

**UNIVERZITA PARDUBICE**  
Fakulta elektrotechniky a informatiky

**IDENTIFIKACE A ŘÍZENÍ SYSTÉMU MOTOR GENERÁTOR**

Bc. Vladislav Kořínek

Diplomová práce  
2017

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Vladislav Kořínek**  
Osobní číslo: **I15181**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Řízení procesů**  
Název tématu: **Identifikace a řízení systému motor generátor**  
Zadávající katedra: **Katedra řízení procesů**

### Z á s a d y p r o v y p r a c o v á n í :

#### Cíl:

Metodami experimentální identifikace určit diskrétní dynamický model systému motor generátor, navrhnout jednoduchý regulátor a implementovat ho v řídicím systému AMiNi4DS.

#### Teoretická část:

- a) experimentální identifikace diskrétního dynamického systému
- b) určení kritických hodnot

#### Praktická část:

- a) seznámení s vývojovým systémem DetStudio
- b) určit maximální použitelnou vzorkovací frekvenci řídicího systému a navrhnout způsob získání naměřených dat
- c) s využitím AMiNi4DS získat data pro experimentální identifikaci
- d) navrhnout a implementovat jednoduchý regulátor a ověřit jeho funkci

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**BALÁTĚ, Jaroslav. Automatické řízení. 2., přeprac. vyd. Praha: BEN, 2004, 663 s. ISBN 80-730-0148-9.**

**OGATA, Katsuhiko. Discrete-time control systems. 2nd ed. Englewood Cliffs, N.J.: Prentice Hall, c1995, xi, 745 p. ISBN 01-303-4281-5.**

**Amit - Home [online]. (c)2016 [cit. 2016-09-29]. Dostupné z:**

**<https://www.amit.cz/>**

**Arduino - ArduinoDue. Arduino - Home [online]. (c)2015 [cit. 2015-10-06].**

**Dostupné z: <https://www.arduino.cc/en/Guide/ArduinoDue>**

Vedoucí diplomové práce:

**doc. Ing. František Dušek, CSc.**

Katedra řízení procesů

Datum zadání diplomové práce: **26. října 2016**

Termín odevzdání diplomové práce: **17. května 2017**



Ing. Zdeněk Němec, Ph.D.  
děkan



L.S.



Ing. Daniel Honc, Ph.D.  
vedoucí katedry

V Pardubicích dne 15. listopadu 2016

## **Prohlášení**

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 17. 5. 2017

Bc. Vladislav Kořínek

### **Poděkování**

V této části bych chtěl poděkovat panu doc. Ing. Františkovi Duškovi, CSc. za spolupráci při zpracování diplomové práce a také za cenné rady, připomínky a poskytnuté materiály, které byly důležité pro dokončení této práce.

V Pardubicích dne 17. 5. 2017

Bc. Vladislav Kořínek

## **ANOTACE**

*Cílem práce je určit metodami experimentální identifikace diskrétní dynamický model systému motor generátor. Navrhnout jednoduchý regulátor pro řízení systému a implementovat ho do řídicího systému AMiNi4DS. Obsluha řídicího systému se provádí ve vývojovém prostředí DetStudio. Pro experimentální identifikaci je využit výpočtový software MATLAB a tabulkový editor Excel.*

## **KLÍČOVÁ SLOVA**

*Motor generátor, regulátor, AMiNi4DS, DetStudio, experimentální identifikace.*

## **TITLE**

*IDENTIFICATION AND CONTROL OF THE MOTOR GENERATOR*

## **ANNOTATION**

*The aim of this thesis is to determine the methods of experimental identification of the discrete dynamic model of the motor generator system. Design a simple controller for system control and implement it into the AMiNi4DS control system. Operation of the control system is carried out in the DetStudio development environment. The MATLAB computational software and the Excel spreadsheet are used for experimental identification.*

## **KEYWORDS**

*Motor generator, controller, AMiNi4DS, DetStudio, experimental identification.*

## OBSAH

|       |  |    |
|-------|--|----|
|       | Seznam zkratk a značek .....                     | 9  |
|       | Seznam symbolů proměnných veličin a funkcí ..... | 10 |
|       | Seznam ilustrací .....                           | 11 |
|       | Seznam tabulek .....                             | 13 |
|       | ÚVOD .....                                       | 14 |
| 1     | TEORIE ŘÍZENÍ SYSTÉMU .....                      | 16 |
| 1.1   | Regulovaná soustava .....                        | 17 |
| 1.2   | Popis regulovaných soustav .....                 | 18 |
| 1.2.1 | Diferenciální rovnice .....                      | 19 |
| 1.2.2 | Diferenční rovnice .....                         | 19 |
| 1.2.3 | Spojité přenosové funkce .....                   | 20 |
| 1.2.4 | Diskrétní přenosové funkce .....                 | 21 |
| 1.3   | Regulátor .....                                  | 22 |
| 1.4   | PSD regulátor .....                              | 24 |
| 1.5   | Metody nastavení regulátorů .....                | 25 |
| 1.6   | Perioda vzorkování .....                         | 26 |
| 2     | SOUSTAVA MOTOR-GENERÁTOR .....                   | 27 |
| 3     | METODY IDENTIFIKACE SYSTÉMŮ .....                | 30 |
| 3.1   | Matematicko-fyzikální analýza .....              | 30 |
| 3.2   | Experimentální identifikace .....                | 30 |
| 3.3   | Vstupní testovací signál .....                   | 31 |
| 3.4   | Výběr vyhodnocovací metody .....                 | 33 |
| 3.4.1 | Metoda nejmenších čtverců .....                  | 33 |
| 3.5   | Určení parametrů systému .....                   | 34 |
| 4     | ŘÍDICÍ SYSTÉM .....                              | 36 |
| 4.1   | AMiNi4DS .....                                   | 36 |
| 4.1.1 | Vstupy a výstupy AMiNi4DS.....                   | 37 |
| 4.1.2 | Komunikační rozhraní .....                       | 38 |
| 5     | PROGRAMOVÉ VYBAVENÍ .....                        | 39 |
| 5.1   | DetStudio .....                                  | 39 |
| 5.2   | MATLAB .....                                     | 41 |
| 5.3   | Excel .....                                      | 43 |

|       |   |    |
|-------|---|----|
| 6     | PŘIPOJENÍ ŘÍDICÍHO SYSTÉMU .....            | 44 |
| 6.1   | Nastavení komunikace .....                  | 45 |
| 6.2   | Komunikace .....                            | 47 |
| 6.3   | OPC server .....                            | 47 |
| 6.4   | Komunikační ovladač ATOUCH .....            | 48 |
| 7     | PROGRAMOVÉ ŘEŠENÍ .....                     | 51 |
| 7.1   | Měření experimentálních dat .....           | 51 |
| 7.1.1 | Proces ProcInit .....                       | 53 |
| 7.1.2 | Proces ProcIDLE .....                       | 53 |
| 7.1.3 | Proces HI01 .....                           | 53 |
| 7.1.4 | Obrazovka Menu .....                        | 54 |
| 7.1.5 | Obrazovka Info .....                        | 54 |
| 7.1.6 | Obrazovka Mereni .....                      | 54 |
| 7.1.7 | Obrazovka Ovladani .....                    | 56 |
| 7.2   | Export dat .....                            | 56 |
| 7.3   | Identifikace .....                          | 58 |
| 7.4   | Relé ve zpětné vazbě .....                  | 59 |
| 7.5   | PSD regulátor .....                         | 60 |
| 8     | VÝSLEDKY .....                              | 61 |
| 8.1   | Statická charakteristika .....              | 61 |
| 8.2   | Dynamická charakteristika .....             | 62 |
| 8.3   | Vzorkovací frekvence řídicího systému ..... | 63 |
| 8.4   | Identifikace .....                          | 63 |
| 8.5   | Relé ve zpětné vazbě .....                  | 67 |
| 8.6   | Nastavení regulátoru .....                  | 69 |
| 9     | VÝSLEDKY REGULACE .....                     | 71 |
| 10    | ZÁVĚR .....                                 | 78 |
|       | POUŽITÁ LITERATURA .....                    | 79 |
|       | PŘÍLOHY .....                               | 80 |

## SEZNAM ZKRATEK A ZNAČEK

|        |                                     |
|--------|-------------------------------------|
| DPS    | Deska plošných spojů                |
| HW     | Hardware                            |
| LA     | Logické adresy                      |
| MATLAB | MATrix LABoratory                   |
| MIMO   | Multi Input, Multi Output           |
| MNČ    | Metoda nejmenších čtverců           |
| OPC    | OLE for Process Control             |
| PID    | Proporcionálně-integračně-derivační |
| PLC    | Programovatelný logický automat     |
| PNBS   | Pseudonáhodný binární signál        |
| PSD    | Proporcionálně-sumačně-diferenční   |
| RS     | Reléová schéma                      |
| SISO   | Single Input, Single Output         |
| ST     | Strukturovaný text                  |
| SW     | Software                            |
| VBA    | Visual Basic for Applications       |
| XOR    | eXklusivní součet                   |
| ZN     | Ziegler-Nichols                     |
| ZOBD   | Zpětná obdélníková difference       |

## SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ

|            |  |
|------------|--|
| $a_i, b_i$ | koeficienty přenosu                        |
| $d$        | poruchová veličina                         |
| $e$        | regulační odchylka                         |
| $F$        | matice známých měřených hodnot             |
| $G(s)$     | spojitý obrazový přenos                    |
| $G(z)$     | diskrétní obrazový přenos                  |
| $K$        | kritérium MNČ                              |
| $n$        | řád soustavy                               |
| $P$        | vektor neznámých parametrů                 |
| $q_i$      | parametry diskrétního regulátoru           |
| $r_{0k}$   | kritické zesílení                          |
| $r_i$      | parametry spojitého PID regulátoru         |
| $s$        | komplexní proměnná Laplaceovy transformace |
| $T$        | perioda vzorkování                         |
| $t_0$      | počáteční časový okamžik, s                |
| $T_D$      | derivační časová konstanta, s              |
| $T_I$      | integrační časová konstanta, s             |
| $t_k$      | diskrétní čas                              |
| $T_k$      | kritická perioda, s                        |
| $T_s$      | časová konstanta, s                        |
| $u$        | akční veličina                             |
| $U(s)$     | Laplaceův obraz vstupního signálu          |
| $U(z)$     | z obraz vstupního signálu                  |
| $w$        | žádaná veličina                            |
| $y$        | regulovaná veličina                        |
| $Y(s)$     | Laplaceův obraz výstupního signálu         |
| $Y(z)$     | z obraz výstupního signálu                 |
| $y_F$      | filtrovaná výstupní veličina               |
| $Y_N$      | vektor naměřených hodnot                   |
| $z$        | komplexní proměnná Z transformace          |
| $R(s)$     | spojitý obrazový přenos PID regulátoru     |

## SEZNAM ILUSTRACÍ

|   |    |
|---|----|
| Obrázek 1.1 – Dopředné řízení .....                     | 16 |
| Obrázek 1.2 – Zpětnovazební řízení .....                | 17 |
| Obrázek 1.3 – Regulovaná soustava .....                 | 18 |
| Obrázek 1.4 – Blokové schéma PID regulátoru .....       | 23 |
| Obrázek 2.1 – Soustava motor-generátor .....            | 27 |
| Obrázek 2.2 – Technologické schéma soustavy .....       | 28 |
| Obrázek 3.1 – Princip posuvných registrů pro PNBS ..... | 32 |
| Obrázek 4.1 – Řídicí systém AMiNi4DS .....              | 36 |
| Obrázek 4.2 – Schématická značka AMiNi4DS .....         | 38 |
| Obrázek 5.1 – Návrhové prostředí DetStudio .....        | 39 |
| Obrázek 5.2 – Výpočtový software MATLAB .....           | 42 |
| Obrázek 6.1 – Mechanický převodník .....                | 44 |
| Obrázek 6.2 – Nastavení komunikace v DetStudios .....   | 46 |
| Obrázek 6.3 – OPC server .....                          | 47 |
| Obrázek 7.1 – Vývojový diagram .....                    | 52 |
| Obrázek 7.2 – Seznam proměnných .....                   | 53 |
| Obrázek 7.3 – Příkaz na přečtení vstupu .....           | 54 |
| Obrázek 7.4 – Příklad zápisu příkazů programu .....     | 54 |
| Obrázek 7.5 – Obrazovka Menu .....                      | 55 |
| Obrázek 7.6 – Obrazovka Info .....                      | 55 |
| Obrázek 7.7 – Obrazovka Merení .....                    | 56 |
| Obrázek 7.8 – Obrazovka Ovládání .....                  | 56 |
| Obrázek 7.9 – Metoda InitFromFile .....                 | 57 |
| Obrázek 7.10 – Metoda DbGetInfo .....                   | 57 |
| Obrázek 7.11 – Metoda GetDataMtx .....                  | 58 |
| Obrázek 7.12 – Metoda Done .....                        | 58 |
| Obrázek 7.13 – Funkce MNČ .....                         | 59 |
| Obrázek 7.14 – Relé ve zpětné vazbě .....               | 60 |
| Obrázek 7.15 – PSD regulátor .....                      | 60 |
| Obrázek 8.1 – Statická charakteristika .....            | 61 |
| Obrázek 8.2 – Dynamické vlastnosti soustavy .....       | 62 |
| Obrázek 8.3 – Měření pro identifikaci AMiNi4DS .....    | 63 |

|  |    |
|--|----|
| Obrázek 8.4 – Identifikace AMiNi4DS .....  | 64 |
| Obrázek 8.5 – Měření pro identifikaci soustavy .....                             | 65 |
| Obrázek 8.6 – Identifikace soustavy a řídicího systému pro 1 ms .....            | 66 |
| Obrázek 8.7 – Identifikace soustavy a řídicího systému pro 10 ms .....           | 66 |
| Obrázek 8.8 – Relé ve zpětné vazbě pro 1 ms .....                                | 68 |
| Obrázek 8.9 – Relé ve zpětné vazbě pro 10 ms .....                               | 68 |
| Obrázek 9.1 – PS regulátor pro identifikaci při 1 ms .....                       | 71 |
| Obrázek 9.2 – Akční zásah PS regulátoru pro identifikaci při 1 ms .....          | 71 |
| Obrázek 9.3 – PSD regulátor pro identifikaci při 1 ms .....                      | 72 |
| Obrázek 9.4 – Akční zásah PSD regulátoru pro identifikaci při 1 ms .....         | 72 |
| Obrázek 9.5 – PS regulátor pro relé ve zpětné vazbě při 1 ms .....               | 73 |
| Obrázek 9.6 – PSD regulátor pro relé ve zpětné vazbě při 1 ms .....              | 73 |
| Obrázek 9.7 – Akční zásah PSD regulátoru pro relé ve zpětné vazbě při 1 ms ..... | 74 |
| Obrázek 9.8 – PS regulátor pro identifikaci při 10 ms .....                      | 74 |
| Obrázek 9.9 – Akční zásah PS regulátoru pro identifikaci při 10 ms .....         | 75 |
| Obrázek 9.10 – PSD regulátor pro identifikaci při 10 ms .....                    | 75 |
| Obrázek 9.11 – Akční zásah PSD regulátoru pro identifikaci při 10 ms .....       | 76 |
| Obrázek 9.12 – PS regulátor pro relé ve zpětné vazbě při 10 ms .....             | 76 |
| Obrázek 9.13 – PSD regulátor pro relé ve zpětné vazbě při 10 ms .....            | 77 |

## SEZNAM TABULEK

|  |    |
|--|----|
| Tabulka 1.1 – Přepočet parametrů regulátoru .....              | 25 |
| Tabulka 2.1 – Popis signálů soustavy .....                     | 29 |
| Tabulka 3.1 – Funkce XOR .....                                 | 32 |
| Tabulka 3.2 – Primitivní polynomu pro PNBS .....               | 32 |
| Tabulka 4.1 – Technické parametry .....                        | 37 |
| Tabulka 4.2 – Technické parametry Ethernetu .....              | 38 |
| Tabulka 6.1 – Význam signálů svorkovnice motor generátor ..... | 45 |
| Tabulka 8.1 – Chyby aproximace .....                           | 64 |
| Tabulka 8.2 – Chyby aproximace systému .....                   | 67 |
| Tabulka 8.3 – Kritické hodnoty z identifikace .....            | 67 |
| Tabulka 8.4 – Kritické hodnoty z relé .....                    | 68 |
| Tabulka 8.5 – Parametry spojitých regulátorů .....             | 69 |
| Tabulka 8.6 – Parametry diskrétních regulátorů .....           | 70 |
| Tabulky 9.1 – Směrodatné odchylky chyby regulace .....         | 77 |

# ÚVOD

Řízení procesů je v současné době jeden z nejvyužívanějších oborů v automatizaci. Technologické procesy ve většině průmyslových odvětví procházejí modernizací a je tedy potřeba zautomatizovat jejich řízení a ovládání. Automatizace procesů spočívá ve využití řídicích systémů pro zvýšení jejich efektivity a bezpečnosti provozu.

Tato práce se zabývá řízením laboratorní úlohy tvořené reálnou soustavou motor generátor. Cílem je identifikovat soustavu motor generátor a na základě výsledků identifikace navrhnout jednoduchý regulátor pro její řízení.

V první teoretické části je popsán úvod do problematiky, kde je rozebrán pojem řízení a jeho typy. V další části jsou popsány prvky potřebné pro řízení, mezi které patří hlavně regulovaná soustava a regulátor. Prvky jsou popsány ve spojitém i diskrétním tvaru, jelikož řídicí systém pracuje v diskrétní oblasti.

Práce se dále zabývá popisem soustavy motor generátor a možnými metodami identifikace popsané soustavy. Pro identifikaci soustavy motor generátor byla vybrána experimentální identifikace a metoda nejmenších čtverců pro určení neznámých parametrů soustavy. Jako vstupní signál je vybrán pseudonáhodný binární signál, kterému je zde věnována další podkapitola.

Soustava bude řízena řídicím systémem od firmy AMiT, konkrétně typem AMiNi4DS. I tento řídicí systém je popsán v další kapitole společně s návrhovým prostředím DetStudio, které je vyvinuto přímo pro programování aplikací pro řídicí systémy firmy Amit. Další využívaný software je výpočtový software MATLAB a tabulkový editor Excel.

Praktická část se nejprve zabývá připojením řídicího systému k soustavě motor generátor. Dále je vysvětlen způsob a nastavení komunikace mezi řídicím systémem a PC.

Následně přijde na řadu samotné programování, kdy je nejprve potřeba vytvořit program AMiNi4DS pro měření dat na soustavě a program pro přenos těchto měřených dat do počítače pro další zpracování. Vyčtení naměřených dat je realizováno programem v tabulkovém procesu Excel, který využívá ovladač ATOUCHX (ActiveX). Tento ovladač je k dispozici jako součást instalace návrhového prostředí DetStudio.

V další části bude z naměřených dat provedena experimentální identifikace diskrétního přenosu soustavy a určení parametrů diskrétního regulátoru. Nastavení hodnot regulátoru bude provedeno na základě kritických hodnot určených nejprve z diskrétního přenosu a také z experimentu s relé ve zpětné vazbě. Realizace experimentu s relé bude řešena programově v AMiNi4DS. Diskrétní PSD regulátor bude implementován do řídicího systému a budou

provedeny experimenty pro ověření jeho chování pro dva intervaly vzorkování a sady parametrů vypočtené z kritických hodnot získaných odlišným způsobem.

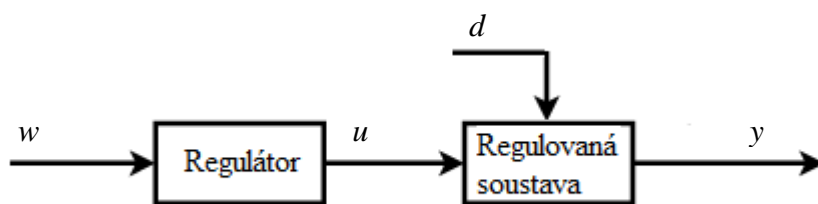
Na závěr je zhodnoceno splnění cílů diplomové práce a shrnutí výsledků provedené identifikace a funkčnosti různých variant nastavení PSD regulátoru.

# 1 TEORIE ŘÍZENÍ SYSTÉMU

Základním oborem v automatizaci je řízení technologických procesů. Jako technologický proces se uvažuje řízený systém, na který je cílevědomě působeno za účelem dosažení předem stanoveného cíle. Systém lze charakterizovat stavovým popisem nebo vstupně výstupním popisem. Stavový popis je charakterizován strukturou systému, která určuje jeho vnitřní funkční stavy. Ty určují momentální stav systému. Naopak vstupně výstupní popis charakterizuje chování systému jako vnější vztah k okolí. Zde je systém chápán jako černá krabice, kde se stavy neberou v úvahu, ale je dán změnou vstupu a příslušnými odezvami objevujícími se na příslušném výstupu.

Pro návrh řízení systému je nutné znát jeho dynamické a statické vlastnosti. Dynamický systém znamená, že hodnoty výstupních veličin nezávisí pouze na aktuálních hodnotách vstupních veličin, ale i na předešlých hodnotách vstupů a výstupů. Statické vlastnosti systému udávají závislost výstupních veličin na veličinách vstupních v ustáleném stavu. U statických systémů nezáleží na historii vstupů a výstupů. Tedy v systému nejsou žádné dynamické změny veličin a systém se nachází v ustáleném stavu. Ovšem ustálený stav nemusí existovat pro všechny dynamické systémy.

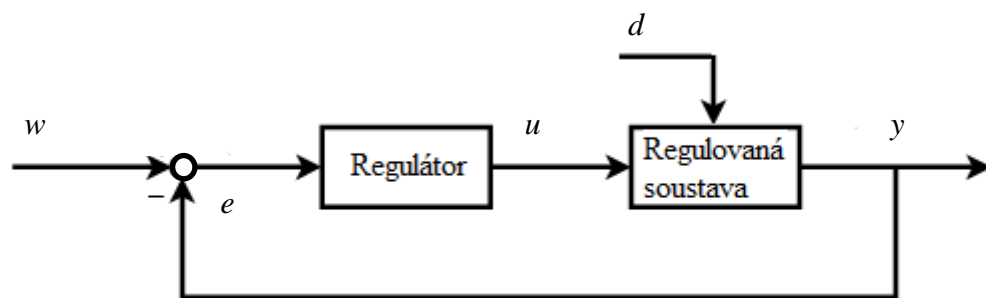
Řízení je obecně děleno na dopředné a zpětnovazební. Oba způsoby řízení se liší použitím zpětné vazby u zpětnovazebního řízení. Řízení se zpětnou vazbou odpovídá regulaci a dopředné řízení je spíše jen ovládání systému.



Obrázek 1.1 - Dopředné řízení

U dopředného řízení se regulátorem nastavuje akční veličina  $u$ , bez informace o skutečné hodnotě regulované veličiny  $y$ . Takže regulátor působí na soustavu přímo, bez informace o předchozím stavu soustavy a je snaha se na výstupu co nejvíce přiblížit žádané hodnotě  $w$ . Na regulovanou soustavu dále mohou působit vnější poruchové veličiny  $d$ . Ty pokud nejsou konstantní nebo předem známé, tak je nelze nijak odstranit. Obvodu, kde probíhá dopředné řízení, se také říká otevřený regulační obvod.

Vliv poruchové veličiny dokáže odstranit zpětnovazební řízení.



Obrázek 1.2 - Zpětnovazební řízení

U zapojení se zápornou zpětnou vazbou regulátor nastavuje akční veličinu  $u$  na základě informace o žádané veličině  $w$  a skutečné hodnotě regulované veličiny  $y$ . Rozdíl mezi žádanou a regulovanou veličinou se nazývá regulační odchylka  $e$ . Regulátor se tedy snaží minimalizovat regulační odchylku vhodným nastavením akční veličiny, kde platí

$$e = w - y, \quad (1.1)$$

kde  $e$  je regulační odchylka,  
 $w$  je žádaná hodnota,  
 $y$  je regulovaná veličina.

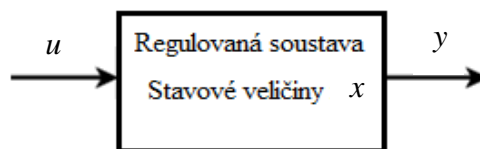
Zpětná vazba nám umožňuje dosažení požadovaného průběhu výstupní veličiny i bez přesné znalosti parametrů soustavy. Zde se využívá záporná zpětná vazba, protože kladná zpětná vazba neodstraňuje regulační odchylku, ale naopak způsobuje nestabilitu regulačního obvodu (Balátě, 2003).

## 1.1 REGULOVANÁ SOUSTAVA

Vlastní technologický proces, který je řízen regulátorem, probíhá v regulované soustavě. Jedná se o řízený systém, který je součástí regulačního obvodu.

U těchto systémů se rozlišují různé vstupní nebo výstupní veličiny v závislosti na interakci s okolím. Mezi vstupní veličiny se zařazují veličiny poruchové a akční. Poruchové veličiny mohou vstupovat do regulovaného systému, ale nejsou ovlivnitelné. To působí na regulovaný systém negativně, protože odklání výsledek procesu od požadovaného cíle řízení. Akční veličiny jsou takové, kterými je řízena regulovaná soustava. Tyto veličiny jsou výstupem regulátoru a vstupem pro regulační soustavu (Balátě, 2003).

