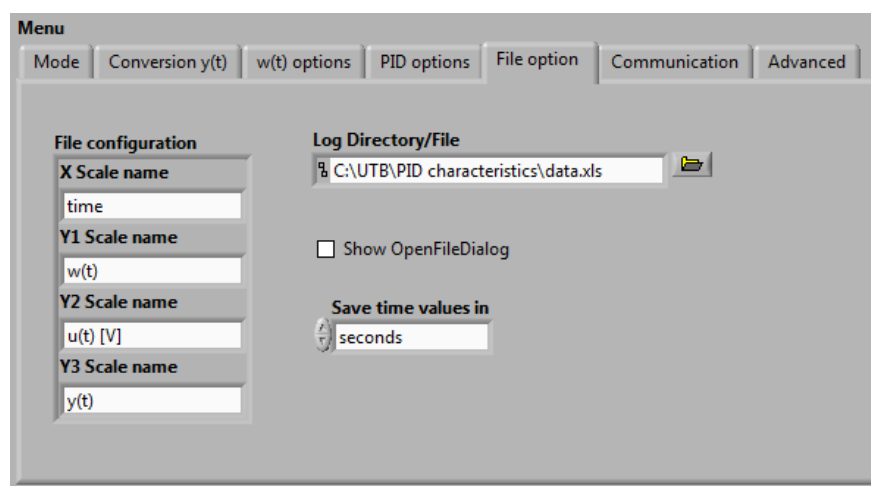
Parameters of PID nám umožňuje nastavit minimální/maximální hodnotu na výstupu zdroje (akční veličiny) a tím ochránit výkonový člen. Při ukončení regulace je navíc napětí

na zdroji automaticky shozeno na 0 V. Parametry P, I, D slouží pro nastavení přenosu regulátoru.

Pozn.: PID parametry nemusí být nastaveny pouze konstantou. Parametry je také možné zadat polynomem či tabulkou (viz blok Advanced), což umožňuje řídit také časově invariantní nelineární soustavy.

File option

Slouží k nastavení cesty k souboru a k nastavení hlavičky souboru.



Obrázek 55 – Nastavení parametrů hlavičky a souboru

Význam jednotlivých prvků je následující.

File configuration – určuje, jaký tvar bude mít hlavička souboru. Řetězce, které jsou zadány, budou uloženy do hlavičky souboru jako názvy jednotlivých měřených veličin (viz kapitola Soubor).

Ostatní prvky jsou shodné s prvky v aplikaci Identifikace.

Příklad:

Budeme-li chtít regulovat tepelnou soustavu pomocí termoelektrického článku a použijeme-li transformaci na °C, vyplníme do *File configuration* následující údaje.

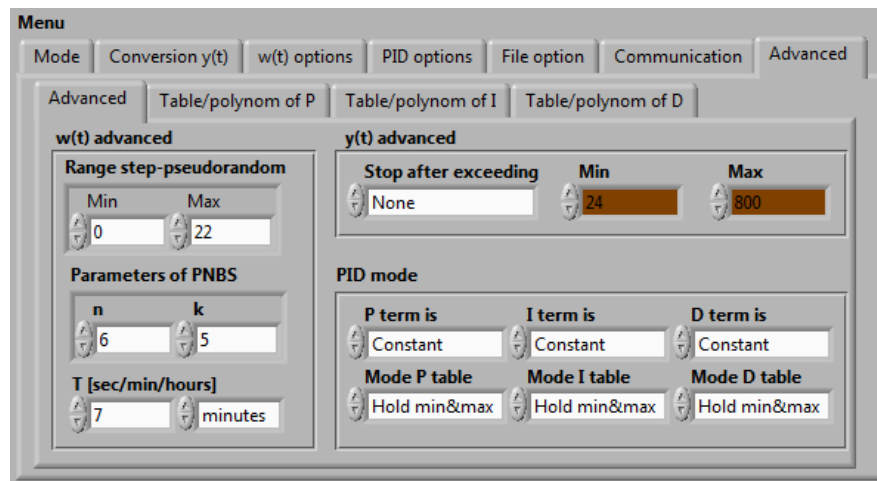
X Scale name:	Čas [s]
Y1 Scale name:	Žádaná teplota [°C]
Y2 Scale name:	Napětí zdroje [V]
Y3 Scale name:	Teplota soustavy [°C]

Communication

Tento blok je totožný s blokem v aplikaci Identifikace.

Advanced

Tento blok slouží k pokročilejšímu nastavení aplikace.



Obrázek 56 – Pokročilé nastavení aplikace regulace

Význam jednotlivých prvků je následující.

w(t) advanced – pokročilé nastavení žádané hodnoty (shodné s aplikací Identifikace)

y(t) advanced – pokročilé nastavení výstupní veličiny (shodné s aplikací Identifikace)

PID mode – pokročilé nastavení PID parametrů

P/I/D term is – způsob zadání PID parametrů

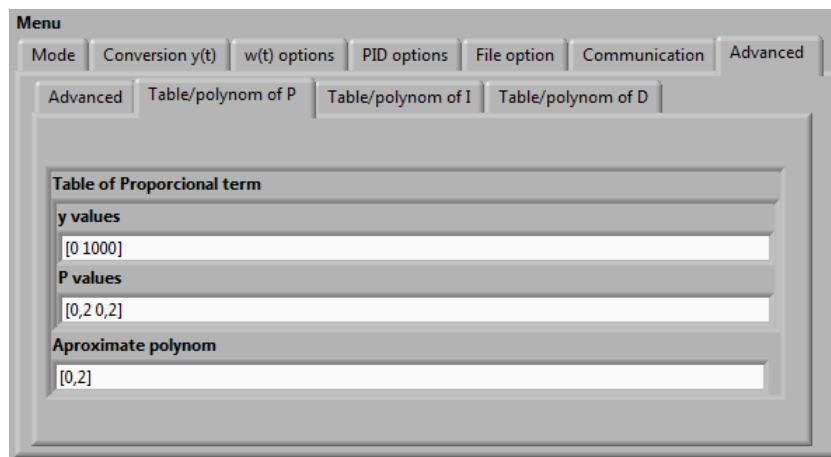
Constant – zadání P/I/D parametru konstantou

