

**UNIVERZITA PARDUBICE**  
Fakulta elektrotechniky a informatiky

**VYTVOŘENÍ UŽIVATELSKÉHO ROZHRANÍ PRO SOUSTAVU GUNT**

Milan Míšenský

Bakalářská práce  
2017

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2016/2017

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Milan Míšenský**  
Osobní číslo: **I14049**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Řízení procesů**  
Název tématu: **Vytvoření uživatelského rozhraní pro soustavu GUNT**  
Zadávací 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: Vytvoření uživatelského rozhraní pro laboratorní soustavu GUNT, které umožní proměřit statické a dynamické vlastnosti soustavy a provést regulační experimenty.

Obsah teoretické části: Statické a dynamické vlastnosti soustavy, regulace, PID regulátor, GUI výpočetního systému MATLAB.

Obsah implementační části: Návrh SW, komunikace soustavy GUNT se systémem MATLAB, příklady použití SW s výsledky experimentů.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**BALÁŤE, J. Automatické řízení. 2 vyd. Praha: BEN - technická literatura, 2004.**  
**DUŠEK, F., HONC, D. Matlab a Simulink, Úvod do používání. skriptum, Univerzita Pardubice, vydání první, Pardubice, 2005.**  
**HONC, D., DUŠEK, F. MATLAB/Simulink Support for GUNT Control Units. In: 19th International Conference on Process Control 2013, Štrbské Pleso, Slovakia, June 18 - 21 2013, Slovak University of Technology in Bratislava, 2013, pp. 534-539.**

Vedoucí bakalářské práce:

**Ing. Daniel Honc, Ph.D.**

Katedra řízení procesů

Datum zadání bakalářské práce:

**7. prosince 2016**

Termín odevzdání bakalářské práce:

**12. května 2017**



Ing. Zdeněk Němec, Ph.D.  
děkan



L.S.



Ing. Daniel Honc, Ph.D.  
vedoucí katedry

V Pardubicích dne 15. prosince 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 5. 3. 2017

Milan Míšenský

### **Poděkování**

Rád bych touto cestou poděkoval vedoucímu práce Ing. Danielu Honcovi, Ph.D. za pomoc, poskytnuté rady a poznatky k řešené problematice během tvorby této práce.

V Pardubicích dne 5. 3. 2017

Milan Mišenský

## **ANOTACE**

*Cílem této práce bylo vytvořit uživatelské rozhraní pro laboratorní soustavu GUNT RT 010. GUI je vytvořeno v prostředí MATLAB. Uživatelské rozhraní umožňuje změřit statické a dynamické vlastnosti soustavy a provádět regulační experimenty. Funkce navrženého GUI jsou demonstrovány na příkladech regulačních pochodů.*

## **KLÍČOVÁ SLOVA**

*Návrh SW, GUI systému MATLAB, regulace, PID regulátor.*

## **TITLE**

*USER INTERFACE FOR LABORATORY SYSTEM GUNT*

## **ANNOTATION**

*The aim of this study was to develop a user interface for laboratory system GUNT RT 010. The GUI is created in MATLAB. The user interface allows to measure static and dynamic properties of the system and perform control experiments. Functionality of designed GUI is demonstrated on selected control examples.*

## **KEYWORDS**

*SW design, MATLAB GUI, Control, PID controller.*

## OBSAH

SEZNAM ZKRATEK A ZNAČEK .....	9
SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ.....	10
SEZNAM ILUSTRACÍ .....	11
SEZNAM TABULEK.....	12
ÚVOD .....	13
1 TEORETICKÁ ČÁST .....	14
1.1 SYSTÉM.....	14
1.1.1 Vlastnosti systému .....	14
1.1.2 Určování vlastností systému vyhodnocováním přechodových charakteristik .....	17
1.2 REGULACE .....	18
1.2.1 Regulované soustavy.....	18
1.2.2 Regulátory .....	20
1.2.3 PID regulátor .....	21
1.2.4 Parametry regulátorů.....	23
1.3 MATLAB.....	24
1.3.1 Příkazy a proměnné.....	25
1.3.2 M-soubory .....	25
1.3.3 Grafické objekty.....	26
1.3.4 Identifikátory objektů.....	28
1.3.5 Vlastnosti objektů .....	28
1.3.6 GUI systému MATLAB.....	29
1.4 LABORATORNÍ STANICE GUNT RT 010.....	32
1.5 MĚŘICÍ KARTA LABJACK U12 .....	33
1.5.1 Analogové vstupy .....	35
1.5.2 Analogové výstupy .....	35
2 IMPLEMENTAČNÍ ČÁST .....	36
2.1 KOMUNIKACE SOUSTAVY GUNT SE SYSTÉMEM MATLAB.....	36
2.1.1 Propojení soustavy GUNT s LabJack U12 .....	36
2.1.2 Využití knihoven při komunikaci s LabJack U12.....	37
2.2 NÁVRH SOFTWARE.....	38
2.2.1 Koncept .....	38
2.2.2 Vizuální návrh.....	39

2.2.3	Práce s GUI .....	42
2.2.4	Využití PID regulátoru.....	46
2.3	PŘÍKLADY POUŽITÍ GUI.....	47
2.3.1	Experiment v ručním režimu.....	48
2.3.2	Měření odezvy na skokovou změnu žádané hodnoty výšky hladiny z 5 na 10 cm .....	49
2.3.3	Měření odezvy na skokovou změnu žádané hodnoty výšky hladiny z 12 na 5 cm .....	50
2.3.4	Měření odezvy na skokové změny žádané hodnoty výšky hladiny .....	51
3	ZÁVĚR .....	53
	POUŽITÁ LITERATURA .....	54
	PŘÍLOHY .....	55

## SEZNAM ZKRATEK A ZNAČEK

ASCII	American Standard Code for Information Interchange
EOL	End Of Line
GUI	Graphical User Interface
ID	Identification
PC	Personal Computer
PID	proporcionálně integračně derivační regulátor
R	regulátor
S	system
USB	Universal Serial Bus

## SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ

$a$	konstantní reálný koeficient
$e$	regulační odchylka
$E(s)$	přenos odchylky
$G(s)$	obrazový přenos
$G_R(s)$	přenos regulátoru
$G_S(s)$	přenos soustavy
$G_{S_{Vi}}(s)$	přenos poruchy regulované soustavy
$h$	výška hladiny
$I$	elektrický proud, A
$n$	nejvyšší stupeň derivace výstupní veličiny
$pp$	pásmo proporcionality, %
$q$	akční zásah
$Q$	průtok, $\text{m}^3 \cdot \text{s}^{-1}$
$R$	elektrický odpor, $\Omega$
$r_0$	proporcionální konstanta regulátoru
$r_{-1}$	integrační konstanta regulátoru
$r_1$	derivační konstanta regulátoru
$s$	komplexní proměnná
$t$	čas, s
$T$	perioda vzorkování, časová konstanta systému, s
$T_d$	derivační časová konstanta, s
$T_i$	integrační časová konstanta, s
$u$	vstup systému, akční veličina
$U$	elektrické napětí, V
$U(s)$	Laplaceův obraz akční veličiny
$v$	poruchová veličina
$V$	objem, $\text{m}^3$
$V_i$	přenos poruchové veličiny
$w$	žádaná hodnota
$y$	výstup systému, regulovaná veličina
$Y(s)$	Laplaceův obraz výstupní veličiny

## SEZNAM ILUSTRACÍ

Obr. 1.1 – Statická charakteristika.....	15
Obr. 1.2 – Heavisideův jednotkový skok.....	16
Obr. 1.3 – Diracův impulz .....	16
Obr. 1.4 – Regulační obvod .....	18
Obr. 1.5 – Přejchodová charakteristika ideálního PID regulátoru .....	22
Obr. 1.6 – Struktura grafických objektů .....	26
Obr. 1.7 – Vkládání objektů v editoru GUI .....	30
Obr. 1.8 – Atributy objektu Edit Text.....	30
Obr. 1.9 – Stanice GUNT RT 010 .....	32
Obr. 1.10 – USB měřicí karta LabJack U12 .....	33
Obr. 2.1 – Základní podoba GUI .....	39
Obr. 2.2 – Panel: Průběh měření.....	40
Obr. 2.3 – Panel: Podmínky experimentu .....	40
Obr. 2.4 – Panel: Manual/Automat .....	41
Obr. 2.5 – Panel: Volba režimu .....	41
Obr. 2.6 – Panel: Ukazatel výšky hladiny .....	42
Obr. 2.7 – Soubor data.txt.....	44
Obr. 2.8 – Soubor manual.txt.....	45
Obr. 2.9 – Soubor automat.txt.....	45
Obr. 2.10 – GUI, měření v ručním režimu.....	48
Obr. 2.11 – GUI, odezva na změnu žádané hodnoty výšky hladiny z 5 na 10 cm .....	49
Obr. 2.12 – GUI, odezva na změnu žádané hodnoty výšky hladiny z 12 na 5 cm .....	50
Obr. 2.13 – GUI, odezva na skokové změny žádané hodnoty výšky hladiny .....	51
Obr. 2.14 – Vstupní data, odezva na skokové změny.....	52

## SEZNAM TABULEK

Tab. 1.1 – Zapojení vývodů svorkovnic .....	34
Tab. 1.2 – Zapojení vývodů konektoru .....	34
Tab. 1.3 – Napěťový rozsah analogových vstupů.....	35
Tab. 1.4 – Závislost výstupní úrovně na odebíraném proudu.....	35
Tab. 2.1 – Vstupní hodnoty experimentu .....	48

# ÚVOD

Při použití laboratorních zařízení ve výuce vzniká požadavek na tvorbu vlastního SW pro monitorování a řízení. Lze použít standardní programovací jazyky a prostředí. Jedním z nich je grafické uživatelské rozhraní (GUI) MATLAB. Pomocí GUI lze vytvářet uživatelské prostředí na míru požadavkům pro použití zařízení, takovým způsobem, který SW dodávaný výrobcem často ani neumožňuje.

Cílem této práce je vytvoření uživatelského rozhraní pro laboratorní soustavu GUNT RT 010, které umožní proměřit statické a dynamické vlastnosti soustavy a provést regulační experimenty.

Teoretická část obsahuje popis statických a dynamických vlastností soustavy, regulace, použitého PID regulátoru a tvorbu GUI pomocí výpočetního systému MATLAB.

V praktické části je uveden návrh GUI pro laboratorní soustavu GUNT, způsob komunikace soustavy se systémem MATLAB a příklad využití GUI s výsledky experimentu.

# 1 TEORETICKÁ ČÁST

## 1.1 SYSTÉM

Jako systém považujeme soubor prvků, mezi nimiž existují vzájemné vztahy a jako celek má určité vazby ke svému okolí (Balátě, 2003).

Každý systém obsahuje jednotlivé prvky, které jsou jeho základními částmi. Těmito prvky rozumíme relativně uzavřený systém nižšího řádu rozlišovací úrovně vzhledem k uvažovanému systému S. Prvek se vyznačuje tím, že z hlediska příslušné rozlišovací úrovně (na níž je systém S definován) tvoří nedělitelný celek.

Rozlišovací úroveň je myšlena jako úroveň omezující rozeznávací schopnosti jen k určité hranici odpovídající požadavkům vyšetřování daného systému S.

Prvek tedy pro danou rozlišovací úroveň již dále nečleníme i kdybychom jeho strukturu znali (Balátě, 2003).

Analýzou rozumíme vyšetřování chování daného existujícího systému. Syntéza je pak typem úlohy, která umožní tvorbu návrhu vhodné struktury nového systému, případně upravení struktury za účelem dosažení požadovaného chování, u již existujícího systému.

Schopnost řešit změny chování systému je společným předpokladem k dosažení vlastního cíle analýzy i syntézy. Změna chování daného systému je podmíněna změnou jeho struktury. Toho lze docílit:

- Změnou počtu prvků systému, to má za následek nejen změnu množiny chování prvků, ale často i změnu organizace vazeb jednotlivých prvků systému.
- Změnou chování prvků systému, při zachování jejich počtu. To má za následek pouze kvalitativní změnu množiny chování prvků struktury při zachování organizace vazeb.
- Změnou organizace vazeb u prvků systému při zachování jejich počtu.
- Kombinací předchozích možností.

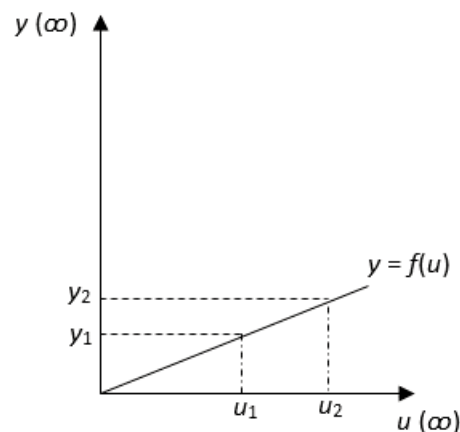
### 1.1.1 Vlastnosti systému

Dynamické vlastnosti systému určují jeho chování. Statické vlastnosti jsou limitním případem dynamických vlastností po doznění přechodového děje. Chování je vyjádřeno závislostí mezi výstupním a vstupním signálem systému. Průběh v čase vstupního signálu se někdy nazývá akcí nebo vzruchem, resp. podnětem a vyvolaný průběh výstupního signálu – reakcí nebo častěji odezvou systému (Balátě, 2003).

Statické a dynamické vlastnosti souvisí s matematickým popisem systému. Statické vlastnosti se projevují v ustáleném stavu systému, zatímco dynamické při přechodech mezi ustálenými stavy. Systém je v ustáleném stavu tehdy, pokud v něm nedochází ke změnám veličin v čase. Avšak existují i systémy které ustálený stav nemají.

Dynamický systém je takový, který se v čase vyvíjí a jehož veličiny se v čase mění. Výsledkem analýzy dynamického systému jsou tedy jeho dynamické vlastnosti. Pro určitým způsobem definované vstupy jsou vyjádřením těchto vlastností dynamické charakteristiky, tj. časové průběhy výstupních veličin pro danou třídu vstupů (Třídění systémů, 2006).

Statické vlastnosti systému v ustáleném stavu se dají vyjádřit statickou charakteristikou, která vyjadřuje závislost ustálených hodnot výstupní veličiny na ustálených hodnotách vstupní veličiny. U lineárních dynamických členů statická charakteristika, pokud existuje, je vždy přímka procházející počátkem souřadnic. Obecně však bývá tato závislost nelineární, pak provádíme její linearizaci v okolí pracovního bodu (Vrožina, 2012).