Výstupní veličiny systému jsou takové, které nesou nějaký informační obsah a tím působí na svoje okolí. Podstatná je řízená regulovaná veličina  $y$ , která vystupuje z regulované soustavy a nese informace o výsledku řízení. Regulovaná soustava obsahuje také stavové



Obrázek 1.3 - Regulovaná soustava

veličiny, které jsou známe při stavovém popisu. Měřené stavové veličiny odpovídají výstupním veličinám.

Podle povahy veličin lze rozdělit regulované systémy na systémy spojité a diskrétní. Spojitý systém je takový, který má všechny veličiny spojité. Mezi spojitými veličinami patří také čas. Diskrétní systém nemá hodnoty veličin spojité a to ani čas. Diskrétní čas lze chápat jako posloupnost kroků.

Naprostá většina řízených systémů je spojitá, ale pokud je potřeba řízení průmyslovým počítačem, je potřeba nalézt diskrétní model spojitého systému. Diskrétní model by měl v místě vzorkování co nejvíce odpovídat spojitému systému.

## 1.2 POPIS REGULOVANÝCH SOUSTAV

Tato práce se zabývá lineárním dynamickým systémem. Dynamické nebo statické vlastnosti systému lze získat z experimentálních dat naměřených na reálném systému. Pro popis lineárního dynamického systému se z pravidla používá vnější nebo vnitřní popis. V našem případě je důležitý vnější popis systému, který vyjadřuje dynamické vlastnosti mezi vstupem a výstupem. Dynamické vlastnosti systému vyjadřují jeho chování.

Podle počtu vstupních a výstupních veličin se dynamické systémy dělí na SISO systém (Single Input, Single Output), tedy systém pouze s jedním vstupem a jedním výstupem. Druhým typem je systém MIMO (Multi Input, Multi Output), který je tvořen vektory vstupů i výstupů, takže soustava obsahuje více vstupů i výstupů.

Dynamický systém typu SISO je možné popsat různými způsoby. Zde jsou vypsány pouze ty nejdůležitější, s kterými se dále pracuje. Mezi způsoby popisu řadíme diferenciální rovnice a spojitý přenos soustavy, diferenční rovnice a diskrétní přenos pro diskrétní popis

soustavy, dále sem řadíme frekvenční přenos a frekvenční charakteristiky, přechodovou a impulsní charakteristiku, póly a nuly přenosu (Balátě, 2003).

### 1.2.1 Diferenciální rovnice

Vztah mezi vstupem  $u(t)$  a výstupem  $y(t)$  lineárního spojitého dynamického systému lze v časové oblasti popsat jednou lineární diferenciální rovnicí  $n$ -tého řádu s konstantními koeficienty

$$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \dots + a_1y'(t) + a_0y(t) = b_mu^{(m)}(t) + \dots + b_1u'(t) + b_0u(t), \quad (1.2)$$

kde  $a_i, b_j$  jsou konstantní koeficienty,

$u(t)$  je vstupní veličina,

$y(t)$  je výstupní veličina a

$n$  je řád diferenciální rovnice.

Pro správné řešení diferenciální rovnice je nutné znát počáteční podmínky systému a vstupní veličiny

$$y^{(i)}(0) = y_0^{(i)}, \quad i = 0, 1, \dots, n-1, \quad u^{(j)}(0) = u_0^{(j)}, \quad j = 0, 1, \dots, m-1. \quad (1.3)$$

Pro řády derivací na levé a pravé straně rovnice popisující fyzikálně realizovatelný systém musí platit podmínka

$$m \leq n. \quad (1.4)$$

To znamená, že stupeň nejvyšší derivace výstupní veličiny musí být větší nebo roven nejvyššímu stupni derivace vstupní veličiny. Tuto podmínku splňují všechny reálné fyzikální systémy. V případě nesplnění by výstupní veličina reagovala na změnu vstupního signálu dříve než změna nastala – neplatil by princip kausality. To v reálných systémech není možné. Řád diferenciální rovnice  $n$  určuje také řád systému (Balátě, 2003).

### 1.2.2 Diferenční rovnice

Diferenční rovnice popisuje stejně jako spojitý přenos vlastnosti dynamického systému ovšem s tím rozdílem, že pro uvažované hodnoty signálu vzorkované pouze ve vybraných časových okamžicích, tedy diskrétních časech  $t_k$ ,

$$t_k = kT + t_0, \quad (1.5)$$

kde  $T$  je perioda vzorkování,

$k$  je násobek periody vzorkování,

$t_0$  je počáteční časový okamžik začátku vzorkování.

Diferenční rovnice vyjadřuje vztah mezi diskretní výstupní veličinou  $y(kT)$  a diskretní vstupní veličinou  $u(kT)$ . U diskretních systémů je pro analýzu a syntézu systémů řízení použita Z transformace namísto Laplaceovy transformace jako je tomu u systémů spojitých. Z transformace je diskretní Laplaceova transformace, kde vztah komplexních proměnných je ve vztahu podle vzorce

$$z = e^{sT}, \quad (1.6)$$

kde  $z$  je komplexní proměnná Z transformace,

$s$  je komplexní proměnná Laplaceovy transformace a

$T$  je perioda vzorkování.

Pro sestavení diferenční rovnice je možné použít dva způsoby definování diferencí. Jedná se buď o dopřednou nebo zpětnou diferenci. Zpětná diference je na rozdíl od dopředné fyzikálně realizovatelná. Diferenční rovnice s použitím dopředné difference má rekurentní tvar

$$a_n y[(k+n)T] + \dots + a_1 y[(k+1)T] + a_0 y(kT) = b_m u[(k+m)T] + \dots + b_1 u[(k+1)T] + b_0 u(kT) \quad (1.7)$$

a s danými počátečními podmínkami

$$y(0), y(T), \dots, y[(n-1)T], u(0), u(T), \dots, u[(m-1)T]. \quad (1.8)$$

Diferenční rovnice s použitím zpětné difference je ve tvaru

$$\begin{aligned} a_n y(kT) + a_{n-1} y[(k-1)T] + \dots + a_1 y[(k-n+1)T] + a_0 y[(k-n)T] = \\ = b_m u(kT) + b_{m-1} u[(k-1)T] + \dots + b_1 u[(k-m+1)T] + b_0 u[(k-m)T] \end{aligned} \quad (1.9)$$

I pro tento tvar platí stejné počáteční podmínky, jako jsou uvedeny v (1.8).

### 1.2.3 Spojitá přenosová funkce

Spojitý přenos je definován jako podíl Laplaceových obrazů výstupu ku Laplaceovým obrazům vstupu při nulových počátečních podmínkách. Přenos je podíl dvou polynomů komplexní proměnné  $s$  a získá se Laplaceovou transformací diferenciální rovnice (1.2),

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b(s)}{a(s)} = \frac{b_m s^m + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}, \quad (1.10)$$

kde  $a_i, b_j$  jsou konstantní koeficienty,

$s$  je komplexní nezávisle proměnná.

Lineární diferenciální rovnice po provedení Laplaceovy transformace při splnění podmínek (1.3) má tvar

$$(a_n s^n + \dots + a_1 s + a_0)Y(s) = (b_m s^m + \dots + b_1 s + b_0)U(s), \quad (1.11)$$

kde  $Y(s)$  je Laplaceův obraz výstupní veličiny,

$U(s)$  je Laplaceův obraz vstupní veličiny.

Polynom  $a(s)$  ve jmenovateli přenosu se nazývá charakteristický polynom. Pokud je roven nule, je nazván jako charakteristická rovnice přenosu systému. Kořeny této charakteristické rovnice se nazývají póly přenosu. Naopak kořeny získané z polynomu  $b(s)$ , který je také položen nule, se nazývají nuly přenosu.

Pokud kořeny čitatele přenosu  $b(s)$  jsou označeny jako  $c_j$ , kde  $j = 1, \dots, m$  a kořeny charakteristické rovnice  $a(s)$  jako  $s_i$ , kde  $i = 1, \dots, n$ , pak je přenos vyjádřen ve tvaru

$$G(s) = \frac{b_m (s - c_1)(s - c_2) \dots (s - c_m)}{(s - s_1)(s - s_2) \dots (s - s_n)}. \quad (1.12)$$

Jestliže všechny póly i nuly přenosu jsou reálné, pak se zavádějí časové konstanty jako záporné převrácené hodnoty právě pólů a nul a přenos lze vyjádřit s časovými konstantami ve tvaru

$$G(s) = \frac{b_0 (\tau_{c1}s + 1)(\tau_{c2}s + 1) \dots (\tau_{cm}s + 1)}{a_0 (\tau_{1s} + 1)(\tau_{2s} + 1) \dots (\tau_{ns} + 1)}, \quad (1.13)$$

kde  $\tau_{cm}$  jsou časové konstanty čitatele,

$\tau_n$  jsou časové konstanty jmenovatele.

Zesílení systému, pokud existuje, je vyjádřeno jako poměr koeficientů čitatele  $b_0$  a jmenovatele  $a_0$  (Balátě, 2003),

$$Z_s = \frac{b_0}{a_0}. \quad (1.14)$$

#### 1.2.4 Diskrétní přenosová funkce

Diskrétní přenos je definován jako podíl z obrazu výstupní diskrétní veličiny ku z obrazu vstupní diskrétní veličiny při nulových počátečních podmínkách. Je to podíl dvou polynomů komplexní proměnné  $z$  a získá se Z transformací diferenční rovnice (1.7), která musí být převedena na algebraický tvar. Přenos má pak tvar

$$G(z) = \frac{Y(z)}{U(z)} = \frac{M(z)}{N(z)} = \frac{b_m z^m + \dots + b_1 z + b_0}{a_n z^n + \dots + a_1 z + a_0}, \quad (1.15)$$

kde  $a_i, b_j$  jsou konstantní koeficienty,

$z$  je komplexní proměnná,

$Y(z)$  je  $z$  obraz diskretní výstupní veličiny,

$U(z)$  je  $z$  obraz diskretní vstupní veličiny.

Algebraický tvar diferenční rovnice po provedení Z transformace má tvar

$$(a_n z^n + \dots + a_1 z + a_0)Y(z) - L(z) = (b_m z^m + \dots + b_1 z + b_0)U(z) - R(z), \quad (1.16)$$

kde  $L(z)$  je mnohočlen  $n$ -tého stupně určený počátečními podmínkami levé strany diferenční rovnice,

$R(z)$  je mnohočlen  $n$ -tého stupně určený počátečními podmínkami pravé strany diferenční rovnice.

Jmenovatel diskretního přenosu je charakteristický mnohočlen diskretního lineárního dynamického systému

$$N(z) = a_n z^n + \dots + a_1 z + a_0 = a_n (z - z_1)(z - z_2) \dots (z - z_n), \quad (1.17)$$

kde  $z_1, z_2, \dots, z_n$  jsou kořeny charakteristické rovnice a tedy póly přenosu.

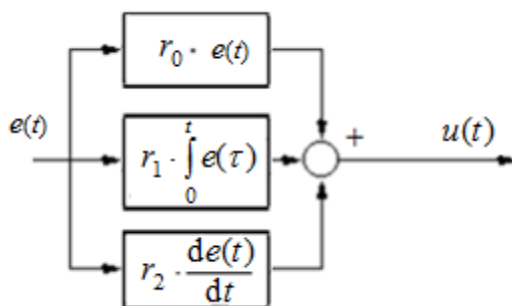
Modifikovaná verze diskretního přenosu je vyjádřena v záporných mocninách komplexní proměnné a má tvar (Balátě, 2003)

$$G(z) = \frac{Y(z)}{U(z)} = \frac{(b_m + b_{m-1} z^{-1} + \dots + b_1 z^{-m+1} + b_0 z^{-m}) z^{-n+m}}{a_n + a_{n-1} z^{-1} + \dots + a_1 z^{-n+1} + a_0 z^{-n}}. \quad (1.18)$$

### 1.3 REGULÁTOR

Regulátor je zařízení v regulačním obvodu, které slouží k řízení regulované soustavy za účelem snížení regulační odchylky podle vzorce (1.1). Regulační odchylka je pro regulátor ve formě vstupního signálu a výstupním signálem je akční veličina, kterou nastaví regulátor tak, aby regulační odchylka bylo co nejmenší nebo úplně eliminována. Je tedy požadováno dosažení výstupní regulované veličiny na hodnotu žádané veličiny a to nezávisle na působení vnějších poruchových veličin.

Nejpoužívanější typ regulátorů pro průmyslové využití je spojitý tzv. proporcionálně-integračně-derivační regulátor neboli PID. Hlavní výhodou tohoto regulátoru je jeho jednoduchost a použitelnost i pro složité nelineární systémy. Ovšem kvalita řízení těchto systémů není zaručena. V mnoha případech se používá pouze proporcionální a integrační složka regulátoru, zatímco derivační složka se nenastavuje. Lze ho realizovat mechanicky nebo elektronickými obvody (Kořínek, 2015).



Obrázek 1.4 – Blokové schéma PID regulátoru

Paralelní tvar PID regulátoru lze zapsat ve tvaru

$$u(t) = r_0 \cdot e(t) + r_1 \cdot \int_0^t e(\tau) d\tau + r_2 \cdot \frac{de(t)}{dt}, \quad (1.19)$$

kde  $r_0$  označuje proporcionální složku,

$r_1$  je integrační složka,

$r_2$  je derivační složka

$e(t)$  je regulační odchylka.

V praxi se častěji pracuje s tvarem, který místo parametrů  $r_i$ , obsahuje časové konstanty. Jejich hodnoty a význam odpovídají fyzikálnímu chování regulátoru. Parametry pro určení časových konstant je nejlepší určit z naměřené přechodové charakteristiky, kde se určí zesílení soustavy a časová konstanta  $T_s$ , jako doba, za kterou výstup dosáhne 63 % své ustálené hodnoty. Integrační časová konstanta se zvolí stejná, jako naměřená časová konstanta soustavy  $T_s$ . Zmenšením zesílení bude výsledná odezva na skokovou změnu žádané hodnoty bez překmitu. Zesílení naopak průběh zrychlí, ale sníží tlumení.

Standardní tvar PID regulátoru s časovými konstantami je

$$u(t) = r \cdot \left[ e(t) + \frac{1}{T_I} \cdot \int_0^t e(\tau) d\tau + T_D \cdot \frac{de(t)}{dt} \right]. \quad (1.20)$$

A následný přenos regulátoru je v tomto případě

$$R(s) = r \cdot \left(1 + \frac{1}{T_I \cdot s} + T_D \cdot s\right), \quad (1.21)$$

kde  $r$  je zesílení soustavy,

$T_I$  je integrační časová konstanta,

$T_D$  je derivační časová konstanta.

PID regulátor tedy obsahuje tři složky. Proporcionální složku (P), která určuje reakci regulátoru na regulační odchylku. Představuje jednoduchý zesilovač, který zesiluje zápornou zpětnou vazbu. Zvyšování zesílení sice zmenšuje regulační odchylku, ale při dosažení mezní hodnoty zesílení se regulovaná soustava stává nestabilní. Proto při použití samostatného P regulátoru vzniká malá trvalá regulační odchylka.

Integrační složka (I) umožňuje dosáhnout nulové regulační odchylky. Regulační pochod se tedy ustálí, pokud je regulační odchylka nulová. Ale zároveň se regulační pochod protahuje, což je nežádoucí pro regulované soustavy, které je potřeba ustálit v krátkém čase.

Derivační složka (D) urychluje regulační pochod i pro soustavy vyššího řádu nebo soustavy s dopravním zpožděním. Samostatná derivační složka je pouze idealizovaná, protože výstupem skokové změny na vstupu by měl být Diracův impuls, který je technicky nerealizovatelný. Proto se vždy používá jako součást PD nebo PID regulátoru. Nevýhodou derivační složky je zesilování šumu působícího na měření (Kořínek, 2015).

## 1.4 PSD REGULÁTOR

Tento typ regulátoru odpovídá standardnímu tvaru spojitého PID regulátoru přepočítanému do diskretní oblasti. Při odvozování rovnic diskretního PID regulátoru se vychází z rovnic spojitého PID regulátoru s přenosem (1.20). Je třeba provést diskretní aproximaci spojitě integrace a derivace pro určitý interval vzorkování  $T$ . Pro náhradu integrálu bylo použito zpětné obdélníkové metody ZOBD.

Pro návrh regulátoru je důležitý výběr správného typu regulátoru a nastavení jeho parametrů. Pro PSD regulátor bude využit modifikovaný přírůstkový tvar regulátoru, který má klidnější průběh při změně žádané hodnoty, oproti klasickému přírůstkovému tvaru. Modifikovaný přírůstkový tvar je

$$u(k) = u(k-1) + q \cdot w(k) - q_0 \cdot y(k) - q_1 \cdot y(k-1) - q_2 \cdot y(k-2), \quad (1.22)$$

kde  $u(k)$  je hodnota akčního zásahu regulátoru,

$w(k)$  je žádaná hodnota,  
 $y(k)$  je hodnota regulované veličiny,  
 $q_i$  jsou parametry PSD regulátoru.

Parametry PSD regulátoru ve tvaru  $q_i$  jsou snadno získatelné přepočtem ze spojitých parametrů PID regulátoru  $T_I$ ,  $T_D$  a  $r_0$ . Parametry diskrétního regulátoru tedy odpovídají spojitému podle velikosti zvolené periody vzorkování. Výpočet parametrů diskrétního regulátoru ze spojitých parametrů je uveden v tabulce 1.1 (Balátě, 2003).

Tabulka 1.1 – Přepočet parametrů regulátoru

|     | $q_0$  | $q_1$  | $q_2$                     | $q$                   |
|-----|--|--|---------------------------|-----------------------|
| PSD | $r_0 \cdot \left(1 + \frac{T_D}{T} + \frac{T}{T_I}\right)$ | $-r_0 \cdot \left(1 + \frac{2T_D}{T}\right)$ | $r_0 \cdot \frac{T_D}{T}$ | $q = q_0 + q_1 + q_2$ |

## 1.5 METODY NASTAVENÍ REGULÁTORŮ

Praktický problém průmyslové regulace je nekvalitní nastavování a seřizování parametrů regulátoru. Nastavení regulátoru je závislé na požadavku na průběh regulované veličiny, např. zda je potřeba kmitavý tlumený průběh, kdy se regulační pochod ustálí v co nejkratší době, ale obvykle za cenu překmitu. Nebo zda je potřeba získat aperiodický děj, který netrpí překmitem, ale doba ustálení regulačního pochodu je delší.

Mezi nejpoužívanější metody návrhu parametrů regulátoru patří metoda Zieglera a Nicholse. Existují dvě varianty metody a obě vycházejí ze zjednodušeného popisu regulované soustavy. První vychází z popisu soustavy pomocí kritických hodnot, tj. kritického zesílení a kritické periody. Do uzavřeného regulačního obvodu je vložen P regulátor do zpětné vazby a jeho zesílení je postupně zvyšováno, dokud se celý obvod nedostane na mez stability a objeví se netlumené kmity s konstantní amplitudou. Frekvence kmitů je rovna kritické periodě  $T_k$  a zesílení P regulátoru je rovno kritickému zesílení  $r_{0k}$  (Hlava, 2000).

Tuto metodu je možné modifikovat, kdy místo P regulátoru je použito relé ve zpětné vazbě. U relé je třeba nastavit hodnotu  $\pm R$ , která bude realizovat spínání a rozepínání relé. Tím budou na výstupu soustavy stabilní kmity, kde jejich perioda bude rovna kritické periodě  $T_k$  a kritické zesílení  $r_{0k}$  se spočte podle vzorce

$$r_{0k} = \frac{4 \cdot M}{\pi \cdot A}, \quad (1.23)$$

kde  $M$  je amplituda na výstupu relé,  
 $A$  je maximální amplituda vstupního signálu.

Druhá metoda vychází ze změřené přechodové charakteristiky regulované soustavy, kdy může být nepřípustné uvést obvod na mez stability. Metoda je založena na vyhodnocení přechodové charakteristiky v okolí pracovního bodu. Pro obě metody existují vzorce pro nastavení parametrů z měřených a odečtených hodnot. Tabulky vzorců pro výpočet parametrů regulátoru jsou uvedeny v literatuře (Hlava, 2000).

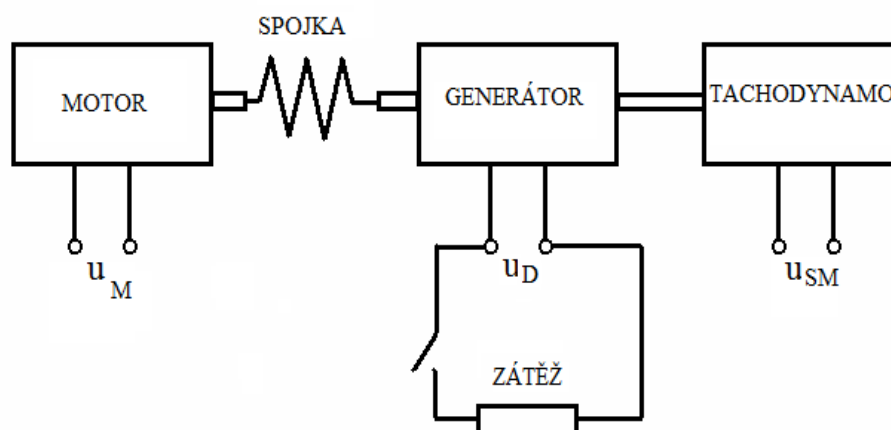
## 1.6 PERIODA VZORKOVÁNÍ

Perioda vzorkování je důležitý faktor při návrhu jakéhokoliv diskrétního regulátoru, tj. i při návrhu PSD regulátoru. Při velkém intervalu vzorkování může docházet až k nestabilnímu chování PSD regulátoru, ačkoliv jeho parametry byly určeny na základě parametrů stabilního spojitého PID regulátoru. Velikost vzorkovací periody má tedy podstatný vliv na stabilitu diskrétního regulačního obvodu (Balátě, 2003).

## 2 SOUSTAVA MOTOR GENERÁTOR

Tato práce se zabývá řízením soustavy tvořené stejnosměrným motorem spojeným pevnou spojkou se zátěží tvořenou generátorem. Na společnou osu je připojeno tachodynamo, které umožňuje měřit otáčky. Otáčky se dají snižovat na základě zátěže generátoru. Tím vznikne porucha působící na soustavu, která zvětší brzdný moment generátoru. Tento brzdný moment je úměrný proudu odebíranému z generátoru a vyvolá snížení otáček generátoru.

Zátěž může pracovat ve třech režimech podle kombinace připojení obou svorek ke generátoru. V prvním režimu při aktivních svorkách je možno měnit velikost zátěže manuálně ve vytvořeném programu. Ve druhém režimu, kdy je aktivní první z kontaktů, odpovídá zátěž odporu 50 % z maxima a v posledním režimu, kdy je aktivní druhý kontakt, je odpor zátěže maximální a odpovídá tedy 100 %. Je doporučeno zátěž nepoužívat příliš dlouho, protože může dojít k velkému zahřátí soustavy (Modrlák, 2010).

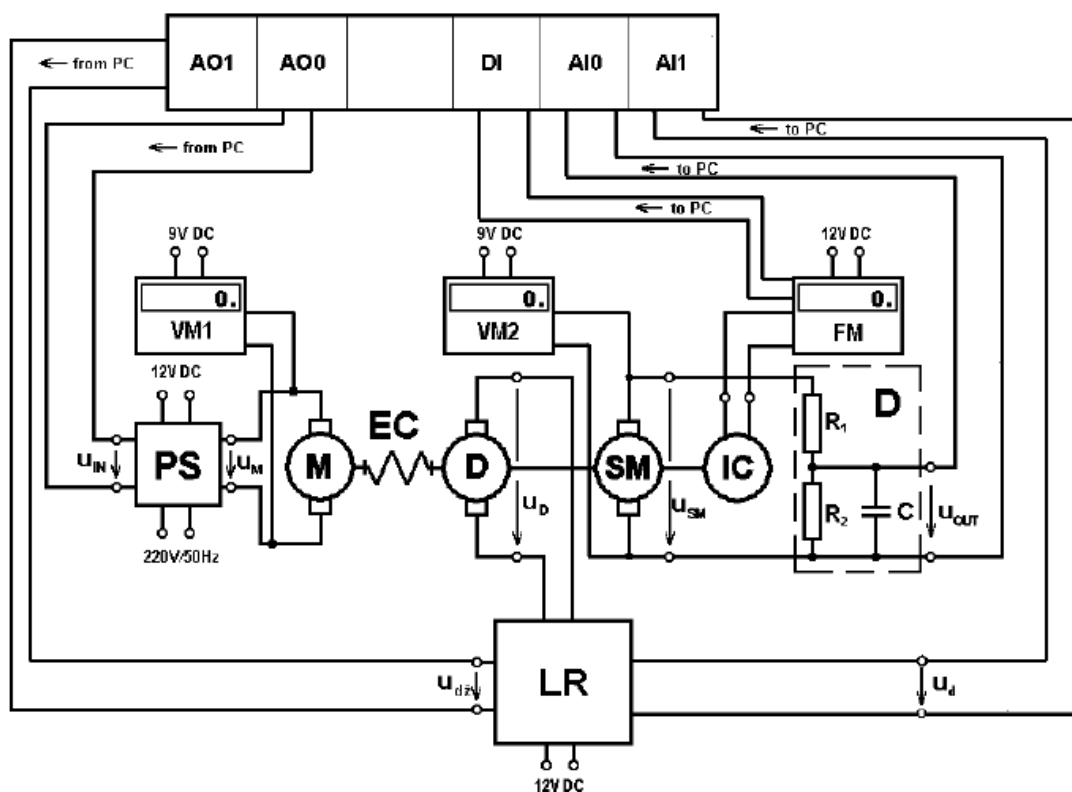


Obrázek 2.1 – Soustava motor generátor

Vstupy a výstupy soustavy jsou vyvedeny na šroubovací část svorkovnice ukončené konektorem na plochý 23 pinový kabel. Protože svorkovnice pro připojení signálů k měřicímu řídicímu systému AMiNi4DS jsou typu WAGO. Byl vytvořen mechanický mezikus, který slouží pro jednoduché spojení s řídicím systémem. Mechanický mezikus je vytvořen, aby nebylo nutné mechanické uchycování propojovacích vodičů pod šroubovací část svorkovnice zařízení.