Function – zadání P/I/D parametru polynomem (v závislosti na převedené výstupní veličině)

Table – zadání P/I/D parametru tabulkou

Mode P/I/D table – chování aplikace při výstupní veličině mimo P/I/D table interval (shodné s Mode table val boundary)

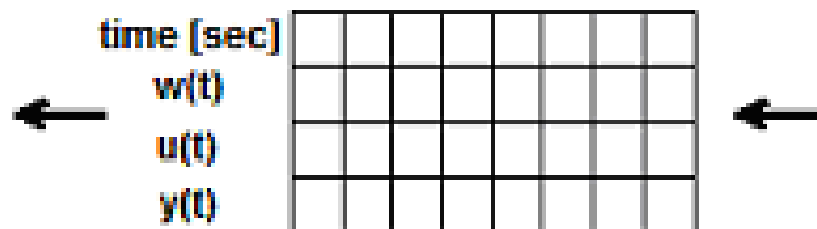
Tabulky a polynomy je možné zadat na dalších záložkách vnitřního Tab control.



Obrázek 57 – Zadání P parametru tabulkou či polynomem

6.4.1 Zobrazení

Veličiny $time$, $w(t)$, $u(t)$ a $y(t)$ se zobrazují v XY grafu (viz Front panel). Tyto hodnoty se vkládají do vyrovnávací paměti (Obrázek 58), která udržuje tolik hodnot, kolik odpovídá jednomu modelovanému průběhu $w(t)$ (podle definice). Paměť je typu FIFO, takže po dosažení jednoho průběhu žádané hodnoty budou následující hodnoty „vytlačovat“ ty původní z fronty.



Obrázek 58 – Vyrovnávací paměť pro hodnoty zobrazované v XY grafu

6.4.2 Fronta

Také u aplikace regulace je fronta typu cluster polí. Kromě naměřených veličin t , $w(t)$, $u(t)$, $y(t)$ (převedená) se předávají také PID parametry regulátoru nastaveny na počátku měření.

6.4.3 Soubor

Ukládání dat do souboru funguje úplně stejně, jako v případě aplikace Identifikace. Hlavička souboru však obsahuje kromě Integration time také parametry PID regulátoru nastavené na začátku regulace. Soubor je pak ukládán v tomto tvaru

```

*****
Integration Time [NPLC]
10
Min          Max          P          I          D
0           26          0,144077  3,04298   0,000116
-----
čas [s]      žádaná teplota [°C] napětí zdroje [V]   teplota
soustavy [°C]
-0,001      150          18,209084  24
0,499029   150          18,258799  24
1,001057   150          18,308712  24
1,502086   150          18,358527  24
2,001114   150          18,408142  24
...

```

6.4.4 Generování žádané veličiny $w(t)$

Pro generování signálu podle definice se používá stejné subVI, jako pro převod výstupní veličiny $y(t)$ pomocí tabulky. SubVI je přepnuto do módu *Mod* což umožňuje periodické opakování signálu. Aktuální hodnota $w(t)$ je spočítána za běhu jako funkce času t (viz $u(t)$ options).

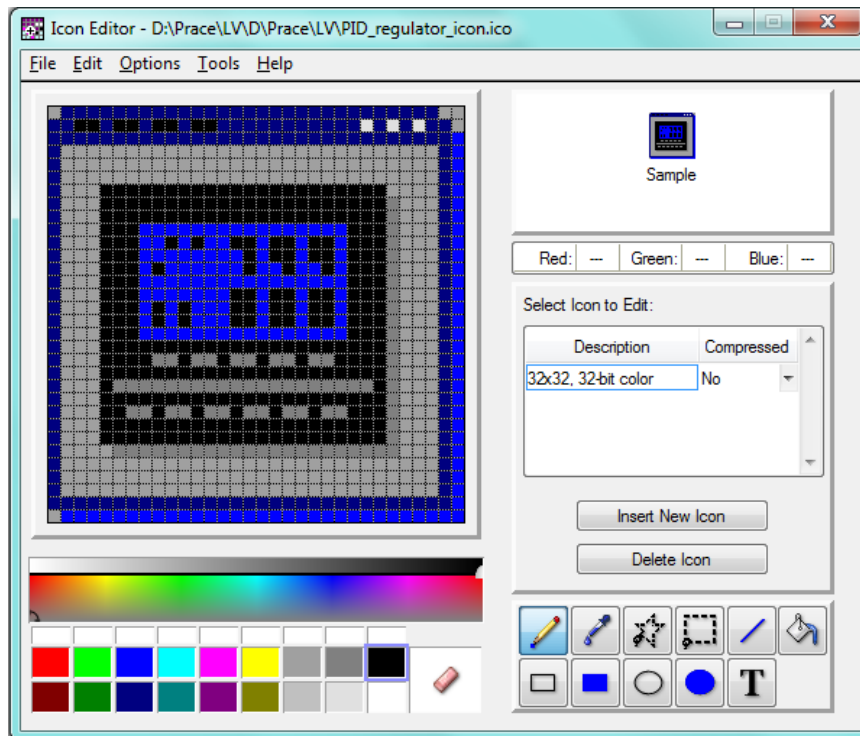
6.5 Vývoj aplikace Regulace

Kód aplikace regulace má velmi podobnou stavbu jako aplikace Identifikace. Obsahuje však navíc PID regulátor z toolboxu „PID and Fuzzy Logic Toolbox“. Tato aplikace byla ve skutečnosti vyvinuta před vývojem aplikace Identifikace, neboť původně byla požadována pouze regulace tepelné soustavy. Na základě potřeby identifikovat soustavu a provádět další měření elektrických veličin byla tato aplikace upravena do podoby aplikace Identifikace. Obě aplikace pak prošly několika verzemi, přičemž vylepšení jedné z nich vedlo k implementaci této funkčnosti i do druhé aplikace.

6.5.1 Tvorba spustitelného a instalačního souboru

Také u této aplikace je nutné vytvořit nejdříve spustitelný soubor a až poté instalační soubor. Parametry spustitelného a instalačního jsou nastaveny podobně jako u aplikace Identifikace. Místo klíčového slova Identification bylo použito slovo PID_regulator.

Ikona aplikace byla vytvořena pomocí Icon Editoru.