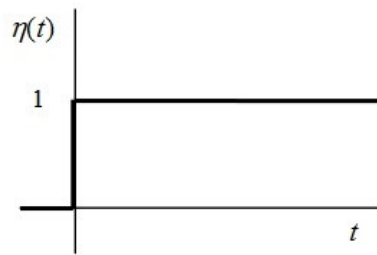
Obr. 1.1 – Statická charakteristika

U dynamických systémů je nejjednodušším způsobem zjišťování této závislosti metoda určování bod po bodu. Postupně nastavujeme hodnoty vstupní veličiny a měříme odpovídající hodnoty výstupní veličiny. Přitom dbáme, aby se neuplatnila dynamika systému. To zajistíme odečítáním výstupní veličiny až po určité době od změny vstupní veličiny, tj. až se soustava dostane do ustáleného stavu (Vrožina, 2012).

Pro úplný popis systému musíme zjistit také dynamické vlastnosti systému, tedy závislost výstupní veličiny v čase. Pro jejich určení zavádíme na vstup soustavy předem definované signály. Ty jsou buď neperiodické, nebo periodické.

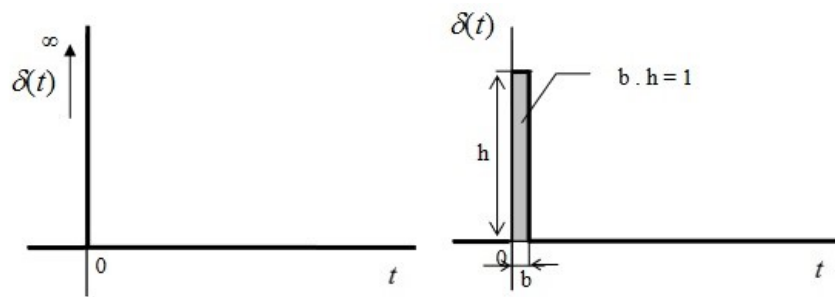
Nejvíce využívané neperiodické signály jsou:

Heavisideův jednotkový skok  $\eta(t)$ ,



Obr. 1.2 – Heavisideův jednotkový skok

Diracův jednotkový impulz  $\delta(t)$ ,



Obr. 1.3 – Diracův impulz

Nejvíce využívané periodické signály jsou:

- sinusový průběh,
- sled pravoúhlých impulsů,
- sled lichoběžníkových impulsů,
- sled trojúhelníkových impulsů.

Odezva na jednotkový Heavisideův skok je přechodová funkce  $h(t)$ . Jejím grafickým vyjádřením je přechodová charakteristika. Odezvou systému na vstupní signál ve tvaru Diracova impulsu je impulsní funkce  $g(t)$  a jejím grafickým vyjádřením je impulsní charakteristika (Vrožina, 2012).

### 1.1.2 Určování vlastností systému vyhodnocováním přechodových charakteristik

Měřením se zjišťuje odezva  $y(t)$  soustavy při změně vstupního signálu  $u(t)$  skokem známé velikosti. Před provedením změny musí být soustava v ustáleném stavu. Změna vstupního signálu se obvykle provede přestavením regulačního členu. Průběh výstupní veličiny  $y(t)$  se zaznamenává vhodným měřicím zařízením (Vrožina, 2012).

Skoková změna vstupní veličiny musí proběhnout tak rychle, aby doba jejího přechodu z výchozí do konečné polohy byla mnohem kratší, než je odezva zkoumaného členu. Vlastní přechod musí být monotónní. Odezva měřicího zařízení musí být mnohem rychlejší, než je odezva měřeného členu. Při měření charakteristik soustavy s krátkými časovými konstantami je nutno přivádět na vstup soustavy opakující se impulzy (Vrožina, 2012).

Pro svou jednoduchost a nenákladnost se nejčastěji používá měření přechodových charakteristik. Tvar přechodové charakteristiky nezávisí jen na řádu soustavy, ale i na hodnotách nul a pólů přenosu. Většina reálných soustav neobsahuje v přenosu nuly, ale reálné póly.

Použití těchto metod je vhodné, je-li náhodný šum na výstupu soustavy zanedbatelný a jedná-li se o soustavy s velkými časovými konstantami. Důležité informace nalezneme v samotném okolí počátku přechodové charakteristiky (rozhodují o řádu soustavy). Velikost přestavení regulačního členu (nastavení hodnoty vstupní veličiny) se volí podle skutečných podmínek, za kterých sledovaná soustava pracuje. Musí být dostatečně veliká, aby vedlejší poruchy neovlivňovaly průběh odezvy. Velká přestavení ale nejsou žádoucí, jelikož mohou silně narušit režim soustavy a mohou se nepříznivě projevit i nelinearity systému. Na základě naměřeného průběhu odvozujeme (přepočítáváme) přechodovou charakteristiku jako odezvu na jednotkový skok, tj. na změnu vstupní veličiny o jednotku.

Přesné určení dynamických vlastností soustavy podle záznamů přechodových charakteristik běžně používanými měřicími přístroji je prakticky nemožné. Proto se vyhodnocování přechodových charakteristik zpravidla spojuje s aproximací skutečných vlastností soustavy (Vrožina, 2012).

## 1.2 REGULACE

### 1.2.1 Regulované soustavy

Regulace je technologický proces řízený regulátorem, jenž působí v regulované soustavě. Regulovaná soustava je jednou z částí uzavřeného regulačního obvodu,

kde  $S$  – regulovaná soustava (řízený systém),

$R$  – regulátor (řídící systém),

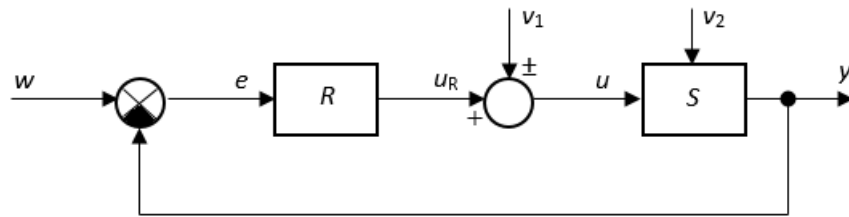
$y$  – regulovaná veličina,

$u$  – akční veličina,

$v_1, v_2$  – poruchové veličiny,

$w$  – žádaná hodnota,

$e$  – regulační odchylka.



Obr. 1.4 – Regulační obvod

Podle obr. 1.4 můžeme snadno definovat dynamické vlastnosti regulované soustavy, tj. přenos soustavy  $G_S(s)$  a přenos poruchy soustavy  $G_{Sv_i}(s)$  z následující rovnice, vyjádřené v oblasti komplexní proměnné (v prostoru obrazů)

$$Y(s) = G_S(s) \cdot U(s) + G_{Sv_i}(s) \cdot V_i(s), \quad (1.1)$$

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

$G_S(s)$  – přenos regulované soustavy,

$U(s)$  – Laplaceův obraz akční veličiny,

$G_{Sv_i}(s)$  – přenos poruchy soustavy,

$V_i(s)$  – přenos poruchové veličiny.

Přenos regulované soustavy je tedy

$$G_S(s) = \frac{Y(s)}{U(s)}, \quad (1.2)$$

za předpokladu, že

$$V_i(t) = 0, \quad (1.3)$$

a přenos poruchy regulované soustavy

$$G_{S_{V_i}}(s) = \frac{Y(s)}{V_i(s)}, \quad (1.4)$$

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

$G_{S_{V_i}}(s)$  – přenos poruchy soustavy,

$V_i(s)$  – přenos poruchové veličiny,

za předpokladu, že

$$u_R(t) = 0 \text{ a } i = 1, 2, \dots \quad (1.5)$$

kde  $u_R(t)$  – akční veličina regulátoru,

$i$  – index poruchové veličiny.

Častým způsobem vyjadřování dynamických vlastností regulované soustavy je lineární diferenciální rovnice s konstantními koeficienty spolu se zadanými počátečními podmínkami, která pro  $n$ -tý řád setrvačnosti má tvar

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \dots + a_2 y''(t) + a_1 y'(t) + a_0 y(t) = u(t), \quad (1.6)$$

kde  $y^{(n)}(t)$  – výstupní veličina,

$u(t)$  – akční veličina,

$a$  – konstantní reálný koeficient,

$n$  – řád derivace,

a z které pro nulové počáteční podmínky (celkem  $n$  počátečních podmínek)

$$y(0) = y'(0) = \dots = y^{(n-1)}(0) = 0, \quad (1.7)$$

a známý tvar vstupního signálu  $u(t)$  můžeme použitím Laplaceovy transformace stanovit přenos regulované soustavy

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{a_0 + a_1 s + a_2 s^2 + \dots + a_n s^n}, \quad (1.8)$$

kde  $G(s)$  – přenos regulované soustavy,

$Y(s)$  – Laplaceův obraz výstupní veličiny,

$U(s)$  – Laplaceův obraz akční veličiny.

U regulovaných soustav je důležitým koeficientem tzv. součinitel autoregulace  $a_0$ , který určuje, zda se jedná o regulovanou soustavu proporcionální nebo integrační a to:

při  $a_0 \neq 0$  – jedná se o proporcionální soustavu,

$a_0 = 0$  – jedná se o integrační soustavu.

## 1.2.2 Regulátory

Regulátorem v regulačním obvodu je nazýváno zařízení, kterým se provádí proces automatické regulace. Regulátor pracuje na základě vyhodnocení regulační odchylky

$$e(t) = w(t) - y(t), \quad (1.9)$$

kde  $e(t)$  – regulační odchylka,

$w(t)$  – žádaná veličina,

$y(t)$  – výstupní veličina.

Tu regulátor zpracovává a změnou akční veličiny se regulační odchylku pokouší odstranit, nebo alespoň minimalizovat.

Regulátory dělíme podle několika kritérií. Pro základní rozdělení regulátorů můžeme uvažovat vliv přívodu energie na:

– přímé regulátory,

– nepřímé regulátory.

Přímé regulátory nevyžadují pro svoji činnost vnější zdroj energie, k ovládní akční veličiny v tomto případě postačí vlastní energie snímače. Zatímco pro činnost nepřímých regulátorů je nutný vnější přívod energie. Dnes se používají převážně nepřímé regulátory (Balátě, 2003).

Podle druhu nositele signálu v regulátoru lze rozlišit:

– elektrické regulátory,

– hydraulické regulátory,

– pneumatické regulátory a jiné.

Často se používají i kombinované regulační systémy, které využívají předností zkombinovaných systémů.

Dále můžeme rozlišit regulátory podle průběhu výstupní veličiny, a to na:

- spojité regulátory,
- nespojité regulátory.

Při práci se spojitými regulátory jsou všechny veličiny spojité v čase, u nespojitých regulátorů pracuje některý z jeho členů nespojitě (Balátě, 2003).

### 1.2.3 PID regulátor

Patří k nejpoužívanějším regulátorům. PID regulátor je tvořen třemi složkami. Jedná se o paralelní spojení proporcionální, integrační a derivační složky regulátoru. Nastavením parametrů těchto složek dosáhneme požadovaných vlastností regulátoru. Jedná se tedy o regulátor kombinovaný.

Činnost kombinovaného regulátoru lze popsat lineární integrodiferenciální rovnicí

$$\dots + T_2^2 u''(t) + T_1 u'(t) + u(t) = r_0 e(t) + r_{-1} \int_0^t e(\tau) d\tau + r_1 \frac{de(t)}{dt}, \quad (1.10)$$

kde  $r_0 e(t)$  – proporcionální složka regulátoru,

$r_{-1} \int_0^t e(\tau) d\tau$  – integrační složka regulátoru,

$r_1 \frac{de(t)}{dt}$  – derivační složka regulátoru,

$\dots, T_2^2 u''(t), T_1 u'(t)$  – zpožďující členy regulátoru.

Provedeme-li Laplaceovu transformaci rovnice (1.2), při nulových počátečních podmínkách, získáme přenos skutečného PID regulátoru

$$G_R(s) = \frac{U(s)}{E(s)} = \frac{r_0 + \frac{r_{-1}}{s} + r_1 s}{1 + T_1 s + T_2^2 s^2 \dots} = \frac{r_0 \left( 1 + \frac{r_{-1}}{r_0} \cdot \frac{1}{s} + \frac{r_1}{r_0} s \right)}{1 + T_1 s + T_2^2 s^2 \dots} = \frac{r_0 \left( 1 + \frac{1}{T_i s} + T_d s \right)}{1 + T_1 s + T_2^2 s^2 \dots}, \quad (1.11)$$

kde  $r_0$  – proporcionální konstanta regulátoru,

$r_{-1}$  – integrační konstanta regulátoru,

$r_1$  – derivační konstanta regulátoru,

$T_i = \frac{r_0}{r_{-1}}$  – integrační časová konstanta regulátoru, s,

$T_d = \frac{r_1}{r_0}$  – derivační časová konstanta regulátoru, s,

$G_R(s)$  – přenos PID regulátoru,

$E(s)$  – Laplaceův obraz regulační odchylky,

$U(s)$  – Laplaceův obraz akční veličiny.

Pokud časové konstanty zpožďujících členů ( $T_1, T_2, \dots$ ) položíme rovny nule, dostaneme pohybovou rovnici i přenos ideálního PID regulátoru (Balátě, 2003).

Pohybová rovnice ideálního PID regulátoru

$$u(t) = r_0 e(t) + r_{-1} \int_{\tau=0}^t e(\tau) d\tau + r_1 \frac{de(t)}{dt}, \quad (1.12)$$

kde  $u(t)$  – akční veličina,

$r_0 e(t)$  – proporcionální složka regulátoru,

$r_{-1} \int_0^t e(\tau) d\tau$  – integrační složka regulátoru,

$r_1 \frac{de(t)}{dt}$  – derivační složka regulátoru.

Přenos ideálního PID regulátoru

$$G_R(s) = \frac{U(s)}{E(s)} = r_0 \left( 1 + \frac{1}{T_i s} + T_d s \right), \quad (1.13)$$

kde  $G_R(s)$  – přenos PID regulátoru,

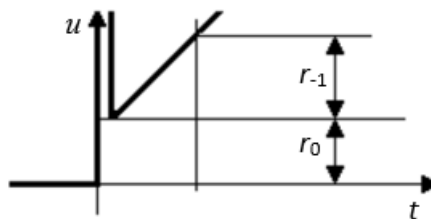
$E(s)$  – Laplaceův obraz regulační odchylky,

$U(s)$  – Laplaceův obraz akční veličiny,

$r_0$  – proporcionální konstanta regulátoru,

$T_i$  – integrační časová konstanta, s,

$T_d$  – derivační časová konstanta, s.