Na obrázku 2.2 je zobrazeno technologické schéma zařízení motor generátor. Schéma znázorňuje reálné propojení bloků soustavy a připojení šroubovací části svorkovnice. Šipky zobrazují, zda se jedná o vstup do soustavy nebo výstup soustavy na svorkovnici a dále do

řídícího systému. Detailní popis částí soustavy a veškeré použité signály jsou uvedeny v literatuře (Modrlák, 2010).



Obrázek 2.2 – Technologické schéma soustavy

kde PS je zdroj DC motoru řízený vstupním napětím,  
M je DC motor s permanentními magnety,  
EC je pevná spojka motoru a generátoru,  
D je generátor tvořený DC motorem,  
SM je tachodynamo,  
LR je elektronická zátěž,  
IC je inkrementální čidlo, nezapojeno,  
VM jsou připojené voltmetry,  
AI0, AI1 jsou analogové výstupy systému,  
DI je digitální výstup systému,  
AO0, AI1 jsou analogové vstupy soustavy.

Použité signály soustavy podle obrázku 2.2 jsou vypsány v tabulce 2.1, kde je vysvětlen také význam jednotlivých signálů.

Tabulka 2.1 - Popis signálů soustavy

| Signál    | Význam                                     | Rozsah       |
|-----------|--|--------------|
| $U_{IN}$  | řídící napětí otáček motoru                | 0 V ÷ 10 V   |
| $U_{dž}$  | řídící napětí nastavující proud el. zátěží | 0 V ÷ 10 V   |
| $U_M$     | napětí na svorkách DC motoru               | 0 V ÷ 28,8 V |
| $U_D$     | napětí na svorkách DC generátoru           |              |
| $U_{SM}$  | napětí na svorkách tachodynamu             |              |
| $U_{OUT}$ | měřené výstupní napětí otáček motoru       | 0 V ÷ 10 V   |
| $U_d$     | napětí odpovídající proudu el. zátěže      | 0 V ÷ 10 V   |

### **3 METODY IDENTIFIKACE SYSTÉMŮ**

Identifikace soustavy slouží pro získání popisu chování – modelu, obvykle ve formě matematického modelu. Je to činnost, při které se určují parametry a struktura modelu. Matematický model dynamického systému je soustava obyčejných diferenciálních nebo diferenčních rovnic popisující vztahy mezi zvolenými veličinami. Matematický model systému je vhodný pro simulace na místo experimentování přímo na reálném systému. Je možnost napodobovat různé situace chování reálného systému a ověřit je na matematickém modelu. Vytvoření přesného matematického modelu je obtížné, ale stačí provést takovou identifikaci systému, aby vytvořený model umožnil analýzu statických i dynamických vlastností reálného systému.

Zkoumaný systém lze identifikovat analyticky, čemuž odpovídá matematicko-fyzikální analýza nebo empiricky, kam patří experimentální identifikace.

#### **3.1 MATEMATICKO-FYZIKÁLNÍ ANALÝZA**

Tento analytický postup vychází z materiálových a energetických bilancí a z konstrukčních, technologických a provozních údajů o daném systému. Podle fyzikálních nebo chemických zákonů jsou popisovány jevy, které probíhají v systému. Získané parametry modelu mají fyzikální význam a model platí v celém rozsahu pro různé provozní stavy. Analytická identifikace je náročná nejen časově, ale také vzhledem k matematickému popisu systému. Čím přesnější je požadavek na výsledný model, tím složitější je provedení analýzy i následné používání. Stavy systému a vnitřní procesy musí být známy a matematicky popsitelné. Výsledný nelineární model je složitý a je nutností použít různé zjednodušující předpoklady pro získání jednoduššího modelu. Některé dílčí procesy se naproti tomu mohou zanedbat. Výhodou je souvislost mezi parametry výsledného modelu a konstrukčními parametry systému a dynamickými vlastnostmi.

#### **3.2 EXPERIMENTÁLNÍ IDENTIFIKACE**

Experimentální identifikace vychází z naměřených experimentálních dat na daném systému. Je tedy nutná existence zkoumaného systému pro naměření experimentálních dat. Výsledný model je poté popsán pouze na základě vztahu mezi vstupem a výstupem a vnitřní relace systému jsou nepodstatné. Proto není potřeba znát přesnou strukturu identifikovaného systému. Problémem může být správná volba struktury modelu a zjištěné parametry

odpovídající této struktuře, protože zvolená struktura nemá fyzikální základ. Nalezené parametry nemají většinou fyzikální význam.

V této práci se pracuje se soustavou motor generátor, jehož identifikace je provedena na základě experimentálních dat. Analytický způsob je vzhledem k složitosti matematického popisu celé soustavy složitý a časově velmi náročný (Balátě, 2003).

### 3.3 VSTUPNÍ TESTOVACÍ SIGNÁL

Pro získání měřených dat použitelných pro experimentální identifikaci musí být použit vhodný průběh vstupního signálu. Teoreticky ideální vybuzující signál je bílý šum, který je ale fyzikálně nerealizovatelný. Dobrou aproximací je využití generátoru náhodných čísel. Zde by bylo možné generovat náhodně čísla v daném omezení hodnot. Jelikož použitý řídicí systém neobsahuje generátor náhodných čísel, lze využít pseudonáhodný binární signál (PNBS).

Pseudonáhodný signál se od bílého šumu odlišuje hlavně tím, že jde o deterministický a periodicky se opakující signál nabývající pouze dvou úrovní. Pseudonáhodný signál má stejně jako bílý šum konstantní výkonovou spektrální hustotu v určitém intervalu frekvencí.

Binární pseudonáhodný signál je takový signál, který nabývá pouze dvou hodnot. K přechodu mezi těmito hodnotami dochází v časových krocích, které odpovídají intervalu vzorkování  $T$ . Pseudonáhodný binární signál se periodicky opakuje v čase  $NT$ , kde  $N$  označuje periodu pseudonáhodného binárního signálu. Ovšem posloupnost pseudonáhodného binárního signálu lze realizovat pouze pro určité periody vzorkování  $N$ . První způsob generování této posloupnosti je s periodou

$$N = 4K - 1, \quad (3.1)$$

kde  $K$  je celé číslo.

Druhý způsob, který je využit i v této práci pracuje s periodou

$$N = 2^n - 1, \quad (3.2)$$

kde  $n$  je celé číslo.

Jelikož je výsledná hodnota periody vždy liché číslo, je střední hodnota posloupnosti pseudonáhodného binárního signálu nenulová (Dušek, 2015).

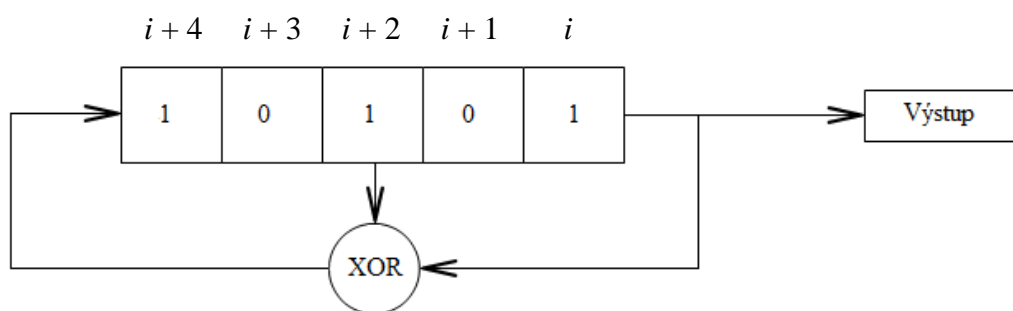
Generování posloupnosti tohoto signálu se nejčastěji provádí pomocí posuvných registrů se zpětnou vazbou. To znamená, že generátor je tvořen posuvným registrem, který obsahuje  $n$  hodnot, které mají velikost 1 nebo 0. Počtem bitů  $n$  je dána maximální perioda a

odpovídající charakteristický polynom. Podle tvaru charakteristického polynomu je generátor složitější, protože je zapotřebí více posuvných registrů, čím je vyšší polynom a tedy i zvolený počet bitů  $n$ . Sumátor modulo 2 odpovídá logické funkci XOR.

Tabulka 3.1 - Funkce XOR

| $b$ | $a$ | XOR |
|-----|-----|-----|
| 0   | 0   | 0   |
| 0   | 1   | 1   |
| 1   | 0   | 1   |
| 1   | 1   | 0   |

Výsledná hodnota funkce XOR je rovna jedné, pokud hodnoty proměnných  $a$ ,  $b$  jsou různé.



Obrázek 3.1 - Princip posuvných registrů pro PNBS

Tabulka 3.2 zobrazuje souvislost se zvoleným počtem bitů  $n$ , výsledné hodnotě maximální periody vzorkování a tvaru charakteristického polynomu. V této práci je použit polynom 5. stupně, to znamená, že perioda pseudonáhodného binárního signálu je tvořena 31 hodnotami. Tvorba posloupnosti hodnot je zobrazena na obrázku 3.1 (Dušek, 2015).

Tabulka 3.2 - Primitivní polynomy pro XOR

| Počet bitů $n$ | Charakteristický polynom    | Perioda: $2^n - 1$ |
|----------------|-----------------------------|--------------------|
| 2              | $x^2 + x + 1$               | 3                  |
| 3              | $x^3 + x^2 + 1$             | 7                  |
| 4              | $x^4 + x^3 + 1$             | 15                 |
| 5              | $x^5 + x^3 + 1$             | 31                 |
| 6              | $x^6 + x^5 + 1$             | 63                 |
| 7              | $x^7 + x^6 + 1$             | 127                |
| 8              | $x^8 + x^6 + x^5 + x^4 + 1$ | 255                |

### 3.4 VÝBĚR VYHODNOCOVACÍ METODY

Popis časové závislosti vzorků výstupního signálu na vzorcích vstupního bude ve tvaru lineární diferenční rovnice

$$y(k) + a_1 y(k-1) + \dots + a_{n_a} y(k-n_a) = b_1 u(k-1) + \dots + b_{n_b} u(k-n_b) \quad (3.3)$$

kde  $y(k)$  je zkrácené označení hodnoty signálu v čase  $t_k = t_0 + kT$ ,

$T$  je interval vzorkování.

Předmětem identifikace systému je odhad určení řádu diferenční rovnice a jejích koeficientů  $a_i, b_j$ .

K odhadu parametrů lineární diferenční rovnice se nejčastěji používají metody regresní analýzy. Metody pracují na principu minimalizování kritéria shody mezi experimentálně získanými daty a výstupem modelu. Jako kritérium se ve většině případů používá suma kvadrátů odchylek (MŇČ). Odhady parametrů jsou počítány tak, aby hodnota kritéria byla co nejmenší.

#### 3.4.1 Metoda nejmenších čtverců

Metoda nejmenších čtverců (MŇČ) je matematicko-statistická metoda používaná mimo jiné pro řešení soustav přeurčených rovnic. To znamená, že soustava má více rovnic, než je neznámých parametrů. Je možné využít vektorového a maticového zápisu, který umožňuje jednoduchý zápis složitých výrazů a zjednodušuje zápis algoritmů. Její výhodou je, že za splnění předpokladů (model lineární v parametrech, aditivní náhodný šum s nulovou střední hodnotou) poskytuje nevychýlené odhady určovaných parametrů.

Touto metodou budou určeny takové neznámé parametry  $a_i, b_i$  diferenční rovnice, minimalizující hodnotu kritéria (3.3), tj. suma kvadrátů odchylek naměřených a aproximovaných hodnot

$$K = \sum_{i=1}^n (\hat{y}_i - y_i)^2 \approx \min, \quad (3.4)$$

kde  $n$  je počet naměřených bodů,

$\hat{y}_i$  jsou aproximované hodnoty,

$y_i$  jsou naměřená experimentální data.

Pro nalezení minimální hodnoty kritéria  $K$  se vypočítají parciální derivace tohoto kritéria podle počtu neznámých parametrů  $a_i$ ,  $b_i$ , které se položí rovny nule, pak tedy získáme parciální derivace ve tvaru

$$\frac{\partial K}{\partial a_i} = \dots = 0, \frac{\partial K}{\partial b_i} = \dots = 0, \quad (3.4)$$

$$\text{kde} \quad i = 0, 1, \dots, n.$$

Potom získáme řešení minima kvadrátů odchylek jako soustavu normálních  $n + 1$  rovnic o  $n + 1$  neznámých (Vrožina, 2012).

### 3.5 URČENÍ PARAMETRŮ

Z naměřených dat použitím metody nejmenších čtverců získáme nejlepší odhad parametrů  $a_i$ ,  $b_i$ , pro diferenční rovnici (1.7) zvoleného řádu.

Pro převod do maticového tvaru se diferenční rovnice přepíše tak, aby na levé straně byly jen aktuálně měřené hodnoty  $y(k)$  a na pravé straně převeden zbytek rovnice obsahující starší vzorky. Tím získáme soustavu rovnic

$$\begin{aligned}
k = n + 1: \quad & y(n + 1) = -a_1 \cdot y(na) - \dots - a_{na} \cdot y(1) + b_1 \cdot u(nb) + \dots + b_{nb} \cdot u(1) \\
k = n + 2: \quad & y(n + 2) = -a_1 \cdot y(na + 1) - \dots - a_{na} \cdot y(2) + b_1 \cdot u(nb + 1) + \dots + b_{nb} \cdot u(2) \\
& \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\
k = N: \quad & y(N) = -a_1 \cdot y(N - 1) - \dots - a_{na} \cdot y(N - na) + b_1 \cdot u(N - 1) + \dots + b_{nb} \cdot u(N - nb)
\end{aligned} \tag{3.6}$$

kde  $N$  je počet měření,

$n$  je celkový počet parametrů,

$na$  je počet hledaných parametrů  $a$ ,

$nb$  je počet hledaných parametrů  $b$ .

Pro vytváření této soustavy rovnic je potřeba použít první aktuální hodnotu až od rovnice  $n + 1$ , protože teprve až pro tento vzorek je známa potřebná historie.

Maticový zápis pak je ve tvaru

$$\begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} -y(n) & \dots & -y(1) & u(n) & \dots & u(1) \\ -y(n+1) & \dots & -y(2) & u(n+1) & \dots & u(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y(N-1) & \dots & -y(N-n) & u(N-1) \dots u(N-n) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ \vdots \\ a_{na} \\ b_1 \\ \vdots \\ b_{nb} \end{bmatrix}. \quad (3.7)$$

Tato matice lze zapsat jako

$$\mathbf{Y}_N = \mathbf{F} \cdot \mathbf{P}, \quad (3.8)$$

kde  $\mathbf{Y}_N$  je vektor naměřených výstupních hodnot soustavy,

$\mathbf{F}$  je známá matice naměřených hodnot,

$\mathbf{P}$  je hledaný vektor neznámých parametrů soustavy.

Potom se pro řešení této rovnice pro regulární matici dá využít inverzní matice, kde výsledný neznámý vektor parametrů se vypočítá podle vzorce

$$\begin{aligned} \mathbf{F}^{-1} \cdot \mathbf{Y}_N &= \mathbf{F}^{-1} \cdot \mathbf{F} \cdot \mathbf{P} \\ \mathbf{P} &= \mathbf{F}^{-1} \cdot \mathbf{Y}_N. \end{aligned} \quad (3.9)$$

Řešení pro singulární matici, tedy když se jedná o přeurčený systém, je výsledné řešení složitější

$$\begin{aligned} \mathbf{F}^T \cdot \mathbf{Y}_N &= \mathbf{F}^T \cdot \mathbf{F} \cdot \mathbf{P} \\ (\mathbf{F}^T \cdot \mathbf{F})^{-1} \cdot \mathbf{F}^T \cdot \mathbf{Y}_N &= (\mathbf{F}^T \cdot \mathbf{F})^{-1} \cdot \mathbf{F}^T \cdot \mathbf{F} \cdot \mathbf{P} \\ (\mathbf{F}^T \cdot \mathbf{F})^{-1} \cdot \mathbf{F}^T \cdot \mathbf{Y}_N &= \mathbf{I} \cdot \mathbf{P} \\ \mathbf{P} &= (\mathbf{F}^T \cdot \mathbf{F})^{-1} \cdot \mathbf{F}^T \cdot \mathbf{Y}_N \end{aligned} \quad (3.10)$$

Stejný výpočetní vztah dostaneme i při maticovém zápisu kvadratického kritéria a následném standardním řešením MNČ pomocí parciálních derivací dle jednotlivých parametrů (Dušek, 2015).

## 4 ŘÍDICÍ SYSTÉM

Programovatelný logický automat je jednoúčelový počítač vybavený prostředky na měření a generování signálů, který se používá při řízení procesů v reálném čase. Práce v reálném čase je také hlavní rozdíl od klasických stolních počítačů. Kromě řízení výrobních linek různých továren se používá i pro řízení menších strojů, jako jsou kotle nebo spotřební elektronika. V současné době je využit pro řízení inteligentních domů, kde může řídit například topení, osvětlení nebo sloužit pro ovládání spotřebičů.

Další důležitou vlastností a odlišností od stolních počítačů je zpracování programu cyklicky a také množství vstupů a výstupů uzpůsobených pro připojení technologických procesů. Existuje velké množství PLC, které se liší počtem periférií a výkonem procesoru. Většina periférií je tvořena digitálními vstupy a digitálními výstupy. Pro zpracování spojitých signálů obsahuje také analogové vstupy a analogové výstupy. Pro zvýšení počtu periférií se k samotnému PLC dají přidávat různé moduly obsahující další periférie, komunikační procesory nebo moduly určené pro sběr a přenos dat.

V závislosti na možnosti rozšíření o různé typy modulů se PLC dělí na modulární nebo kompaktní systémy (Honc, 2016).

### 4.1 AMiNi4DS

AMiNi4DS je programovatelný automat z rodiny malý kompaktní řídicí systém firmy AMiT. Patří mezi nejuniverzálnější volně programovatelné logické automaty pro komplexní autonomní řízení a ovládání malých systémů.



Obrázek 4.1 – Řídicí systém AMiNi4DS

V tabulce 4.1 jsou zobrazeny důležité technické parametry řídicího systému (AMiT, 2009).

Tabulka 4.1 - Technické parametry

| CPU                   |                                |
|-----------------------|--------------------------------|
| CPU                   | ST10F269                       |
| FLASH                 | 256 + 1024 kB                  |
| RAM                   | 1024 kB                        |
| EEPROM                | 2 kB                           |
| Zálohování RAM, RTC   | Lithiová baterie,<br>Panasonic |
| Životnost baterie     | 5 let                          |
| RTC (Real Time Clock) |                                |
| Typ                   | RTC72423A                      |
| Přesnost 25 °C        | ±20 ppm                        |
| Displej               |                                |
| Displej               | Grafický 122 × 32 bodů         |
| Podsvětlení           | Zelené LED                     |
| Klávesnice            | 8 tlačítek                     |
| Napájení              |                                |
| Napájení              | 24 V DC, ±20%                  |
| Odběr                 | Max. 200 mA při 24 V DC        |

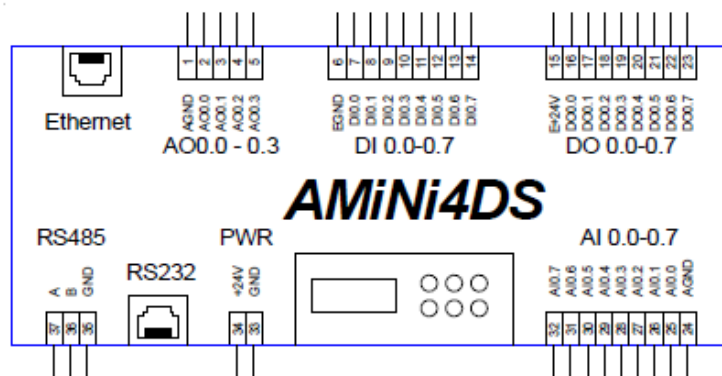
#### 4.1.1 Vstupy a Výstupy AMiNi4DS

Mezi základní periferie řídicího systému patří 8 číslicových vstupů. Číslicové vstupy jsou galvanicky odděleny a jejich označení na přístroji je od DI 0.0 do DI 0.7. Dále obsahuje 8 číslicových výstupů, které jsou také galvanicky odděleny. Jejich přístrojové označení je od DO 0.0 do DO 0.7. Číslicové výstupy je možné použít také jako frekvenční a impulsní výstupy. To znamená, že dokáží generovat požadovaný počet pulsů nebo požadované frekvenci řádově do kilohertzů.

Stejně jako digitálních vstupů obsahuje AMiNi4DS také stejný počet analogových vstupů. Konfigurace těchto vstupů je možná pro napěťové (od 0 V do 10 V) a proudové (od 4 mA do 20 mA) unifikované rozsahy nebo pro přímé připojení odporových čidel Ni1000 a Pt1000 pomocí konfiguračních propojek. V závislosti na zpracovávaném signálu lze analogové vstupy používat i jako vstupy digitální. Záleží na programu vyhodnocování. Přístrojové označení je AI 0.0 do AI 0.3 a AI 0.4 do AI 0.7. Dále obsahuje 4 napěťové analogové výstupy jejichž výstupní rozsah je od 0 V do 10 V. Označení je AO 0.0 do AO 0.3.

Systém dále obsahuje 8 tlačítek pro ovládání a černobílý LCD displej, který zobrazuje obrazovku naprogramovanou v aplikaci.

Napájení tohoto řídicího systému je možné pouze ze stejnosměrného zdroje. Napájecí napětí je 24 V. Uspořádání periférií je zobrazeno na schématické značce (AMiT, 2009a).



Obrázek 4.2 – Schématická značka pro AMiNi4DS

#### 4.1.2 Komunikační rozhraní

Dále řídicí systém obsahuje 3 komunikační linky. Sériové rozhraní RS232 bez galvanického oddělení a galvanicky oddělené sériové rozhraní RS485 a rozhraní Ethernet.

Rozhraní RS232 je převážně určeno pro spojení dvou zařízení. Pouze pomocí tohoto rozhraní lze do řídicího systému AMiNi4DS nahrát operační systém NOS.

Rozhraní RS485 je průmyslové poloduplexní sériové rozhraní. Lze použít pro spojení až 32 zařízení.

Rozhraní Ethernet slouží pro připojení řídicího systému do počítačové sítě. Ethernet lze použít i pro nahrání programů do systému přes internet. Ke komunikaci je použit protokol TCP/IP (AMiT, 2012).

Tabulka 4.2 - Technické parametry Ethernetu

|                     |                   |
|---------------------|-------------------|
| Přenosová rychlost  | 10 Mbps           |
| Použitý radič       | RTL8019AS         |
| Indikace funkce     | LED na panelu     |
| Galvanické oddělení | Ano               |
| Izolační pevnost    | 300 V AC/1 minuta |
| Přípojné místo      | Konektor RJ45     |

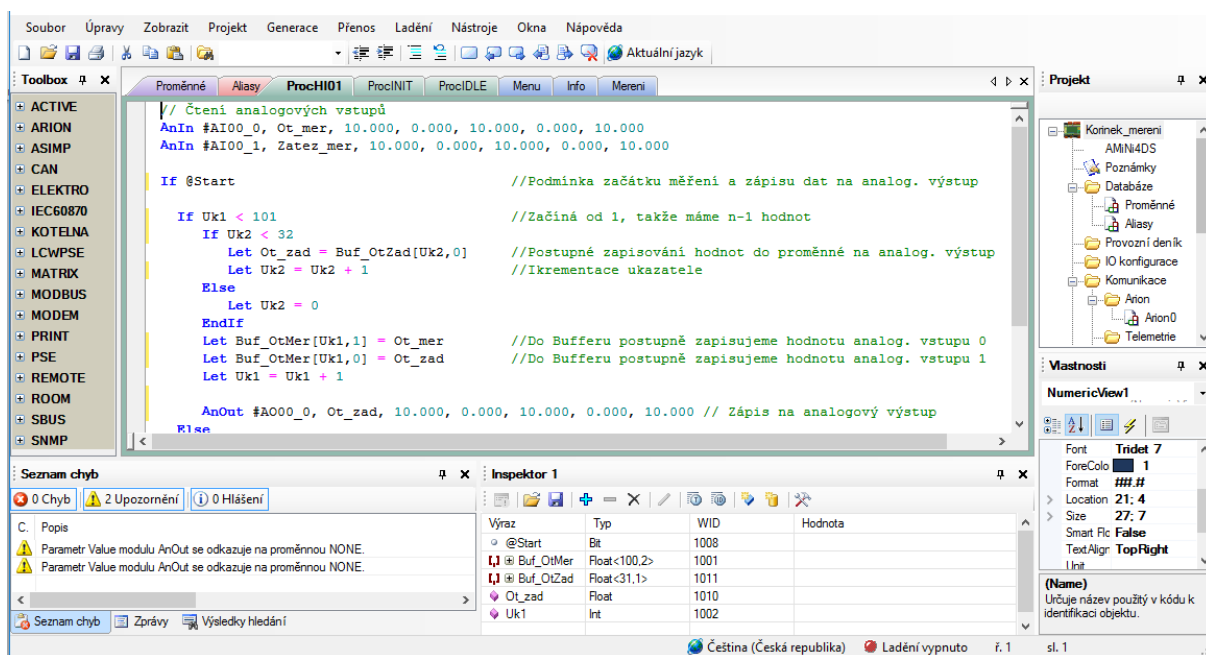
## 5 PROGRAMOVÉ VYBAVENÍ

V této práci je ve větší míře využito návrhové prostředí pro ovládání řídicího systému firmy AMiT a výpočtový software pro identifikaci regulované soustavy MATLAB. Je zde využít také tabulkový procesor EXCEL s možností využití maker a programovacím jazykem VisualBasic.

