

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Simulátor chování uzavřeného regulačního obvodu

Tomáš Benda

Bakalářská práce

2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš Benda**
Osobní číslo: **I09326**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Řízení procesů**
Název tématu: **Simulátor chování uzavřeného regulačního obvodu**
Zadávatel katedra: **Katedra řízení procesů**

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

Cílem je vytvořit simulátor chování uzavřeného regulačního obvodu - zpětnovazebního zapojení různých regulovaných soustav a regulátorů.

Obsah teoretické části: Zpětnovazební regulace - typické regulované soustavy a regulátory (dvupolohový regulátor a PID regulátory).

Obsah implementační části: Simulace regulačních pochodů v Simulinku. Diskretizace a převod řešení do MATLABu - vytvoření uživatelského rozhraní umožňujícího volbu typu a parametrů soustavy a regulátoru. Mělo by být možné definovat experiment - průběh žádané hodnoty a působení poruch a šumu měření.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

DUŠEK, F., HONC, D. Matlab a Simulink, Úvod do používání. skriptum, Univerzita Pardubice, vydání první, Pardubice, 2005. ISBN 80-7194-776-8.
Dorf, R. C., Bishop, R. H.: Modern Control Systems. 10 vyd. New Jersey: Prentice Hall, 2005. ISBN 0-13-127765-0.
Balátě, J.: Automatické řízení. 2 vyd. Praha: BEN - technická literatura, 2004. ISBN80-7300-148-9.

Vedoucí bakalářské práce: **Ing. Daniel Honc, Ph.D.**
Katedra řízení procesů

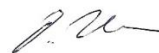
Datum zadání bakalářské práce: **21. prosince 2012**
Termín odevzdání bakalářské práce: **10. května 2013**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 29. března 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci 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.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne

Tomáš Benda

Poděkování

Tímto bych rád poděkoval panu Ing. Danielu Honcovi, Ph.D za poskytnuté teoretické rady a připomínky pro vývoj simulátoru chování regulačního obvodu. Dále bych rád poděkoval kamarádům a hlavně mé rodině za podporu během studia.

Anotace

Cílem práce je pomocí GUI ve výpočetním prostředí Matlab navrhnout aplikaci Simulátor chování regulačního obvodu. Teoretická část je věnována teorii automatického řízení, problematice simulace a popisu výpočetního systému Matlab.

V praktické části je popsána aplikace jako celek, jak ji ovládat, jaké parametry je možno zadávat do editačních polí a jsou uvedeny příklady, které mohou sloužit jako laboratorní úlohy pro studenty.

Klíčová slova

Matlab, automatické řízení, simulace

Title

Simulator of the closed loop behaviour

Anotation

The aim of the work is to design an application for the closed loop behaviour simulator in GUI in Matlab computing environment. The theoretical part is focused to the theory of automatic control, the issue of simulation and describes computing environment Matlab.

The practical part describes the whole application, how to control it, what parameters can be specified into the editable fields and provides examples that can serve as a laboratory exercise for students.

Keywords

Matlab, automatic control, simulation

Obsah

Seznam obrázku	9
Seznam zkratek a symbolů.....	13
1 Úvod.....	14
2 Teoretická část.....	15
2.1 Teorie automatického řízení	15
2.1.1 Uzavřený regulační obvod	15
2.1.2 Diferenciální rovnice systému a přenos.....	16
2.1.3 PID regulátor.....	16
2.1.4 Dvoupolohový regulátor	18
2.1.5 Přejchodová funkce a charakteristika	19
2.1.6 Impulsní funkce a charakteristika	20
2.1.7 Diskretizace spojitého systému.....	21
2.1.8 Diferenční rovnice	22
2.1.9 Číslicové regulátory	23
2.2 Simulace.....	25
2.2.1 Co je to simulace.....	25
2.2.2 Typy simulačních programů	25
2.2.3 Princip simulace v <i>Simulátoru chování regulačního obvodu</i>	26
2.3 Základní orientace v Matlabu	28
2.4 Princip vytváření GUI aplikací	30
2.4.1 Předpoklady k vytvoření GUI.....	30
2.4.2 Co je to systém Handle Graphics.....	30
2.4.3 Struktura M-File v GUI	31
3 Praktická část	32
3.1 Popis aplikace.....	32
3.1.1 Hlavní okno aplikace	32
3.1.2 Nastavení žádané hodnoty	32
3.1.3 Nastavení soustavy	38
3.1.4 Nastavení PID a dvoupolohového regulátoru.....	40
3.1.5 Nastavení doby simulace	42

3.1.6	Zobrazení nastavených parametru	42
3.1.7	Zobrazení odezvy systému.....	42
3.1.8	Spuštění aplikace v Matlabu	43
3.2	Příklady.....	46
3.2.1	Příklad 1 – Soustava 1. řádu s P regulátorem	46
3.2.2	Příklad 2 – Soustava 1. řádu s PI regulátorem	51
3.2.3	Příklad 3 – Soustava 1. řádu s Dvupolohovým regulátorem	57
3.2.4	Příklad 4 – Soustava 3. řádu s P a PI regulátorem	59
3.2.5	Příklad 5 – Soustava 3. řádu s Dvupolohovým regulátorem	62
4	Závěr.....	64
	Literatura	65