Obrázek 59 – Tvorba ikony aplikace PID regulátor v Icon Editoru

6.6 Regulace tepelné soustavy

Aplikace regulace byla použita pro řízení systému tepelné soustavy – pece. Žádaná a výstupní veličina je teplota ve stupních Celsia a akční zásah je napětí zdroje ve voltech.

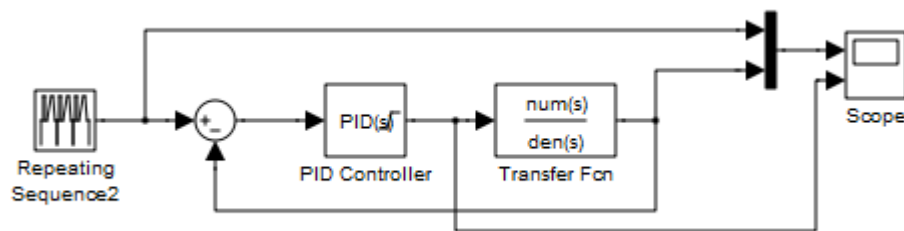
6.6.1 Výpočet regulátoru PID

Z určeného přenosu tepelné soustavy byly vypočítány složky PID regulátoru pomocí Naslinovy metody (kapitola 4.2). Požadované maximální přeregulování bylo zvoleno jako $\sigma = 1\%$, čemuž odpovídá koeficient $\alpha = 2,4$. Odhad přeregulování je pouze přibližný. Při výpočtu Naslinovou metodou získáme dvě rovnice o třech neznámých, takže je nutné jednu složku regulátoru zvolit a ostatní dopočítat. V našem případě byla zvolena derivační složka na hodnotu $r_1 = 0,001$. Dosazením do rovnic jsme získali zbylé 2 složky.

$$r_0 = 0,144077, r_{-1} = 0,000789, r_1 = 0,001 \quad (54)$$

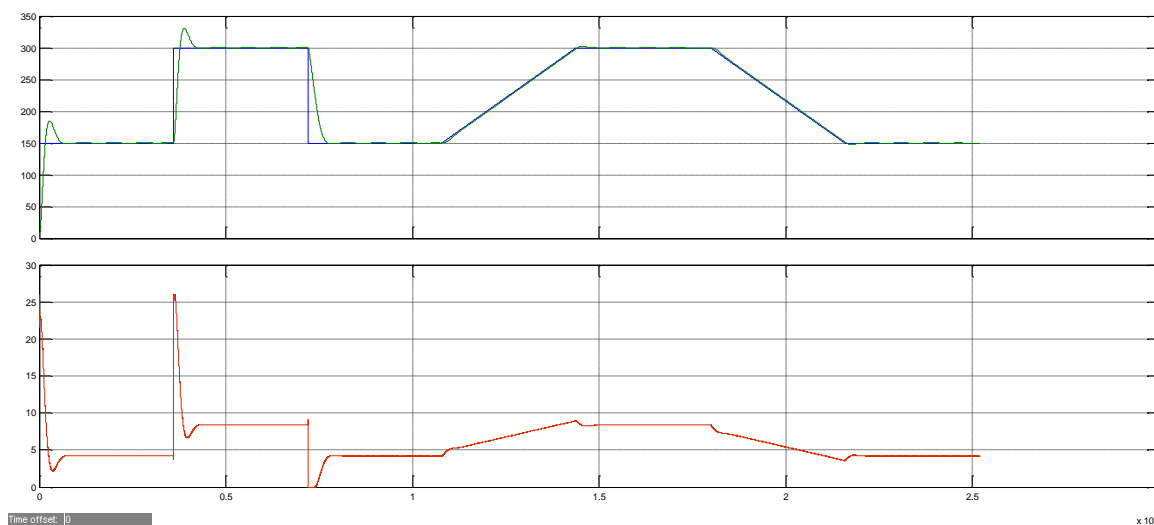
6.6.2 Simulace regulace

Před samotnou regulací bylo nutné provést simulace s modelem. Omezí se tím situace, že by se reálná soustava dostala do nestability nebo vykazovala jiné neobvyklé chování. Tvar žádané hodnoty byl zvolen tak, aby průběh signálu obsahoval skoky i lineární rampy. Signál obsahuje celkem 7 hodinových intervalů, takže celková simulační doba je 7 hodin. Pro model regulátoru byly použity vypočtené (přenosové) složky PID regulátoru. Jelikož reálná soustava má napěťové omezení (stejně tak i zdroj), byl akční zásah omezen v povoleném rozsahu. Tepelná soustava může pracovat až do teploty 800 °C. Pro testování maxima akčního zásahu byla využita aplikace Identifikace a bylo zjištěno, že teplotě 800 °C odpovídá akční zásah v intervalu 26 - 27 V. Rozsah akční veličiny regulátoru byl proto nastaven na interval $\langle 0, 26 \rangle$ V. Aby nedocházelo k windup efektu, byla zvolena anti-windup metoda back-calculation.



Obrázek 60 – Simulace regulace – model

Žádaná hodnota, akční i výstupní veličina pak byly zobrazeny v grafu.



Obrázek 61 – Simulace regulace - graf

Z grafu je patrné, že regulátor při skokové změně žádané veličiny generuje maximální či minimální akční zásah. Windup efekt se neprojevuje. Přeregulování je však dosti velké, avšak podregulování nepatrné. V oblasti lineární rampy výstupní veličina drží stejný trend jako žádaná veličina, avšak nedosahuje nulové regulační odchylky. Pro účely aplikace to ovšem není podstatné.

6.6.3 Regulace – reálný experiment

Pro regulaci tepelné pece byla použita aplikace Regulace. Rozdílový člen i omezení výstupu má již PID regulátor v sobě obsažen. Windup efekt je tedy ošetřen.

Jednotlivé PID složky jsou požadovány jako koeficient zesílení a časové konstanty. Proto je nutné přepočítat výsledky získané Naslinovou metodou.