Obr. 1.5 – Přejchodová charakteristika ideálního PID regulátoru

## 1.2.4 Parametry regulátorů

Pásmo proporcionality

$$pp = \frac{1}{r_0} \cdot 100, \quad (1.14)$$

kde  $pp$  – pásmo proporcionality, %,

$r_0$  – proporcionální konstanta regulátoru,

určuje hodnotu v procentech o kterou se musí změnit vstupní signál regulátoru, aby se akční člen přestavil z jedné krajní polohy (0 %) do druhé (100 %), nebo naopak. Při zmenšujícím se pásmu proporcionality (intervalu, ve kterém se mění skutečná hodnota regulované veličiny) se zvětšuje zesílení regulátoru (Balátě, 2003).

Integrační časová konstanta ( $T_i$ , s) je čas, který by potřeboval čistě integrační regulátor, aby přestavil akční člen (vstupní signál) do polohy, které dosáhne PI regulátor v čase  $t = 0$  s vlivem své proporcionální složky. Určíme ji z přechodové charakteristiky PI regulátoru, tj. pro vstupní signál rovný jednotkovému skoku (Balátě, 2003).

Derivační časová konstanta ( $T_d$ , s) je čas, který by potřeboval čistě proporcionální regulátor, aby přestavil akční člen do polohy, které dosáhne PD regulátor v čase  $t = 0$  s, vlivem své derivační složky. Určíme ji u PD regulátoru pro vstupní signál rovný jednotkové rychlosti (Balátě, 2003).

Podle toho, které z konstant  $r_0$ ,  $r_{-1}$ ,  $r_1$  položíme rovno nule, dostáváme základní druhy regulátorů. Základními druhy regulátorů jsou P regulátor, I regulátor a kombinované regulátory PD, PI a PID. Tyto regulátory můžeme uvažovat jako ideální (bez zpožďujících členů), nebo jako skutečné (se zpožďujícími členy) (Balátě, 2003).

Integrační regulátor se nepoužívá, jelikož v uzavřeném regulačním obvodu pracuje pouze s přechodnou regulační odchylkou a nevyhoví podmínkám stability regulačního obvodu. Derivační regulátor není schopen samostatné funkce jako regulátor připojený k regulované soustavě, protože vstupním signálem je derivace regulační odchylky, a tedy neví nic o velikosti (hodnotě) odchylky  $e(t)$  (Balátě, 2003).

PID regulátor v uzavřeném regulačním obvodu dosáhne vlivem I složky odstranění trvalé regulační odchylky a D složka zlepšuje stabilitní vlastnosti regulačního obvodu.

## 1.3 MATLAB

MATLAB je vysoce výkonný jazyk pro technické výpočty. Integruje výpočty, vizualizaci a programování do jednoduše použitelného prostředí, kde problémy i řešení jsou vyjádřeny v přirozeném tvaru (Dušek, 2000).

Používá se často v univerzitním prostředí, ale i v průmyslu jako vysoce efektivní nástroj využívaný k výzkumu, vývoji a analýze dat.

Využívá se především pro:

- inženýrské výpočty,
- vývoj algoritmů,
- modelování, simulace a vývoj prototypů,
- analýza dat a jejich vizualizace,
- inženýrská grafika,
- vývoj aplikací včetně tvorby grafického uživatelského rozhraní.

MATLAB integruje numerickou analýzu, maticové výpočty, zpracování signálů a grafiku do uživatelsky příjemného prostředí, ve kterém se problémy a řešení zapisují stejně jako v matematice (bez tradičního programování) (Heringová, 1995).

MATLAB je interaktivní systém, jehož základní datový prvek je matice, u které se nezadáva rozměr. Díky tomu můžeme řešit mnoho numerických problémů podstatně rychleji než při použití klasických programovacích jazyků (Fortran, Basic, nebo C).

Jednou z nejoceňovanějších vlastností systému MATLAB je jeho snadná rozšiřitelnost, která vám umožňuje doplňovat systém o vámi napsané funkce (M-soubory) i celé aplikace. K systému MATLAB si můžeme navíc pořídit celou řadu specificky zaměřených nadstaveb. Tyto nadstavby nazýváme toolboxy, což je kolekce M-souborů, která je určena pro řešení jistých tříd problémů. Například jsou k dispozici nadstavby pro následující okruhy problémů:

- zpracování signálů,
- zpracování obrazů,
- teorie řízení,
- identifikace systémů,
- optimalizace,
- neuronové sítě,
- splíny,
- statistika,
- symbolická matematika (Heringová, 1995).

### 1.3.1 Příkazy a proměnné

MATLAB využívá výrazový jazyk. Výrazy, které napíšeme, jsou interpretovány a vyhodnoceny. Příkazy systému MATLAB jsou obvykle ve tvaru:

*Proměnná = výraz,*

nebo jen

*výraz.*

Výraz je obvykle složen z operátorů, funkcí a jmen proměnných. Výsledkem vyhodnocení výrazu je matice, která se zobrazí na obrazovce a je přiřazena do proměnné. V případě, že název proměnné a rovnítko chybí, je automaticky vytvořena proměnná ans (answer), do které je výsledek uložen.

Příkaz je obvykle ukončen znakem konce řádky (EOL), který je vyvolán stiskem tlačítka Enter. Pokud je posledním znakem příkazu středník, je potlačeno zobrazení výsledku. Toto se využívá především v M-souborech a v situacích, kde je lokální výsledek značně rozsáhlý nebo nezajímavý.

Názvy proměnných a funkcí musí začínat písmenem, za kterým následuje libovolný počet písmen, číslic a podtržíték. MATLAB však rozlišuje pouze prvních 19 znaků jména (Heringová, 1995).

MATLAB rozlišuje malá a velká písmena, tedy „X“ a „x“ nejsou stejné proměnné.

### 1.3.2 M-soubory

MATLAB se obvykle používá v příkazovém módu. Když zadáte jednořádkový příkaz, MATLAB ho okamžitě provede a zobrazí výsledky. Kromě toho může MATLAB také spouštět posloupnosti příkazů, které jsou uloženy v souborech.

Soubory, které obsahují příkazy prostředí MATLAB, se nazývají M-soubory, neboť mají příponu „.m“. M-soubory obsahují posloupnost běžných příkazů prostředí MATLAB, které se mohou dále odkazovat na jiné M-soubory. M-soubor může volat rekursivně sám sebe. M-soubor můžete vytvořit libovolným textovým editorem.

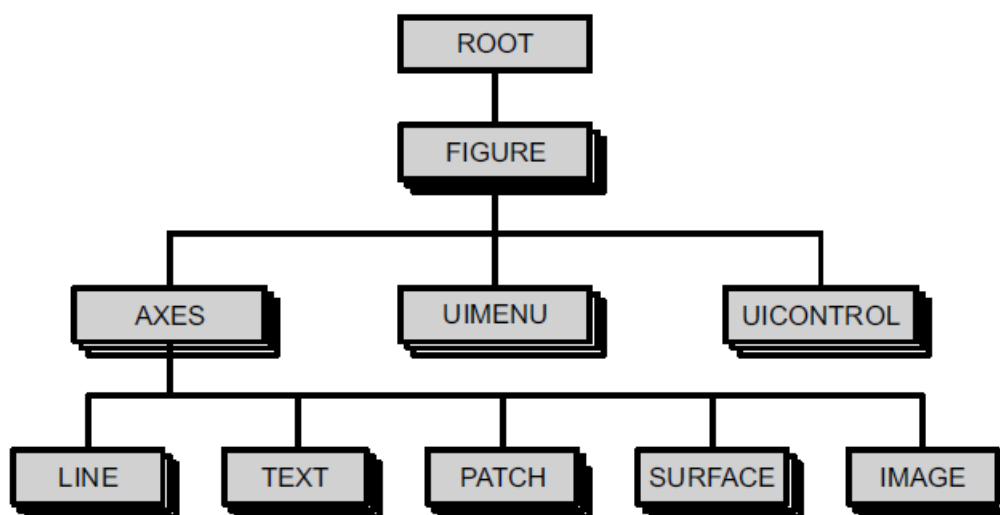
Rozlišujeme dva typy M-souborů: skripty a funkce. Skripty neboli skriptové soubory, automatizují dlouhé posloupnosti příkazů. Funkce neboli funkční soubory, poskytují prostředí MATLAB rozšiřitelnost. Funkce vám umožňují přidávat nové funkce k funkcím existujícím. Za svoji oblíbenost vděčí MATLAB v mnohém právě své schopnosti vytvářet nové funkce,

kteří řeší uživatelem specifikované problémy. Skripty i funkce jsou textové (ASCII) soubory (Heringová, 1995).

### 1.3.3 Grafické objekty

MATLAB definuje grafické objekty jako základní grafické jednotky svého grafického systému a organizuje je do stromově strukturované hierarchie. Tyto objekty zahrnují root (obrazovka), figure (grafické okno), axes (osy), line (čára), patch (vyplněný mnohoúhelník), surface (plocha), image (fotografie), text, uicontrol (uživatelem řízené rozhraní) a uimenu (uživatelské menu) (Heringová, 1995).

Následující obrázek ukazuje hierarchickou strukturu těchto objektů.



**Objekt root** je kořenem této hierarchie a odpovídá obrazovce počítače. Objekt root je jediný, všechny další objekty jsou jeho potomky.

**Objekty figure** jsou samostatná grafická okna. Těchto objektů může existovat libovolný počet, všechny jsou potomky objektu root. Všechny další grafické objekty jsou potomky objektu figure a jsou zobrazeny v tomto objektu (grafickém okně). Všechny funkce, které generují objekty, a všechny funkce vyšší úrovně vytvoří objekt figure i v případě, kdy tento objekt ještě neexistuje. Pomocí příkazu **figure** můžeme objekt figure vytvořit přímo.

**Objekty axes** definují oblast v grafickém okně a orientují své děti uvnitř této oblasti. Axes jsou dětmi objektů figure a jsou rodiči pro objekty line, surface, text, image a patch. Všechny funkce, které generují tyto objekty, a všechny funkce vyšší úrovně vygenerují objekt

axes i v případě, kdy tento objekt ještě neexistuje. Přímo lze objekt axes generovat funkcí axes (Heringová, 1995).

**Objekty line** jsou základní grafické jednotky užívané pro generování většiny 2-D a některých 3-D grafů. Jsou dětmi objektu axes a jejich umístění je určeno souřadným systémem, který byl definován jejich rodiči. Objekty line jsou vytvářeny funkcemi **plot**, **plot3**, **contour** a **contour3**.

**Objekty patch** jsou vyplněné mnohoúhelníky s hranami. Jsou dětmi objektu axes a jejich umístění je určeno souřadným systémem, který byl definován jejich rodiči. Tyto objekty mohou být vybarveny. Grafické objekty patch jsou generovány funkcemi **fill** a **fill3**.

**Objekty surface** jsou třírozměrnou reprezentací matice dat. Jsou složeny ze čtyřúhelníků, jejichž vrcholy jsou definovány prvky matice. Objekty surface mohou být kresleny jako plné nebo barevně interpolované, nebo pouze drátové (mesh). Jsou dětmi objektu axes a jejich poloha je určena souřadným systémem, který byl definován jejich rodiči. Objekty surface jsou generovány funkcemi **pcolor** a **surf** a skupinou funkcí **mesh**.

**Objekty image** jsou výsledkem transformace prvků matice na indexy aktuální mapy barev. Objekty image mívají zpravidla své vlastní mapy barev, palety barev, které definují barvy použité v konkrétním objektu image. Objekty image jsou zpravidla dvourozměrné, a proto nemohou být vidět z jiného úhlu, než je implicitní 2-D pohled. Jsou dětmi objektu axes a jejich poloha je určena souřadným systémem, který byl definován jejich rodiči. Tyto objekty jsou generovány funkcí **image** (Heringová, 1995).

**Objekty text** jsou znakové řetězce. Jsou dětmi objektu axes a jejich poloha je určena souřadným systémem, který byl definován jejich rodiči. Objekty text jsou generovány funkcemi **text**, **gtext**, **title**, **xlabel**, **ylabel** a **zlabel**.

**Objekty uicontrol** jsou uživatelem řízená rozhraní, která vykonávají příslušné funkce, jsou-li k tomu uživatelem vyzvány. Jsou dětmi objektů figure, a jsou proto nezávislé na objektu axes.

**Objekty uimenu** jsou uživatelská menu, která nám umožňují vytvořit si v objektu figure vlastní menu. Jsou dětmi objektu figure, a tedy jsou nezávislé na objektech axes (Heringová, 1995).

### 1.3.4 Identifikátory objektů

Každý samostatný grafický objekt má při jeho vytvoření přiřazen svůj vlastní identifikátor, tzv. handle. Identifikátor objektu `root` je vždy nulový. U objektu `figure` je to celé kladné číslo, které je implicitně zobrazeno v názvu grafického okna. Identifikátory ostatních objektů jsou reálná čísla, která obsahují informace používané prostředím MATLAB (Heringová, 1995).

Nutnou podmínkou při odkazování na tyto identifikátory je zachování jejich úplné přesnosti. Při práci s identifikátory se doporučuje je ukládat do proměnné a s ní poté pracovat, než je přepisovat ručně.

K jednoduchému přístupu k identifikátorům objektů využíváme v prostředí MATLAB následující funkce:

**gcf** – vrátí identifikátor aktuálního objektu `figure`,

**gca** – vrátí identifikátor aktuálního objektu `axes`.

Funkce **gcf** a **gca** můžeme používat jako vstupní argumenty pro jiné funkce, které požadují identifikátor objektů `figure` nebo `axes`.

Libovolný objekt lze zrušit funkcí **delete** s použitím identifikátoru tohoto objektu jako argumentu. Například můžeme vymazat aktuální osy, a tím i všechny jejich děti, příkazem:

```
delete (gca).
```

Všechny funkce prostředí MATLAB, které vytvářejí objekty, vrací jejich identifikátory (nebo vektor identifikátorů). A to jak funkce vyšší úrovně (např. **surf** – generuje jak plochu, tak čáry), tak i funkce nižší úrovně, které generují pouze jeden objekt (např. **surface**) (Heringová, 1995).

### 1.3.5 Vlastnosti objektů

Všechny objekty mají vlastnosti, které rozhodují o tom, jak budou tyto objekty zobrazeny. Tyto vlastnosti obsahují jak obecné informace (typ objektu, jeho rodiče a děti, jestli je nebo není objekt viditelný), tak i informace jedinečné pro daný typ objektu (například rozsah osy x grafu objektu `axes`).