Seznam obrázku

Obrázek 1: Blokové schéma uzavřeného regulačního obvodu	15
Obrázek 2: Přechodová charakteristika PID regulátoru	18
Obrázek 3: Blokové schéma regulačního obvodu s dvoupolohovým regulátorem.....	18
Obrázek 4: Přepínací charakteristika dvoupolohového regulátoru.....	19
Obrázek 5: Jednotkový skok.....	20
Obrázek 6: Jednotkový impuls	21
Obrázek 7: Funkce pro výpočet diferenciální rovnice.....	25
Obrázek 8: Volání funkce – rovnice(x,Y)	25
Obrázek 9: Metoda určení přenosu z diferenciální rovnice	26
Obrázek 10: Vytvoření spojitého přenosu v Matlabu	26
Obrázek 11: Diskretizace spojitého systému	27
Obrázek 12: Úvodní okno Matlabu	28
Obrázek 13: Nastavbové prostředí Simulink	29
Obrázek 14: Rozhraní pro tvorbu uživatelských aplikací.....	30
Obrázek 15: Hierarchie systému Handle Graphics	30
Obrázek 16: Vzhled celé aplikace	32
Obrázek 17: Žádaná hodnota	33
Obrázek 18: Žádaná hodnota - možnosti volby.....	33
Obrázek 19: Dialogové okno - nastavení jednotkového skoku	33
Obrázek 20: Vygenerovaný signál - Jednotkový skok	34
Obrázek 21: Dialogové okno - nastavení rampy	34
Obrázek 22: Vygenerovaný signál – Rampa	35
Obrázek 23: Dialogové okno - Libovolná žádaná hodnota a porucha.....	35
Obrázek 24: Rozšíření aplikace - modelování signálu	36
Obrázek 25: Detekce bodu - změna kurzoru	36
Obrázek 26: Nastavení bodu - pravé tlačítko myši.....	36
Obrázek 27: Pohyb bodu.....	37
Obrázek 28: Nastavení signálů - možnosti volby.....	37
Obrázek 29: Barevné odlišení signálů	37
Obrázek 30: Dialogové okno – nastavení šumu	38
Obrázek 31: Příklad vygenerovaného šumu	38

Obrázek 32: Soustava – možnosti volby.....	39
Obrázek 33: Dialogové okno – přenos 1. řádu	39
Obrázek 34: Dialogové okno – přenos libovolného řádu	40
Obrázek 35: Regulátor – možnosti volby	40
Obrázek 36: Dialogové okno – nastavení PID regulátoru.....	40
Obrázek 37: Dialogové okno – nastavení Dvoupolohového regulátoru.....	41
Obrázek 38: Grafické vyobrazení parametru v přepínací charakteristice u nastavení dvoupolohového regulátoru	41
Obrázek 39: Editační pole – doba simulace.....	42
Obrázek 40: Zobrazení nastavených hodnot.....	42
Obrázek 41: Zobrazení odezvy systému – základní režim	43
Obrázek 42: Zobrazení odezvy systému – rozšířený režim	43
Obrázek 43: Spuštění GUIDE Quick Start	44
Obrázek 44: Vyhledání aplikace.....	44
Obrázek 45: Vyhledání aplikace.....	44
Obrázek 46: Rozmístění ovládacích prvku v GUI.....	45
Obrázek 47: Dvojí způsob spuštění aplikace.....	45
Obrázek 48: Dialogové okno změna Current Folder	46
Obrázek 49: Spuštěná aplikace	46
Obrázek 50: Nastavení jednotkového skoku - Příklad 1.....	49
Obrázek 51: Nastavení přenosu soustavy – Příklad 1.....	50
Obrázek 52: Nastavení PID regulátoru – Příklad 1	50
Obrázek 53: Odvozené vztahy – Příklad 1	51
Obrázek 54: Simulátor zpětnovazebního obvodu – Příklad 1	51
Obrázek 55: Nastavení jednotkového skoku - Příklad 2.....	55
Obrázek 56: Nastavení přenosu soustavy – Příklad 2.....	55
Obrázek 57: Nastavení PID regulátoru – Příklad 2	55
Obrázek 58: Odvozené vztahy – Příklad 2	56
Obrázek 59: Simulátor zpětnovazebního obvodu – Příklad 2	56
Obrázek 60: Nastavení jednotkového skoku - Příklad 3.....	57
Obrázek 61: Nastavení přenosu soustavy – Příklad 3.....	57
Obrázek 62: Nastavení Dvoupolohového regulátoru – Příklad 3	58
Obrázek 63: Simulátor zpětnovazebního obvodu – Příklad 3	58
Obrázek 64: Zpětnovazební obvod na hranici stability – netlumené kmity	59

Obrázek 65: Soustava 3. řádu – seřízení PI regulátorem	62
Obrázek 66: Soustava 3. řádu – seřízení P regulátorem	62
Obrázek 67: Soustava 3. řádu – Dvoupolohový regulátor	63

Seznam tabulek

Tabulka 1: Diskretizace spojitého systému	21
Tabulka 2: Nastavení dvoupolohového regulátoru Příklad 3	57
Tabulka 3: Průběh žádané hodnoty Příklad 4	59
Tabulka 4: Seřízení PID regulátoru - Ziegler-Nichols.....	61
Tabulka 5: Nastavení dvoupolohového regulátoru Příklad 5	62

Seznam zkratk a symbolů

PP	Počáteční podmínka
GUI	Graphical User Interface
y	Regulovaná veličina
w	Žádaná hodnota
u	Akční veličina
e	Regulační odchylka
r_K	Kritické zesílení
T_K	Kritická perioda kmitu
G_{Yw}	Přenos regulované veličiny z žádané hodnoty
G_{Uw}	Přenos akční veličiny z žádané hodnoty
G_{Ew}	Přenos regulační odchylky z žádané hodnoty
G_S	Přenos soustavy
G_R	Přenos regulátoru

1 Úvod

Cílem této bakalářské práce je navrhnout aplikaci s názvem *Simulátor chování regulačního obvodu* ve výpočetním systému Matlab pomocí Graphical User Interface (GUI).

Teoretická část je rozdělena na tři podkapitoly - teorie automatického řízení, simulace a základní orientaci v Matlabu. V první podkapitole jsou popsány základní pojmy týkající se zpětnovazebního obvodu, matematického popisu spojitého systému, základní typy regulátorů a jejich vhodnost použití, diferenční rovnice a základní typy odezev. Dále je zde uveden princip diskretizace spojitého systému a odvození algoritmu číslicového PID regulátoru. Druhá podkapitola je věnována simulaci. Je v ní uvedeno, co je to simulace, jaké máme typy simulačních programů a také je věnována základní orientaci ve výpočetním systému Matlab a tvorby GUI aplikací.