### 5.1 DETSTUDIO

DetStudio je návrhové prostředí, ve kterém se vytvářejí aplikace (programy) pro řídicí systémy firmy AMiT. Toto prostředí má podporu českého jazyka a je distribuované zdarma na webových stránkách firmy AMiT. Prostedí se skládá z několika nástrojů, kde jejich kombinace dovoluje vytvářet aplikace a také ovládací a informační rozhraní člověk – stroj. Pro vybrané systémy podporuje nastavení obrazovek podle přání uživatele (AMiT, 2011).

V prostředí se pracuje vytvořením projektů, ve kterých se otevře pracovní okno, kde se odehrává většina programových a editačních činností při tvorbě aplikace. Hlavním nástrojem pracovního okna je editor procesů, ve kterém se do jednotlivých procesů zapisují příkazy zvoleného programovacího jazyka. Tím se vytvoří algoritmická část aplikace. DetStudio obsahuje také nástroj pro editaci obrazovek, kde se vytváří grafické prvky a pomocí proměnných se spojují s algoritmickou částí aplikace v editoru procesů.



Obrázek 5.1 – Návrhové prostředí DetStudio

Nejdůležitější částí návrhového prostředí je vpravo umístěno okno Projekt, které obsahuje seřazení ve stromové struktuře a zobrazuje základní položky projektu. Struktura a zobrazované položky jsou závislé na typu zvoleného řídicího systému.

Důležitou položkou projektu je databáze. Databáze obsahuje dvě položky, a to proměnné a aliasy. Ve správci proměnných je vytvořena tabulková databáze, ve které se musí vytvořit proměnné podle funkce, kterou má program vykonávat. Proměnná může být typu integer (celé číslo 16 bitů), long (celé číslo 32 bitů) a float (reálné číslo 32 bitů). Další tři typy jsou matice těchto jednoduchých typů proměnných. U proměnné se nastavuje jméno, WID (jednoznačný identifikátor proměnné v síti DB-Net), datový typ, inicializační hodnota (pokud je potřeba), číslo stanice a komentář. WID proměnné by neměl být editován uživatelem, je řešeno automaticky návrhovým systémem. Alias slouží pro přístup k proměnným jako k šestnácti nebo třiceti dvěma hodnotám 0/1. Jedná se o přístup k jednotlivým bitům dané celočíselné proměnné. Aliasy mohou nabývat pouze binárních hodnot.

Další položkou projektu jsou procesy. Zde jsou zobrazeny vytvořené procesy v daném projektu a v programovacím jazyce a tyto procesy jsou na sobě nezávislé a probíhají postupně podle zvoleného typu procesu. Důležitou vlastností procesů je schopnost jednotlivých procesů přerušovat chod ostatních, což se provádí na základě určení jejich typu a jejich priority. Každý proces má své jedinečné jméno a existuje několik typů procesů, které se liší podle času periody vykonávání a priority, kdy se který proces má vykonat (AMiT, 2011).

Základní proces je typu Normal a je nejčastěji využíván pro regulaci nebo měření na běžných technologických procesech. Perioda procesu Normal je nastavitelná od 100 ms až do  $10^6$  s.

Mezi procesy s vyšší prioritou patří proces Quick a HiSpeed. Tyto procesy mohou přerušovat normální procesy a je potřeba na to dávat pozor při stavbě aplikace. Proces Quick je speciální a slouží k obsluze rychlých dějů v procesní stanici. Procesy HiSpeed jsou dva a od procesu Quick se liší volbou periody, která je v rozmezí od 1 ms do 1677 ms. Lze vytvořit ještě proces typu Interrupt, ve kterém vytvořený program na základě vnější události, dokáže vyvolat přerušení.

Po zapnutí obsluhované stanice je spuštěn proces Init, který je spuštěn pouze jednou ihned po startu stanice. Zde se umísťují inicializační sekvence, kterými se zajišťují počáteční podmínky běhu aplikace. Poslední možný je proces Idle, který je spuštěn, pouze když si jiný proces nenárokuje čas na procesoru stanice. Zde se uvádí příkaz pro obsluhu obrazovek.

Samotný proces je možno psát v pracovním okně programovacím jazykem ST (strukturovaný text), LA (jazyk logických adres) nebo RS (grafický jazyk reléových schémat).

Programová část této práce je napsána pouze ve strukturovaném textu, ve kterém se pracuje pomocí vkládání modulů. V každém modulu je různý počet proměnných. Většina používaných modulů je typu funkčních modulů, kde se pracuje se signály vstupu nebo výstupu technologického procesu a zvolenou proměnnou. Další skupinou modulů je pseudojazyk, který obsahuje specializované funkční moduly pro podmíněné vykonávání funkčních modulů. Poslední skupinou je modul Let, který umožňuje zapsání logických nebo aritmetických funkcí.

Další položkou po procesech jsou podprogramy. Ty na rozdíl od procesů nejsou spouštěny periodicky, ale přímo z jednotlivých procesů. Poslední položkou jsou obrazovky. Zde se provádí tvorba i editace jednotlivých vytvořených obrazovek pro zobrazení na displej použité stanice.

Pod oknem Projekt se nachází okno Vlastnosti, kde jsou zobrazovány parametry modulů nebo vlastnosti procesů. Okno se mění podle právě vybraného objektu v programu.

V levé části se nachází okno ToolBox. Zde se zobrazuje seznam objektů, které můžeme použít do programu ve zvoleném procesu. Seznam objektů je seříděn podle možnosti použití.

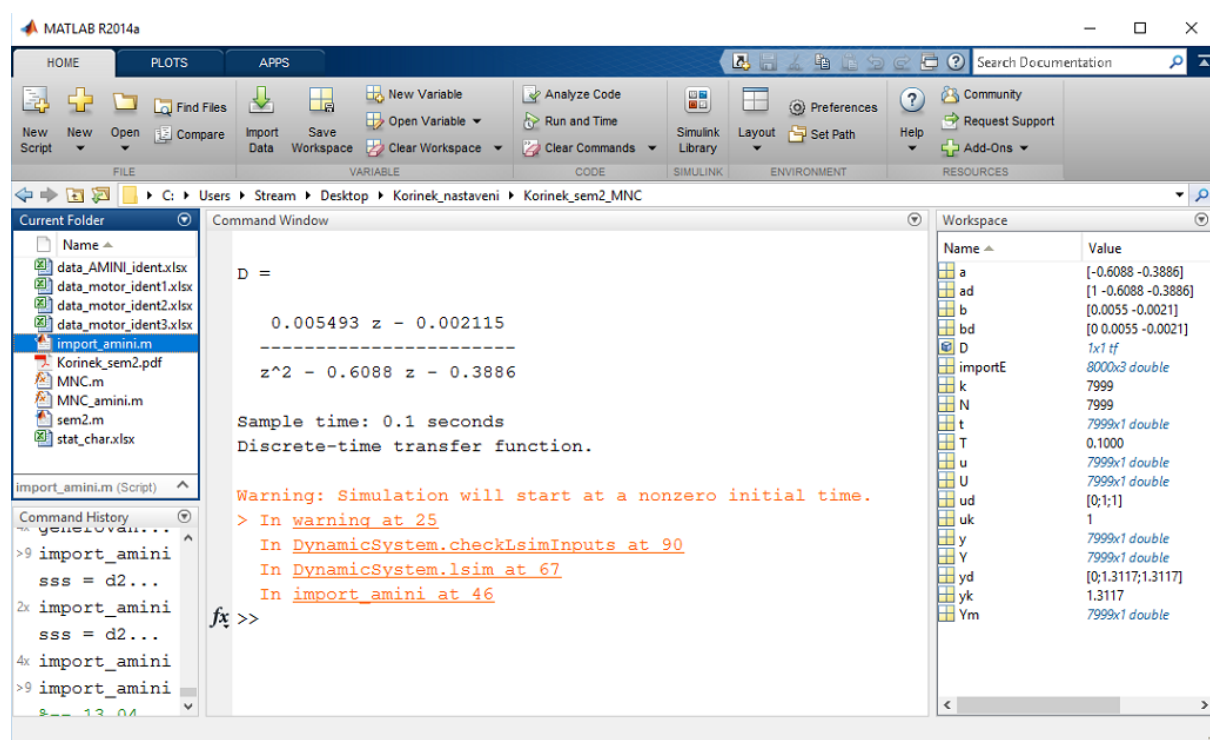
Dole v návrhovém prostředí se nachází okno se seznamem chyb. Okno slouží k zobrazení hlášení o chybách, o průběhu prováděných činností a slouží pro zobrazení výsledků při hledání v projektu.

Vedle seznamu chyb je možné přidat okno s inspektorem, který slouží pro sledování aktuálního stavu ve stanici a také pro ladění aplikace v reálném čase. Do tabulky je možné si zvolit proměnné, které chceme sledovat a je možnost zde měnit jejich stávající hodnoty.

## **5.2 MATLAB**

Matlab je výpočtový software, který je založen na práci převážně s maticemi. Může být použit pro vykreslení 2D i 3D grafů, počítačovou simulaci, modelování, návrhy řídicích nebo komunikačních systémů, analýzu dat nebo vytvoření uživatelského rozhraní. Možnosti Matlabu mohou být rozšířeny pomocí knihoven nazývaných toolboxy. Ty rozšiřují funkce Matlabu o další funkce v dané oblasti, pro kterou je toolbox určen. Dalším rozšířením Matlabu je nadstavba Simulink. Ten je určen pro simulaci a modelování dynamických systémů. Umožňuje vytvářet modely dynamických soustav ve formě blokových schémat a rovnic. Prostředí je objektově orientované, takže po vybrání objektu je možné zobrazit seznam metod, které lze u vybraného objektu použít (Dušek, 2005).

Po startu aplikace se objeví hlavní okno programu, které je rozděleno na čtyři důležitá podokna. Seskupení hlavního okna je zobrazeno na obrázku 5.2.



Obrázek 5.2 - Výpočtový software MATLAB

Hlavním zobrazeným oknem je Command Window. Zde se zapisují příkazy, které se mají vykonat a vypisují se zde také výsledky v textové podobě. V pravé části je okno Workspace, kde se zobrazují použité proměnné v aktuálním pracovním prostoru a jejich velikost. V levém horním rohu je okno Current Folder, které obsahuje seznam souborů v aktuálně zvolené složce. Pod tímto oknem je okno Command History. Zde se vypisuje historie zapsaných příkazů.

Pro přehledné uspořádání zapsaných příkazů se používá soubor m.file. Zde se zapisují příkazy stejně jako do okna Command Window, ale jsou zde v přehlednější formě a dají se zde přidávat i komentáře (Dušek, 2005).

MATLAB bude využit pro výpočet parametrů diferenciální rovnice z experimentálních dat uložených v souboru.

### 5.3 EXCEL

Excel je tabulkový editor od firmy Microsoft. Převážně slouží k vytváření a úpravám tabulek, protože obsahuje řadu funkcí umožňujících různé výpočty a je možné na základě zadaných dat vytvářet přehledné grafy.

Pro složitější funkce lze v tabulkovém editoru využít programovací jazyk VBA (Visual Basic for Application). Ten vychází z klasického Visual Basic s modifikacemi pro práci s produkty MS Office. VBA slouží především ke zrychlení a k zautomatizování aplikací. Pro práci se složitějšími aplikacemi se pracuje v prostředí s využitím maker, což jsou aplikace napsané právě v jazyce VBA.

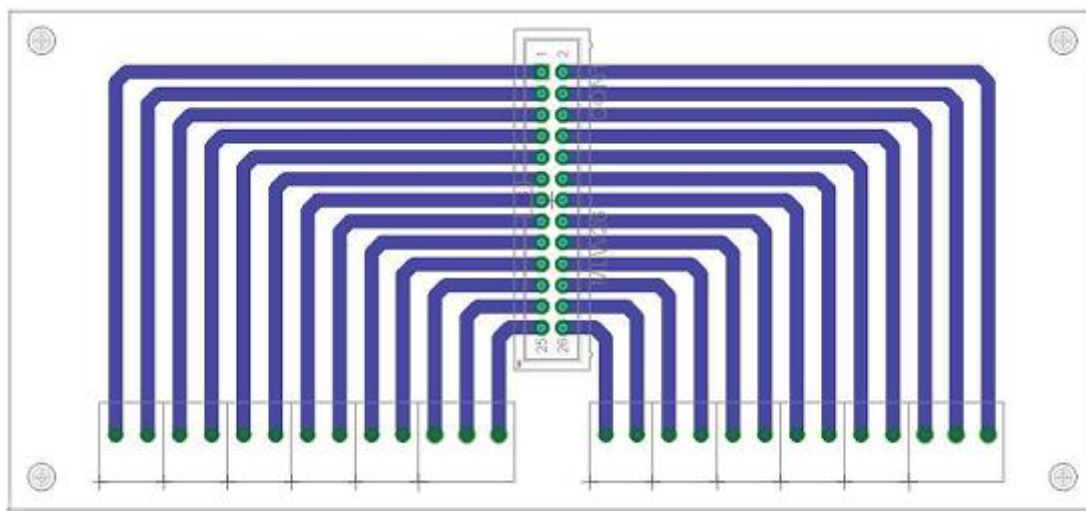
Excel, respektive funkce ve VBA volající služby ovladače ATOUCH (ActiveX), budou využity k přenosu naměřených dat z AMiNi4DS do listu Excelu.

## 6 PŘIPOJENÍ ŘÍDICÍHO SYSTÉMU

Základním úkolem pro připojení soustavy motor generátor k řídicímu systému AMiNi4DS je vytvoření mechanického převodníku. Převodník je použit z důvodu jiné koncovky řídicího systému, která je realizována jako konektor typu WAGO. Řízená soustava obsahuje pouze svorkovnici s konektorem pro 26 pinový plochý kabel, který je stabilně používán pro připojení motorů. Svorkovnice obsahuje pouze vyvedené drátové propojky.

Nejprve bylo provedeno ověření pinů svorkovnice, kde byly měřením nalezeny jednotlivé výstupy a vstupy soustavy a jím odpovídající výstup na svorkovnici. Pro úkol této práce bylo potřeba zjistit minimálně polohu pinu země, dvou vstupních napětí, zvlášť pro motor a generátor a polohu pinu pro výstupní napětí soustavy.

Samotný převodník je tedy realizován pouze jako jednoduchý DPS. Na jedné straně obsahuje konektor pro plochý kabel, který je spojen s odpovídajícími šroubovacími svorkovnicemi na straně druhé. Tento převodník je zobrazen na obrázku níže.



Obrázek 6.1 – Mechanický převodník

Po ověření jednotlivých pinů svorkovnice je tedy možné připojit motor generátor přes plochý kabel na zhotovený DPS, kde vyvedené drátové propojky jsou spojeny s WAGO konektorem, který je již možné připojit do řídicího systému.

Následující tabulka obsahuje identifikované piny svorkovnice a jím odpovídající parametry konektoru s napěťovou úrovní.

Tabulka 6.1 - Význam signálů svorkovnice motor generátor

| Význam pinu              | Piny převod. | Piny svorkovnice horní řada | Piny svorkovnice dolní řada | Piny převod. | Význam pinu             |
|--------------------------|--------------|-----------------------------|-----------------------------|--------------|-------------------------|
| Napětí otáčky, (0; 10) V | 1            | A0                          | 0V                          | 26           | Nezapojeno              |
| Nezapojeno               | 2            | /                           | I3                          | 25           | Nezapojeno              |
| Napětí zátěž, (0; 10) V  | 3            | A1                          | 0V                          | 24           |                         |
| Nezapojeno               | 4            | V+                          | I2                          | 23           | Nezapojeno              |
| Nezapojeno               | 5            | A2                          | 0V                          | 22           |                         |
| Signál zátěž, 0/5 V      | 6            | S0                          | I1                          | 21           | Nezapojeno              |
| Nezapojeno               | 7            | A3                          | 0V                          | 20           |                         |
| Signál zátěž, 0/5 V      | 8            | S1                          | I0                          | 19           | Nezapojeno              |
| Nezapojeno               | 9            | /                           | 5V                          | 18           | Nezapojeno              |
|                          | 10           | S2                          | 0V                          | 17           |                         |
| Nezapojeno               | 11           | VE                          | N1                          | 16           | Signál zátěž, (0; 10) V |
| Nezapojeno               | 12           | S3                          | 0V                          | 15           | GND                     |
|                          | 13           | 0V                          | N0                          | 14           | Signál motor, (0; 10) V |

## 6.1 NASTAVENÍ KOMUNIKACE

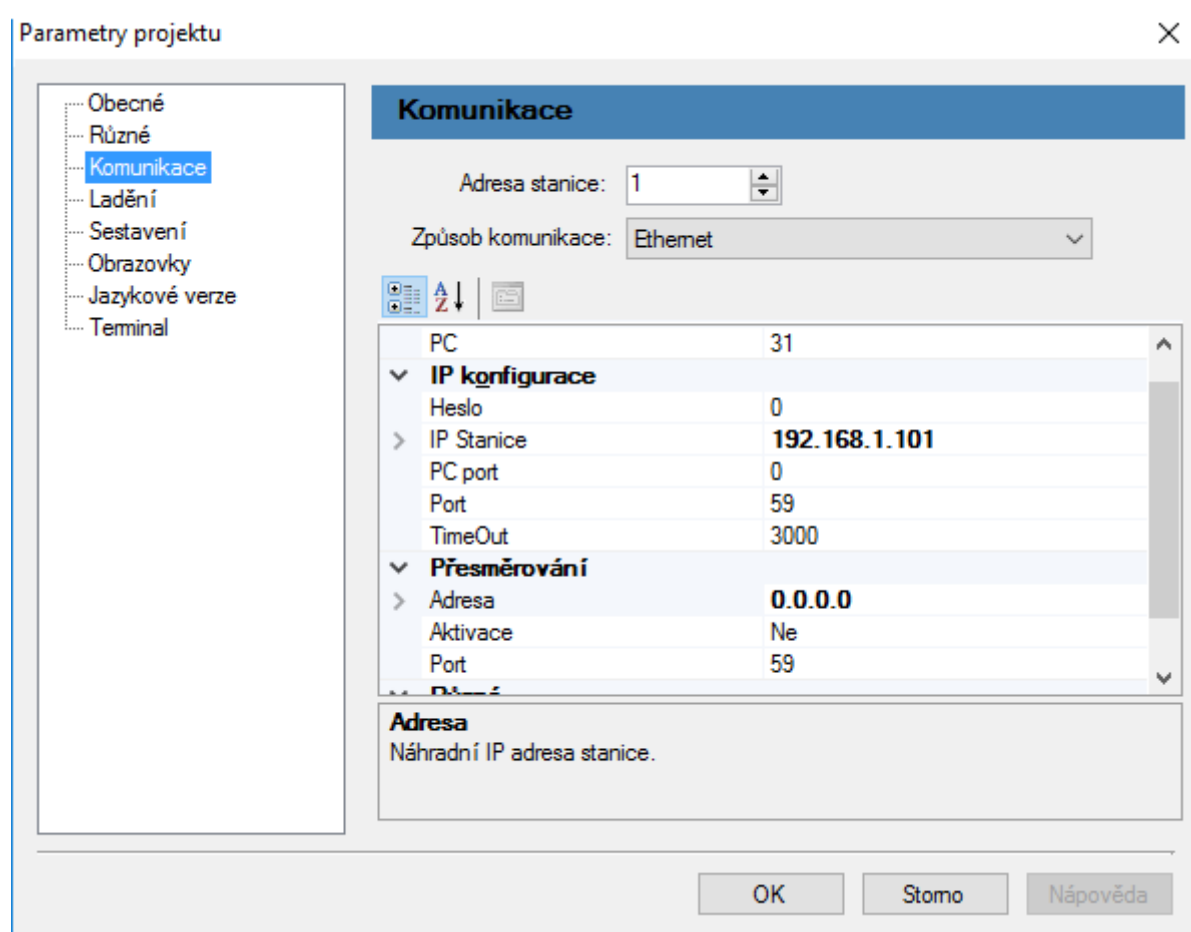
Řídicí systém obsahuje vlastní operační systém označovaný jako NOS. Verze operačního systému předpokládaného v DetStudiosu při vytváření programu by se měla shodovat s verzí v řídicím systému. Nahrání nejnovější verze se provádí pouze pomocí sériové linky RS232. Samotné nahrání se provede v prostředí DetStudio. Operační systém řídicího systému umožňuje spouštění aplikací vytvořených v DetStudiosu. Dále umožňuje komunikaci mezi řídicími systémy pomocí sériových linek nebo pomocí ethernetu.

Po nahrání operačního systému je potřeba nakonfigurovat používanou stanici. To lze provést přímo na řídicím systému pomocí příslušných tlačítek, kde se nastavení zobrazuje na displeji. Do nastavení se lze dostat držením tlačítka enter při připojení napájení. Vhodnější nastavení parametrů se provádí v DetStudiosu, kde je nastavení uživatelsky přehlednější.

Nastavení v DetStudiosu se provede při založení nového projektu, kde je potřeba nejprve vybrat typ řídicího systému, v tomto případě tedy AMiNi4DS. Otevřením okna nastavení a výběrem položky komunikace je možné nastavit komunikační parametry PC se stanicí. Změna konfigurace komunikace během chodu stanice nemá žádný vliv na chování, je tedy nutno po změně konfigurace stanici restartovat.

První položka co je potřeba nastavit, je adresa stanice, na které se bude pracovat. Pokud práce probíhá pouze na jedné stanici, je zpravidla nastavena hodnota jedna a nemusí se tedy editovat. Tato hodnota je důležitá pro komunikaci mezi řídicím systémem a PC s komunikačním ovladačem ATOUCH, který umožňuje zápis nebo čtení hodnot proměnných řídicího systému. Dále již je potřeba změnit způsob komunikace ze sériové linky na Ethernet. Výhodou ethernetového rozhraní je nahrávání programu do stanice vzdáleně.

Ethernetové rozhraní umožňuje přes PC nahrát program, provést identifikaci, získat data nebo ladit program nahraný ve stanici. Dále je v položce Adresace uvedena adresa PC. Přednastavená hodnota 31 udává poslední stanici v síti DB-Net, takže není nutné tuto hodnotu měnit. V položce IP konfigurace jediné co je potřeba modifikovat, je parametr IP stanice. Zde se nastavuje adresa procesní stanice, se kterou bude probíhat komunikace.



Obrázek 6.2 – Nastavení komunikace v DetStudio

Správné nastavení komunikačních parametrů lze následně ověřit pomocí funkce identifikace stanice.

## 6.2 KOMUNIKACE

Aby bylo možné v AMiNi4DS měřená data dále zpracovávat, je potřeba vybraná data exportovat. Řídicí systémy firmy AMiT umožňují vzájemnou komunikaci v proprietární síti DB-Net, která může být realizována přes libovolnou komunikační linku zařízení. Pokud se využije Ethernet, může být do sítě začleněno i standardní PC. Komunikace s dalšími stanicemi zajišťuje prostřednictvím sítě DB-Net nad Ethernetem na straně PC ovladač ATOUCH. Vybraná data jednotlivých stanic lze získat či zapsat voláním služeb ovladače ATOUCH.

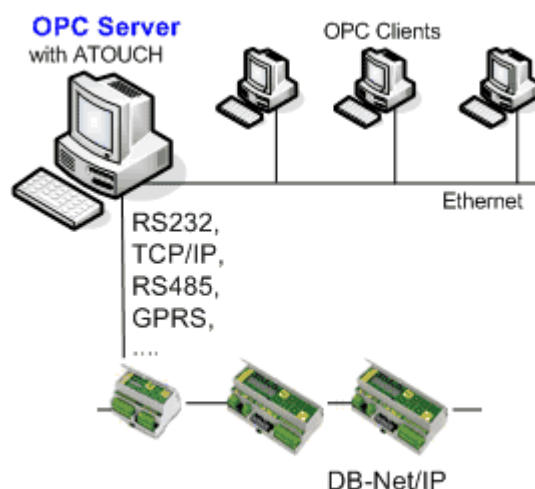
## 6.3 OPC SERVER

První možností komunikace s řídicím systémem je možnost využití OPC serveru. OPC server je software, který komunikuje s řídicími systémy pomocí jejich komunikačního protokolu a zajišťuje obousměrný přenos dat mezi řídicími systémy a vizualizačními systémy. Hlavním cílem OPC je vytvořit jednotné komunikační rozhraní mezi hardware a software průmyslové automatizace. Výhodou pro výrobce hardware je, že vystačí pouze s jedním souborem software, který má stále stejné ovladače, i když se změní vlastnosti hardware v nových verzích.