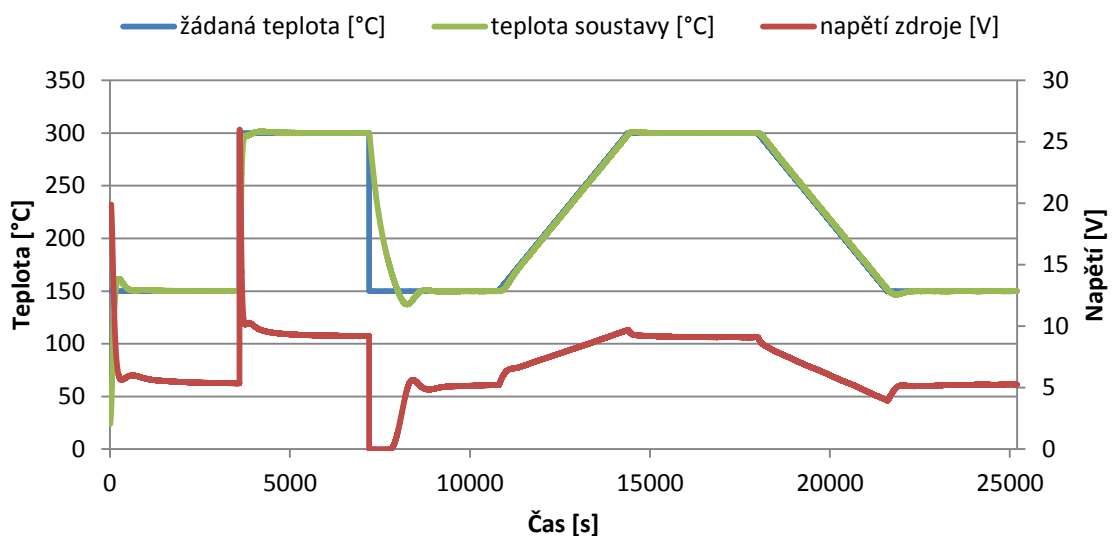
$$R(s) = r_0 + \frac{r_{-1}}{s} + r_1 s = K_R \left(1 + \frac{1}{T_I s} + T_D s \right)$$

$$\begin{aligned} K_R &= r_0 = 0,144077 \\ \Rightarrow T_I &= \frac{K_R}{r_{-1}} = 182,579 \\ T_D &= \frac{r_1}{K_R} = 0,00694071 \end{aligned} \quad (55)$$

Časové konstanty však musí být zadány v minutách, proto je nutné ještě derivační i integrační složku podělit 60.

$$\begin{aligned} K_{R_{LV}} &= K_R = 0,144077 \\ T_{I_{LV}} &= \frac{T_I}{60} = 3,04298 \\ T_{D_{LV}} &= \frac{T_D}{60} = 0,000115679 \end{aligned} \quad (56)$$

Po nastavení všech potřebných parametrů byl spuštěn experiment a po dosažení výsledků byly všechny veličiny zobrazeny v grafu.



Obrázek 62 – Regulace soustavy – reálný experiment

Z grafu je patrné, že při prvním skoku nebyla akční veličina maximální. Přeregulování je ovšem menší než při simulaci. Naopak podregulování je celkem velké, neboť soustava nemá aktivní chlazení a tedy dynamika soustavy pro ohřev a chlazení je rozdílná. V oblasti lineárních ramp se teplota soustavy chová podle očekávání a kopíruje průběh žádané hodnoty. Není však dosaženo nulové regulační odchylky $e(t)$, což však pro účel experimentu není podstatné. Výstupní veličina zaostává za žádanou hodnotou asi o 2 °C. Pro dosažení nulové regulační odchylky v oblasti rampy by bylo nutné použít 2 integrační regulátory.

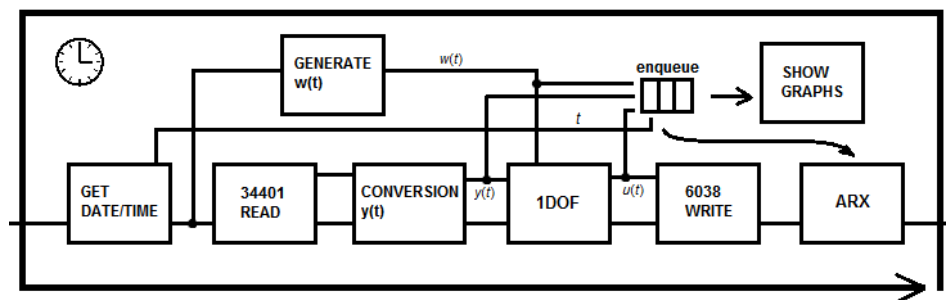
7 ADAPTIVNÍ REGULACE SOUSTAVY

Aplikace pro adaptivní regulaci vznikla jako požadavek na možnost řízení složitějších systémů v budoucnosti. Pro řízení tepelné soustavy však nemá velký význam, avšak lze ji využít k ověření funkčnosti aplikace. Tato aplikace je tedy rozšířením aplikace pro regulaci s pevnými parametry, takže zde budou uvedeny pouze rozdíly. Tato aplikace sdílí projekt s aplikací pro řízení s pevnými parametry.

7.1 Ovládání aplikace Regulace

7.1.1 Popis programu

Blokové schéma aplikace je velmi podobné blokovému schématu pro regulaci s pevnými parametry. Na následujícím obrázku je tedy zobrazena pouze měřicí smyčka.



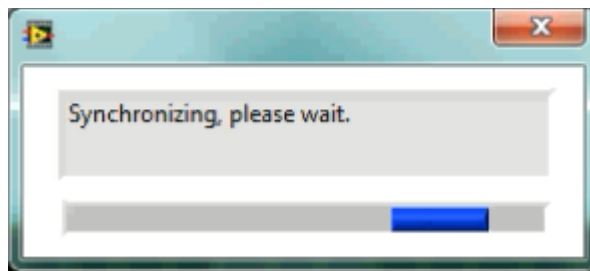
Obrázek 63 – Měřicí smyčka aplikace pro adaptivní řízení

Na konci měřicí smyčky je modul pro průběžnou identifikaci. K identifikaci je využita Metoda nejmenších čtverců a k výpočtu přenosu je použit ARX model. Vypočtené koeficienty vstupují spolu se žádanou a výstupní veličinou do regulátoru s jedním stupněm volnosti (dále jen 1dof regulátor). Regulátor se ve skutečnosti skládá ze dvou samostatných modulů, kde jeden modul vypočítá parametry regulátoru a druhý modul vygeneruje akční zásah (diferenční rovnice). Celá smyčka je navíc synchronizována s multiplikativní periodou časovače, takže perioda měření je ekvidistantní.