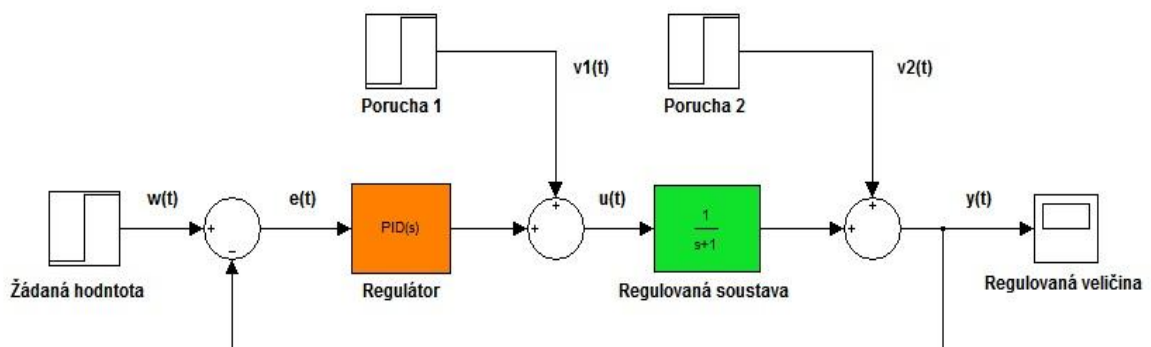
Praktická část je rozdělena do dvou podkapitol. První podkapitola je věnována podrobnému popisu aplikace, jaké jsou celkové možnosti nastavení, co se týče žádaných hodnot, přenosů soustavy, PID regulátorů, dvoupolohového regulátoru, doby simulace a možnosti vyobrazení regulačních pochodů. Dále je u každého tohoto nastavení uvedeno, co uživatel smí, respektive nesmí zadat. V druhé podkapitole je uvedeno několik praktických příkladů včetně jejich analytických řešení, na kterých si uživatel může vyzkoušet celou aplikaci.

2 Teoretická část

2.1 Teorie automatického řízení

2.1.1 Uzavřený regulační obvod

Úkol regulace je nastavení technických parametru například udržování teploty v místnosti [1, str. 42-43], výška hladiny v nádobě a mnohé další elektrické i neelektrické veličiny na základě požadované hodnoty. Snažit se tuto hodnotu dodržovat i v případě výskytu poruchy.



Obrázek 1: Blokové schéma uzavřeného regulačního obvodu

Regulovaná soustava je objekt nebo skupina objektu, která podléhá regulaci a je ovládána regulátorem. Příklad regulované soustavy může být nádrž s přítokem a odtokem. Výstupem regulované soustavy je regulovaná veličina označuje se většinou písmenem $y(t)$.

Žádaná hodnota označena písmenem $w(t)$ je to předepsaná hodnota, na které má být regulovaná veličina udržována. V uzavřeném regulačním obvodu se tato veličina neustále porovnává s regulovanou veličinou. Tomuto rozdílu mezi regulovanou veličinou a žádanou hodnotou říkáme regulační odchylka $e(t)$ a platí následující rovnice.

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

Regulační odchylka by měla být v ideálním případě nulová nebo co nejmenší. Jestliže je tento rozdíl nenulový je proveden akční zásah. To provádí výstupní veličina regulátoru akční veličina $u(t)$. Regulátor musí být zapojen tak, aby akční veličina snižovala regulační odchylku.

Další příčiny proč regulovat je výskyt poruch – poruchová veličina $v_1(t)$ a $v_2(t)$, tyto poruchy nám velice výrazně ovlivňují regulovanou soustavu a celkový průběh regulované veličiny. Poruchová veličina má většinou obdobný průběh jako žádaná veličina.

2.1.2 Diferenciální rovnice systému a přenos

Lineární spojité systém [1, str. 50-51] se vstupem $u(t)$ a výstupem $y(t)$ je možno obecně popsat následující diferenciální rovnicí.

$$a_n y^{(n)} + a_{n-1} y^{(n-1)} + \dots + a_1 y' + a_0 y = b_n u^{(m)} + b_{n-1} u^{(m-1)} + \dots + b_1 u' + b_0 u \quad (2)$$

Mezi nejdůležitější podmínku realizovatelnosti jakéhokoli systému patří, že řád nejvyšší derivace výstupní veličiny je vždy vyšší nebo rovno než nejvyšší řád derivace vstupní veličiny.

$$m \leq n \quad (3)$$

Jestliže je známa obecná rovnice systému, průběh vstupní veličiny a počáteční podmínky, lze určit odezvu libovolného systému. Mezi nejobvyklejší a pro nás přirozenější zápis této rovnice je přenos.