Spodní komunikační vrstva OPC serveru směrem k řídicímu systému je zajištěna komunikačním ovladačem ATOUCHX. Proto OPC server podporuje všechny typy komunikací, které jsou nabízeny ovladačem ATOUCH. Konfigurace je tvořena dvěma soubory s koncovkou ini. Jeden soubor udává hardware konfiguraci a druhý udává konfiguraci vybraných proměnných. Jsou zde podporovány všechny datové typy proměnných používané v řídicích systémech (INT, LONG, FLOAT a matice těchto typů). Frekvence čtení proměnných z řídicího systému je určena požadavky klienta OPC serveru. Při ztrátě komunikace se OPC server snaží průběžně znovunavázat ztracené spojení (AMiT, 2009b).

Nevýhodou tohoto způsobu komunikace při použití OPC serveru firmy Merz pro DB-Net/IP je, že data stanice ve formě matice zpřístupňuje pouze jako samostatné pojmenované proměnné (automaticky). Protože je potřeba získat při identifikačním experimentu velký počet dat, jsou ukládána do matice. To znamená značný problém pro další zpracování dat, protože by k datům byl obtížný přístup. Tisíce hodnot by se muselo vyčítat jednotlivě.

Na obrázku 6.3 je zobrazeno propojení klasického PC s řídicími systémy firmy AMiT přes OPC server Merz.



Obrázek 6.3 - OPC server (AMiT, 2009b)

## 6.4 KOMUNIKAČNÍ OVLADAČ ATOUCH

Tyto ovladače existují ve dvou verzích, Atouch32 a AtouchX. Atouch32 je určen pro použití z programovacího jazyka používající formát předávání parametrů jazyka C (stdcall). AtouchX je ActiveX objekt a jeho použití je možné ve Visual Basicu.

ATOUCHX je komunikační ovladač ve formě Active X objektu, který zajišťuje vazbu mezi řídicím systémem a zvolenou aplikací na osobním PC. Slouží pro přenos dat z řídicího systému a také pro zápis dat opačným směrem. V práci byly jeho služby volány z Excelu s využitím VBA pro přenos naměřených dat a pro start identifikačních či řídicích experimentů (AMiT, 2013).

V návrhovém prostředí existuje možnost export dat, která vytváří soubory s informacemi potřebnými pro komunikaci s ovladači Atouch32 nebo AtouchX. Po otevření položky export dat se zobrazí okno, kde jsou vypsány všechny použité proměnné v daném projektu. Zde se zaškrtnou zvolené proměnné, jejichž hodnoty bude možné vyčíst či zapsat a tím se vytvoří tři soubory s informacemi o datech v proměnných s koncovkou ini.

První soubor HW popisuje informace pro připojení. Jsou zde zobrazeny komunikační parametry využívané pro komunikaci se stanicí, jako je například IP adresa nebo číslo připojené stanice. Druhý soubor DB obsahuje popis vybraných proměnných. Je zde název

proměnné, identifikační číslo WID, typ proměnné, velikost a popis. Poslední soubor ARC obsahuje popis archivů a provozního deníku.

AtouchX zajišťuje vazbu mezi řídicími systémy a aplikacemi na PC. Obsahuje několik ActiveX objektů, které jsou určeny pro přístup k datům v řídicích systémech. Kromě základního přenosu dat mezi řídicím systémem a aplikací na PC umožňuje také zpětný přenos nebo detekci stavu stanice. Parametrizace komunikačního ovladače se provede pomocí dvou souborů s koncovkou ini, které se exportují z DetStudia. Pomocí těchto souborů jsou zadány proměnné řídicího systému, ze kterých chceme číst data a rozhraní o komunikaci s řídicím systémem. Práce s ovladačem AtouchX probíhá v prostředí Excel, kde je nutné mít aktivovanou práci s makry a musí být umožněno spouštění maker. V Excelu se s knihovnou modulů AtouchX pracuje ve spojení s programovacím jazykem Visual Basic.

Práce se všemi objekty ovladače AtouchX probíhá pomocí metod, protože objekty ovladače mají pouze funkcionální charakter. To znamená, že nemají téměř žádné vlastnosti. Všechny metody vracejí číslo chyby, které určuje důvod selhání metody. Všechny chyby pro daný modul jsou přehledně sepsány v tabulkách v nápovědě.

Objekty knihovny jsou dvojího druhu, podle složitosti využití a zkušenosti uživatelů. Pro méně zkušené se pracuje s objektem AtouchDir a pokročilejší uživatelé pracují s objekty AtouchApp. Pro tuto práci je využit jednodušší způsob práce s objekty, protože výsledný program neplní složitou funkci a není třeba používat složitější přístup. Hlavním rozdílem mezi oběma objekty je způsob vykonávání jednotlivých metod.

Objekt AtouchDir má připojení nastaveno jako direktivní nebo také synchronní. Při tomto způsobu uživatel vyvolá komunikační metodu a metoda zabezpečí přenos dat a počká na konec komunikace. To znamená, že metody tohoto objektu čekají na ukončení komunikace tak, že celý volající proces je uspán do doby, než komunikace skončí. Po ukončení komunikace je k dispozici údaj o průběhu komunikace a přenesená data.

Naopak objekt AtouchApp pracuje v asynchronním režimu. Zde je přenos dat výrazně pomalejší než počítač a probíhá proto na pozadí. Při použití tohoto způsobu je důležité dbát na synchronizaci komunikace. Uživatel vyvolá komunikační metodu, která pouze poznamená a vyšle požadavek na přenos dat, ale sám o sobě nic nevykoná. Metoda nečeká a vrací odpověď. Komunikační požadavek je vyřízen na pozadí, a jakmile je vyřízen, tak teprve potom je vyvolána událost. Ve vyvolané události je informace o výsledku komunikace a jsou zde k dispozici přečtená data.

Oba typy objektů se používají tak, že po vytvoření se nejdříve musí vyvolat některá z inicializačních metod pro inicializaci k síti podle daného popisu uloženého v souboru

HW.ini. Pokud inicializace nebude provedena správně, ostatní metody objektu nelze používat. Pro ukončení objektu se nejprve musí vyvolat metoda Done a až poté lze daný objekt zrušit. Tato metoda ukončí komunikaci se stanicí a uvolní všechny HW prostředky pro další práci. Metoda se musí vyvolat i v případě, že inicializační metoda dopadla neúspěšně s libovolnou chybovou hláškou (AMiT, 2013).

Oba objekty, jak AtouchDir tak i AtouchApp jsou na stejné úrovni z hlediska aplikace. Oba poskytují aplikaci své metody pro komunikaci se sítí a se stanicí. Liší se především způsobem komunikace.

## 7 PROGRAMOVÉ ŘEŠENÍ

Když je správně nastavena komunikace s řídicím systémem, může se přistoupit k samotnému psaní programového řešení.

Pro tvorbu aplikací na měření a identifikaci soustavy je důležité si nejprve rozvrhnout metodiku a správný postup řešení. K tvorbě postupu slouží vývojový diagram, kde se pomocí grafických značek různého tvaru, zapisuje funkce programu pro lepší přehlednost i pro snadné dodatečné úpravy. Vývojový diagram tedy znázorňuje průběh a stavbu programu.

Pro zadanou úlohu této práce je vhodné vytvořit vývojový diagram pro program na měření experimentálních dat z řídicího systému, který bude ukládat odezvu systému na pseudonáhodný binární signál na vstupu systému. Nejprve je tedy důležité si rozmyslet, jak bude program na měření dat pracovat, co vše by měl obsahovat pro správnou funkci. Pro přehlednost je vývojový diagram programu zobrazen na obrázku 7.1.

Po naměření dat je třeba proměnnou s daty uložit do vhodného formátu pro jednodušší zpracování. K tomu slouží komunikační ovladač AtouchX, který převede data z řídicího systému do zvoleného formátu pro práci s daty, například v Excelu. Zde je program pro přenos dat napsán v jazyce Visual Basic for Application.

V další části je již vytvořena samotná identifikace měřených dat získaných z řídicího systému a soustavy motor generátor. Data uložená v editoru Excel jsou převedena do výpočtového software Matlab, kde je vytvořen skript na načtení těchto dat a jejich identifikaci pro získání odpovídající diferenční rovnice.

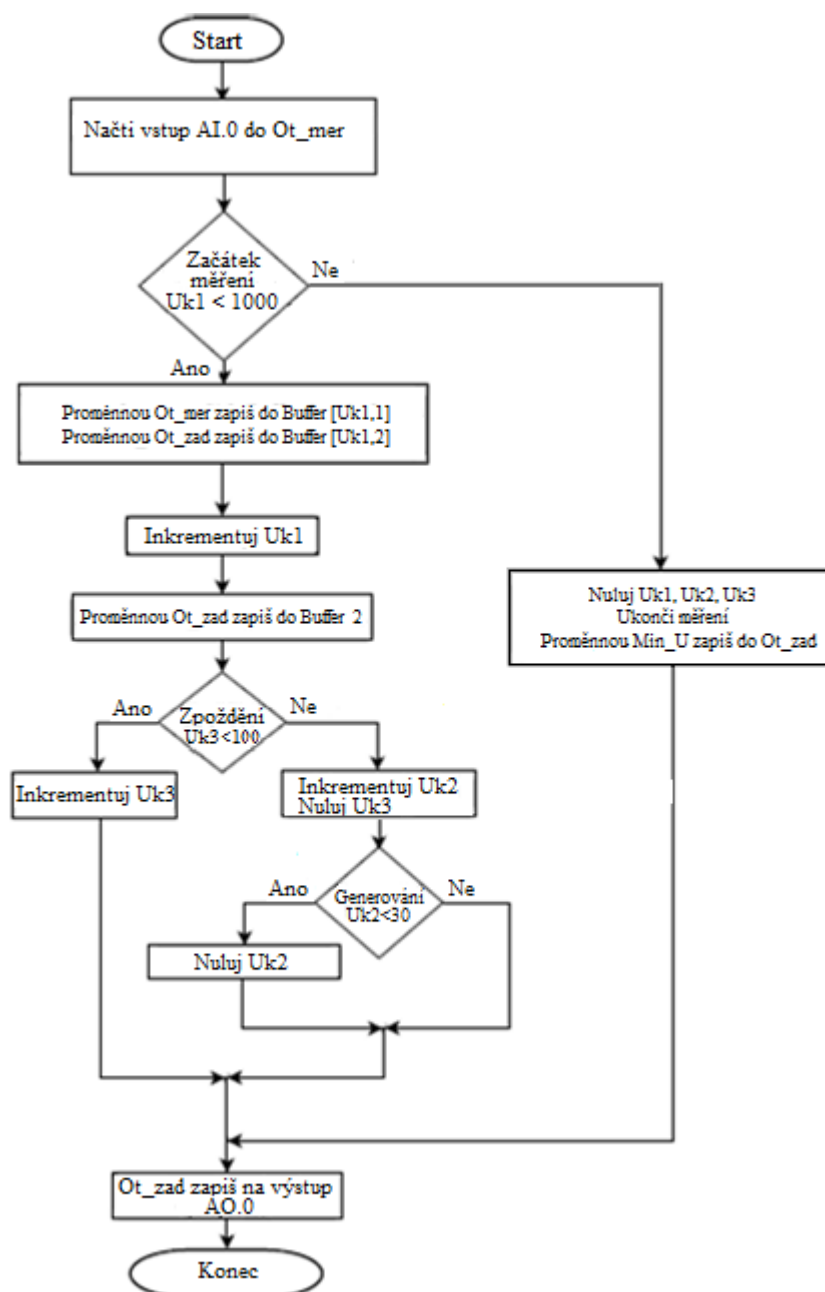
### 7.1 MĚŘENÍ EXPERIMENTÁLNÍCH DAT

Protože způsob realizace analogových výstupů kompaktního řídicího systému AMiNi4DS vykazuje při vzorkování 1 ms pozorovatelnou dynamiku, byla změřena data pro její vyhodnocení. Toto měření dovolí určit dynamické chování samostatného systému motor generátor bez ovlivnění dynamickými vlastnostmi řídicího systému.

Program je napsán v rychlém procesu HiSpeed s periodou opakování 1 ms. Ovšem základní interval generování PNBS 1 ms je příliš krátký i pro zjištění dynamiky řídicího systému AMiNi4DS. Proto je potřeba část programu generujícího PNBS rozšířit o smyčku se zpožděním. Perioda programu a měření signálů zůstává 1 ms, ale generovaná hodnota PNBS bude stejná po zvolený počet volání programu. Takže program poběží s periodou 1 ms, ale změna signálu pro určení dynamiky řídicího systému bude např. 10 ms. Tato doba již je

dostačující pro ustálení odezvy v nejdelším neměnném úseku PNBS a tedy i získání dat pro provedení identifikace.

Pro měření dat na soustavě motor generátor je potřeba zpoždění zvýšit, protože motor je pomalejší než řídicí systém a při prodloužení na 10 ms není dosaženo ustálení. Proto bylo prodloužení zvětšeno na hodnotu 100 a doba 100 ms základního intervalu PNBS je již dostačující.



Obrázek 7.1 - Vývojový diagram

### 7.1.1 Proces ProcInit

Proces je vykonávaný pouze jednou a to vždy jako první po teplém nebo studeném startu řídicího systému. Proces zjistí nastavenou IP adresu, hodnotu vstupního napětí řídicího systému a také při spuštění řídicího systému zapíše na výstup hodnotu, která odpovídá nejnižší hodnotě generovaného pseudonáhodného binárního signálu.

### 7.1.2 Proces ProcIDLE

Tento proces je spuštěn, když si žádný ze zbývajících procesů nenárokuje procesorový čas. Proces obsahuje příkaz na obsluhu obrazovek.

### 7.1.3 Proces HI01

Tento proces obsahuje samotný program pro měření dat. Podle vývojového grafu na obrázku 7.1 je nejprve vytvořen seznam proměnných, s kterými je dále pracováno a seznam aliasů.

| Proměnné  |     |       |         |      |                          |                 |         |   |
|-----------|-----|-------|---------|------|--------------------------|-----------------|---------|---|
| Alias     |     |       |         |      |                          |                 |         |   |
| ProcHI01  |     |       |         |      |                          |                 |         |   |
| ProcINIT  |     |       |         |      |                          |                 |         |   |
| ProcIDLE  |     |       |         |      |                          |                 |         |   |
| Menu      |     |       |         |      |                          |                 |         |   |
| Info      |     |       |         |      |                          |                 |         |   |
| Mereni    |     |       |         |      |                          |                 |         |   |
| Ovladani  |     |       |         |      |                          |                 |         |   |
| Jméno     | Typ | Řádků | Sloupců | WID  | Wam                      | Init hodnota    | Stanice | Komentář                                  |
| alfa      | F   |       |         | 1019 | <input type="checkbox"/> | 0.3             | 1       | Volba filtrace (od 0 do 1)                |
| Ot_mer    | F   |       |         | 1006 | <input type="checkbox"/> |                 | 1       | Hodnota měření otáček                     |
| Ot_merF   | F   |       |         | 1018 | <input type="checkbox"/> |                 | 1       | Měřená hodnota filtrovaná                 |
| Ot_zad    | F   |       |         | 1010 | <input type="checkbox"/> |                 | 1       | Hodnota žádaných otáček                   |
| Pwr       | F   |       |         | 1015 | <input type="checkbox"/> |                 | 1       | napájecí napětí                           |
| Min_U     | F   |       |         | 1014 | <input type="checkbox"/> | 2               | 1       | Minimalni hodnota na vystupu              |
| mer       | I   |       |         | 1008 | <input type="checkbox"/> |                 | 1       | Měření                                    |
| Uk1       | I   |       |         | 1002 | <input type="checkbox"/> |                 | 1       | Ukazatel pro posun hodnoty otáček         |
| Uk2       | I   |       |         | 1003 | <input type="checkbox"/> |                 | 1       | Ukazatel pro posun hodnoty zátěží         |
| u_in      | I   |       |         | 1000 | <input type="checkbox"/> |                 | 1       | Vstupní napětí                            |
| Uk3       | I   |       |         | 1007 | <input type="checkbox"/> |                 | 1       | Ukazatel pro zpoždění generovaných hodnot |
| I         | I   |       |         | 1017 | <input type="checkbox"/> |                 | 1       | Nastavení hodnoty čítače                  |
| Buf_OtZad | MF  | 31    | 1       | 1011 | <input type="checkbox"/> | 8,2,8,2,8,2(... | 1       | Buffer pro žádanou hodnotu otáček (PNBS)  |
| Buf_OtMer | MF  | 8000  | 2       | 1001 | <input type="checkbox"/> |                 | 1       | Buffer pro měřenou hodnotu otáček         |
| ip_adr    | MI  | 1     | 4       | 1004 | <input type="checkbox"/> |                 | 1       | IP adresa stanice                         |

Obrázek 7.2 - Seznam proměnných

Program je napsán ve strukturovaném jazyce pro větší přehlednost v programu. Podle vývojového diagramu na obrázku 7.1, kde je shrnuto principiální řešení programu, začíná program zápisem měřené hodnoty analogového vstupu do proměnné. Na konci programu je naopak zápis další proměnné na analogový výstup systému. Rozsah napětí je nastaven

od 0 do 10 V. Příkazy pro zápis i čtení běží nezávisle na začátku měření, protože nejsou součástí podmíněného výrazu pro začátek měření. To znamená, že v proměnných je zatím nedefinovaná hodnota, tedy nula. To proto, aby se tam při začátku měření neobjevovala nechtěná hodnota.

```
// Čtení analogového vstupu
AnIn #AI00_0, Ot_mer, 10.000, 0.000, 10.000, 0.000, 10.000
```

Obrázek 7.3 - Příkaz na přečtení vstupu

Dále je v programu podmínka začátku měření, která je propojena s vytvořenou obrazovkou. Program je spuštěn každou milisekundu, ale ukládání dat do bufferu proběhne 8000 při nenulové hodnotě proměnné Start a pak se proměnná Start vynuluje.

Naměřená data v proměnných zapíše do jednoho bufferu do dvou sloupců, aby odpovídající data byla pohromadě. Tato hodnota je zvolena tak, aby dostačovala vnitřní paměť řídicího systému pro uložení takového počtu hodnot. Jelikož buffer má dva sloupce, tak se bude jednat o šestnáct tisíc hodnot datového typu float.

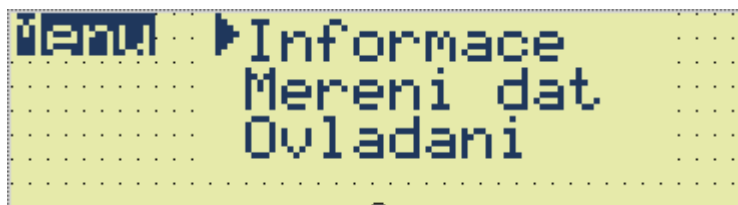
Další podmínka je vytvořena z důvodu zpoždění zápisu generovaného pseudonáhodného binárního signálu do proměnné. Zpoždění pro soustavu motor generátor má hodnotu sto. Třetí, poslední, podmínka udává posuvník pro jednotlivé načtení PNBS do proměnné a zároveň na výstup. Jelikož PNBS obsahuje 31 hodnot, tak i tato podmínka je nastavena na tuto hodnotu, aby po zapsání poslední hodnoty začalo zapisování opět od první hodnoty. Takže jedna generovaná hodnota PNBS se na výstupu objeví stokrát a celková perioda, kdy se protočí všech 31 hodnot PNBS, bude mít hodnotu 3100.

```
If @Start // Podmínka začátku měření a zápisu dat na Buffer
  If Uk1 < 8000 // Začínáme od 0 a měříme 1000 hodnot
    Let Buf_OtMer[Uk1,1] = Ot_mer // Do Bufferu postupně zapisujeme hodnotu AI.0
    Let Buf_OtMer[Uk1,0] = Ot_zad // Do Bufferu postupně zapisujeme hodnotu AO.0
    Let Uk1 = Uk1 + 1
    Let Ot_zad = Buf_OtZad[Uk2,0] // Postupné zapisování hodnot PNBS do proměnné pro AO.0
    If Uk3 < 100 // 100 krát zpoždění další zapisované hodnoty do Ot_zad
      Let Uk3 = Uk3 + 1
    Else
      Let Uk2 = Uk2 + 1
      Let Uk3 = 0
      If Uk2 > 30 // Kontrola pro hlídání přetečení Bufferu
        Let Uk2 = 0
      EndIf
    EndIf
  EndIf // End Uk3
```

Obrázek 7.4 - Příklad zápisu příkazů programu

### 7.1.4 Obrazovka Menu

Tato obrazovka je úvodní, zobrazená při načtení programu do řídicího systému. Umožňuje přístup k dalším obrazovkám, které jsou zde vypsány v přehledném seznamu. Na jednotlivé obrazovky se dostaneme vybráním požadované šipky a potvrzením tlačítkem Enter. Návrat zpět na obrazovku Menu je realizován stisknutím tlačítka Esc.



Obrázek 7.5 - Obrazovka Menu

### 7.1.5 Obrazovka Info

Po vybrání položky *Informace* v obrazovce Menu se objeví obrazovka Info. Ta zobrazuje IP adresu připojeného řídicího systému a vstupní napětí systému. Tyto dvě hodnoty jsou aktualizovány v procesu ProcInit. Takže jsou načteny pouze jednou při startu systému a nemění se.



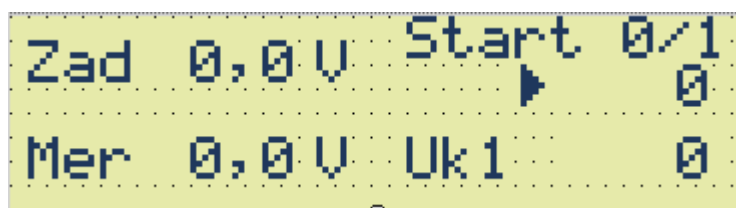
Obrázek 7.6 - Obrazovka Info

### 7.1.6 Obrazovka Mereni

Obrazovka Mereni slouží primárně k spuštění měření dat. Obsahuje editovatelnou hodnotu Start 0/1, která při změně hodnoty na jedničku zaktivuje podmínku programu pro měření. Tato hodnota je synchronizována s bitovou hodnotou aliasu Start.

Obrazovka dále obsahuje hodnotu ukazatele 1, který bude po startu měření nabývat hodnot 0 až 8000. V levé části obrazovky se zobrazuje hodnota *Zad*, která udává hodnotu proměnné představující hodnotu PNBS. Hodnota *Mer* naopak udává hodnotu čtenou na

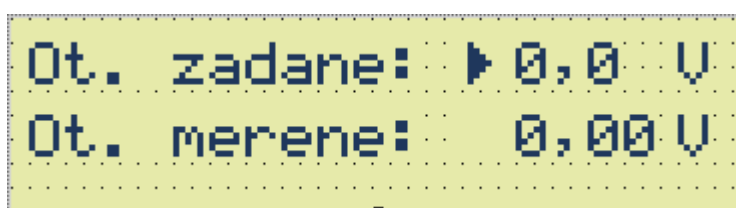
vstupu, která odpovídá zapisované hodnotě *Zad*. Tyto hodnoty se budou měnit s časovým intervalem 300 ms.



Obrázek 7.7 - Obrazovka Mereni

### 7.1.7 Obrazovka Ovladani

Obrazovka Ovladani slouží k manuálnímu zadávání žádané hodnoty na výstup řídicího systému a čtení odpovídající hodnotě na vstupu systému, která je filtrovaná exponenciálním filtrem.



Obrázek 7.8 - Obrazovka Ovladani

## 7.2 EXPORT DAT

Po provedeném naměření experimentálních dat je třeba data přenést do vhodného programu pro další zpracování, jak bylo uvedeno v kapitolách výše. Přenos dat do listu Excelu je realizován uživatelskými funkcemi ve VBA používajícími metody objektů komunikačního ovladače AtouchX.

S jazykem VBA se pracuje lehce a intuitivně, využívá se zde programování obsluh událostí. Pro přenos dat je vytvořeno v editoru Excel pět tlačítek, která po stisku vykonají příslušné činnosti. Obsluhy událostí tlačítek využívají objekt AtouchDir, tj. synchronní (direktivní) způsob komunikace s AMiNi4DS.