Každý vytvořený grafický objekt je inicializován množinou implicitních hodnot vlastností. Aktuální hodnoty všech vlastností objektů můžeme získat a většinu z nich specifikovat. Některé z vlastností měnit nelze, jsou určeny pouze pro čtení. Změna vlastností se aplikuje jednoznačně na konkrétní objekt, nastavením hodnot jednoho objektu, neovlivníme hodnoty u jiných objektů stejného typu (Heringová, 1995).

### 1.3.6 GUI systému MATLAB

Vzhledem k tomu, že složitost i objem dat stále roste s rostoucí složitostí datových zdrojů a algoritmů, je stále důležitější intuitivní reprezentace těchto dat a výsledků. Grafické znázornění je často nejúčinnější forma reprezentace získaných dat. Umožňuje nám například snadnou identifikaci důležitých oblastí, kde mohou data vykazovat kuriózní chování. Zde je zřejmá výhoda grafického zobrazení dat, kdy nejsme zahlceni velkým množstvím číselných hodnot (Marchand, 2003).

Pro tvorbu vlastního grafického uživatelského rozhraní slouží v systému MATLAB editor GUI. Ten umožňuje snadno navrhnout uživatelské prostředí, pomocí vkládání objektů na pracovní plochu editoru. GUI obvykle obsahuje objekty, pomocí kterých můžeme například zadávat vstupní data, měnit hodnoty proměnných veličin, nebo vyhodnocovat získaná data. Dále se v GUI běžně setkáme s ovládacími prvky, jako jsou například tlačítka.

Základní prostředek používaný pro grafické zobrazení dat je 2-D graf. K zobrazení dat ve tvaru 2-D grafů včetně popisů a komentářů je v prostředí MATLAB k dispozici řada funkcí (Heringová, 1995).

Následující seznam obsahuje přehled funkcí, které vytvářejí graf daných dat:

**plot** – vytváří graf užitím lineární stupnice pro obě osy,

**loglog** – vytváří graf užitím logaritmické stupnice pro obě osy,

**semilogx** – vytváří graf s logaritmickou stupnicí pro osu x a lineární stupnicí pro osu y,

**semilogy** – vytváří graf s logaritmickou stupnicí pro osu y a lineární stupnicí pro osu x.

Tyto funkce se liší pouze užitím jiné stupnice os. Všechny akceptují vstupní data ve tvaru vektorů nebo matic a automaticky provádějí transformaci os podle rozsahu hodnot vstupních dat (Heringová, 1995).

Do grafu lze přidat nadpis, popis os, text nebo zobrazit mřížku pomocí následujících funkcí:

**title** – přidá nadpis do grafu (doprostřed nad graf),

**xlabel** – přidá popis osy x (doprostřed nad osu),

**ylabel** – přidá popis osy y (doprostřed podél osy),

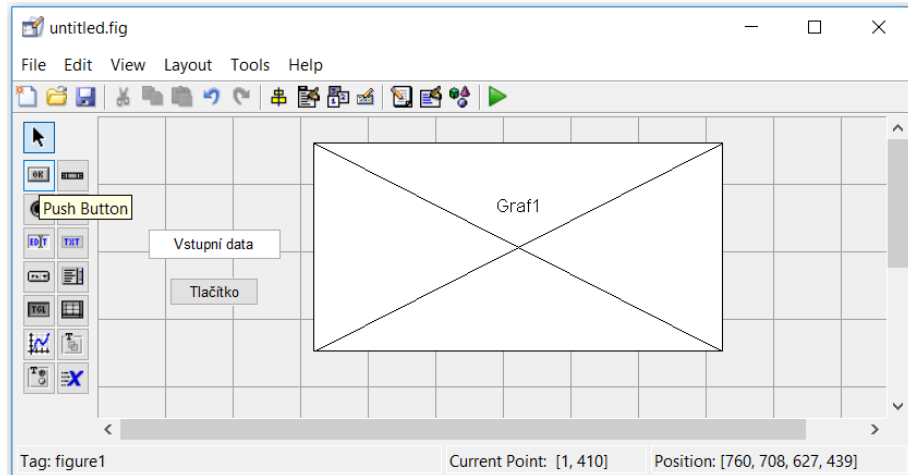
**text** – přidá textový řetězec na určenou pozici,

**gtext** – umístí text do grafu na místo vybrané myší,

**grid** – zobrazí mřížku.

Do GUI můžeme graf vložit pomocí tlačítka **Axes** a poté ho libovolně umístit. Graf je jeden z objektů, které můžeme do GUI vkládat.

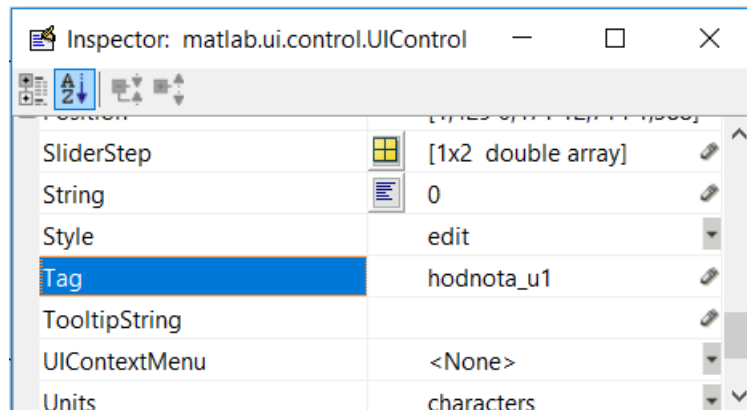
Objekty do GUI vkládáme vždy stejným způsobem. Vybereme požadovaný objekt



Obr. 1.7 – Vkládání objektů v editoru GUI

z nabídky v levé části editoru GUI a následně ho umístíme.

Kromě již zmíněného objektu **Axes**, můžeme zvolit **ActiveX Control**, **Button Group**, **Edit Text**, **Check Box**, **Listbox**, **Panel**, **Pop-up Menu**, **Push Button**, **Radio Button**, **Slider**, **Static Text**, **Table**, nebo **Toggle Button**.



Obr. 1.8 – Atributy objektu Edit Text

U každého ze zmíněných objektů lze nastavovat různé vlastnosti, z nichž nejdůležitější je atribut **Tag**. Ta definuje název objektu, pod kterým k němu budeme přistupovat.

Atributy objektů můžeme jednoduše nastavovat pomocí příkazu **set**. Jako příklad může být uvedena změna textu v objektu **Edit Text** (obr. 1.8). Na obrázku je vidět, že atribut „String“ je nastaven na hodnotu 0, tato hodnota je tedy vidět v textovém poli.

Pro změnu tohoto atributu „String“ použijeme příkaz:

```
set (hodnota_u1, 'String', 'text'),
```

kde hodnota\_u1 – název objektu,

String – atribut objektu, který chceme nastavit,

text – nový obsah textového pole.

Obdobně lze nastavovat i ostatní atributy objektu. Pokud bychom chtěli například vytvořit podmínku na základě stavu některého z atributů, můžeme zjistit jeho aktuální hodnotu pomocí příkazu **get**. Jako příklad můžeme brát v úvahu opět atribut „String“ objektu „hodnota\_u1“. Použitím příkazu:

```
x = get (hodnota_u1, 'String'),
```

zapišeme do proměnné „x“ aktuální textovou hodnotu zapsanou v objektu „hodnota\_u1“. V této bakalářské práci je objekt „hodnota\_u1“ využíván pro zadávání hodnoty napětí, avšak příkazem **get** získáme tuto hodnotu pouze v podobě textu, je tedy nutné ji převést na číselnou hodnotu, vhodnou pro další zpracování. Tento převod můžeme provést příkazem **str2double**.

Přidáním objektu do GUI vzniká tzv. Callback na daný objekt. Ten vytváří odkaz na objekt při vzniku události. Jako objekt si můžeme představit tlačítko (Push Button). Událost představuje například stisk tlačítka, nebo zápis hodnoty do textového pole. Callback odkazuje na funkci která se vykoná, jakmile událost nastane. Pro ukázkou použijeme opět objekt „hodnota\_u1“.

Funkce pak může vypadat následovně:

```
function hodnota_u1_Callback(hObject, eventdata, handles)
```

```
% hObject handle to hodnota_u1 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of hodnota_u1 as text
```

```
% str2double(get(hObject,'String')) returns contents of hodnota_u1 as a double
```

```
handles.u1 = str2double (get (handles.hodnota_u1, 'String'));
```

```
guidata (hObject, handles);
```

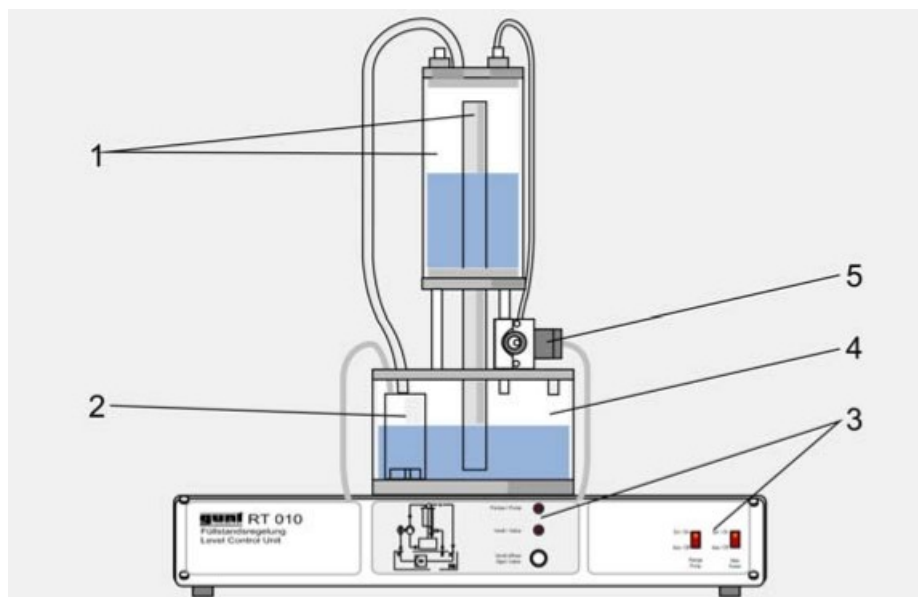
Znak „%“ definuje veškerý následující text na řádku jako komentář, ten tedy nemá žádný vliv na běh programu. Komentář uvedený výše je ukázkou automatického komentáře, který vzniká v prostředí MATLAB při vytvoření funkce (Callback) pro odkaz na objekt.

Na posledním řádku této funkce si můžeme všimnout příkazu **guidata**, s jehož pomocí se ukládají všechny proměnné do skupiny „handles“. Proměnné v této skupině jsou poté využívány jako globální proměnné v celém GUI. Název této skupiny byl zvolen záměrně, jelikož jako handle je označován identifikátor objektu. Stejně tak by mohl být zvolen jiný název, například „data“.

Program pro jednotlivé funkce si píšeme sami, v textové formě, podle jejich účelu. Při tvorbě programu se doporučuje využívat nápovědu systému MATLAB. Ta obsahuje přehledný souhrn všech použitelných příkazů, jejich parametrů a jednoduché praktické ukázky.

## 1.4 LABORATORNÍ STANICE GUNT RT 010

Laboratorní soustava představuje model provozní aparatury s možností měření a regulace výšky hladiny vody v provozní nádrži. Schéma technologického zařízení je vyobrazeno na přední straně zařízení s dalšími ovládacími prvky.



Obr. 1.9 – Stanice GUNT RT 010

Laboratorní soustava je propojena s počítačem pomocí rozhraní USB. Laboratorní modely obsahují potřebné senzory pro snímání měřené veličiny, akční členy a další ovládací prvky. Proces je řízen regulátorem, který je realizován pomocí softwaru počítače. Na přední straně zařízení jsou umístěny ovládací a signalizační prvky (3) (Kadlec, 2012).

Základ laboratorní stanice tvoří dvě nádrže z organického skla (obr. 1.9). Voda ze zásobní nádrže (4) o objemu 3,7 l je pomocí čerpadla (2) přiváděna do provozní nádrže s přepadem (1) s objemem  $V = 1,2$  l, která představuje regulovanou soustavu. Čerpadlo dokáže

vytvořit maximální průtok  $Q = 8 \text{ l/min}$ . Regulovanou veličinou je výška hladiny v provozní nádrži. Poloha hladiny je měřena snímačem hydrostatického tlaku s tenzometrickým čidlem, který poskytuje elektrický signál pro porovnávací člen regulátoru. Pracovní rozsah snímače tlaku je  $0 \div 3 \text{ kPa}$ . Pro vizuální kontrolu je na provozní nádrži k dispozici milimetrové měřítko.

V dolní zásobní nádrži je umístěno ponorné čerpadlo, které čerpe vodu do horní provozní nádrže. Čerpadlo plní funkci akčního členu v regulačním obvodu. Akční veličina se mění se změnou otáček motoru čerpadla. Voda z provozní nádrže odtéká zpět do zásobní nádrže přes elektricky ovládaný proporcionální ventil (5), nebo přes přepad. Změna v otevření proporcionálního ventilu představuje poruchovou veličinu regulačního obvodu (Kadlec, 2012).

## 1.5 MĚŘICÍ KARTA LABJACK U12



Obr. 1.10 – USB měřicí karta LabJack U12

Měřicí karta LabJack U12 je kompaktní systém osazený dvěma konektory – USB typu B pro komunikaci s nadřizovaným PC a 25východovým konektorem pro externí připojení expanderu, nebo přímé aplikace. Systém je napájen z PC z USB portu. Nepotřebuje tedy připojení vnějšího zdroje, naopak může napájet další periferie. Proudové omezení je podle charakteru PC až do  $I = 500 \text{ mA}$  (Závidčák, 2005).