Přenos je definován jako poměr Laplaceova obrazu výstupní veličiny ku Laplaceova obrazu vstupní veličiny, při nulových počátečních podmínkách a platí rovnice.

$$G(s) = \frac{L\{y(t)\}}{L\{u(t)\}} = \frac{Y(s)}{U(s)} \quad (4)$$

S použitím znalosti Laplaceovy transformace přejdu k odvození obecného přenosu.

$$\begin{aligned} L\{a_n y^{(n)} + a_{n-1} y^{(n-1)} + \dots + a_1 y' + a_0 y\} &= L\{b_n u^{(m)} + b_{n-1} u^{(m-1)} + \dots + b_1 u' + b_0 u\} \\ a_n s^{(n)} Y(s) + a_{n-1} s^{(n-1)} Y(s) + \dots + a_1 s Y(s) + a_0 Y(s) &= b_n s^{(n)} U(s) + b_{n-1} s^{(n-1)} U(s) + \dots + b_1 s U(s) + b_0 U(s) \\ Y(s) [a_n s^{(n)} + a_{n-1} s^{(n-1)} + \dots + a_1 s + a_0] &= U(s) [b_n s^{(n)} + b_{n-1} s^{(n-1)} + \dots + b_1 s + b_0] \end{aligned} \quad (5)$$

Dosažením tohoto výsledku do definice přenosu lze získat velice důležitý vzorec pro výpočet přenosu z diferenciální rovnice.

$$G(s) = \frac{b_n s^{(n)} + b_{n-1} s^{(n-1)} + \dots + b_1 s + b_0}{a_n s^{(n)} + a_{n-1} s^{(n-1)} + \dots + a_1 s + a_0} \quad (6)$$

2.1.3 PID regulátor

Regulátor je zařízení [1, str. 100-101], které slouží k dosažení a udržování požadované hodnoty. Vstupní veličina regulátoru je regulační odchylka a výstupní veličina akční zásah. Regulátor může tuto vstupní veličinu buďto zesilovat, integrovat nebo derivovat.

P-regulátor

Je nejjednodušší, protože se realizuje prostým zesilovačem. Akční veličina je proporcionální, tedy přímo úměrná regulační odchylce. Z toho vyplývá, čím větší regulační odchylku detekuje tím, je větší jeho akční zásah. Akční veličinu můžeme popsat vztahem.

$$u(t) = r_0 e(t) \quad (7)$$

Není vhodný k regulaci soustav vyšších řádu a soustav s dopravním zpožděním. Jeho výhody je dostatečná rychlost regulace a stabilita. Mezi jeho hlavní nevýhody patří trvalá regulační odchylka.

I-regulátor

Z matematického hlediska se jedná o složitější typ regulátoru, protože akční veličina je úměrná integrálu z regulační odchylky. Akční veličinu vyjádřím tímto vztahem.

$$u(t) = r_{-1} \int e(t) dt \quad (8)$$

Jestliže tímto regulátorem budu regulovat proporcionální soustavy lze zcela odstranit regulační odchylku.

D-regulátor

Z technického pohledu nelze tento typ regulátoru realizovat, kde by byla akční veličina úměrná derivaci regulační odchylky, protože by se v ustáleném stavu zesilovali šumy v obvodu a to je pro nás nežádoucí. Tento typ regulátoru lze použít pouze v kombinaci PD nebo PID. Akční veličinu vyjádřím tímto vztahem.

$$u(t) = r_1 e'(t) \quad (10)$$

Pro určení rovnice ideálního PID regulátoru stačí pouze sečíst tyto tři složky a dostanu tvar.

$$u(t) = r_0 e(t) + r_{-1} \int e(t) dt + r_1 e'(t) \quad (11)$$

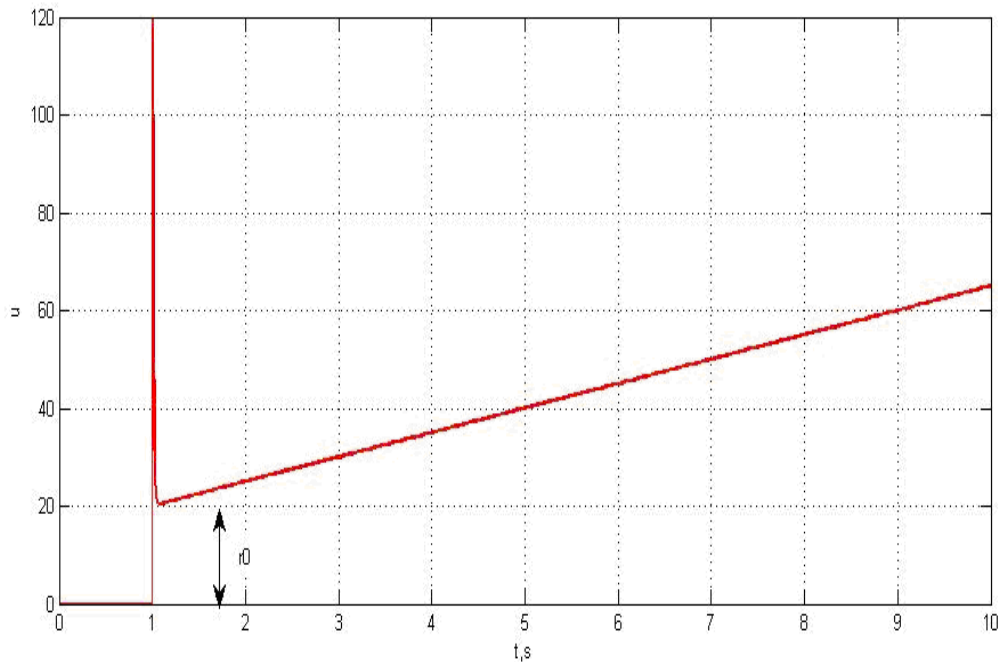
Nyní je nutné přenos z přecházející rovnice. Konstanty r_0, r_{-1}, r_1 určují vliv jednotlivé složky tedy proporcionální, integrační a derivační. Tyto konstanty můžu měnit podle potřeby, aby regulační obvod splňoval to, co od něj potřebuji.

$$G_R(s) = \frac{U(s)}{E(s)} = r_0 + \frac{r_{-1}}{s} + r_1 s \quad (12)$$

Jestliže z tohoto přenosu vytknu konstantu r_0 dostanu se k následujícímu přenosu o tvaru.

$$G_R(s) = \frac{U(s)}{E(s)} = r_0 + \frac{r_{-1}}{s} + r_1 s = r_0 \left(1 + \frac{1}{\frac{r_0}{r_{-1}} s} + \frac{r_1}{r_0} s \right) = r_0 \left(1 + \frac{1}{T_I s} + T_D s \right) \quad (13)$$