Na začátku programu musí být vytvořen ActiveX objekt typu AtouchDir, který je pojmenován ATC. První makro, které je spojeno s tlačítkem *Inicializace* slouží k inicializaci napojení na vybrané proměnné v zadaných stanicích sítě DB-Net/IP. Potřebné informace jsou uloženy v souborech HW.ini a DB.ini. Inicializace je vytvořena pomocí metody InitFromFile, která provede počáteční inicializaci objektu. Pokud není inicializace úspěšná, nelze používat ostatní metody. V případě, že inicializace proběhne v pořádku, vypíše se do okna v Excelu „Inicializace OK“. Pokud inicializace neproběhne, otevře se dialogové okno s číslem hlášené chyby.

```
Ret = ATC.InitFromFile(HW.ini, Db.ini)
If (Ret <> arrOK) Then
    Call MsgBox("Chyba inicializace !" + Hex(Ret))
Else
    Me.Cells(5, 4) = "Inicializace OK"
End If
```

Obrázek 7.9 - Metoda InitFromFile

Druhé tlačítko slouží k získání informací o proměnné, z které budou čtena data. K tomu je použita metoda DbGetInfo. Ta vytváří proměnnou *INFO* jako matici o velikosti 6 prvků. Každý prvek matice přitom nese jinou vlastnost použité proměnné. Pokud metoda proběhne správně, tak se vypíše počet řádků a sloupců proměnné. Jinak se otevře okno s textem „Proměnná neexistuje!“.

```
Ret = ATC.DbGetInfo(WID, INFO)|
If (Ret = arrOK) Then
    R = INFO(4)
    S = INFO(5)
    Me.Cells(5, 6) = "Řádky"
    Me.Cells(6, 6) = INFO(4) 'Počet řádků
    Me.Cells(5, 7) = "Sloupce"
    Me.Cells(6, 7) = INFO(5) 'Počet sloupců
Else
    Call MsgBox("Proměnná neexistuje! ")
End If
```

Obrázek 7.10 - Metoda DbGetInfo

Další tlačítko již slouží k načtení dat ze zvolené proměnné. K tomu slouží metoda GetDataMtx, která přečte obsah proměnné podle zvoleného WIDu. WID proměnné je uložen v souboru DB.ini, který je načten již v inicializaci. Metoda vrátí proměnnou *DATA* ve formátu matice. Prvky matice odpovídají typu čtené proměnné. Pokud je proměnná správně načtena,

tak se vypíše do vybraných oken v Excelu. Pokud ne, tak se otevře dialogové okno s textem „Čtení dat selhalo!“.

```
Ret = ATC.GetDataMtx(WID, 0, 0, R, S, DATA, Res)
If (Ret = arrok) And ((Res And atfOk) <> 0) Then
    For i = 0 To R - 1
        For j = 0 To S - 1
            Me.Cells(i + 7, j + 8) = DATA(i, j)
        Next j
    Next i
Else
    Call MsgBox("Čtení dat selhalo! ")
End If
```

Obrázek 7.11 - Metoda GetDataMtx

Další tlačítko je vytvořeno, kdyby bylo potřeba zapsat data do řídicího systému. K tomu slouží metoda PutData, ale toto makro není v této práci využito.

Poslední tlačítko slouží k ukončení inicializace. Zde je vytvořena metoda Done, která ukončí připojení k síti. To znamená, že ukončí komunikaci s řídicím systémem a uvolní všechny přidělené HW prostředky. Tato metoda se musí vyvolat vždy před zrušením objektu a to bez ohledu na to, zda inicializace proběhla úspěšně či neúspěšně. Po ukončení připojení nelze používat ostatní metody v objektu.

```
'***** UKONČENÍ OBJEKTU *****
Public Sub AtouchKonec()
    On Error Resume Next
    'jestli objekt existoval, tak ho zrušíme
    If Not (ATC Is Nothing) Then ATC.Done
    'uvolníme paměť
    Set ATC = Nothing
    MsgBox "Ukončení komunikace. "
    'Call ATC.Done

End Sub
```

Obrázek 7.12 - Metoda Done

## 7.3 IDENTIFIKACE

Pro identifikaci naměřených dat byl vytvořen program ve výpočtovém prostředí Matlab. Nejdříve jsou načtena data z Excelu, kde jsou vypsaná data z databázové proměnné z řídicího systému obsahující vstupní a výstupní data. Načtení je rozlišeno v souboru, zda se

jedná o data pro identifikaci řídicího systému nebo identifikaci soustavy motor generátor. Identifikace je provedena metodou nejmenších čtverců, pro kterou je vytvořena funkce, která je následně volána z hlavního programu.

```
function [b,a]=MNC_amin(nb,na,u,y)
    ly = length(y); % delka vektoru y
    lu = length(u); % delka vektoru u
    X = zeros(ly,na+nb); % matice vystupu y a vstupu u

    % Cyklus naplneni matice vystupem
    for i = 1:na
        X(i+1:end,i) = -y(1:ly-i);
    end

    % Cyklus naplneni matice vstupem
    for i = 1:nb
        X(i+1:end,na+i) = u(1:lu-i);
    end

    P = inv(X'*X)*X'*y; % vypočet odhadu parametru a,b
    a = P(1:na)';
    b = P(na+1:na+nb)';
end
```

Obrázek 7.13 - Funkce MNČ

Jako vstupní parametry vytvořené funkce *MNC\_amin* je počet neznámých parametrů *na*, *nb*, načtené vstupní hodnoty *u* a výstupní hodnoty *y*. Výstupem této funkce budou hodnoty neznámých parametrů *a*, *b*.

V hlavním programu je poté příkaz na volání této funkce, kam se za počty neznámých parametrů dosadí konkrétní číslo. Daná čísla udávají počet neznámých parametrů diferenční rovnice a určují tedy její řád. Pro ověření správnosti řešení je na konci hlavního programu vykreslení proložení aproximovaných a měřených dat.

## 7.4 RELÉ VE ZPĚTNÉ VAZBĚ

Získání kritických hodnot pro výpočet parametrů diskrétního regulátoru je realizováno pomocí relé ve zpětné vazbě. Programová realizace relé je řešena v návrhovém prostředí DetStudio.

Relé ve zpětné vazbě je jeden z nejsnadnějších způsobů pro získání kritických hodnot. Programová realizace je provedena jako přepínání kolem žádané hodnoty, která je určena jako odezva na skokovou změnu v proměnné *Skok*. Podle statické charakteristiky je pro 5 V v proměnné *Skok* žádaná hodnota 3,58 V. Přepínání je realizováno pro proměnnou *Skok* s  $\pm Re$ , jejíž hodnota je 3 V. Relé bude spínat nebo rozepínat po překročení žádané hodnoty *Ot\_zad* hodnotou měřenou *Ot\_mer*.

```
// Realizace relé u0 = 5V, y0 = 3,58 V
Let Re_p = Skok + Re
Let Re_m = Skok - Re

If Ot_mer > Ot_zad
    Let Vyst = Re_m
EndIf
If Ot_mer < Ot_zad
    Let Vyst = Re_p
EndIf
```

Obrázek 7.14 - Relé ve zpětné vazbě

## 7.5 PSD REGULÁTOR

Realizace PSD regulátoru je v prostředí DetStudio vytvořena na základě znalosti modifikovaného přírůstkového tvaru regulátoru. Po vypočítání parametrů regulátoru  $q_i$  a dosazení do dané rovnice (1.21), je vytvořen příkaz na zápis tohoto tvaru regulátoru. Výsledná hodnota akčního zásahu je poté pouze zapisována na výstup řídicího systému a je čtena odezva.

```
Let PSD_akcni = PSD_akcni + q*Ot_Zad - q0*Ot_Mer - q1*y1 - q2*y2
Limiter PSD_akcni, 0.000, 10.000
```

Obrázek 7.15 - PSD regulátor

Příkaz Limiters je použit z důvodu omezení velikosti akčního zásahu regulátoru. Je to z důvodu přípustného napětí použitého pro řízení soustavy, které je v rozmezí od 0 do 10 V.

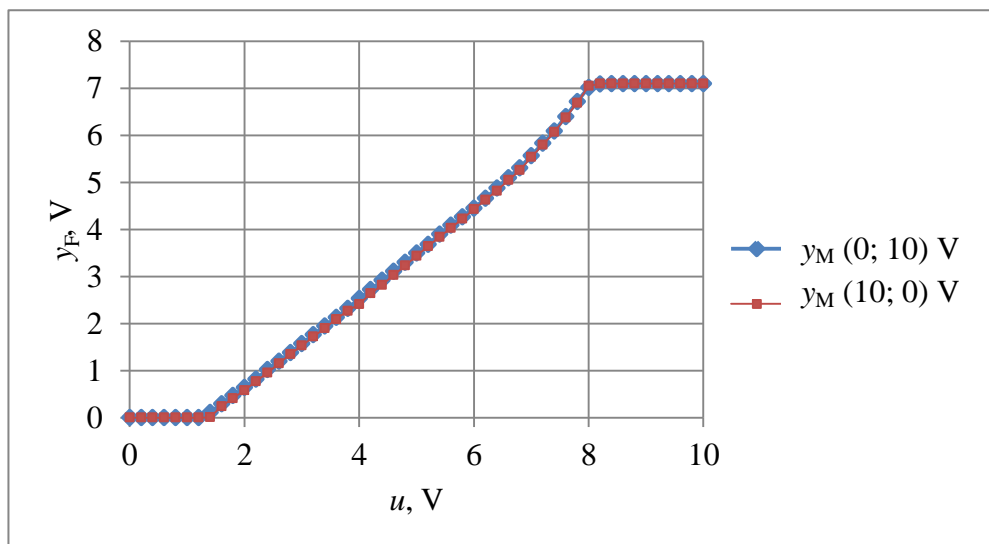
## 8 VÝSLEDKY

Po naměření experimentálních dat je důležité naměřit i statickou a dynamickou charakteristiku systému motor generátor.

### 8.1 STATICKÁ CHARAKTERISTIKA

Statická charakteristika se měří z důvodu určení mezních hodnot měření a pro zvolení vhodného rozsahu PNBS. Slouží pro určení rozsahu napětí měření, ve kterém je zvolen nejlepší pracovní bod a krajní body, které značí konec měřicího rozsahu a nedefinované hodnoty. Měření probíhalo od 0 V do 10 V, což je rozsah přípustného napětí soustavy motor generátor.

Měření proběhlo nejprve od nejnižší k nejvyšší hodnotě napětí a poté zpět od nejvyšší k nejnižší. Je to z důvodu ověření, zda dynamika systému nevykazuje hysterezi. Tímto měřením se potvrdilo, že hystereze v systému není, protože měřené hodnoty v obou směrech jsou podobné, lišící se o setiny voltu. Statická charakteristika je zobrazena na obrázku 8.1. Podle průběhu statické charakteristiky lze usoudit, že v použitém rozsahu je téměř lineární.



Obrázek 8.1 - Statická charakteristika

Pro snazší odečtení naměřených hodnot zatížených šumem měření byl v programu realizován filtr, který filtruje měřenou veličinu. Diskrétní filtr je typu exponenciálního ve tvaru

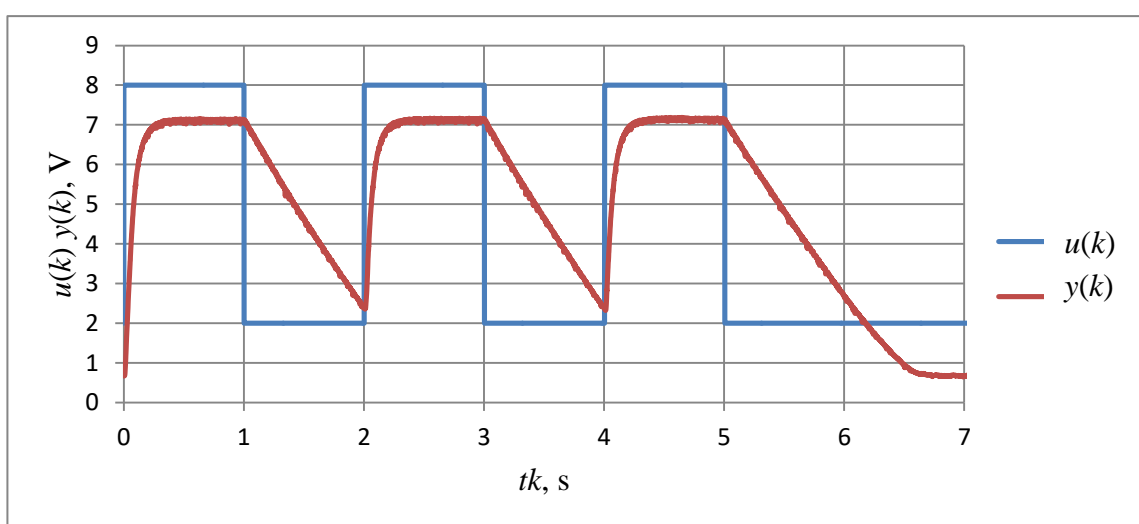
$$y_F(k) = \alpha \cdot y(k) + (1 - \alpha) \cdot y_F(k-1), \quad (8.1)$$

kde  $\alpha$  je filtrovací konstanta o hodnotě 0,3,  
 $y(k)$  jsou měřená data,  
 $y_F(k)$  jsou filtrovaná data.

Motor začne pracovat při minimální hodnotě vstupního napětí 1,4 V. Maximální hodnota, nad kterou motor již nezvyšuje otáčky je 7 V při minimálně 8 V vstupního napětí. Měřicí rozsah je tedy od 1,4 V do 8 V.

## 8.2 DYNAMICKÁ CHARAKTERISTIKA

Pro získání představy o dynamických vlastnostech soustavy byla změřena odezva systému na skokovou změnu vstupního signálu při intervalu vzorkování 1 ms.



Obrázek 8.2 – Dynamické vlastnosti soustavy

Z charakteristiky je patrné, že soustava se chová jinak při reakci na náběžnou hranu a jinak při reakci na sestupnou hranu vstupního signálu. Takže soustava vykazuje asymetrické dynamické chování a jedná se tedy o nelinearitu. Soustavu lze chápat jako systém se dvěma vstupy, kdy při působení obou vstupů je soustava rychlejší na náběžné hraně signálu, ale při sestupné hraně je soustava pomalejší, protože je brzděna při působení jednoho vstupu.

Toto zjištění asymetrického chování je velmi důležité pro identifikaci soustavy. Také je zřejmé, že lineární diferenční rovnice nemůže takovéto chování dobře popsat a získané výsledky jsou sice nejlepší aproximací z hlediska MNČ, ale nepopisují chování reálného systému dobře.

### 8.3 VZORKOVACÍ FREKVENCE ŘÍDICÍHO SYSTÉMU

Po určení dynamiky regulovaného systému je třeba zvolit vzorkovací frekvenci řídicího systému. Volba vzorkování je důležitá pro provedení identifikace, aby byla patrná dynamika systému. Vzorkovací frekvence se následně promítne i do nastavení PSD regulátoru a bude určovat časové okamžiky akčních zásahů regulátoru.

Při nastavení velké vzorkovací frekvence by dynamika systému nemusela být vůbec rozeznatelná. Proto při zjištění dynamické charakteristiky podle obrázku 8.2 můžeme určit dobu ustálení při náběhu soustavy na 300 ms a dobu ustálení při snížení otáček motoru na 1500 ms. Proto je zbytečné použít příliš velkou vzorkovací frekvenci.

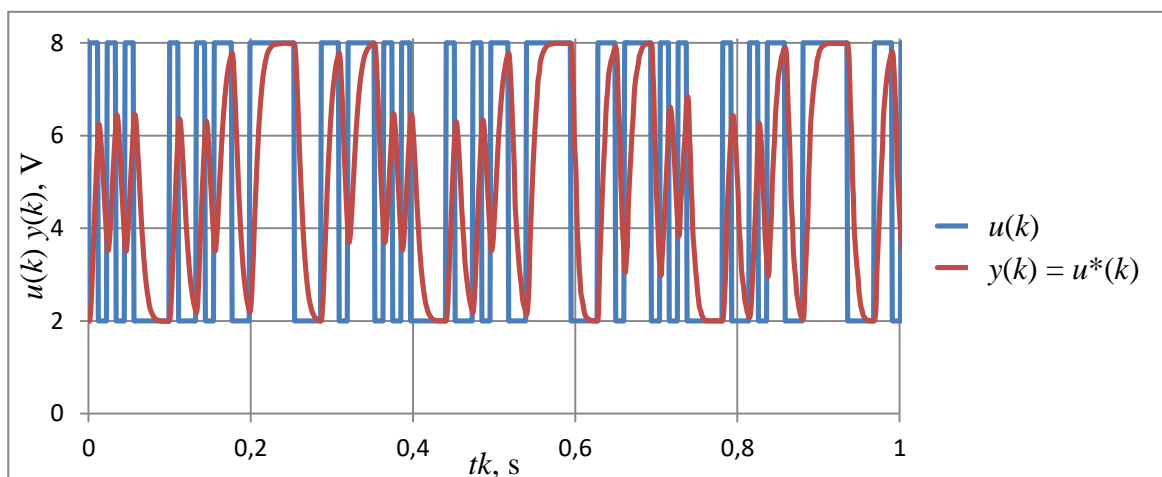
Jelikož připojený řídicí systém AMiNi4DS má také vlastní dynamiku generování analogových výstupů, tak byly naměřeny data při dvou vzorkovacích frekvencích. Je zvolena nejmenší možná, kterou řídicí systém dokáže nastavit, tedy 1 ms. Pro menší vzorkovací rozlišení je také použita vzorkovací perioda 10 ms.

### 8.4 IDENTIFIKACE

Než se začne s identifikací samotné soustavy motor generátor, je nutné nejprve identifikovat samotný řídicí systém AMiNi4DS a určit řád diferenční rovnice.

Identifikace dynamiky řídicího systému proběhne při odpojené soustavě motor generátor. S intervalem vzorkování 1 ms je měřen průběh výstupního napětí jako odezva na programově nastavované hodnoty generované PNBS.

Při vyhodnocování dat naměřených na řídicím systému připojeném naprázdno je třeba určit řád diferenční rovnice. Pro druhý řád vyšla diferenční rovnice ve tvaru



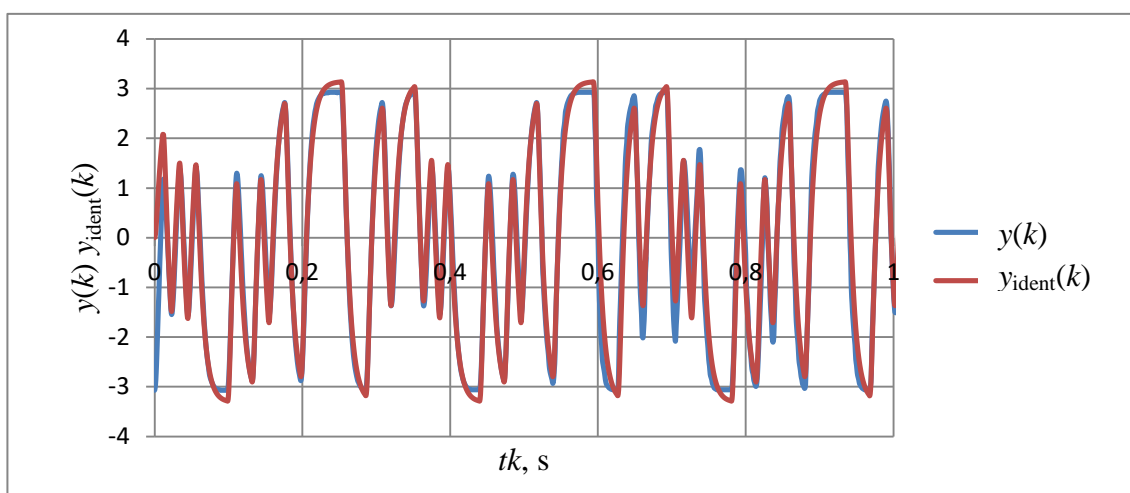
Obrázek 8.3 – Měření pro identifikaci AMiNi4DS

$$y(k) - 1,501y(k-1) + 0,547y(k-2) = 0,022u(k-1) + 0,027u(k-2), \quad (8.4)$$

které odpovídá výsledný přenos řídicího systému ve tvaru

$$F_A = \frac{0,027 \cdot z + 0,022}{z^2 - 1,501 \cdot z + 0,547}. \quad (8.5)$$

Po zvolení vyššího řádu diferenční rovnice se chyba aproximace výrazně nezměnila. To znamená, že zvolený druhý řád je dostatečný a vyšší řády jsou zbytečně složité a jejich řešení již výsledek citelně nevylepší.



Obrázek 8.4 – Identifikace AMiNi4DS

Pro porovnání výsledků aproximace pro 2. řád a 3. řád diferenční rovnice je vypočtena směrodatná odchylka chyby aproximace.

Tabulka 8.1 - Chyby aproximace

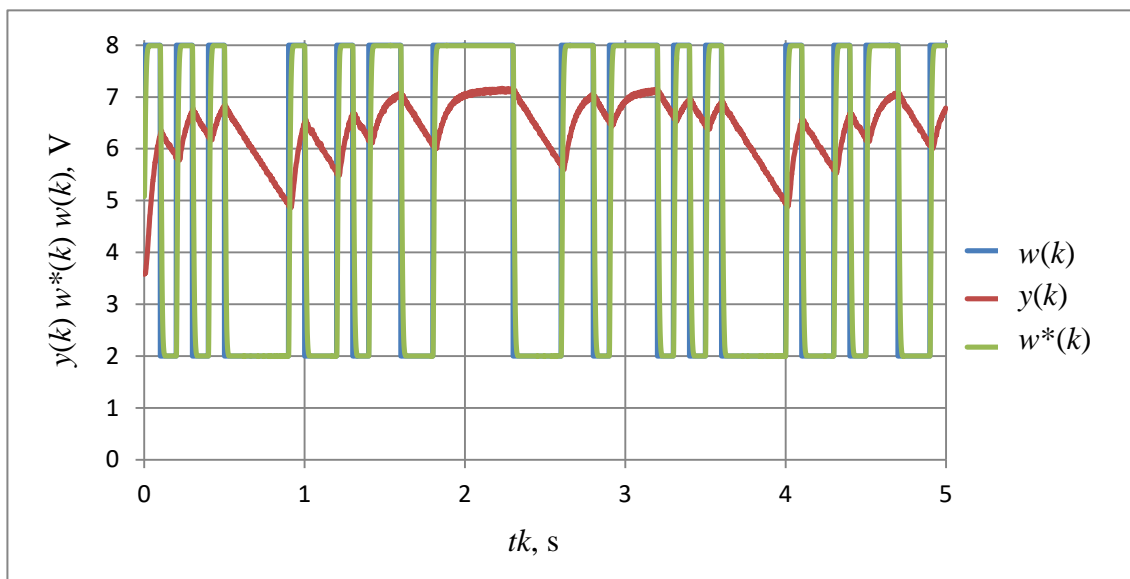
|        | směr. odch. 1 ms |
|--------|------------------|
| 2. řád | 0,3784           |
| 3. řád | 0,3235           |

Pro identifikaci měření a určení odpovídající diferenční rovnice soustavy motor generátor byla naměřena data pro celou soustavu zapojení, tedy motor generátor a řídicí systém. Pro určení diferenční rovnice pouze pro soustavu motor generátor je třeba z naměřených dat určit výstupní data řídicího systému, které působí na soustavu.

Měření dat pro identifikaci proběhlo pro obě zvolené periody vzorkování řídicího systému, tedy pro 1 ms a pro 10 ms. Identifikace je provedena pro průběh vstupního PNBS v celém rozsahu měření, tedy od 2 do 8 V.

Samotná identifikace je provedena z dat odpovídajících jedné periodě PNBS.

Pro identifikaci celkového systému lze použít data získaná s periodou vzorkování 1 ms a pro periodu vzorkování 10 ms, brát v úvahu každou desátou hodnotu z tohoto měření.



Obrázek 8.5 – Měření pro identifikaci soustavy

Výpočet parametrů diferenční rovnice je potřeba provádět pro data v odchylkovém tvaru, aby mohl být použit tvar diferenční rovnice (3.3), tj. aby bylo splněno, že pro nulové hodnoty vstupu jsou i nulové hodnoty výstupu. Je vhodné začít měřit data pro identifikaci z ustáleného stavu, kdy pro hodnotu vstupu  $u_0$  je měřena hodnota  $y_0$ . Identifikace se pak počítá z vstupů a výstupů v odchylkách

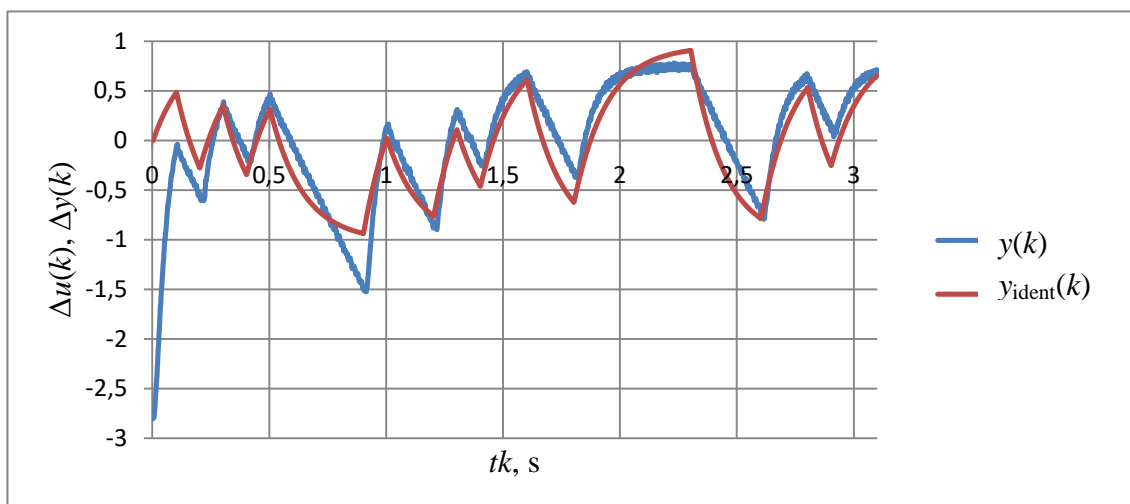
$$\Delta u = u - u_0, \quad \Delta y = y - y_0, \quad (8.2)$$

kde  $u, y$  jsou aktuální měřené hodnoty,  
 $u_0, y_0$  jsou hodnoty posunutí.