7.2 Chování programu

Po spuštění programu je aplikace v nastavovacím módu a je možné nastavit hodnoty prvků Front panelu. Po stisknutí tlačítka **Start** dochází k synchronizaci se systémovým časovačem spuštěním smyčky naprázdno. To má za následek zpoždění začátku měření na náhodnou dobu. Pokud je nastavená perioda velká, jsou také opožděny reakce na GUI prvky. Při práci s aplikací je tedy nutné s tímto zpožděním počítat. O počáteční

synchronizaci je uživatel informován pomocí dialogu, který automaticky zmizí po spuštění měření (nebo jej může uživatel zavřít a vyčkat na začátek měření sám).



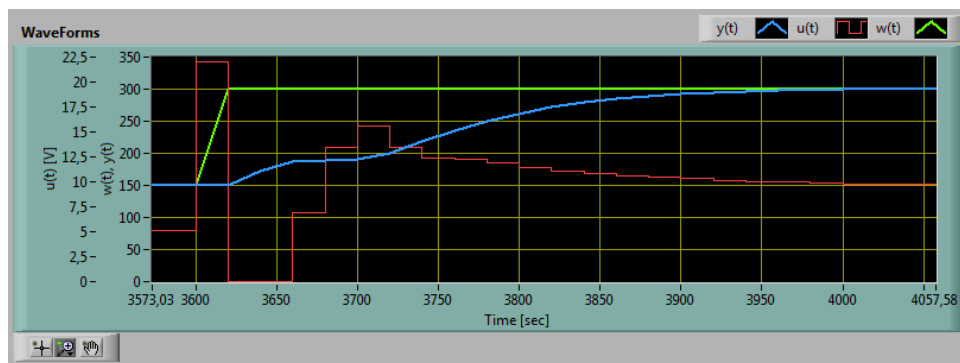
Obrázek 64 – Synchronizační dialog

7.3 Front panel

Front panel je téměř shodný s aplikací pro regulaci s pevnými parametry (PID regulátorem).

Graf

Jelikož je akční veličina skoková a po dobu periody má konstantní hodnotu (ve skutečnosti je skoková i u ostatních aplikací) je průběh akční veličiny zobrazen jako skoková funkce.



Obrázek 65 – Zobrazení časových průběhů v aplikaci pro adaptivní řízení

Menu

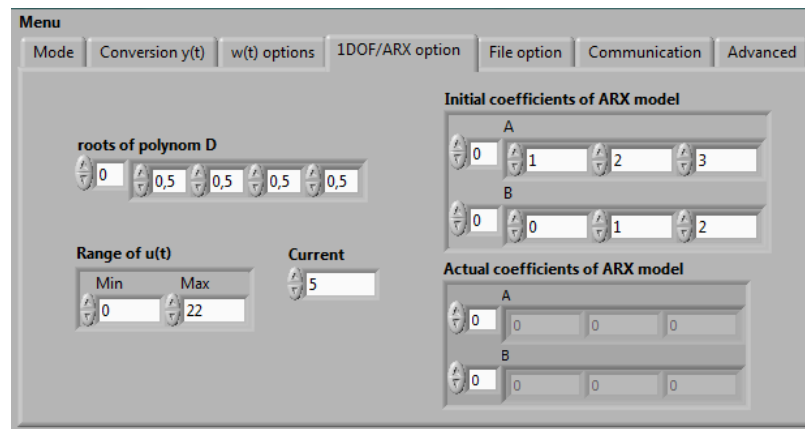
Slouží k nastavení parametrů aplikace. Skládá se z několika dalších subbloků.

Mode

Blok *Mode* obsahuje navíc ovládací prvek, umožňující zadat periodu měření v sekundách. Všechny GUI prvky pak reagují s tímto nastaveným zpožděním (mezitím probíhá měření nebo je aplikace uspana a čeká na další multiplikatívni periodu časovače).

1DOF/ARX option

Slouží k nastavení parametrů regulátoru a k počáteční inicializaci koeficientů ARX modelu.



Obrázek 66 – Nastavení parametrů 1DOF/ARX

Význam jednotlivých prvků je následující.

roots of polynom D – kořeny polynomu D (polynom D je jeden volný parametr regulátoru).

Initial coefficients of ARX model – počáteční hodnoty koeficientů ARX modelu.

Actual coefficients of ARX model – aktuální hodnoty ARX modelu.

Pozn.: Počáteční hodnoty nemusí odpovídat diskrétnímu přenosu soustavy. Aktuální hodnoty koeficientů jsou vypočteny z dat, které jsou uloženy ve FIFO paměti pro zobrazení v XY grafu. To znamená, že model soustavy je neustále zpřesňován, ale po uplynutí jednoho průběhu signálu definicí jsou první data zapomínána (úplně) a velikost pole dat je již konstantní.

7.3.1 Soubor

V hlavičce souboru jsou uloženy kořeny polynomu D (místo PID parametrů). Soubor má potom následující tvar.

```
*****
```

```
Integration Time [NPLC]
```

```
10
```

```
Min          Max          root1      root2      root3      root4
0            22           0,5       0,5       0,5       0,5
```

```
-----
```

```
čas [s]      žádaná teplota [°C] napětí zdroje [V]   teplota
soustavy [°C]
```

```
0            150           22         24
19,999144   150,000000          0,000000   30,465400
39,999288   150,000000          10,407905  38,914375
59,999432   150,000000          17,789332  45,843650
```

```
...
```

Z uložených dat je patrné, že perioda měření je ekvidistantní. Nepřesnosti na čtvrtém a vyšším desetinném je možné zanedbat, neboť časovač pracuje s přesností na ms.