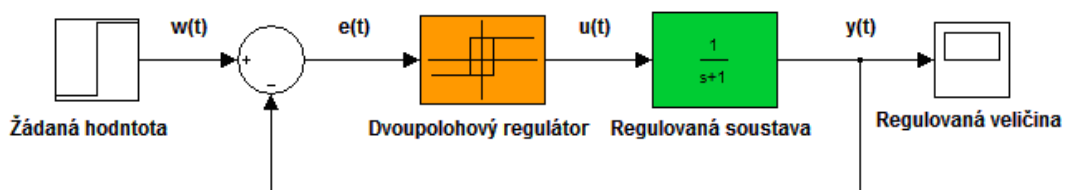
Konstanty r_0 , T_I , T_D v tomto případě znamenají zesílení regulátoru, integrační časovou konstantu a derivační časovou konstantu.



Obrázek 2: Přejchodová charakteristika PID regulátoru

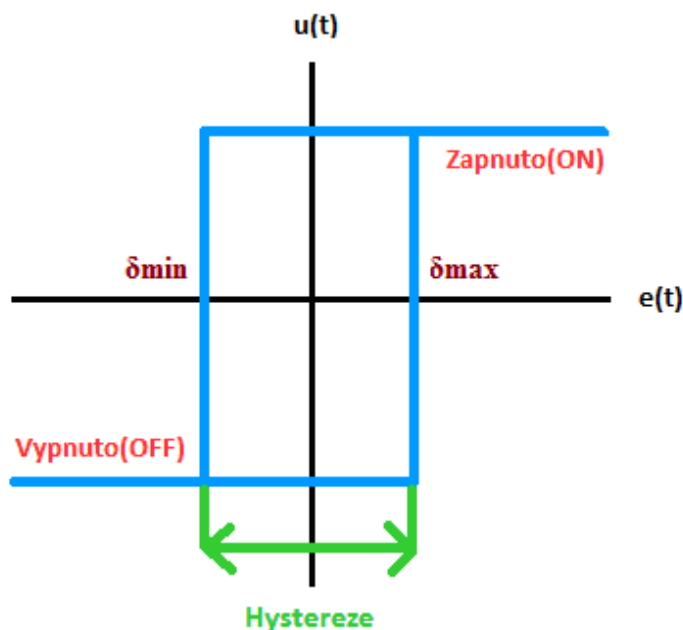
2.1.4 Dvoupolohový regulátor

Používá se všude tam, kde není kladen tak velký důraz na přesnost regulace. Velkou výhodou je jednoduchost použití, nízké pořizovací a provozní náklady. Schéma regulačního obvodu je na obrázku 3.



Obrázek 3: Blokové schéma regulačního obvodu s dvoupolohovým regulátorem

Dvoupolohový regulátor se liší od spojitého regulátoru tím, že akční veličina může nabývat pouze dvou hodnot a to hodnotu „zapnuto“ nebo „vypnuto“. Abychom tento regulátor nezničili, volíme takzvané pásmo necitlivosti neboli hysterezi tak, aby nedocházelo k častému zapínání a vypínání regulátoru. Vlastnosti respektive chování regulátoru je možno zobrazit pomocí přepínací charakteristiky z obrázku 4.



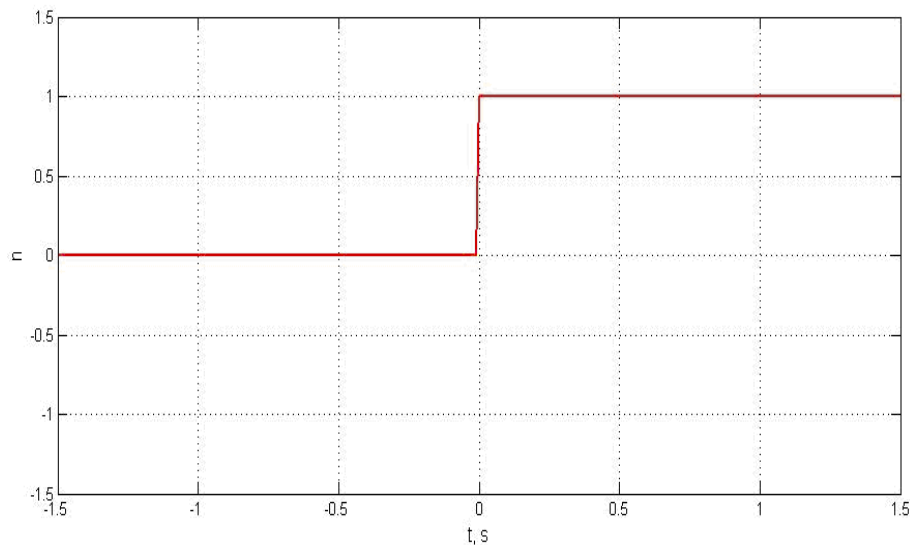
Obrázek 4: Přepínací charakteristika dvoupolohového regulátoru

Jestliže byl minulý stav „zapnuto“ udržuj tento stav, dokud regulační odchylka neklesne pod hodnotu δ_{MIN} a když klesne pod tuto hodnotu, změň stav na „vypnuto“. Jestliže byl minulý stav „vypnuto“ udržuj tento stav, dokud regulační odchylka nepřekročí hodnotu δ_{MAX} a když překročí tuto hodnotu, změň stav na „zapnuto“.

2.1.5 Přechodová funkce a charakteristika

Přechodová funkce [2, str. 77] $y(t) = h(t)$ je odezva systému na jednotkový skok $u(t) = \eta(t)$ na vstupu. Grafem přechodové funkce je přechodová charakteristika. Jednotkový skok je funkce, kterou můžu matematicky popsat. Do času $t(0)$ má nulovou hodnotu a v čase $t(0)$ nabývá hodnoty jedna. A platí následující matematický zápis.

$$\eta(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (14)$$



Obrázek 5: Jednotkový skok

Jestliže je známá diferenciální rovnice systému a vstupní signál. A uvažuje se, že děj začíná v čase $t(0)$. Lze tedy odvodit vztah pro přechodovou funkci. Laplaceův obraz jednotkové ho skoku $L\{u(t)\} = L\{\eta(t)\} = \frac{1}{s}$.