Řád diferenční rovnice popisující chování soustavy motor generátor spolu s řídicím systémem AMiNi4DS při intervalu vzorkování 1 ms, byl zvolen 4. řádu. Vyšší řád již nemá na kvalitu proložení měřených dat identifikovanými žádný vliv.

Výsledná diferenční rovnice pro zapojení soustavy motor generátor a řídicího systému pro vzorkovací periodu 1 ms je

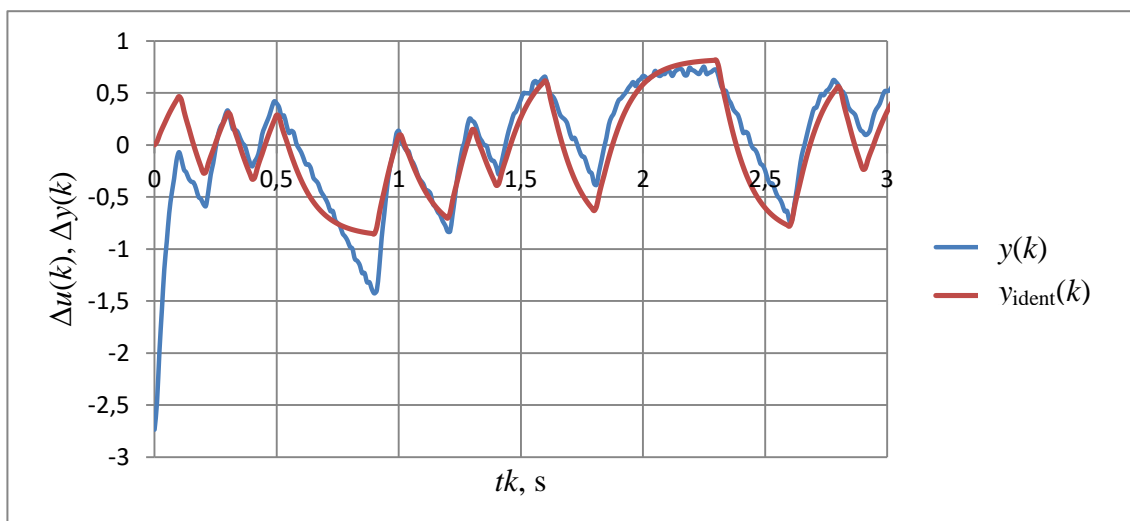
$$y(k) - 1,139y(k-1) + 0,003y(k-2) + 0,195y(k-3) - 0,053y(k-2) = -0,0003u(k-1) + 0,001u(k-2) - 0,001u(k-3) + 0,002u(k-4) \quad (8.3)$$



Obrázek 8.6 – Identifikace soustavy a řídicího systému pro 1 ms

a pro vzorkovací periodu 10 ms je

$$y(k) - 1,354y(k-1) + 0,429y(k-2) - 0,199y(k-3) + 0,162y(k-4) = 0,011u(k-1) + 0,007u(k-2) - 0,006u(k-3) - 0,002u(k-4) \quad (8.4)$$



Obrázek 8.7 - Identifikace soustavy a řídicího systému pro 10 ms

Tabulka 8.2 ukazuje vypočítané směrodatné odchylky pro aproximaci získanou ze zapojení řídicího systému se soustavou motor generátor. Rozdíl hodnot odchylek pro 4. a 5. řád je minimální, proto je zvolen nižší řád, aby výsledná diferenční rovnice nebyla příliš složitá.

Tabulka 8.2 - Chyby aproximace AMiNi4DS a soustavy motor generátor

|        | směr. odch. 1 ms | směr. odch. 10 ms |
|--------|------------------|-------------------|
| 4. řád | 0,3847           | 0,3637            |
| 5. řád | 0,3844           | 0,3610            |

Následně je provedena identifikace samotné soustavy motor generátor, kde jako vstupní signál je použita odezva řídicího systému AMiNi4DS na PNBS.

Výsledná diferenční rovnice pro soustavu motor generátor se vzorkovací periodou 1 ms je

$$y(k) - 1,127y(k-1) + 0,134y(k-2) = -0,006u(k-1) + 0,009u(k-2) \quad (8.5)$$

a pro vzorkovací periodu 10 ms je ve tvaru

$$y(k) - 1,405y(k-1) + 0,456y(k-2) = 0,021u(k-1) - 0,008u(k-2). \quad (8.6)$$

Kritické hodnoty pro nastavení regulátorů jsou z identifikace získány pomocí funkce *margin* ve výpočtovém softwaru Matlab. Hodnoty jsou zobrazeny v tabulce 8.3.

Tabulka 8.3 - Kritické hodnoty z identifikace

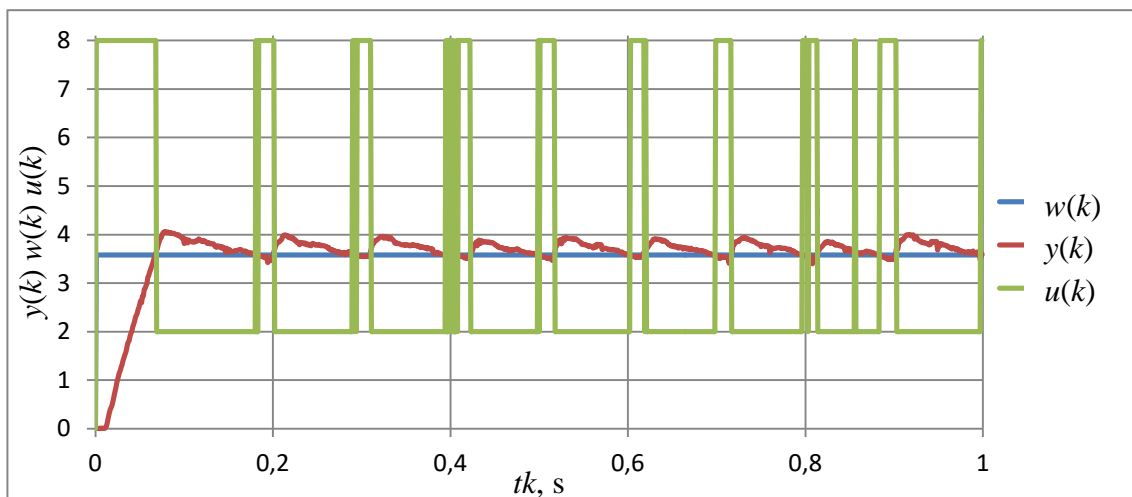
|       | $r_{0k}$ | $T_k, s$ | $Z$    |
|-------|----------|----------|--------|
| 1 ms  | 180,79   | 0,0155   | 0,3274 |
| 10 ms | 76,98    | 0,0385   | 0,2845 |

## 8.5 RELÉ VE ZPĚTNÉ VAZBĚ

Pro porovnání kritických hodnot pro nastavení parametrů regulátoru bylo vytvořeno relé ve zpětné vazbě, podle kterého byly získány kritické hodnoty pro zvolenou periodu vzorkování.

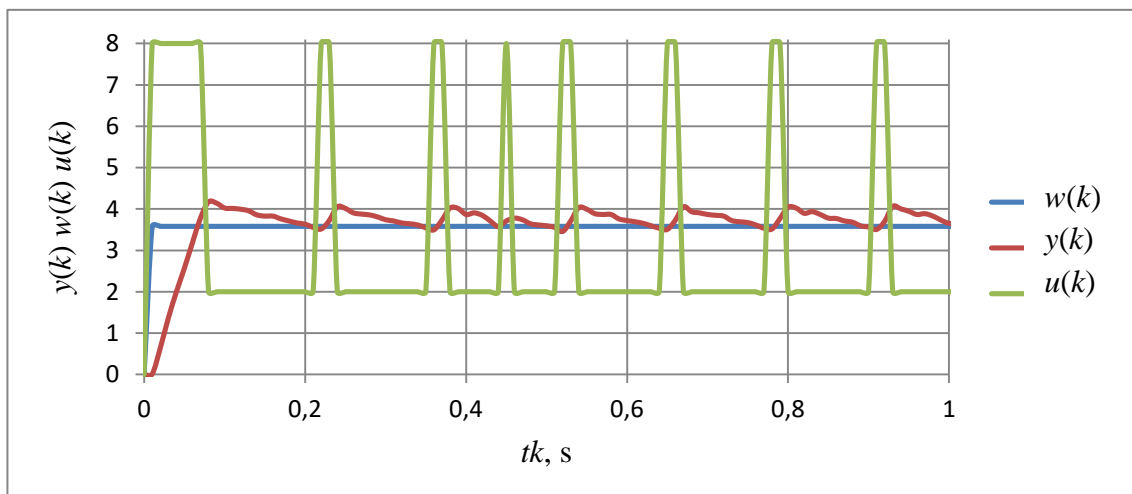
Bylo provedeno nastavení relé pro periodu vzorkování 1 ms a 10 ms. Získané kritické hodnoty jsou uvedeny v tabulce 8.4. Při použité periodě vzorkování 1 ms je získaný průběh přesnější, protože při větší periodě je výsledek zprůměrovaný podle velikosti zvolené periody vzorkování a přepnutí relé je o tuto dobu opožděné.

Proto výsledky pro periodu vzorkování 1 ms a 10 ms vycházejí podobně. Je to z důvodu pomalé dynamiky měřené soustavy, kdy se soustava ustálí za 300 ms, tudíž změna vzorkovací periody na 10 ms je oproti době ustálení stále malá. Pro měření je nastaveno  $u_0 = 5$  V čemuž odpovídá žádaná hodnota  $y_0 = 3,58$  V při přepínání relé  $\pm 3$  V kolem  $y_0$ .



Obrázek 8.8 – Relé ve zpětné vazbě pro 1 ms

Obrázek 8.9 zobrazuje průběh relé pro periodu vzorkování 10 ms.



Obrázek 8.9 - Relé ve zpětné vazbě pro 10 ms

Spočítané kritické hodnoty pro relé ve zpětné vazbě jsou uvedeny v tabulce 8.4.

Tabulka 8.4 – Kritické hodnoty z relé

|       | $r_{0k}$ | $T_k, s$ |
|-------|----------|----------|
| 1 ms  | 10,3235  | 0,102    |
| 10 ms | 8,1268   | 0,133    |

Kritické hodnoty získané z relé v tabulce 8.4 a z identifikace v tabulce 8.3 se liší a je to způsobeno rozdílnou dynamikou soustavy při náběžné a sestupné hraně.

## 8.6 NASTAVENÍ REGULÁTORU

Pro řízení soustavy, které odpovídá diferenční rovnice (8.3) a (8.4), je potřeba vypočítat parametry diskrétního regulátoru. Z předchozí kapitoly jsou známy kritické hodnoty a podle předem známých vzorečků získáme parametry spojitýho PI a PID regulátoru pro periodu vzorkování 1 ms a 10 ms a zvlášť pro hodnoty získané identifikací soustavy a zvlášť pro hodnoty získané pomocí relé ve zpětné vazbě.

Tabulka 8.5 - Parametry spojitých regulátorů

|       |     | $r_0$        | $T_I$  | $T_D$  | $r_0$  | $T_I$  | $T_D$  |
|-------|-----|--------------|--------|--------|--------|--------|--------|
|       |     | Identifikace |        |        | Relé   |        |        |
| 1 ms  | PI  | 81,3539      | 0,0132 |        | 4,6456 | 0,0867 |        |
|       | PID | 108,4721     | 0,0078 | 0,0019 | 6,1941 | 0,0510 | 0,0127 |
|       |     | Identifikace |        |        | Relé   |        |        |
| 10 ms | PI  | 34,6389      | 0,0327 |        | 3,6571 | 0,1105 |        |
|       | PID | 46,1859      | 0,0192 | 0,0048 | 4,8761 | 0,0650 | 0,0163 |

Ze získaných parametrů pro nastavení spojitýho PI a PID regulátoru lze přepočítat parametry pro modifikovaný přírůstkový tvar PS a PSD regulátoru (1.21) podle známých vzorců v tabulce. Přepočtové vzorce jsou

$$q_0 = r_0 \cdot \left( 1 + \frac{T}{2 \cdot T_I} + \frac{T_D}{T} \right), \quad (8.7)$$

$$q_1 = -r_0 \cdot \left( 1 - \frac{T}{2 \cdot T_I} + \frac{2 \cdot T_D}{T} \right), \quad (8.8)$$

$$q_2 = r_0 \cdot \frac{T_D}{T}, \quad (8.9)$$

$$q = q_0 + q_1 + q_2, \quad (8.10)$$

kde  $T$  je perioda vzorkování,  
 $T_I$  je integrační časová konstanta  
 $T_D$  je derivační časová konstanta,  
 $r_0$  je proporcionální zesílení,

$q_i$  jsou parametry diskrétního regulátoru.

Po přepočtu na diskrétní typy regulátorů podle uvedených vzorců jsou hodnoty parametrů uvedeny v tabulce 8.6. Tabulka obsahuje hodnoty pro periodu vzorkování 1 ms a 10 ms získaných z identifikace soustavy a z hodnot relé ve zpětné vazbě.

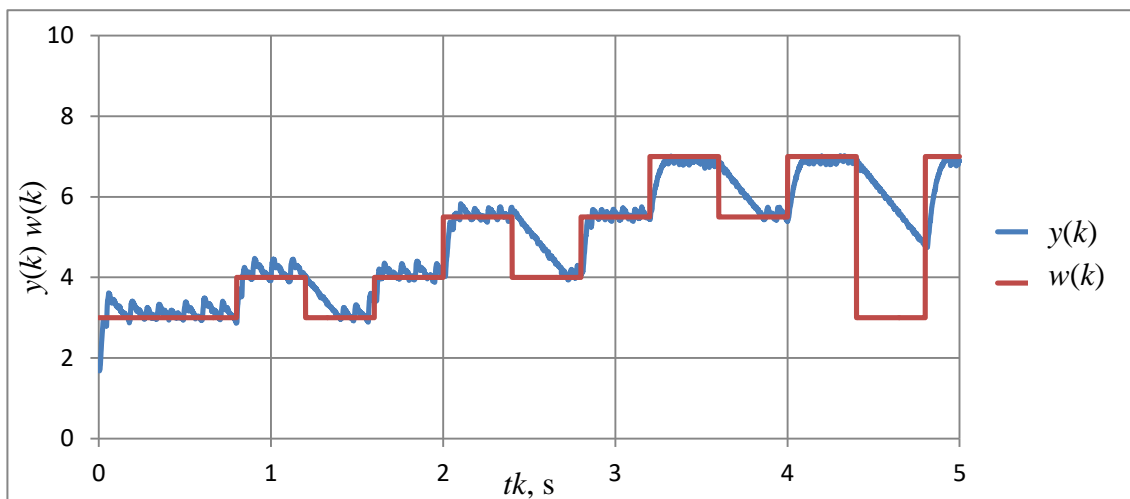
Tabulka 8.6 - Parametry diskrétních regulátorů

|       |     | $q_0$        | $q_1$    | $q_2$   | $q_0$   | $q_1$     | $q_2$   |
|-------|-----|--------------|----------|---------|---------|-----------|---------|
|       |     | Identifikace |          |         | Relé    |           |         |
| 1 ms  | PS  | 84,1947      | -78,055  | 0       | 4,6724  | -4,6188   | 0       |
|       | PSD | 325,303      | -521,564 | 210,178 | 85,2296 | -164,0829 | 78,9748 |
|       |     | Identifikace |          |         | Relé    |           |         |
| 10 ms | PS  | 40,885       | -30,156  | 0       | 3,8225  | -3,4916   | 0       |
|       | PSD | 82,5794      | -81,320  | 23,059  | 13,1748 | -20,3483  | 7,9236  |

## 9 VÝSLEDKY REGULACE

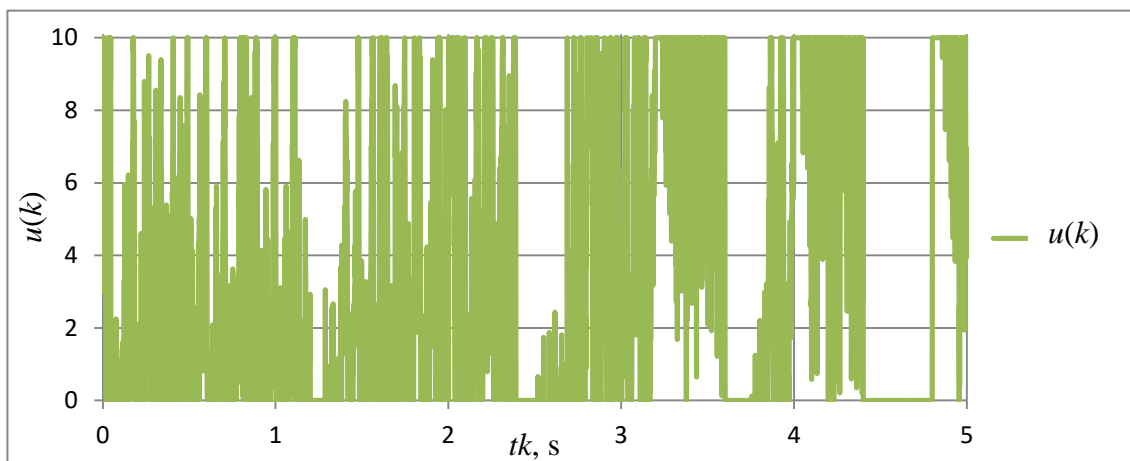
Ze získaných parametrů pro dané struktury diskrétních regulátorů je možné realizovat měření na odzkoušení funkčnosti navržených regulátorů.

Nejprve je implementován PS regulátor s nastavenými parametry pro vzorkovací periodu 1 ms, které jsou získány z provedené identifikace.



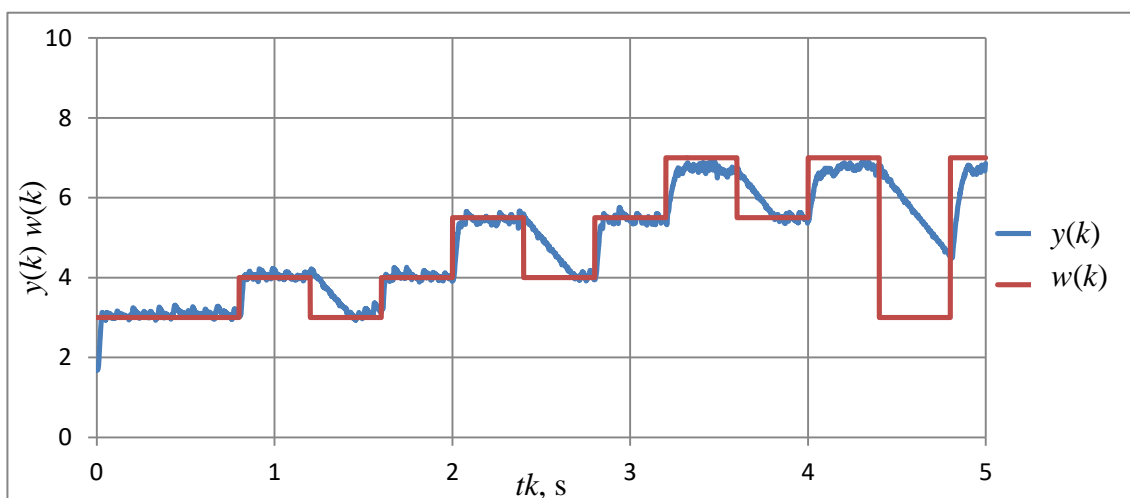
Obrázek 9.1 - PS regulátor pro identifikaci při 1 ms

Z výsledného grafu regulačních pochodů je patrné, že při vzorkování 1 ms regulátor provede časté akční zásahy. Je zřejmé, že i při ustálené hodnotě regulátor stále provádí zásahy akční veličiny, které způsobují překmit regulační veličiny, jejíž hodnota tedy nebude ustálena. To může být způsobeno volbou malé periody vzorkování. Regulátor při působení nulové akční veličiny nemůže zrychlit klesání otáček motoru, což je způsobeno nelineární dynamikou soustavy.



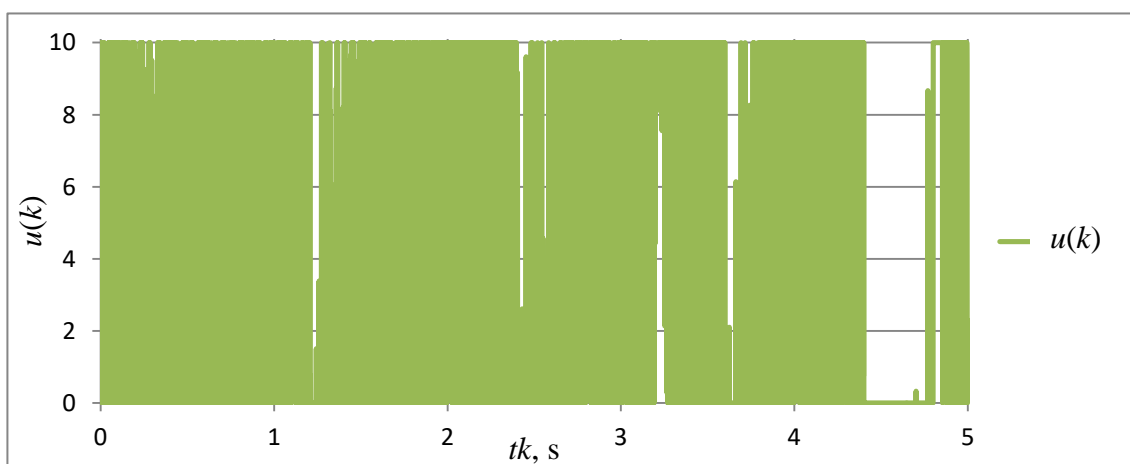
Obrázek 9.2 - Akční zásah PS regulátoru pro identifikaci při 1 ms

Další je PSD regulátor získaný z identifikace soustavy pro vzorkovací periodu 1 ms.



Obrázek 9.3 - PSD regulátor pro identifikaci při 1 ms

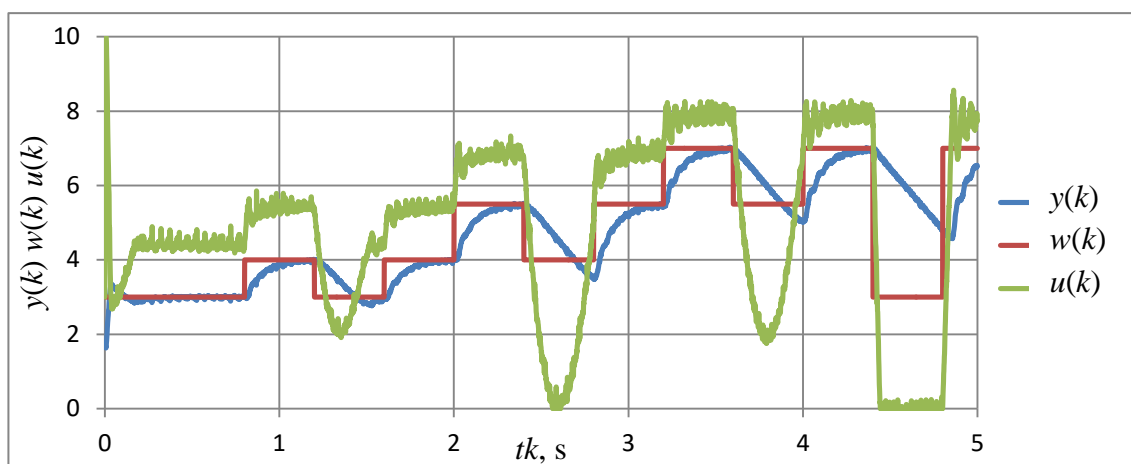
Při použití parametrů PSD regulátoru nejsou oproti PS regulátoru tak velké přeskmy při ustálené hodnotě. Přeskmy jsou způsobeny spíše šumem, na který regulátor reaguje zásahy akční veličiny až do omezení. Tyto přeskmy by se dali minimalizovat použitím filtru, který by minimalizoval vliv působícího šumu.



Obrázek 9.4 - Akční zásah PSD regulátoru pro identifikaci při 1 ms

Regulovaná hodnota se při vyšších hodnotách napětí, které jsou blízké maximální hodnotě, nedostane na žádanou hodnotu i při maximálním působení regulátoru. Příčinou může být omezení akční veličiny podle maximálního přípustného napětí pro soustavu motor generátor, které je od 0 V do 10 V. Regulátor počítá akční zásah z předchozí vypočtené hodnoty namísto hodnoty naposledy realizované, proto se hodnota regulované hodnoty nemusí dostat až na žádanou hodnotu.