7.4 Vývoj aplikace Regulace

Jelikož modul pro průběžnou identifikaci vypočítá parametry diskrétního modelu, bylo potřeba vyřešit synchronizaci měřicí smyčky tak, aby perioda byla ekvidistantní. K tomu byl do smyčky umístěn prvek *Wait until next multiple*. Vstupem do tohoto prvku je počet milisekund, avšak pokud by druhé vlákno trvalo déle než zadaný interval, čekalo by se až na následující periodu a skutečná perioda by byla nejméně dvojnásobná. Z toho vyplývá, že nastavená doba periody musí být vždy větší, než předpokládaná doba skutečného měření. Synchronizace s multiplikativní periodou navíc způsobuje, že první spuštění smyčky může proběhnout kdykoliv v tomto intervalu, a proto je první iterace smyčky kratší než nastavená perioda. Toto je vyřešeno přeskočením této náhodně dlouhé iterace a spuštěním měření až v periodách dalších. Přeskočení prvních iterací však znamená, že počáteční razítko nelze sejmout v inicializační části, ale pouze v aktivní části v první platné

iteraci. O synchronizaci je uživatel informován pomocí dialogu, který byl použit z projektu *Busy Popup Project* staženým z Internetu a upraven pro účely aplikace.

Pro identifikaci systému byl použit modul *SI Estimate ARX Model (SISO Array)*, který je součástí prostředí LabView. Délka vstupních dat musí být větší než $[\max(\text{A order}, \text{B order} + \text{delay} - 1) + (\text{A order} + \text{B order})]$. Na začátku měření, kdy dat je ještě nedostatek, jsou tedy využívány počáteční koeficienty ARX modelu zadané uživatelem.

Pomocí polynomiální metody byla určena diferenční rovnice regulátoru.

$$u(k) = q_0e(k) + q_1e(k-1) + q_2e(k-2) + (1-p_1)u(k-1) + p_1u(k-2) \quad (57)$$

Srovnáním koeficientů diofantické rovnice byly vypočítány rovnice pro výpočet parametrů regulátoru zapsané v maticovém tvaru.

$$\begin{bmatrix} b_1 & 0 & 0 & 1 \\ b_2 & b_1 & 0 & a_1 - 1 \\ 0 & b_2 & b_1 & a_2 - a_1 \\ 0 & 0 & b_2 & -a_2 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ p_1 \end{bmatrix} = \begin{bmatrix} d_1 + 1 - a_1 \\ d_2 + a_1 - a_2 \\ d_3 + a_2 \\ d_4 \end{bmatrix} \quad (58)$$

$$\mathbf{AX} = \mathbf{B} \Rightarrow \mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$$

Obě rovnice byly naprogramovány ve formě modulů a dohromady tvoří regulátor soustavy. Samozřejmě je možné naprogramovat i jiné typy regulátorů, neboť výpočet parametrů regulátoru je proveden na základě koeficientů ARX modelu.

Pro ověřování funkcí a simulaci řízení byl vytvořen testovací program, pomocí kterého byly určeny přesné parametry aplikace před nasazením na reálnou soustavu.

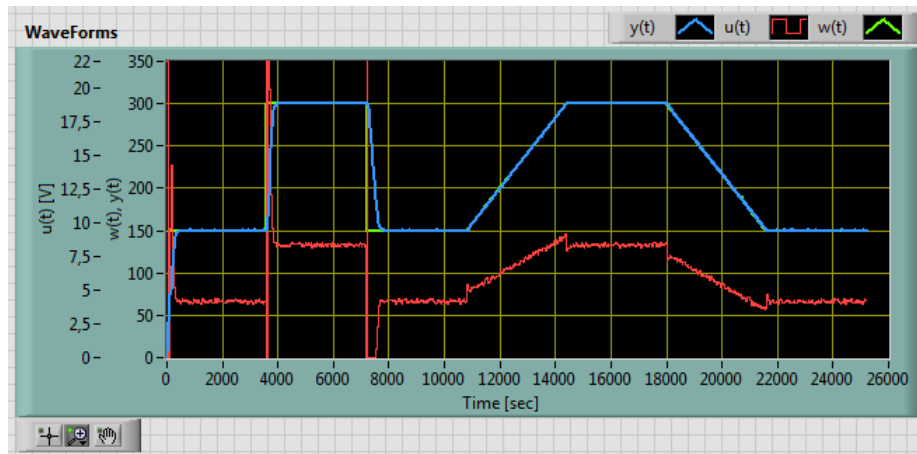
7.5 Regulace reálné soustavy

K regulaci byla využita opět tepelná pec, i když k regulaci této pece není adaptivní řízení potřeba. Vytvořenou aplikaci lze však v budoucnu nasadit na složitější systém.

7.5.1 Simulace regulace

Pro simulaci byl použit testovací program. Tento program umožňuje simulovat dlouhé procesy ve velmi krátkém čase a tato data uložit do souboru. Porovnáním naměřených dat pro různé parametry aplikace můžeme zjistit přesné parametry aplikace. Parametry byly tedy zjištěny experimentálně (simulačně) pro soustavu zadanou spojitým přenosem.

Průběh jedné simulace je na následujícím obrázku.

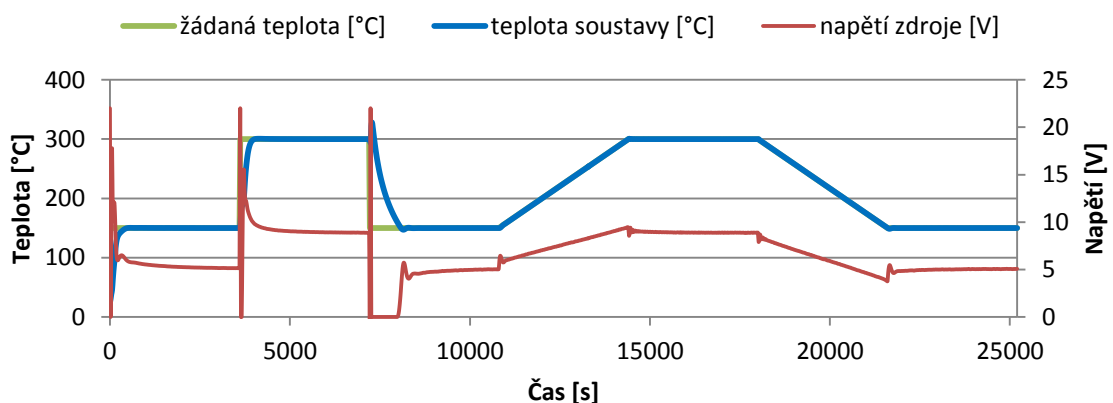


Obrázek 67 – Časový průběh simulace adaptivního řízení

Ze simulací bylo zjištěno, že vhodná perioda měření je 20 s a vhodný polynom D je tvořen násobným kořenem $d = 0,5$. Koeficienty diskrétního přenosu soustavy lze využít jako počáteční koeficienty ARX modelu.