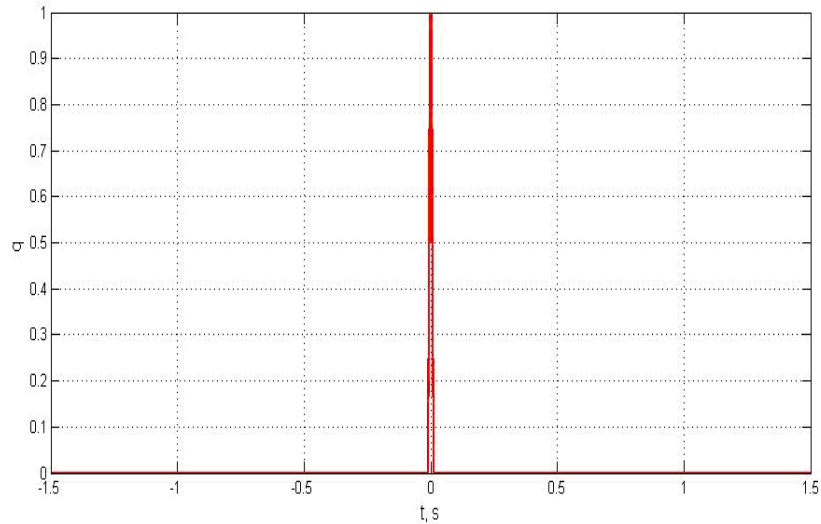
$$Y(s) = H(s) = G(s)U(s) = G(s)\frac{1}{s} = \frac{G(s)}{s} \quad (15)$$

$$y(t) = h(t) = L^{-1}\left\{\frac{G(s)}{s}\right\}$$

2.1.6 Impulsní funkce a charakteristika

Impulsní funkce [2, str. 78] $g(t) = y(t)$ je odezva systému na jednotkový (Diracův) impuls obvykle značíme $\delta(t) = u(t)$, který přivádím na vstup systému. Grafem impulsní funkce je impulsní charakteristika. Jednotkový impuls můžu z hlediska matematiky popsat obdobným způsobem jako jednotkový skok. Jednotkový impuls se jeví jako nekonečně krátký impuls nenulové hodnoty.

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases} \quad (16)$$



Obrázek 6: Jednotkový impuls

Jestliže je známá diferenciální rovnice systému a vstupní signál. A uvažuje se, že děj začíná v čase $t = 0$. Lze tedy odvodit vztah pro impulsní funkci. Laplaceův obraz jednotkového (Diracova) impulsu $L\{u(t)\} = L\{\delta(t)\} = 1$.

$$\begin{aligned}
 Y(s) &= H(s) = G(s)U(s) = G(s) \cdot 1 = G(s) \\
 y(t) &= g(t) = L^{-1}\{G(s)\}
 \end{aligned}
 \tag{17}$$

2.1.7 Diskretizace spojitého systému

Je mnoho způsobu, jak provést diskretizaci spojitého systému. V této podkapitole bude v krátkosti popsána nejpoužívanější metoda. Tato metoda se nazývá přibližná metoda dsterizace [1, str. 185-186], kde se použijí náhrady derivace a integrálu. Čili bude provedena náhrada diferenciální rovnice za diferenční. Tato metoda se používá u malých vzorkovacích period.

Tabulka 1: Diskretizace spojitého systému

Spojité funkce	Diskrétní funkce
$\frac{dx(t)}{dt}$	$\frac{x(k) - x(k-1)}{T}$
$\frac{dx^2(t)}{dt}$	$\frac{x(k) - 2x(k-1) + x(k-2)}{T^2}$

Proces distretizace ukáži na diferenciální rovnici 1. řádu.

$$a_1 y' + a_0 y = b_0 u \quad (18)$$

Provedu náhradu derivací diferencemi podle tabulky 1. A dostanu rovnici v následujícím tvaru.

$$a_1 \frac{y(k) - y(k-1)}{T} + a_0 y(kT) = b_0 u(kT) \quad (19)$$

Jestliže dosadím za vzorkovací periodu $T = 1$ dostanu diferenční rovnici ve tvaru.

$$(a_0 + a_1)y(k) - a_1 y(k-1) = b_0 u(k) \quad (20)$$

2.1.8 Diferenční rovnice

Oblast matematiky nabízí popis systému v diskrétní neboli číslicové formě a to zavedením diferenčních rovnic.

Základním stavebním kamenem [1, str. 185-186] u těchto rovnic je pojem difference funkce $f(kT)$. Kde konstanta k , znamená kolikátý vzorek a konstanta T je vzorkovací perioda. Existují dva typy diferencí a to dopředná nebo zpětná. Matematický zápis vypadá následovně pro $T \neq 1$.

První dopředná difference	První zpětná difference	
$\Delta f(kT) = f[(k+1)T] - f(kT)$	$\Delta f(kT) = f(kT) - f[(k-1)T]$	(21)

Druhá dopředná difference	Druhá zpětná difference	
$\Delta^2 f(kT) = \Delta f[(k+1)T] - \Delta f(kT)$	$\Delta^2 f(kT) = \Delta f(kT) - \Delta f[(k-1)T]$	(22)

n-tá dopředná difference	n-tá zpětná difference	
$\Delta^n f(kT) = \Delta^{n-1} f[(k+1)T] - \Delta^{n-1} f(kT)$	$\Delta^n f(kT) = \Delta^{n-1} f(kT) - \Delta^{n-1} f[(k-1)T]$	(23)

Když tyto všechny difference sečtu, dostanu diferenční rovnici systému n-tého řádu. Rovnici vyjádřím pro hodnotu $T = 1$.