Tab. 1.1 – Zapojení vývodů svorkovnic

Č.	NÁZEV	POPIS	Č.	NÁZEV	POPIS
1	AI0	Analogový vstup 0	16	IO0	Vstup/Výstup 0
2	AI1	Analogový vstup 1	17	IO1	Vstup/Výstup 1
3	GND	Zem	18	GND	Zem
4	AI2	Analogový vstup 2	16	IO2	Vstup/Výstup 2
5	AI3	Analogový vstup 3	20	IO3	Vstup/Výstup 3
6	GND	Zem	21	GND	Zem
7	AI4	Analogový vstup 4	22	AO0	Analogový výstup 0
8	AI5	Analogový vstup 5	23	AO1	Analogový výstup 1
9	GND	Zem	24	GND	Zem
10	AI6	Analogový vstup 6	25	CNT	Čítač
11	AI7	Analogový vstup 7	26	GND	Zem
12	GND	Zem	27	GND	Zem
13	+5 V	Napájecí napětí	28	+5 V	Napájecí napětí
14	+5 V	Napájecí napětí	29	+5 V	Napájecí napětí
15	CAL	Kalibrační napětí 2,5 V	30	STB	Generátor pulzů

Tab. 1.2 – Zapojení vývodů konektoru

Č.	NÁZEV	POPIS	Č.	NÁZEV	POPIS
1	D0	Vstup/Výstup 0	14	GND	Zem
2	D1	Vstup/Výstup 1	15	GND	Zem
3	D2	Vstup/Výstup 2	16	GND	Zem
4	D3	Vstup/Výstup 3	17	GND	Zem
5	D4	Vstup/Výstup 4	18	D8	Vstup/Výstup 8
6	D5	Vstup/Výstup 5	19	D9	Vstup/Výstup 9
7	D6	Vstup/Výstup 6	20	D10	Vstup/Výstup 10
8	D7	Vstup/Výstup 7	21	D11	Vstup/Výstup 11
9	NC	Nezapojen	22	D12	Vstup/Výstup 12
10	+5 V	Napájecí napětí	23	D13	Vstup/Výstup 13
11	+5 V	Napájecí napětí	24	D14	Vstup/Výstup 14
12	+5 V	Napájecí napětí	25	D15	Vstup/Výstup 15
13	+5 V	Napájecí napětí			

### 1.5.1 Analogové vstupy

Měřicí karta LabJack U12 má 8 analogových vstupů s rozlišením 12 bitů. Jednotlivé vstupy jsou použitelné jako standardní vstup (měří napětí vstupu proti zemi), diferenční (měří napětí mezi dvěma vstupy), nebo jejich kombinace.

Tab. 1.3 – Napěťový rozsah analogových vstupů

<b><math>G = 1</math></b>	$\pm 20 \text{ V}$	<b><math>G = 8</math></b>	$\pm 2,5 \text{ V}$
<b><math>G = 2</math></b>	$\pm 10 \text{ V}$	<b><math>G = 10</math></b>	$\pm 2 \text{ V}$
<b><math>G = 4</math></b>	$\pm 5 \text{ V}$	<b><math>G = 16</math></b>	$\pm 1,25 \text{ V}$
<b><math>G = 5</math></b>	$\pm 4 \text{ V}$	<b><math>G = 20</math></b>	$\pm 1 \text{ V}$

Pro standardní vstupy platí základní rozsah  $\pm 10 \text{ V}$ . Volbou faktoru  $G$  nastavujeme úroveň rozsahu diferenčních vstupů. Maximální přípustná napěťová úroveň je  $\pm 40 \text{ V}$ .

### 1.5.2 Analogové výstupy

LabJack U12 má dva analogové výstupy. Oba jsou nastavitelné v rozmezí od  $0 \text{ V}$  do úrovně napájecího napětí ( $5 \text{ V}$ ) v 10bitovém rozlišení (Závidčák, 2005).

Tab. 1.4 – Závislost výstupní úrovně na odebíraném proudu

<b><math>I = 0 \text{ mA}</math></b>	$V_S$
<b><math>I = 1 \text{ mA}</math></b>	$0,99 \cdot V_S$
<b><math>I = 5 \text{ mA}</math></b>	$0,96 \cdot V_S$

Uvedené hodnoty odpovídají při definovaném vnitřním odporu výstupu  $R = 20 \Omega$ . Maximální přípustný proud pro každý výstup je  $30 \text{ mA}$ .

## 2 IMPLEMENTAČNÍ ČÁST

### 2.1 KOMUNIKACE SOUSTAVY GUNT SE SYSTÉMEM MATLAB

Soustava GUNT RT 010 je propojena s počítačem pomocí rozhraní USB a měřicí karty LabJack U12. Měřicí karta je dále připojena k sensorům a akčním členům soustavy GUNT.

#### 2.1.1 Propojení soustavy GUNT s LabJack U12

##### **Zapojení LabJack U12:**

AI0 – výstup snímače tlaku ( $0,5 \div 2,7$  V),

AO0 – vstupní napětí čerpadla ( $0 \div 5$  V),

AO1 – vstupní napětí ventilu ( $0 \div 5$  V),

IO2 – vypínač čerpadla (on/off),

IO3 – tlačítko otevření ventilu (on/off).

Zde je třeba si uvědomit obrácenou logiku vstup/výstup. Tím je myšleno, že vstup do měřicí karty LabJack U12 je výstupem ze soustavy GUNT a naopak. To může být při následné tvorbě programu matoucí. Pokud například chceme zjistit stav na výstupu tlakového čidla, musíme přistupovat ke vstupu AI0 měřicí karty LabJack U12.

Rozsahy možných hodnot napětí pro snímače a akční členy jsou uvedeny výše v popisu jejich zapojení. Pracovní rozsah čerpadla se liší, jelikož k roztočení motoru čerpadla potřebujeme napětí vyšší než 1 V.

Vypínač čerpadla a tlačítko pro ruční otevření ventilu jsou umístěny na ovládacím panelu, na přední straně skřínky soustavy GUNT RT 010. Zapnutí čerpadla pomocí vypínače na ovládacím panelu je nutnou podmínkou k činnosti čerpadla řízeného pomocí navrženého GUI. Tlačítko otevření vypouštěcího ventilu funguje nezávisle na činnosti soustavy, stiskem tlačítka je tedy ventil maximálně otevřen.

## 2.1.2 Využití knihoven při komunikaci s LabJack U12

Při komunikaci měřicí karty LabJack U12 se systémem MATLAB se využívá dvou knihoven. Jedná se o knihovny ljackuw.dll a LJDotNet.dll. Tyto knihovny musí být přítomny ve stejném adresáři jako M-soubor a figure, aby byla zajištěna správná funkce GUI.

Knihovny jsou využity například při nastavování hodnot napětí na výstupech měřicí karty LabJack U12. K tomu slouží funkce EAnalogOut, která nastaví napětí analogových výstupů. Zde slouží k nastavení vstupních napětí pro čerpadlo a ventil v rozsahu  $0 \div 5$  V (AO0 – čerpadlo, AO1 – vypouštěcí ventil).

### Syntaxe funkce EAnalogOut:

[errorcode idnum] = EAnalogOut (idnum, demo, analogOut0, analogOut1).

### Vstupní parametry:

idnum – lokální ID, sériové číslo, nebo -1 pro první nalezený LabJack,

demo – 0 demo vypnuto, >0 demo zapnuto,

analogOut0 – napětí pro AO0 ( $0 \div 5$  V),

analogOut1 – napětí pro AO1 ( $0 \div 5$  V).

### Výstupní parametry:

idnum – lokální ID LabJack, nebo -1 když LabJack není nalezen,

errorcode – číslo poruchy, nebo 0 když žádná porucha nenastala.

Obdobně je využita funkce EAnalogIn. Pomocí této funkce je možné získat aktuální hodnotu napětí na analogovém vstupu (AI0) měřicí karty LabJack U12, tedy hodnotu napětí na výstupu snímače tlaku ( $0,5 \div 2,7$  V).

### Syntaxe funkce EAnalogIn:

[voltage overvoltage errorcode idnum] = EAnalogIn (idnum, demo, channel, gain).

### Vstupní parametry:

idnum – lokální ID, sériové číslo, nebo -1 pro první nalezený LabJack,

demo – 0 demo vypnuto, >0 demo zapnuto,

channel –  $0 \div 7$  pro jednoduché, nebo  $8 \div 11$  pro rozdílové vstupy,

gain – zesílení 1, 2, 4, 5, 8, 10, 16, 20 V/V.

**Výstupní parametry:**

voltage – hodnota napětí na vstupu,

overVoltage – hodnota  $> 0$ , indikuje přepětí na jednom z analogových vstupů,

errorcode – číslo poruchy, nebo 0 když žádná porucha nenastala,

idnum – lokální ID LabJack, nebo -1 když LabJack není nalezen.

Doba vykonání funkce EAnalogOut, nebo EAnalogIn je maximálně 50 ms. Uvedené knihovny pro měřicí kartu LabJack U12 umožňují použití celé řady dalších funkcí. Ty ale při tvorbě GUI nejsou využity.

## **2.2 NÁVRH SOFTWARE**

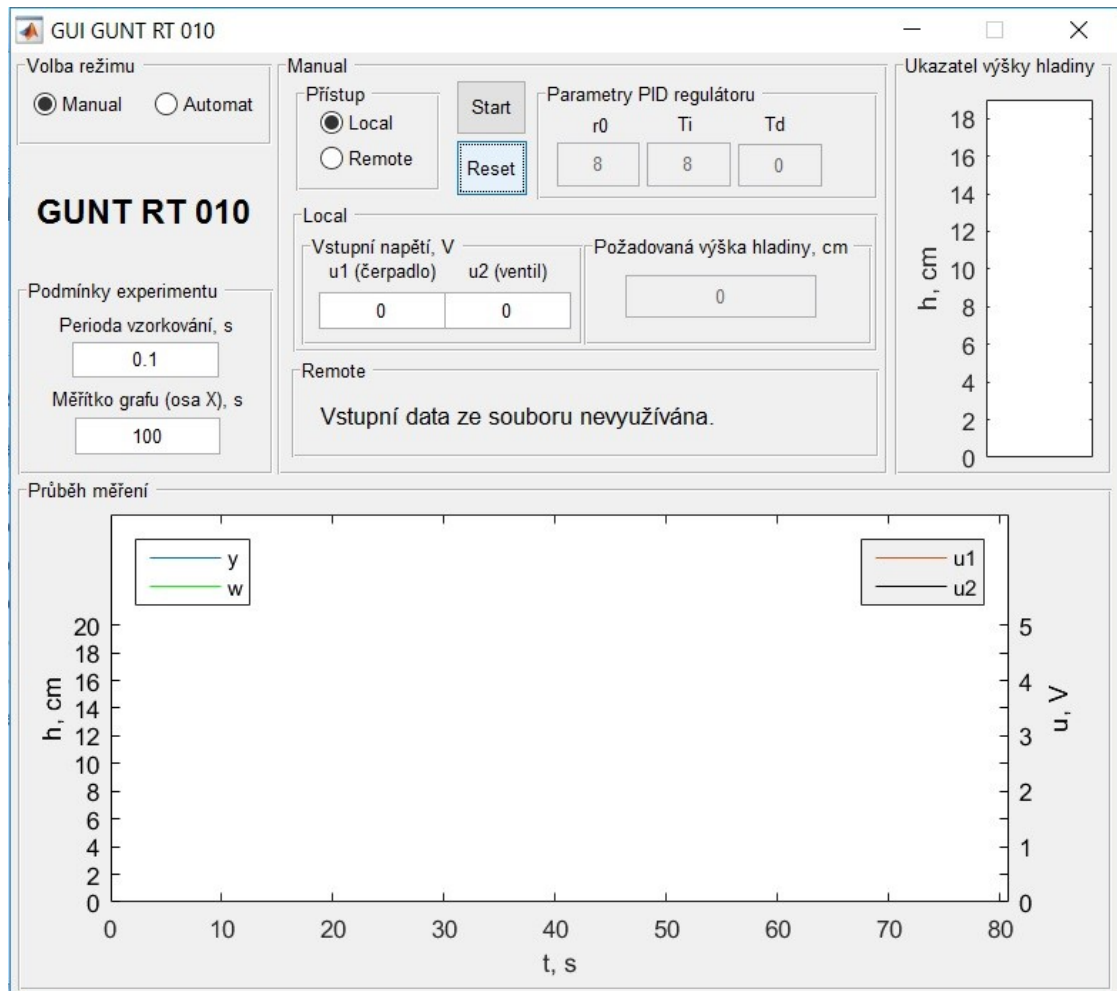
### **2.2.1 Koncept**

Základní myšlenka při tvorbě GUI, byla vytvořit, co možná nejjednodušší grafické rozhraní s přehledným a jednoduchým ovládáním. Takové, aby s ním uživatel mohl snadno pracovat, bez nutnosti dřívějšího seznámení s GUI.

Grafické uživatelské rozhraní pro soustavu GUNT RT 010 jsem vytvořil v systému MATLAB. GUI jako funkční celek je složeno z několika částí. První a nejobsáhlejší částí je M-soubor. Ten obsahuje veškerý program zajišťující činnost grafického uživatelského rozhraní a jeho základní nastavení. Program je zde uveden v textové formě. Druhou částí je figure, čili návrh grafické podoby GUI. Ten slouží především k rozmístění jednotlivých objektů, určení jejich tvaru a pojmenování. Třetí částí jsou knihovny nutné pro komunikaci s měřicí kartou LabJack U12. Poslední částí navrženého GUI jsou textové soubory určené pro vstupní a výstupní data. Všechny výše jmenované části musí být pro správnou funkci GUI umístěny ve stejném adresáři.

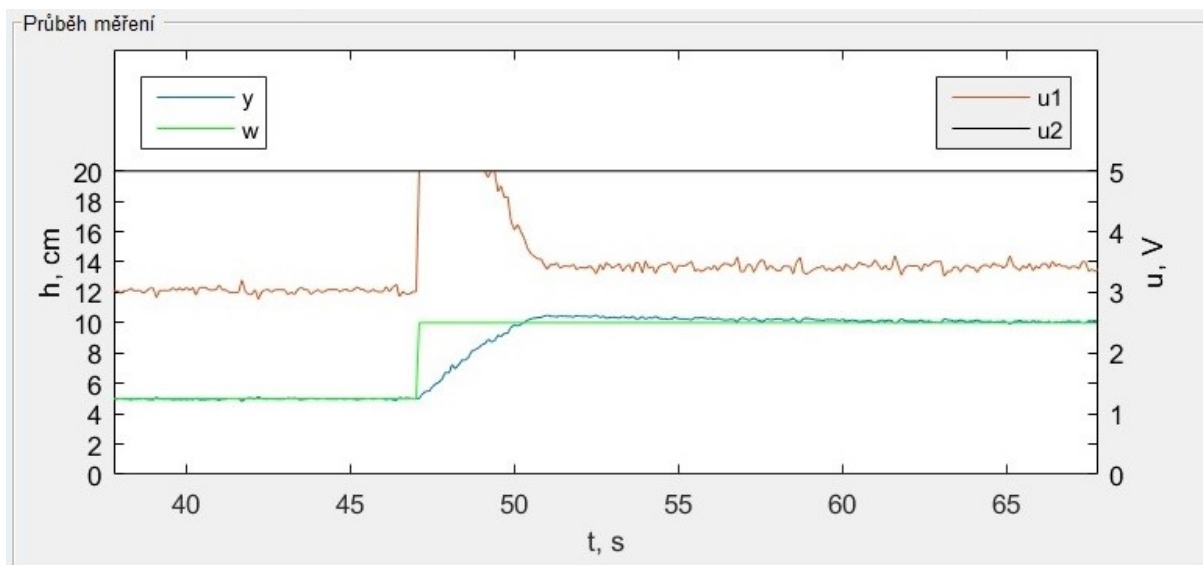
## 2.2.2 Vizuální návrh

Na obrázku (obr. 2.1), vidíme základní podobu GUI. Pro jeho maximální přehlednost byly použity při tvorbě objekty „Panel“. Můžeme je vidět jako ohraničené obdélníkové oblasti s vlastním názvem. Tento název stručně definuje význam určité oblasti, a tedy i objektů v ní umístěných.