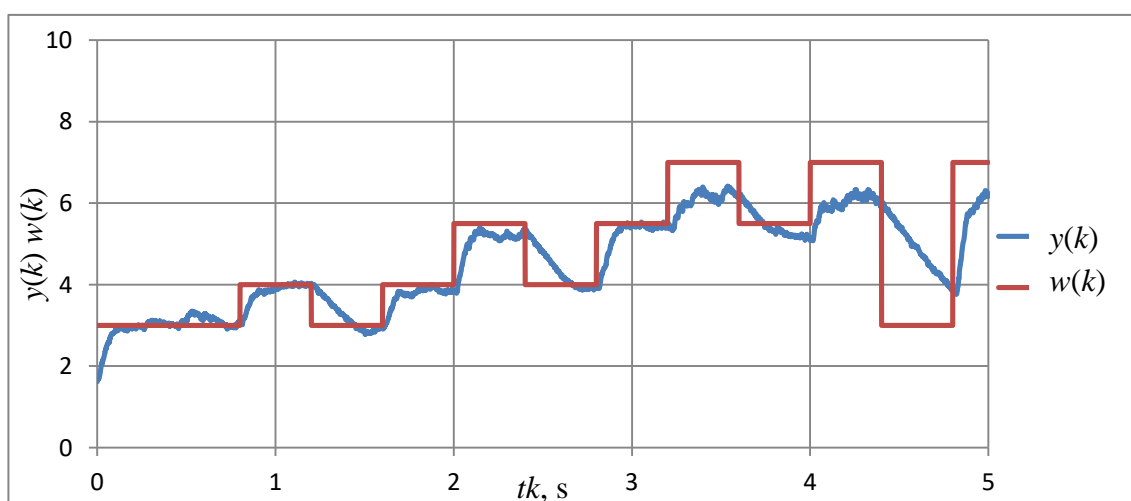
Jako další je použit PS regulátor, jehož parametry jsou získány z kritických hodnot určených z relé ve zpětné vazbě při periodě vzorkování 1 ms.



Obrázek 9.5 - PS regulátor pro relé ve zpětné vazbě při 1 ms

Výsledný regulační pochod je oproti stejnému typu regulátoru získanému z identifikace pro 1 ms nekmitavý. Naopak doba ustálení regulované veličiny je delší. Akční zásah regulátoru je přiměřený regulované veličině a jeho zásahy akční veličiny nejsou až do omezení. Opět se projevuje reakce regulátoru na přítomný šum měření.

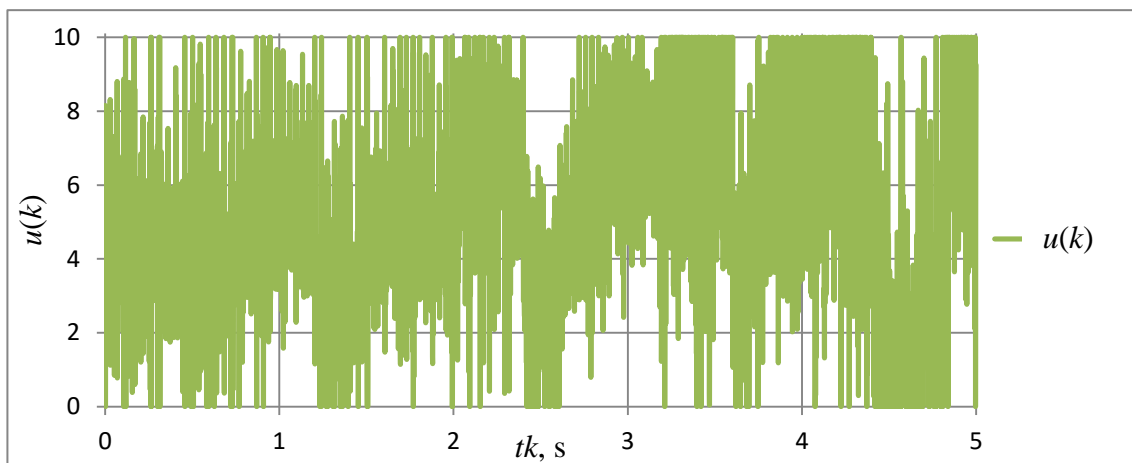
PSD regulátor vypočtený z kritických hodnot získaných z relé ve zpětné vazbě pro periodu vzorkování 1 ms je zobrazen na obrázku 9.6.



Obrázek 9.6 - PSD regulátor pro relé ve zpětné vazbě při 1 ms

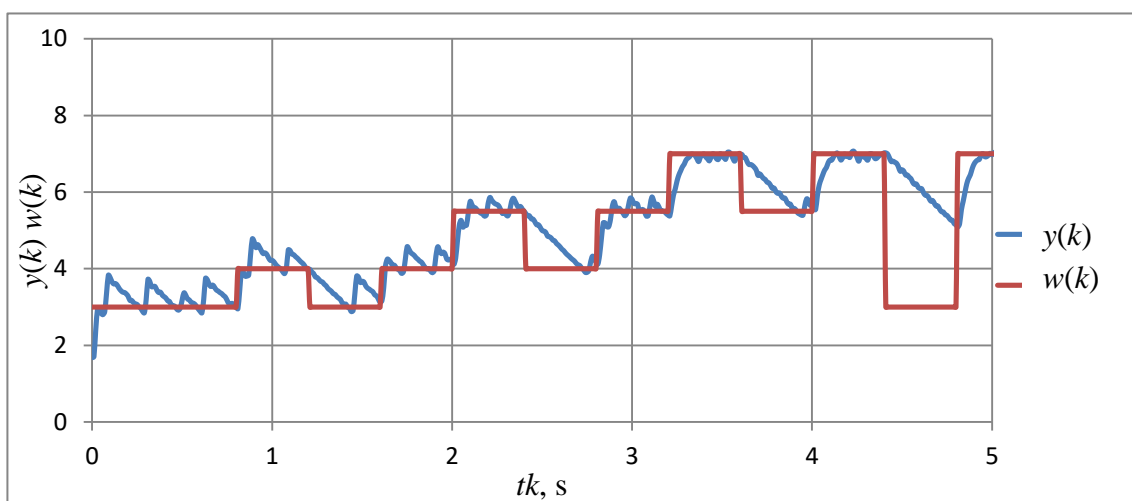
Regulovaná veličina je opět zatížena šumem měření. Stejně jako PSD regulátor získaný z parametrů identifikace u vyšších hodnot napětí žádané veličiny, se hodnoty regulované veličiny nedostanou na požadovanou hodnotu i při působení akční veličiny

regulátoru. Tím je hodnota směrodatné odchylky chyby regulace v tabulce 9.1 zkreslena a tudíž nelze objektivně porovnat regulátory a určit nejvhodnější. Je to z důvodu různé dynamiky soustavy motor generátor při zvyšování a klesání otáček.



Obrázek 9.7 - Akční zásah PSD regulátoru pro relé ve zpětné vazbě při 1 ms

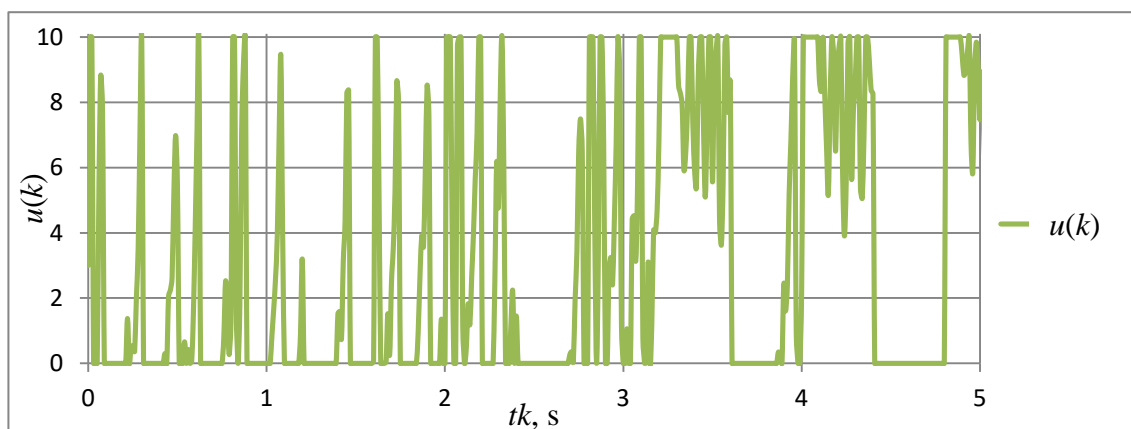
Pro porovnání jsou uvedeny typy regulátorů získaných z identifikace a relé pro periodu vzorkování 10 ms. Jako první je použit PS regulátor získaný z parametrů určených identifikací.



Obrázek 9.8 - PS regulátor pro identifikaci při 10 ms

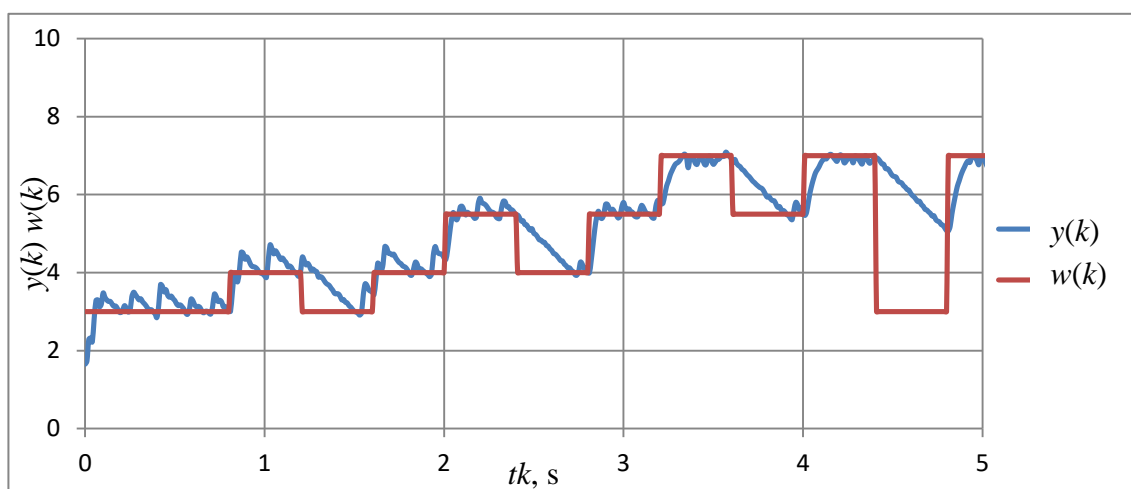
V tomto případě je výsledný regulační pochod obdobný, jako u PS regulátoru, jehož parametry byly získány z identifikace při periodě vzorkování 1 ms. Zde je ovšem překmit regulační veličiny větší. Regulační veličina je také při větších hodnotách napětí menší než žádaná hodnota, ale v tomto případě nastavení je to pouze v řádu setin voltu.

Akční zásah PS regulátoru nastaveného podle kritických hodnot z identifikace je zobrazen na obrázku 9.9.



Obrázek 9.9 - Akční zásah PS regulátoru pro identifikaci při 10 ms

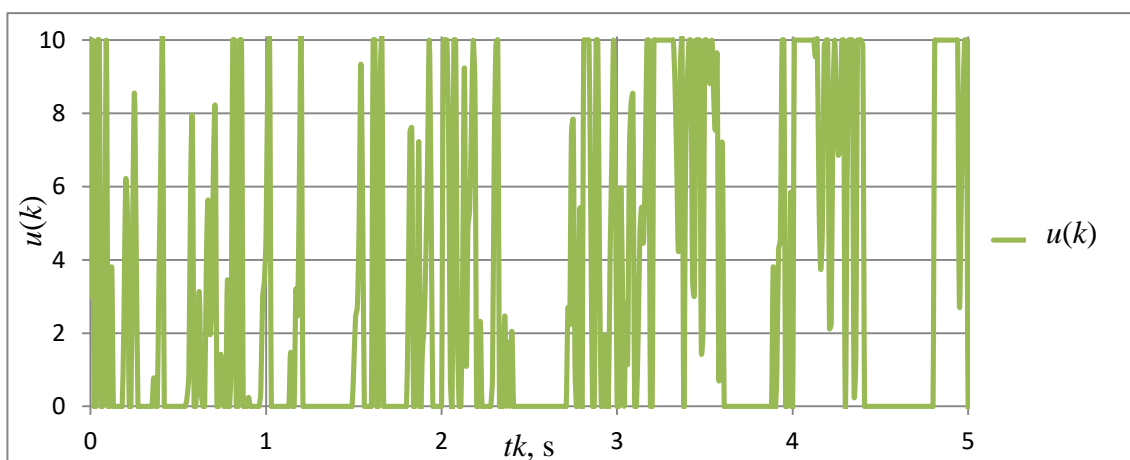
Jako další je použit PSD regulátor spočítaný z kritických hodnot získaných identifikací soustavy.



Obrázek 9.10 - PSD regulátor pro identifikaci při 10 ms

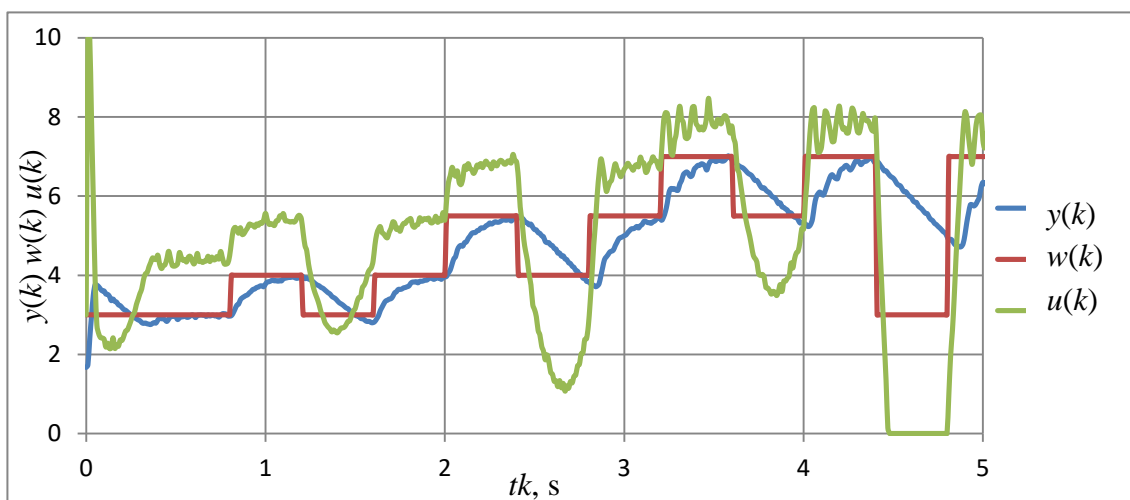
Tento PSD regulátor má podobný regulační pochod jako PS regulátor při stejném způsobu určení kritických hodnot i pro stejnou periodu vzorkování 10 ms. V tomto případě je překmit regulační veličiny menší a doba ustálení je také trochu kratší.

Akční zásah PSD regulátoru získaného z kritických hodnot z identifikace pro periodu vzorkování 10 ms je zobrazen na obrázku 9.11.



Obrázek 9.11 - Akční zásah PSD regulátoru pro identifikaci při 10 ms

Další je PS regulátor nastavený podle kritických hodnot získaných z relé ve zpětné vazbě pro periodu vzorkování 10 ms.

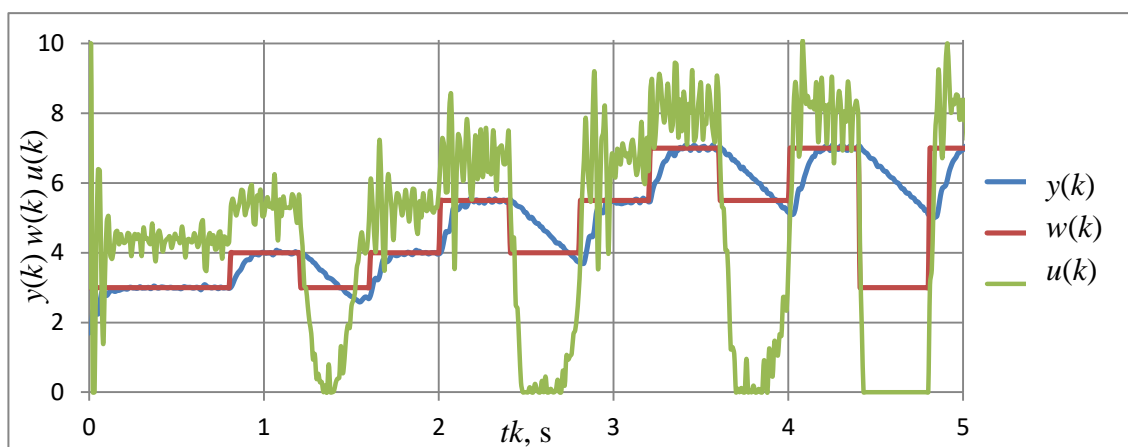


Obrázek 9.12 - PS regulátor pro relé ve zpětné vazbě při 10 ms

Ustálení regulační veličiny je pouze s minimálním překmitem na začátku měření při prvním zásahu regulátoru. Doba ustálení regulační veličiny je poměrně dlouhá a je více zatížena šumem. Oproti stejnému typu regulátoru, jehož parametry jsou získány pro periodu vzorkování 1 ms, je tento průběh výrazně pomalejší s delší dobou ustálení.

V případě soustavy motor generátor je neklidný průběh žádanější, aby regulátor nepůsobil akční veličinou s velkým rozsahem napětí.

A poslední regulátor je typu PSD získaný opět z kritických hodnot určených z relé ve zpětné vazbě pro periodu vzorkování 10 ms.



Obrázek 9.13 - PSD regulátor pro relé ve zpětné vazbě při 10 ms

Volba regulátoru je podle průběhu regulačního pochodu nejlepší. Regulovaná veličina se ustálí na žádané hodnotě za poměrně krátkou dobu. Je zde patrné působení šumu, ale v tomto případě si s tím nastavení regulátoru poradí s minimálním překmitem.

Pro zhodnocení výsledků regulace všech použitých regulátorů je v tabulce 9.1 vypočítána směrodatná odchylka chyby regulace.

Tabulka 9.1 – Směrodatné odchylky chyby regulace

|       | PS           |        | PSD          |        |
|-------|--------------|--------|--------------|--------|
|       | Identifikace | Relé   | Identifikace | Relé   |
| 1 ms  | 0,9409       | 1,0465 | 0,8860       | 0,8730 |
| 10 ms | 1,0327       | 1,0876 | 1,0227       | 1,0705 |

Podle výsledků v tabulce 9.1 lze usoudit, že parametry regulátorů určené pro periodu vzorkování 1 ms vycházejí lépe než pro větší periodu vzorkování 10 ms. A to pro oba typy regulátorů PS a PSD, jejichž parametry jsou naměřeny oběma metodami z kritických hodnot pomocí identifikace a relé ve zpětné vazbě.

U všech použitých nastavení regulátorů se projevuje rozdílná dynamika soustavy motor generátor, jednak při zvyšování otáček a jednak při snižování otáček motoru, kdy snižování otáček je mnohem pomalejší. Proto zvolená identifikace a identifikovaný model nemusí přesně odpovídat dynamice soustavy. A tudíž nastavení regulátorů nemusí být úplně správné a vypočítaná směrodatná odchylka chyby regulace v tabulce 9.1 nemusí být pro porovnání kvality regulace přesná.

## 10 ZÁVĚR

Cílem diplomové práce bylo určit diskrétní dynamický model systému motor generátor. Navrhnout jednoduchý diskrétní regulátor pro určený model a implementovat ho do řídicího systému AMiNi4DS.

Diskrétní dynamický model soustavy motor generátor byl získán experimentální identifikací. Podle zvoleného kritéria model nejlépe odpovídá dynamickému chování soustavy. Identifikovaný model určený metodou nejmenších čtverců ovšem neodpovídá dokonale dynamice systému, protože systém má odlišnou dynamiku při reakci na náběžnou hranu a při reakci na sestupnou hranu vstupního signálu. Soustava tedy vykazuje asymetrické dynamické chování.

Navržený diskrétní regulátor je dvojího typu pro porovnání výsledků identifikované dynamiky systému. Parametry PS a PSD regulátoru jsou získány z analogických spojitých regulátorů, jejichž parametry byly spočteny z kritických hodnot z experimentální identifikace a z použití relé ve zpětné vazbě. Parametry jsou vypočteny pro dva vzorkovací intervaly řídicího systému AMiNi4DS pro 1 ms a pro 10 ms. Regulátory jsou implementovány do řídicího systému AMiNi4DS přes návrhové prostředí DetStudio.

Výsledné regulační pochody pro oba typy regulátorů splňují funkci regulace, ale výsledek je do jisté míry ovlivněn asymetrickou dynamikou systému. Regulátor při reakci na kladnou skokovou změnu reguluje správně, ovšem při skokové změně na nižší hodnotu napětí regulátor neovlivní rychlost snižování otáček. Kvalita regulace byla posouzena na základě výpočtu směrodatné odchylky chyby regulace.

## POUŽITÁ LITERATURA

- AMIT, SPOL. s.r.o. 2011. DetStudio – průvodce první aplikací. [Online] AMiT. [Citace: 2. 5. 2017]. Dostupné z: [http://www.amit.cz/docs/cz/sw/detstudio\\_g\\_cz\\_104.pdf](http://www.amit.cz/docs/cz/sw/detstudio_g_cz_104.pdf).
- AMIT, SPOL. s.r.o. 2009a. Kompaktní řídicí systém s rozhraním Ethernet – návod na obsluhu. [Online] AMiT. [Citace: 2. 5. 2017]. Dostupné z: <http://www.amit.cz/docs/cz/obsolete/aminim.pdf>.
- AMIT, SPOL. s.r.o. 2012. Aplikační poznámka – Komunikace v síti Ethernet. [Online] AMiT. [Citace: 2. 5. 2017]. Dostupné z: [http://www.amit.cz/support/cz/aplikacni\\_poznamky/ap0006\\_cz\\_04.pdf](http://www.amit.cz/support/cz/aplikacni_poznamky/ap0006_cz_04.pdf).
- AMIT, SPOL. s.r.o. 2013. Aplikační poznámka – Parametrizace AtouchX. [Online] AMiT. [Citace: 2. 5. 2017]. Dostupné z: [http://www.amit.cz/support/cz/aplikacni\\_poznamky/ap0013\\_cz\\_02.pdf](http://www.amit.cz/support/cz/aplikacni_poznamky/ap0013_cz_02.pdf).
- AMIT, SPOL. s.r.o. 2009b. Aplikační poznámka – Příklad použití OPC serveru. [Online] AMiT. [Citace: 2. 5. 2017]. Dostupné z: [http://www.amit.cz/support/cz/aplikacni\\_poznamky/ap0020\\_cz\\_01.pdf](http://www.amit.cz/support/cz/aplikacni_poznamky/ap0020_cz_01.pdf).
- BALÁTĚ, J. 2003. *Automatické řízení*. Praha: BEN – technická literatura. 664 s. ISBN 80-7300-020-2.
- CVEJN, J. 2007. Elektronický studijní materiál k předmětu Automatizace I. *Řízení procesů* [online]. Pardubice: Univerzita Pardubice, FEI. 1. 11. 2007, s. 112 [cit. 2015-04-22].
- DUŠEK, F. 2000. *MATLAB a SIMULINK: úvod do používání*. Vyd. 1. Pardubice: Univerzita Pardubice. 146 s., [4] s. barev. obr. příl. ISBN 80-7194-273-1.
- DUŠEK, F. 2017. Osobní sdělení. Pardubice: Univerzita Pardubice.
- HLAVA, J. 2000. *Prostředky automatického řízení II: analogové a číslicové regulátory, elektrické pohony, průmyslové komunikační systémy*. Vyd. 1. Praha: ČVUT, Strojní fakulta. 160 s. ISBN 80-01-02221-8.
- HONC, D. 2016. *Teorie řídicích systémů*. Konzultace v rámci předmětu Průmyslové řídicí systémy. Pardubice: Univerzita Pardubice. Osobní sdělení.
- HORŇÁK, R. 2010. *Příprava laboratorních úloh pro předmět Identifikace náhodných procesů*. Zlín. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně. Vedoucí: M. Kubalčík.
- KOŘÍNEK, V. 2015. *Metody nastavení PID regulátoru pro statické systémy 1. řádu s dopravním zpožděním*. Pardubice. Bakalářská práce. Univerzita Pardubice. Vedoucí : J. Cvejn.
- MathWorks – MATLAB and SIMULINK for Technical Computing* [online]. 1994-2015. [cit. 2015-04-22]. Dostupné z: <https://www.mathworks.com/>
- MODRLÁK, O; HUBKA L. 2010. *Otáčky DC motoru – “DC motor se zátěží”*. Liberec. Laboratorní úloha. Technická Univerzita v Liberci.
- OGATA, K. 1995. *Discrete-time control systems*. 2nd ed. London: Prentice-Hall International. 760 s. ISBN 0-13-328642-8.
- VROŽINA, M; JANČÍKOVÁ, Z; DAVID, J. 2012. *Identifikace systémů: učební text*. Ostrava: Vysoká škola báňská – Technická univerzita. ISBN 978-80-248-2594-6. Vedoucí: F. Němec.

## **PŘÍLOHY**

**A - CD**

**Příloha k diplomové práci**

Identifikace a řízení systému motor generátor

Bc. Vladislav Kořínek

**CD**

## **Obsah**

- 1 Text diplomové práce ve formátu PDF
- 2 Úplný zdrojový kód aplikace (DetStudio, MATLAB, Excel)