S dopředných diferencí

$$\alpha_n \Delta^n y(k) + \alpha_{n-1} \Delta^{n-1} y(k) + \dots + \alpha_1 \Delta y(k) + \alpha_0 y(k) = \beta_n \Delta^n u(k) + \beta_{n-1} \Delta^{n-1} u(k) + \dots + \beta_1 \Delta u(k) + \beta_0 u(k) \quad (24)$$

Ze zpětných diferencí

$$\alpha_n \Delta^n y(k) + \alpha_{n-1} \Delta^{n-1} y(k) + \dots + \alpha_1 \Delta y(k) + \alpha_0 y(k) = \beta_n \Delta^n u(k) + \beta_{n-1} \Delta^{n-1} u(k) + \dots + \beta_1 \Delta u(k) + \beta_0 u(k) \quad (25)$$

Jestliže nahradím za diference konkrétní funkční hodnoty, dostanu se k takzvanému rekurentnímu tvaru.

Z dopředných diferencí

$$a_n y(k+n) + \dots + a_1 y(k+1) + a_0 y(k) = b_n u(k+n) + \dots + b_1 u(k+1) + b_0 u(k) \quad (26)$$

Ze zpětných diferencí

$$a_0 y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_0 u(k) + b_0 u(k-1) + \dots + b_0 u(k-n) \quad (27)$$

Jak u diferenciálních rovnic, tak rovněž u diferenčních rovnic, je nutno znát počáteční podmínky. U dopředných diferencí v následujícím tvaru pro $k = 0 \dots n$ a $T = 1$.

$$\begin{aligned} y(k) &= y(0) = \text{PP} \\ y(k+1) &= y(1) = \text{PP} \\ y(k+2) &= y(2) = \text{PP} \end{aligned} \quad (28)$$

U zpětných diferencí v následujícím tvaru pro $k = 0 \dots n$ a $T = 1$.

$$\begin{aligned} y(k) &= y(0) = \text{PP} \\ y(k-1) &= y(-1) = \text{PP} \\ y(k-2) &= y(-2) = \text{PP} \end{aligned} \quad (29)$$

2.1.9 Číslicové regulátory

Od číslicového regulátoru budu požadovat úplně stejné chování jako u spojitého regulátoru. To znamená [1, str. 202-204], že číslicový regulátor bude schopen zesilovat, integrovat a derivovat a proto můžu vycházet z rovnice spojitého regulátoru.

$$u(t) = r_0 e(t) + r_{-1} \int e(t) dt + r_1 e'(t) \quad (30)$$

Jelikož se jedná o disktrétní formu regulátoru, je nutné použít náhrady derivace a integrace v rovnici. U derivace bude potřeba použít náhradu derivace diferencí a náhradu integrálu sumací. Tomuto algoritmu se říká polohový algoritmus, lze použít tam, kde není sumační složka. A výsledná rovnice disktrétního regulátoru vypadá následovně.

$$\begin{aligned}\frac{de(t)}{dt} &= \frac{\Delta e(k)}{T} = \frac{e(k) - e(k-1)}{T} \\ \int_0^{kT} e(t) dt &= T \sum_{i=1}^k e(i) \\ u(k) &= r_0 \left\{ e(k) + \frac{T}{T_I} \sum_{i=1}^k e(i) + \frac{T_D}{T} [(e(k) - e(k-1))] \right\}\end{aligned}\quad (31)$$

Já se budu zabývat přírůstkovým algoritmem PSD regulátoru. Jak název napovídá, nejedná se o hodnotu akční veličiny v daném okamžiku, ale o přírůstek akční veličiny oproti minulé hodnotě.

$$\Delta u(k) = u(k) - u(k-1) \quad (32)$$

Nyní si stačí vyjádřit přírůstek minulé akční veličiny. K tomuto vyjádření použijí platnosti rovnice polohového algoritmu.

$$u(k-1) = r_0 \left\{ e(k-1) + \frac{T}{T_I} \sum_{i=1}^{k-1} e(i) + \frac{T_D}{T} [(e(k-1) - e(k-2))] \right\} \quad (33)$$

Jestliže je znám přírůstek aktuální a minulé akční veličiny, můžu tedy přejít k odvození přírůstkového algoritmu PSD regulátoru.

$$\begin{aligned}u(k) - u(k-1) &= r_0 \left\{ e(k) - e(k-1) + \frac{T}{T_I} \sum_{i=1}^k e(i) - \frac{T}{T_I} \sum_{i=1}^{k-1} e(i) + \frac{T_D}{T} [(e(k) - e(k-1))] - \frac{T_D}{T} [(e(k-1) - e(k-2))] \right\} \\ u(k) - u(k-1) &= r_0 \left\{ e(k) - e(k-1) + \frac{T}{T_I} e(k) + \frac{T_D}{T} [(e(k) - 2e(k-1) + e(k-2))] \right\} \\ u(k) - u(k-1) &= r_0 \left(1 + \frac{T}{T_I} + \frac{T_D}{T} \right) e(k) - r_0 \left(1 + 2\frac{T_D}{T} \right) e(k-1) + r_0 \frac{T_D}{T} e(k-2)\end{aligned}\quad (34)$$

Z tohoto tvaru nyní můžu určit koeficienty PSD regulátoru q_0, q_1, q_2 .

$$\begin{aligned}q_0 &= r_0 \left(1 + \frac{T}{T_I} + \frac{T_D}{T} \right) \\ q_1 &= -r_0 \left(1 + 2\frac{T_D}{T} \right) \\ q_2 &= r_0 \frac{T_D}{T}\end{aligned}\quad (35)$$

A následně výsledný přenos PSD regulátoru.

$$\begin{aligned}u(k) - u(k-1) &= q_0 e(k) - q_1 e(k-1) + q_2 e(k-2) \\ G_R(z) &= \frac{U(z)}{E(z)} = \frac{q_0 - q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}}\end{aligned}\quad (36)$$

2.2 Simulace

2.2.1 Co je to simulace

Počítačové simulace mají v dnešní době velké uplatnění v mnoha odvětvích, protože nám umožňuje odzkoušet chování daného systému při různých vstupních hodnotách, aniž by bylo nutné mít jeho fyzický respektive reálný model. Velkou výhodou simulace je, že dává možnost odstranění chyb či vyladění našeho programu, před tím než ho nahrajeme do zkonstruovaného zařízení a tím předešli jeho zničení.