7.5.2 Regulace – reálný experiment

Pro účely experimentu byl reálný experiment proveden na tepelné peci. Parametry aplikace byly nastaveny podle výsledků ze simulace experimentu. Časový průběh veličin je zobrazen na následujícím obrázku.



Obrázek 68 – Časový průběh adaptivní regulace – reálný experiment

Z grafu je patrná dobrá shoda výstupní veličiny se žádanou hodnotou. Adaptivní řízení pro tyto parametry funguje podle očekávání a časový průběh celkem odpovídá časovému průběhu simulace řízení.

ZÁVĚR

Cílem této práce bylo vytvořit aplikace pro měření, identifikaci a řízení systému podle konkrétních požadavků na spolupracujícím pracovišti FT UTB. Tyto aplikace byly vytvořeny v prostředí LabView a komunikují se dvěma přístroji programovatelným zdrojem HP6038A a multimetrem HP34401A pomocí GPIB sběrnice.

Před tvorbou této práce byly prostudovány odborné materiály, na základě kterých byla sestavena teoretická část práce. Ta zahrnuje stručný popis prostředí LabView a popis vybraných metod měření, identifikace a řízení.

V praktické části byly popsány možnosti a řešené problémy při vývoji těchto aplikací. Cíle práce bylo dosaženo a v prostředí LabView byly vytvořeny celkem 3 aplikace. Pro měření a identifikaci systému je použita stejná aplikace, neboť principem je měření odezvy systému na určitý signál. Aplikace s pevnými parametry obsahuje PID regulátor z toolboxu *PID and Fuzzy Logic Toolbox*. Díky možnosti převodu PID parametrů lze řídit i nelineární soustavy, které však musí být časově invariantní. Poslední aplikace je aplikace pro adaptivní řízení. Samotné řízení zajišťuje regulátor s jedním stupněm volnosti rozdělený na 2 části. První část vypočítá parametry regulátoru na základě koeficientů ARX modelu, určených průběžnou identifikací metodou nejmenších čtverců. Druhá část provede výpočet akčního zásahu. Součástí práce jsou také některé výsledky měření, řízení a simulací.

ZÁVĚR V ANGLIČTINĚ

The aim of this thesis was to develop applications for measurement, identification and control of systems according to specific requirements of the collaborating laboratory of FT UTB. These applications were developed in the LabView environment. The applications communicate with two devices: HP6038A programmable source and HP34401A multimeter using GPIB bus.

Materials relating to the issue were reviewed prior to creating applications. The theoretical part was based on the obtained information. This part contains a brief description of the LabView environment and description of selected methods of measurement, identification and control.

The practical part contains description of possibilities and problems solved in these applications. The aim of this thesis was achieved and three applications were created in the LabView environment. The same application is used for measurement and identification of systems. The principle of this application is to measure a response to an input signal. The application with fixed parameters contains a PID controller from PID and Fuzzy Logic toolbox. We can control nonlinear systems by conversion of PID parameters. The systems must be time-invariant. The last application is for adaptive control. The controller has the one degree of freedom structure and it is divided into the two parts. The first part calculates controller's parameters on the basis of the ARX model coefficients. The ARX model coefficients are calculated by the least squares method. The second part calculates the controller's output. This thesis also contains some results of measurement, control and simulation.

SEZNAM POUŽITÉ A PŘEHLEDOVÉ LITERATURY

- [1] What is LabView? In *National Instruments*. [online]. [cit. 2013-05-07]. Dostupný z: <<http://sine.ni.com/np/app/main/p/docid/nav-104/lang/cs/fmid/1762/>>
- [2] Automatizované měřicí systémy se sběrnicí IEEE 488 (GPIB) [online]. [cit. 2013-05-07]. Dostupný z: <http://www.rss.tul.cz/download/cms/03_GPIB_HW.pdf>
- [3] Metody identifikace systémů z přechodových charakteristik In *Katedra automatizační techniky a řízení*. [online]. [cit. 2013-05-07]. Dostupný z <http://www.352.vsb.cz/uc_texty/Identifikace/str/metody.htm#ma3>
- [4] Průmyslové PID regulátory: tutorial In *matlab.fei.tuke.sk*. [online]. [cit. 2013-05-07]. Dostupný z: <http://matlab.fei.tuke.sk/raui/doc/PIDTutor_CZ.pdf>
- [5] Naslinova metoda nastavení regulátoru In *Řízení technologických procesů*. [online]. [cit. 2013-05-09]. Dostupný z: <<http://rtp.webzdarma.cz/řízení11.php>>
- [6] Navrátil, Pavel. Polynomiální metoda - systém řízení se zpětnovazebním regulátorem (1DOF konfigurace) In *Počítačová podpora automatického řízení*. [online]. [cit. 2013-05-07]. Dostupný z: <http://195.178.89.122/CAAC_PHP/CAAC/cesky/synteza/1dof_fb/1dof_fb.php>
- [7] Šulc B., Vítečková M.: *Teorie a praxe návrhu regulačních obvodů*, vydavatelství ČVUT, 2004.
- [8] Vlach J., Havlíček J., Vlach M.: *Začínáme s LabView*, BEN-technická literatura, 2008.
- [9] Balátě J.: *Automatické řízení*, BEN-technická literatura, 2003.
- [10] *LabView: Real-time user manual*. Austin: National Instruments Corporation, 2004.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

1DOF	One Degree Of Freedom
GPIB	General Purpose Interface Bus
IEEE	Institute of Electrical and Electronics Engineers
LED	Light-Emitting Diode
MSExcel	Microsoft Excel
NI	National Instruments
NiCr-Ni	Nickel-Chromium-Nickel
OLED	Organic Light-Emitting Diode
PC	Personal Computer
PID	Proportional-Integral-Derivate
PNBS	PseudoNáhodný Binární Signál
PXI	The PCI eXtensions for Instrumentation
TTL	Tranzistor-Tranzistor-Logic
VA	Volt-Ampérová
VI	Virtual Instrument
VISA	The Virtual Instrument Software Architecture
VXI	VME eXtensions for Instrumentation