Obr. 2.1 – Základní podoba GUI

Výjimkou jsou oblasti sloužící k volbě pracovního režimu a přístupu k vstupním datům. Jedná se o oblasti „Volba režimu“ (obr. 2.5) a „Přístup“ (obr. 2.4). V těchto částech GUI je použit objekt „Button Group“. Jeho význam je zde zcela zásadní, ačkoli se na pohled od objektu „Panel“ ničím neliší. Tento objekt je využit z důvodu umístění zaškrtačacích políček „Radio Button“, které slouží k volbě jedné z nabízených možností. Použitím objektu pro skupinu tlačítek vymežíme oblast, ve které smí být v jeden moment zaškrtnuto pouze jedno z tlačítek typu „Radio Button“. Tím se vytvoří prostor, v němž může uživatel zvolit vždy jen jednu z nabízených možností.



Obr. 2.2 – Panel: Průběh měření

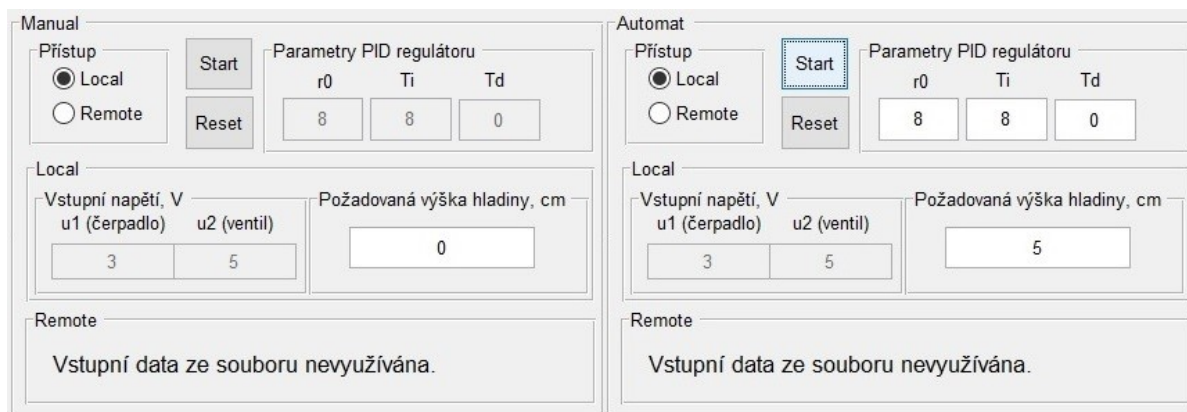
Největší oblast GUI je vymezena pro graf, který zobrazuje průběh měření (obr. 2.3). A to časový průběh hodnot na výstupu soustavy, žádané hodnoty a velikosti akčního zásahu. V závislosti na volbě režimu se udává pouze průběh používaných veličin. To si lze představit při práci v ručním režimu, kde nemůžeme ovlivňovat hodnotu žádané veličiny. Z toho důvodu je vyloučena, respektive nastavena na hodnotu 0. Oblast zobrazovaných dat a periodu vzorků lze kdykoli měnit v panelu „Podmínky experimentu“ (obr. 2.2), který je umístěn uprostřed levé části GUI.

The panel contains two input fields. The first is labeled 'Perioda vzorkování, s' and has the value '0.1'. The second is labeled 'Měřítko grafu (osa X), s' and has the value '200'.

Obr. 2.3 – Panel: Podmínky experimentu

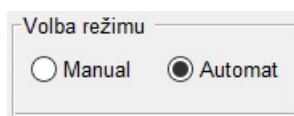
Druhá největší oblast GUI patří panelu „Manual/Automat“. Ta je dále rozdělena na několik menších částí, jejichž dostupnost závisí na zvoleném pracovním režimu. Obsahuje volbu přístupu k vstupním datům (Local, nebo Remote). V režimu Local se vstupní data získávají přímo z GUI, režim Remote umožňuje načítat vstupní data z textového souboru. Hodnoty v lokálním režimu zadáváme z klávesnice, přepsáním aktuálních hodnot. Volba hodnot vstupních napětí pro čerpadlo a ventil je možná pouze v režimu Manual. Při práci v tomto režimu je přístupná pouze volba těchto dvou hodnot napětí, popřípadě změna zdroje vstupních dat. Pro režim Automat se hodnota vstupního napětí čerpadla získává pomocí

výpočtu. Napětí určené k ovládní ventilu je v tomto režimu konstantní (5 V). Obě hodnoty napětí nelze při práci v tomto režimu ručně měnit. Uživatel zvolí požadovanou výšku hladiny a může dle uvážení měnit parametry používaného PID regulátoru. Samozřejmě i v tomto režimu lze kdykoli změnit zdroj vstupních dat. V této části GUI se dále nachází tlačítka Start/Stop a Reset, která slouží k zahájení/zastavení měření a nastavení průběhu měření do výchozích hodnot. Pokud je měření zastaveno, veškeré změny provedené v GUI se vykonají až ve chvíli, kdy uživatel spustí měření.



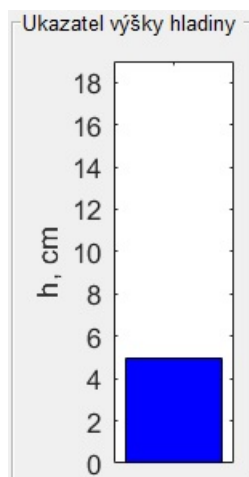
Obr. 2.4 – Panel: Manual/Automat

Další část GUI je panel „Volba režimu“ a obsahuje základní volbu mezi režimy Manual a Automat. Ta určuje, zda bude uživatel ovládat soustavu GUNT pomocí vstupních napětí čerpadla a ventilu, nebo využije PID regulátor a zadá pouze požadovanou hodnotu výšky hladiny v nádrži. Tato volba určuje dostupnost jednotlivých objektů v panelu „Manual/Automat“, jak již bylo popsáno výše.



Obr. 2.5 – Panel: Volba režimu

V poslední oblasti GUI se nachází ukazatel výšky hladiny (obr. 2.6). Jedná se o informativní prvek a slouží jako obdoba milimetrového měřítka na provozní nádrži.



Obr. 2.6 – Panel: Ukazatel výšky hladiny

### 2.2.3 Práce s GUI

Jak již bylo řečeno, GUI bylo navrženo tak, aby s ním mohl uživatel snadno pracovat. Po spuštění je GUI nastaveno do výchozích hodnot.

#### Výchozí hodnoty v GUI:

- vstupní napětí čerpadla – 0 V,
- vstupní napětí ventilu – 0 V,
- požadovaná hodnoty hladiny – 0 cm,
- zesílení regulátoru – 8,
- integrační časová konstanta regulátoru – 8 s,
- derivační časová konstanta regulátoru – 0 s,
- perioda vzorkování – 0,1 s,
- měřítka osy x grafu (počet zobrazených hodnot) – 10,
- režim – manual (ruční režim),
- přístup – local (vstupní data zadávána z GUI).

Použitím tlačítka „Reset“ můžeme při práci v GUI přejít do výchozích hodnot. Tato volba se ale týká pouze hodnot v panelu Manual/Automat. Nastavení pracovního režimu, periody vzorkování a měřítka grafu zůstává nezměněno. Důvodem je rychlejší práce v GUI. Nepředpokládám, že by uživatel měl potřebu tyto hodnoty často měnit a takhle je nemusí po použití resetu opakovaně zadávat, což šetří čas.

Navržené GUI obsahuje jednu volbu parametru, která je dostupná pouze při úpravě M-souboru a v GUI k ní není přístup. Do GUI jsem tuto volbu nezahrnul, protože běžný uživatel

ji zřejmě nevyužije. Jedná se o volbu parametru demo. Její užitečnost je v možnosti simulovaného připojení měřicí karty LabJack U12. Lze tedy pracovat s GUI i přesto, že nejsme reálně připojeni k měřicí kartě. Tedy při volání funkcí EAnalogIn a EAnalogOut nedochází k chybám. Tyto funkce ovšem při použití v demo módu vrací pouze konstantní hodnoty, proto demo mód nemůže nikdy nahradit skutečné připojení soustavy kartou LabJack U12. Tuto možnost lze ale efektivně využívat při tvorbě GUI. Z toho důvodu je vhodné ji zmínit.

Parametr demo lze nastavit v tzv. Opening funkci, která se vykoná při spuštění GUI. Při nastavení demo = 0 je demo mód vypnut a pro funkci GUI je nutné propojení s měřicí kartou LabJack U12. Nastavením parametru demo na hodnotu větší než 0, je demo mód aktivován.

Komunikaci GUI se soustavou GUNT RT 010 zahájíme stiskem tlačítka „Start“, tím se aktivují veškeré změny hodnot v GUI, provedené před zahájením měření. Poté se vlastnost string u tlačítka mění na hodnotu „Stop“. Toto tlačítko plní funkci zahájení i zastavení měření. Pro zahájení nového měření můžeme využít tlačítko „Reset“, popřípadě restartovat GUI. Po stisku tlačítka „Stop“ se tedy pozastaví měření a na vstupech soustavy GUNT RT 010 zůstávají trvale nastaveny hodnoty, které byly aktuální při zastavení měření. V tuto chvíli neprobíhá záznam průběhu měření do grafu, ani indikace výšky hladiny. Soustava není žádným způsobem regulována a časem by se dostala do jednoho z hraničních stavů. Pracovní nádrž by byla buď vypuštěna, nebo zcela napuštěna (voda by odtékala přes přepad do zásobní nádrže), případně by se výška hladiny ustálila na hodnotě definované velikostí odtoku a přítoku vody. Zmíněné stavy soustavy jsou pro účely měření nežádoucí, a proto nedoporučuji pozastavovat měření v průběhu experimentu, ale až po jeho ukončení.

V průběhu měření samozřejmě může uživatel libovolně nastavovat všechny dostupné parametry. Změna režimu a přístupu se provede ihned po jejich přepnutí. Změny vstupních hodnot, parametrů regulátoru, nebo podmínek experimentu se provedou po jejich přepsání a následném stisku klávesy „Enter“, nebo po kliknutí mimo zadávací pole (objekt Edit Text). Dostupnost jednotlivých parametrů je závislá na zvoleném režimu a přístupu k vstupním datům.

Průběh měření může uživatel pozorovat v grafu, který je součástí navrženého GUI. Graf obsahuje časovou osu x a dvě osy y. Osa y umístěná vlevo vyjadřuje hodnotu hladiny provozní nádrže a požadovanou hodnotu výšky hladiny. Osa y vpravo vyjadřuje hodnotu vstupních napětí pro čerpadlo a ventil. Tyto veličiny jsou od sebe navzájem barevně odlišeny následujícím způsobem:

- aktuální výška hladiny (modrá),
- požadovaná hodnota výšky hladiny (zelená),
- vstupní napětí čerpadla (červená),

– vstupní napětí vypouštěcího ventilu (černá).

Jednotlivé barvy veličin můžeme vidět i na obr. 2.1 v legendě grafu.

V ustáleném stavu by měla být hodnota aktuální výšky hladiny rovna žádané hodnotě výšky hladiny, tento stav však nemusí vždy nastat, závisí na nastavení použitého PID regulátoru.

Osa y grafu průběhu měření má nastaven horní limit na 20 cm. Této hodnoty měřená hladina provozní nádrže nikdy nedosáhne. Voda začne odtékat přes přepad při výšce hladiny přibližně 18,1 cm. Velikost maximálního vstupního napětí čerpadla, nebo ventilu (5 V) odpovídá v grafu hodnotě 20 cm výšky hladiny. Měřítko os y grafu průběhu měření je tak nastaveno na hodnotu 1:4 (levá osa y: pravá osa y).

Při měření se jeho průběh zaznamenává nejen do grafu, ale hodnoty se zapisují i do textového souboru. To je výhodné zejména pokud uživatel potřebuje znát přesné hodnoty, které by se mu z grafu nepodařily určit. Tyto data jsou uložena a seříděna v textovém souboru data.txt. Uspořádání dat v souboru můžeme vidět na obr. 2.7,

kde  $t$  – čas měření, s,

$u1$  – vstupní napětí čerpadla, V,

$u2$  – vstupní napětí ventilu, V,

$u_{výst}$  – výstupní napětí snímače tlaku, V,

$h$  – výška hladiny provozní nádrže, cm,

$hp$  – požadovaná výška hladiny provozní nádrže, cm.



t, s	u1, V	u2, V	u_výst, V	h, cm	hp, cm
0.085	0.000	0.000	1.455	7.854	9.000
0.150	1.369	5.000	1.416	7.533	9.000
0.217	1.463	5.000	1.406	7.453	9.000
0.315	1.761	5.000	1.372	7.172	9.000
0.410	2.100	5.000	1.333	6.851	9.000
0.517	2.483	5.000	1.289	6.490	9.000

Obr. 2.7 – Soubor data.txt

Naměřená data se do souboru ukládají postupně, po celou dobu měření, v závislosti na periodě vzorkování. Soubor data.txt vždy uchovává pouze poslední měřená data. Tyto hodnoty jsou v souboru uloženy pouze do doby opětovného spuštění GUI, nebo mohou být vymazány použitím tlačítka „Reset“.

Podobným způsobem jsou řešeny soubory pro vstupní data, využívané v režimu přístupu Remote. Pracovní režim Manual a Automat mají každý svůj vlastní textový soubor pro vstupní data, jelikož se jejich vstupní data navzájem liší. A to nejen jejich hodnoty, ale i počet vstupních veličin. Režimu Manual využívá pro vstupní data textový soubor manual.txt a režim Automat soubor automat.txt. Strukturu souboru manual.txt lze vidět na obr. 2.8,

- kde
1. sloupec –  $t$ , s,
  2. sloupec –  $u_1$ , V (vstupní napětí čerpadla),
  3. sloupec –  $u_2$ , V (vstupní napětí ventilu).