SEZNAM OBRÁZKŮ

Obrázek 1 – Měření VA charakteristiky – varianta 1	14
Obrázek 2 – Měření VA charakteristiky – varianta 2	15
Obrázek 3 - Přechodová charakteristika s aperiodickým průběhem a s vyznačenou dobou náběhu a dobou průtahu	16
Obrázek 4 - Aproximace proporcionální soustavou 2.řádu s rozdílnými časovými konstantami	17
Obrázek 5 - Aproximace proporcionální soustavou 2.řádu se stejnými časovými konstantami	18
Obrázek 6 – Časový průběh pseudonáhodného signálu	20
Obrázek 7 – Generátor PNBS	21
Obrázek 8 – Časový průběh PNBS	21
Obrázek 9 – Základní schéma regulačního obvodu	22
Obrázek 10 – Windup efekt a jeho vliv na výstupní veličinu	24
Obrázek 11 – Systém řízení s jedním stupněm volnosti	26
Obrázek 12 – Instalace aplikace Identifikace – krok 1	30
Obrázek 13 – Instalace aplikace Identifikace – krok 2	31
Obrázek 14 – Instalace aplikace Identifikace – krok 3	31
Obrázek 15 – Instalace aplikace Identifikace – krok 4	32
Obrázek 16 – Obecné blokové schéma aplikací	32
Obrázek 17 – Blokové schéma aplikace Identifikace	33
Obrázek 18 – Front panel aplikace Identifikace	34
Obrázek 19 – Vizualizace dat (Identifikace) – graf XT	35
Obrázek 20 – Vizualizace dat – graf XY	35
Obrázek 21 – Tlačítka	36
Obrázek 22 – Výstupy	36
Obrázek 23 – Mód aplikace	37
Obrázek 24 – Transformace výstupní veličiny $y(t)$	38
Obrázek 25 – Nastavení parametrů akční veličiny $u(t)$	40
Obrázek 26 – Příklad modelovaného signálu $u(t)$	41
Obrázek 27 – Nastavení vlastností souboru	42
Obrázek 28 – Nastavení parametrů komunikace se zdrojem a multimetrem	43
Obrázek 29 – Nastavení pokročilých parametrů aplikace	44

Obrázek 30 – Vyrovnávací paměť pro hodnoty zobrazované v grafech XT a XY	45
Obrázek 31 – Závislost Periody měření na nastavení Integration Time	47
Obrázek 32 – Příklad převodu výstupní veličiny tabulkou.....	50
Obrázek 33 – Tvorba ikony aplikace Identifikace v Icon Editoru.....	59
Obrázek 34 – Schéma zapojení pro měření VA charakteristik.....	61
Obrázek 35 – VA charakteristiky komerčních LED.....	62
Obrázek 36 – VA charakteristiky OLED.....	62
Obrázek 37 – Schéma zapojení pece	63
Obrázek 38 – Tepelná soustava - pec	64
Obrázek 39 – Měření přechodové charakteristiky soustavy – původní signál	65
Obrázek 40 – Přechodová charakteristika soustavy	66
Obrázek 41 – Simulace přechodové charakteristiky - model	67
Obrázek 42 – Simulace přechodové charakteristiky – graf	67
Obrázek 43 – Simulace měření statické charakteristiky – model.....	68
Obrázek 44 – Simulace měření statické charakteristiky – graf	68
Obrázek 45 – Měření statické charakteristiky – reálná soustava.....	69
Obrázek 46 – Statická charakteristika soustavy a lineární aproximace.....	69
Obrázek 47 – Buzení soustavy generátorem PNBS.....	70
Obrázek 48 – Buzení soustavy pseudonáhodným signálem s rovnoměrným rozdělením.....	70
Obrázek 49 – Zjednodušené blokové schéma aplikace Regulace	71
Obrázek 50 – Front Panel aplikace Regulace	73
Obrázek 51 – Graf časového průběhu veličin v aplikaci Regulace	74
Obrázek 52 – Výstupy aplikace Regulace	74
Obrázek 53 – Nastavení parametrů žádané hodnoty $w(t)$	75
Obrázek 54 – Nastavení parametrů PID regulátoru.....	76
Obrázek 55 – Nastavení parametrů hlavičky a souboru	77
Obrázek 56 – Pokročilé nastavení aplikace regulace.....	78
Obrázek 57 – Zadání P parametru tabulkou či polynomem	79
Obrázek 58 – Vyrovnávací paměť pro hodnoty zobrazované v XY grafu.....	79
Obrázek 59 – Tvorba ikony aplikace PID regulátor v Icon Editoru	81
Obrázek 60 – Simulace regulace – model	82
Obrázek 61 – Simulace regulace - graf.....	83

Obrázek 62 – Regulace soustavy – reálný experiment	84
Obrázek 63 – Měřicí smyčka aplikace pro adaptivní řízení	85
Obrázek 64 – Synchronizační dialog	86
Obrázek 65 – Zobrazení časových průběhů v aplikaci pro adaptivní řízení.....	86
Obrázek 66 – Nastavení parametrů 1DOF/ARX	87
Obrázek 67 – Časový průběh simulace adaptivního řízení.....	90
Obrázek 68 – Časový průběh adaptivní regulace – reálný experiment	90

SEZNAM TABULEK

Tabulka 1 – Určení poměru časových konstant.....	18
Tabulka 2 – Stanovení řádu n aproximační soustavy a zpřesnění polohy inflexního bodu.....	19
Tabulka 3 – Závislost maximálního přeregulování na koeficientu α podle Naslina	25
Tabulka 4 – Definice signálu.....	41
Tabulka 5 – Hlavička – popis sloupců.....	42
Tabulka 6 – Závislost Resolution na Integration Time.....	47
Tabulka 7 – Převod výstupní veličiny	50
Tabulka 8 – Stavový automat – tabulka přechodů stavů	56
Tabulka 9 – Převodní tabulka termoelektrického čidla	64

SEZNAM PŘÍLOH

Příloha P I: DVD se zdrojovými kódy, instalačními soubory a manuály k oběma přístrojům