Soubor	Úpravy	Formát	Zobrazení	Nápověda
0	0	0		
5	0	5		
10	3	2		
15	5	5		
20	2	5		

Obr. 2.8 – Soubor manual.txt

Obdobně vypadá i struktura souboru automat.txt na obr. 2.9,

- kde
1. sloupec –  $t$ , s,
  2. sloupec – požadovaná výška hladiny provozní nádrže, cm.

Soubor	Úpravy	Formát	Zobrazení	Nápověda
0	7			
10	6			
20	3			
30	5			

Obr. 2.9 – Soubor automat.txt

Pro úpravu vstupních dat v souborech manual.txt a automat.txt není v GUI zavedena žádná funkce. Uživatel je tedy nucen vstupní data zapsat ručně v textovém editoru, zde je pro ukázkou použit Poznámkový blok. V GUI v panelu Remote je přehledně uvedeno, jestli jsou vstupní data z textového souboru využívána. Pokud ano, je uveden i název právě používaného souboru. Při vyčerpání všech vstupních dat v textovém souboru se GUI automaticky přepne do režimu přístupu Local a jako vstupní data jsou použity hodnoty z posledního řádku textového

souboru. GUI zůstává v tomto stavu až do dalšího zásahu uživatele, přičemž měření stále probíhá.

## 2.2.4 Využití PID regulátoru

Jak již bylo uvedeno, GUI ke své činnosti v režimu Automat využívá PID regulátor. S jeho pomocí se v automatickém režimu počítá vhodná hodnota akční veličiny (vstupní napětí čerpadla) tak, abychom dosáhli požadované hodnoty výstupní veličiny (výška hladiny provozní nádrže). Výpočet probíhá na základě následujících rovnic pro PID regulátor v přírůstkovém tvaru:

$$u(k-1) = r_0 \left( e(k-1) + \frac{T}{T_i} \sum_{i=1}^{k-1} e(i) + T_d \frac{e(k-1) - e(k-2)}{T} \right), \quad (2.1)$$

$$u(k) - u(k-1) = q_0 \cdot e(k) - q_1 \cdot e(k-1) + q_2 \cdot e(k-2), \quad (2.2)$$

$$q_0 = r_0 \left( 1 + \frac{T}{T_i} + \frac{T_d}{T} \right), \quad (2.3)$$

$$q_1 = r_0 \left( 1 + \frac{2 \cdot T_d}{T} \right), \quad (2.4)$$

$$q_2 = r_0 \cdot \frac{T_d}{T}, \quad (2.5)$$

kde  $u$  – akční veličina,

$e$  – regulační odchylka,

$r_0$  – proporcionální konstanta regulátoru,

$T_i$  – integrační časová konstanta regulátoru,

$T_d$  – derivační časová konstanta regulátoru,

$T$  – perioda vzorkování,

$q_0$  – parametr,

$q_1$  – parametr,

$q_2$  – parametr.

Z těchto rovnic můžeme vidět, že k výpočtu hodnoty akční veličiny potřebujeme znát dva předchozí stavy soustavy. Tato skutečnost je v programu GUI vyřešena podmínkou, a to

tím způsobem, že samotný výpočet nastane až ve chvíli, kdy jsou známé tři naměřené hodnoty. Tedy aktuální stav a dva předchozí. Nevýhodou tohoto řešení je jisté zpoždění, které závisí na zvolené periodě vzorkování. Vypočítaná hodnota akčního zásahu se může vymykat použitelnému rozsahu vstupního napětí čerpadla. Proto jsou zde zavedeny omezení těchto hodnot, které se tak mohou pohybovat pouze v mezích rozsahu napětí čerpadla ( $0 \div 5$  V).

## 2.3 PŘÍKLADY POUŽITÍ GUI

Jako příklad použití GUI jsou uvedeny ukázky práce v ručním režimu a měření odezvy uzavřeného regulačního obvodu na skokové změny žádané hodnoty výšky hladiny. U měření odezvy regulačního obvodu na skokové změny (obr. 2.13) je použito zadávání vstupních hodnot z textového souboru.

U všech uvedených měření byly použity následující parametry:

$r_0 = 8$  (proporcionální konstanta regulátoru),

$T_i = 8$  s (integrační časová konstanta regulátoru),

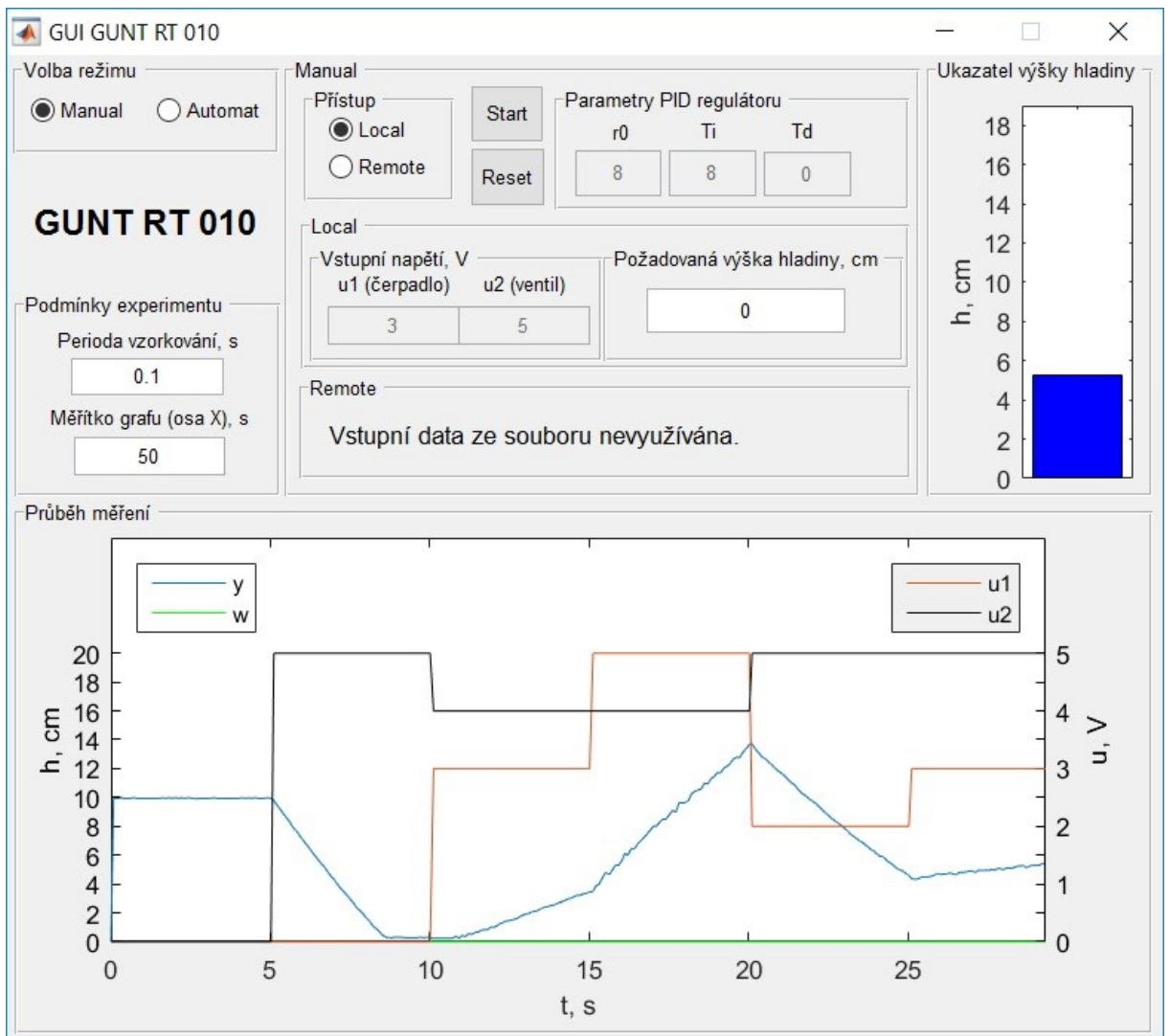
$T_d = 0$  s (derivační časová konstanta regulátoru),

$T = 0,1$  s (perioda vzorkování).

Měřítka osy x grafu průběhu měření byly vždy zvoleny tak, aby graf vždy obsáhl všechny důležité hodnoty.

U první ukázky (obr. 2.10) byl využit ruční režim GUI, u ostatních příkladů režim Automat. Režim Automat je pro ukázku vhodnější, jelikož oproti manuálnímu režimu využívá ke své činnosti navíc i PID regulátor. Díky PID regulátoru můžeme přesně získat požadovanou hodnotu výšky hladiny. V režimu Manual pouze zadáváme hodnoty vstupních napětí a sledujeme chování soustavy. Dosažení požadované výšky hladiny provozní nádrže, nebo ustáleného stavu je za těchto podmínek mnohem náročnější a velmi nepřesné.

### 2.3.1 Experiment v ručním režimu



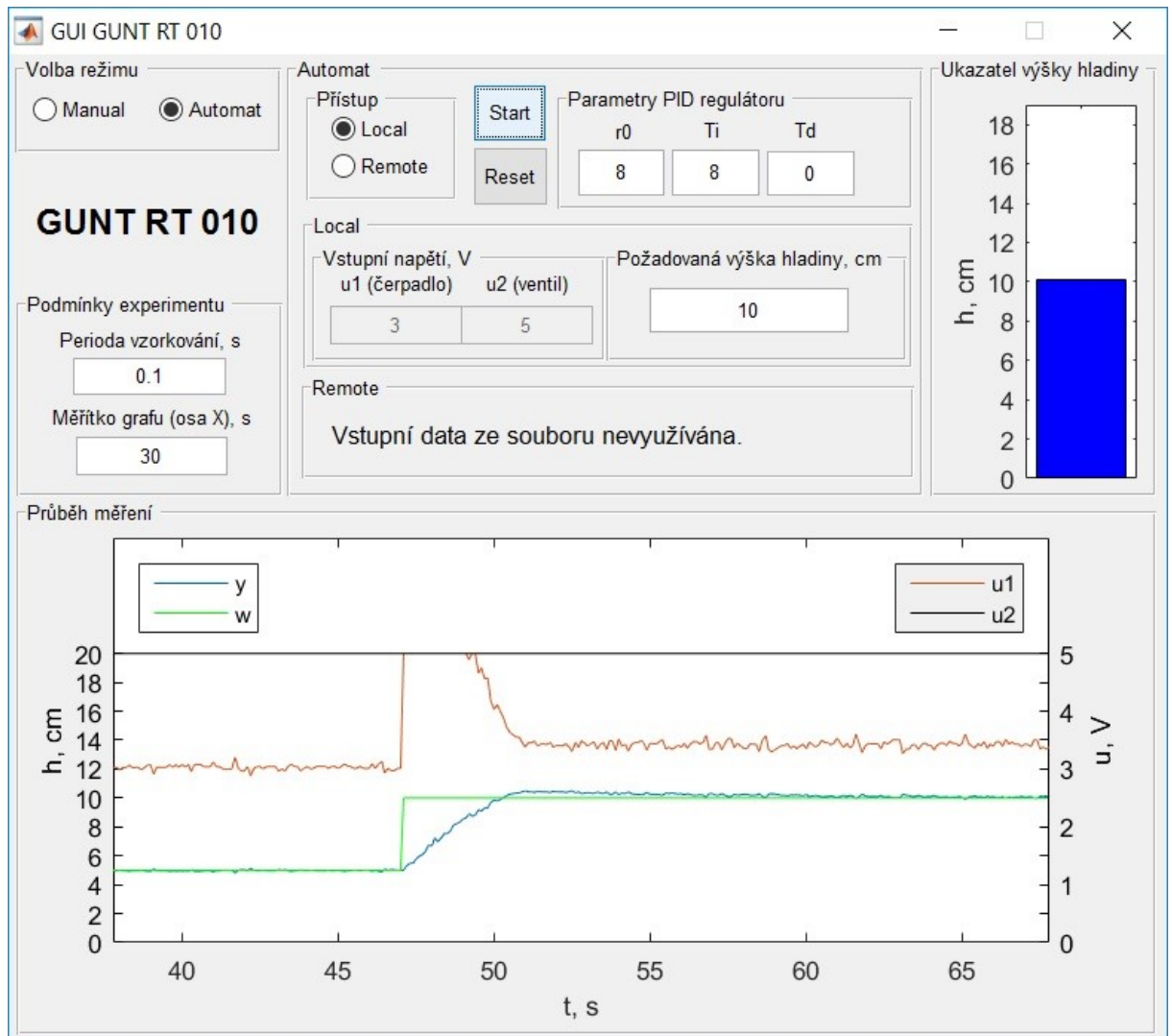
Obr. 2.10 – GUI, měření v ručním režimu

Tento regulační experiment je ukázkou práce v ručním režimu (Manual). Měření bylo zahájeno při počáteční výšce hladiny 10 cm. Následující tabulka ukazuje posloupnost zadávaných hodnot napětí pro čerpadlo ( $u_1$ ) a ventil ( $u_2$ ).

Tab. 2.1 – Vstupní hodnoty experimentu

$t, s$	$u_1, V$	$u_2, V$
0	0	0
5	0	5
10	3	4
15	5	4
20	2	5
25	3	5

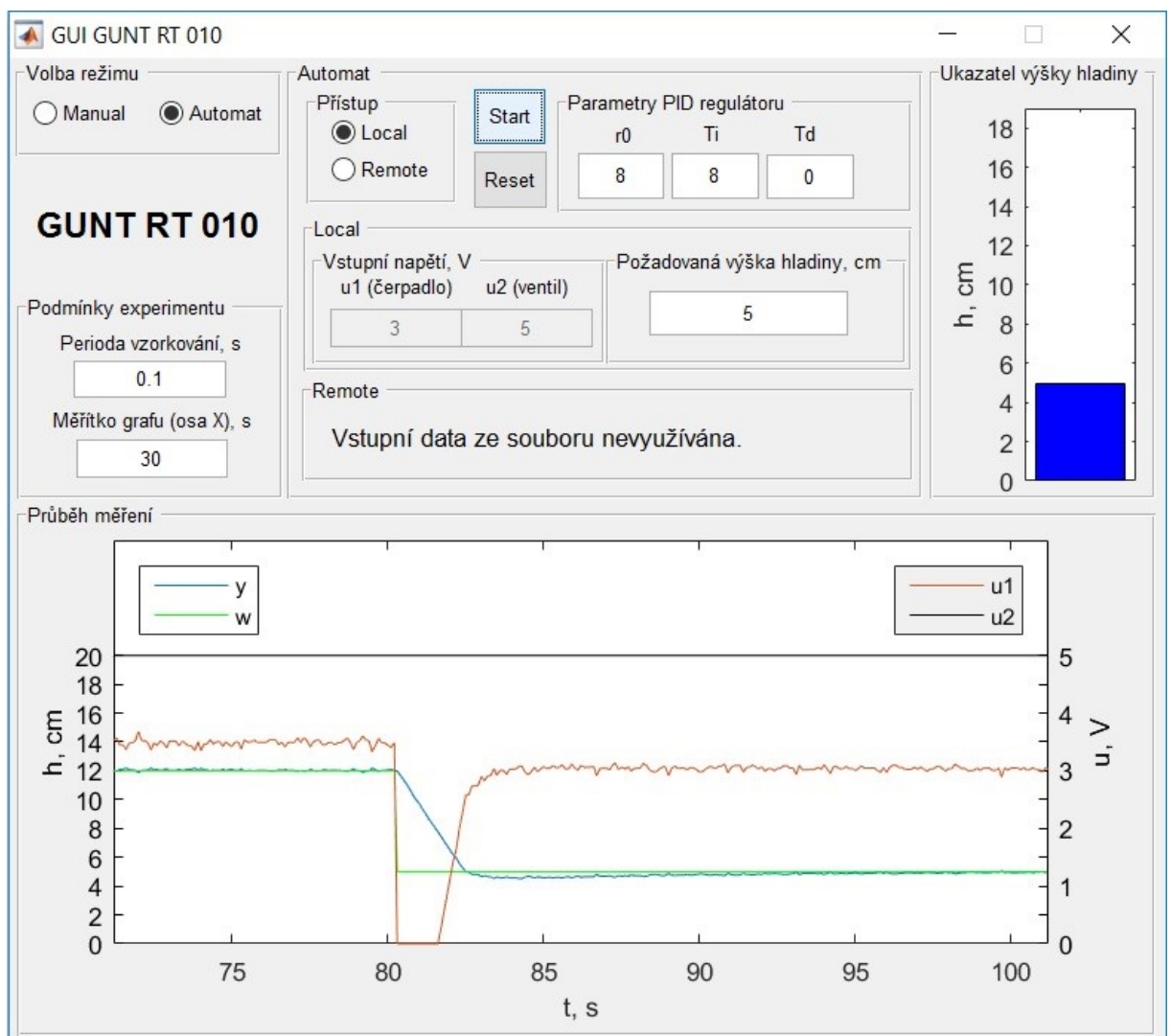
### 2.3.2 Měření odezvy na skokovou změnu žádané hodnoty výšky hladiny z 5 na 10 cm



Obr. 2.11 – GUI, odezva na změnu žádané hodnoty výšky hladiny z 5 na 10 cm

Při tomto regulačním experimentu byl použit režim Automat. Měření bylo zahájeno z počáteční hodnoty výšky hladiny 5 cm. Žádaná hodnota hladiny provozní nádrže byla poté nastavena na 10 cm, seznam ostatních parametrů je uveden výše. Z tohoto měření je vidět omezení vstupního napětí čerpadla, které může být maximálně 5 V.

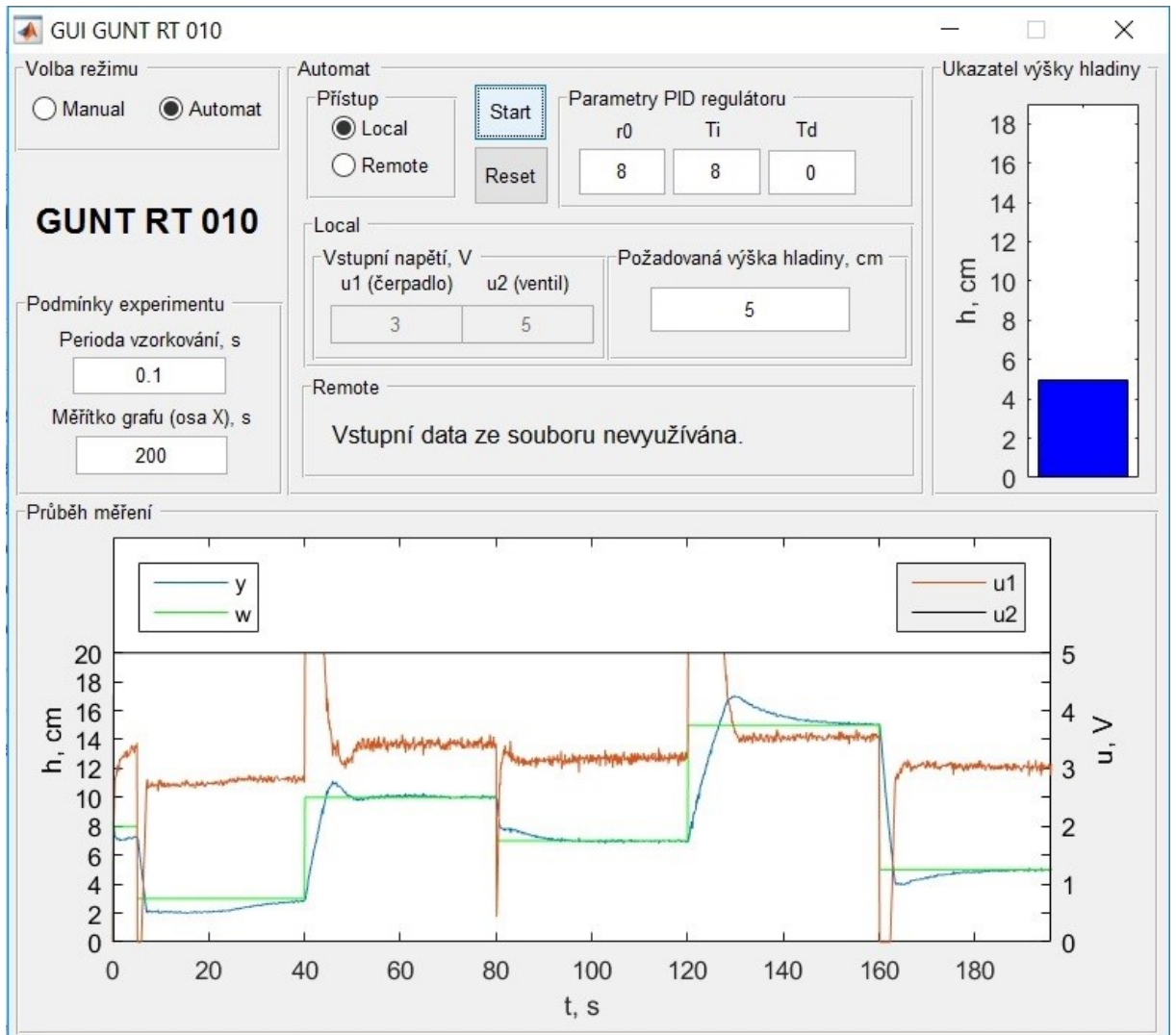
### 2.3.3 Měření odezvy na skokovou změnu žádané hodnoty výšky hladiny z 12 na 5 cm



Obr. 2.12 – GUI, odezva na změnu žádané hodnoty výšky hladiny z 12 na 5 cm

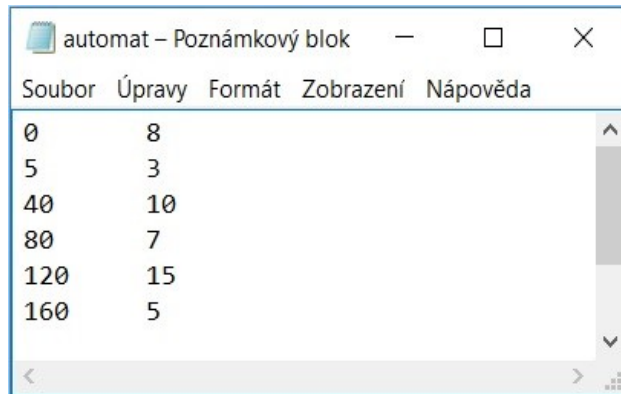
Měření bylo zahájeno při počáteční výšce hladiny 12 cm. Žádaná hodnota hladiny provozní nádrže byla poté nastavena na 5 cm, seznam ostatních parametrů je uveden výše. Pro větší skokové změny žádané hodnoty výšky hladiny, trvá déle, než se výstupní veličina dostane do ustáleného stavu. Při větších skokových změnách žádané hodnoty nastává větší překmit.

### 2.3.4 Měření odezvy na skokové změny žádané hodnoty výšky hladiny



Obr. 2.13 – GUI, odezva na skokové změny žádané hodnoty výšky hladiny

V této ukázce měření je vidět posloupnost několika skokových změn žádané veličiny. Měření bylo zahájeno z počáteční hodnoty výšky hladiny 8 cm. Vstupní data pro tento experiment byla definována v textovém souboru automat.txt. Obsah tohoto souboru je vidět na obr. 2.14, kde první sloupec udává čas a druhý sloupec odpovídající hodnoty požadované výšky hladiny.



The screenshot shows a Notepad window with the title 'automat - Poznámkový blok'. The menu bar includes 'Soubor', 'Úpravy', 'Formát', 'Zobrazení', and 'Nápověda'. The text area contains a table with two columns of data:

0	8
5	3
40	10
80	7
120	15
160	5

Obr. 2.14 – Vstupní data, odezva na skokové změny

Z GUI (obr. 2.13) při měření odezvy na skokové změny žádané hodnoty výšky hladiny, si můžeme všimnout, že ačkoli vstupní data pro experiment získáváme ze souboru automat.txt, v GUI je nastaven režim přístupu k vstupním datům na hodnotu Local a získávání vstupních dat z textového souboru tedy není využito. To je však pouze z důvodu, že obrázek GUI byl pořízen až po dokončení měření. Jakmile GUI vyčerpá všechna data ze souboru, automaticky dojde ke změně režimu z Remote na Local a zůstává nastavena poslední hodnota žádané veličiny. Tato změna nastala v čase  $t = 160$  s.

### 3 ZÁVĚR

Bakalářská práce je věnována tvorbě GUI v prostředí MATLAB pro regulaci a měření výšky hladiny a práci s laboratorní stanicí GUNT RT 010.

V teoretické části byl popsán systém a jeho vlastnosti, regulace a funkce PID regulátoru. Tato část vysvětluje základní princip prvků uzavřeného regulačního obvodu. Ukazuje nejčastěji využívané signály přiváděné na vstup soustavy a popis parametrů PID regulátoru. Dále je zde popsáno prostředí MATLAB, zejména část týkající se grafických objektů. Obsahuje popis M-souborů, využívaných při tvorbě skriptů a funkcí, hierarchii jednotlivých grafických objektů a jejich popis, identifikátory a vlastnosti objektů, popis tvorby jednoduchého GUI a způsob vkládání jednotlivých objektů. Závěrem teoretické části je popis použitých zařízení, a to laboratorní soustavy GUNT RT 010 a měřicí karty LabJack U12.

Implementační část obsahuje vlastní tvorbu GUI a praktické ukázky. Ukázky byly zvoleny tak, aby byla vidět činnost co možná nejvíce prvků v GUI. Ukázky demonstrují chování regulované veličiny při skokové změně žádané veličiny. Z průběhů můžeme posoudit vhodnost použitého PID regulátoru se zvolenými parametry. Pro regulaci výšky hladiny pracovní nádrže soustavy GUNT RT 010 je nejvhodnější využití regulátoru PI, protože se jedná o soustavu prvního řádu. Určení parametrů regulátoru jsem provedl experimentálně metodou pokus omyl, tyto hodnoty jsou v GUI nastaveny jako výchozí a s jejich použitím dosáhneme kvalitních regulačních pochodů. Průběh každého měření je zobrazen v grafu. Pokud potřebujeme záznam měřených hodnot, získáme je z datového souboru, do kterého se průběžně ukládají.

Z uvedených praktických ukázek lze vidět, že překmit regulované výšky hladiny nad požadovanou hodnotu nastává především při velkých skokových změnách žádané hodnoty. V těchto případech trvá déle, než se soustava dostane do ustáleného stavu.

## POUŽITÁ LITERATURA

- BALÁTĚ, J. 2003. *Automatické řízení*. Praha: BEN – technická literatura. ISBN 80-7300-020-2.
- DUŠEK, F.; HONC, D. 2000. *Matlab a Simulink, Úvod do používání*. skriptum, Univerzita Pardubice, vydání první, Pardubice.
- HERINGOVÁ, B.; HORA, P. 1995. *MATLAB: Díl I. – Práce s programem* [online]. [cit. 2017-03-26]. Dostupné z: <http://www.cdm.cas.cz/czech/hora/vyuka/mvs/tutorial.pdf>.
- HONC, D.; DUŠEK, F. 2013. MATLAB/Simulink Support for GUNT Control Units. In: *Proceedings of the 2013 International Conference on Process Control, Štrbské Pleso, Slovakia, červen 18 – 21, Slovak University of Technology in Bratislava*. p. 534 – 539.
- KADLEC, K. 2012. Výukové stanice pro měření a regulaci technologických veličin. *Automa* [online]. 18 (6), 8-10 [cit. 2017-03-12]. ISSN 1210-9592. Dostupné z: <http://www.automa.cz/page-flip/casopis/automa/2012/06/index.html#page/10>.
- MARCHAND, P.; HOLLAND, T. 2003. *Graphics and GUIs with MATLAB*. 3rd ed. Boca Raton, FL: Chapman & Hall/CRC, ISBN 1-58488-320-0.
- Třídění systémů. 2006. *Řízení technologických procesů* [online]. [cit. 2017-03-06]. Dostupné z: <http://rtp.webzdarma.cz/model3.php>.
- VROŽINA, M.; JANČIKOVA, Z.; DAVID, J. 2012. *Identifikace systémů*. Vyd. 1. Ostrava: Vysoká škola báňská – Technická univerzita, 1 CD-ROM. ISBN 978-80-248-2594-6.
- ZÁVIDČÁK, M. 2005. USB měřicí karta LabJack U12. *HW.cz: vše o elektronice a automatizaci* [online]. Praha: HW.cz, [cit. 2017-03-13]. Dostupné z: <http://vyvoj.hw.cz/produkty/usb-merici-karta-labjack-u12.html>.

# **PŘÍLOHY**

**A – CD**

**Příloha k bakalářské práci**

Vytvoření uživatelského rozhraní pro soustavu GUNT

Milan Míšenský

**CD**

## **Obsah**

- 1 Text bakalářské práce
- 2 Příklady použití GUI
- 3 Veškeré soubory nutné pro práci s navrženým GUI