2.2.2 Typy simulačních programů

Abychom mohli vytvořit simulační program, který nám usnadní naši práci, musíme znát příslušný programovací jazyk, program vytvořit a odladit. Simulační programy můžeme rozdělit na blokově orientované a rovnicově orientované.

Rovnicově orientované programy

Model je zadán ve formě algebraických rovnic [8], diferenciálních rovnic nebo diferenčních rovnic. V případě, že máme diferenciální rovnice vyšších řádu, je nutné tyto rovnice převést na soustavu diferenciálních rovnic 1. řádu metodou snižování řádu derivace, čili na levé straně musí být vyjádřena nejvyšší hodnota derivace, můžeme pozorovat na obrázku 7. Rovnicově orientovaný program je Matlab.

```
% y''+5y'+2 = 1|
function dY = rovnice(x,Y)
    y = Y(1);
    v = Y(2);

    dY = [v; (1-5*v-2*y)];
end
```

Obrázek 7: Funkce pro výpočet diferenciální rovnice

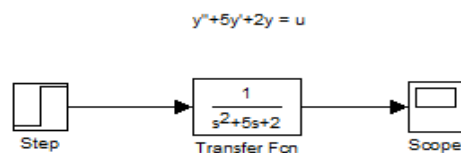
```
>> [x,y] = ode45('rovnice',[0 10],Y0);
>> plot(x',y(:,1)');
```

Obrázek 8: Volání funkce – rovnice(x,Y)

Blokově orientované programy

Velkou výhodou tohoto řešení je, že v těchto programech máme takzvané předdefinované bloky, které jsou jistě pro uživatele příjemnější a názornější, než zadávat model ve formě rovnic. V těchto blocích máme možnost volby parametrů bloku, počáteční podmínky, mezní podmínky a celkové doby simulace apod. Mezi nejznámější blokově orientovaný program je nastavba programu Matlab - Simulink.

Jestliže máme k dispozici diferenciální rovnice systému, je potřeba řešit tyto rovnice pomocí Laplaceovy transformace a vyjádřit si jejich přenosy. Nevýhodou tohoto řešení je, že soustavy diferenciálních rovnic musí být lineární a při vyjádření pomocí přenosu musí být počáteční podmínky nulové. Následně v předdefinovaných knihovnách je nutné vyhledat příslušné bloky.



Obrázek 9: Metoda určení přenosu z diferenciální

2.2.3 Princip simulace v *Simulátoru chování regulačního obvodu*

Jako první co je nutné udělat, je zadání koeficientu čitatele a jmenovatele. Tyto koeficienty jsou programově přiřazeny do spojitého přenosu podle rovnice 6. Vytvoření spojitého přenosu se provádí ve výpočetním systému Matlab pomocí příkazu `tf([koeficienty čitatele], [koeficienty jmenovatele])`.

```
>> B = [1];
>> A = [5 1];
>> Hs = tf(B,A)

Transfer function:
      1
-----
    5 s + 1
```

Obrázek 10: Vytvoření spojitého přenosu v

Tento spojitý přenos je následně převeden do diskrétní podoby pomocí příkazu `c2d(Hs, Vzorkovací perioda)`. Kde Hs je přenos ve spojitě formě a vzorkovací perioda je dána jako podíl celkové doby simulace ku deseti tisícům.

```

>> Hd = c2d(Hs,Vzorkovaci_perioda)

Transfer function:
0.001998
-----
z - 0.998

Sampling time: 0.01

```

Obrázek 11: Diskretizace spojitého systému

Pro výpočet aktuální hodnoty regulované veličiny, je tento diskrétní přenos převeden na diferenční rovnici se záporným posunutím, tak aby na levé straně rovnice byla hodnota s nejnižším posunutím.

$$y(k) = b_0 u(k) + b_0 u(k-1) + \dots + b_0 u(k-n) - [a_1 y(k-1) + \dots + a_n y(k-n)] + d_2(k) \quad (37)$$

Kde je:

- $y(k)$ - aktuální hodnota regulované veličiny
- $u(k)$ - aktuální hodnota akčního zásahu
- $y(k-n)$ - minulé hodnoty regulované veličiny
- $u(k-n)$ - minulé hodnoty akčního zásahu
- $d_2(k)$ - poruchová veličina za soustavou
- b - koeficienty čitatele
- a - koeficienty jmenovatele

V aplikaci je zadání koeficientů regulátoru v následující podobě r_0, r_{-1}, r_1 které jsou v programu přepočítány na koeficienty r_0, T_I, T_D a pomocí nich je proveden výpočet koeficientů PSD regulátoru q_0, q_1, q_2 podle rovnice 35.

$$r_0 = r_0, T_I = \frac{r_0}{r_{-1}}, T_D = \frac{r_1}{r_0} \quad (38)$$

Pro výpočet aktuálního akčního zásahu regulátoru postupuji obdobně, jako při výpočtu aktuální hodnoty regulované veličiny tak, že provedu diskretizaci PID regulátoru a vyjádřím si jeho diferenční rovnici.

$$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + u(k-1) + d_1(k) \quad (39)$$

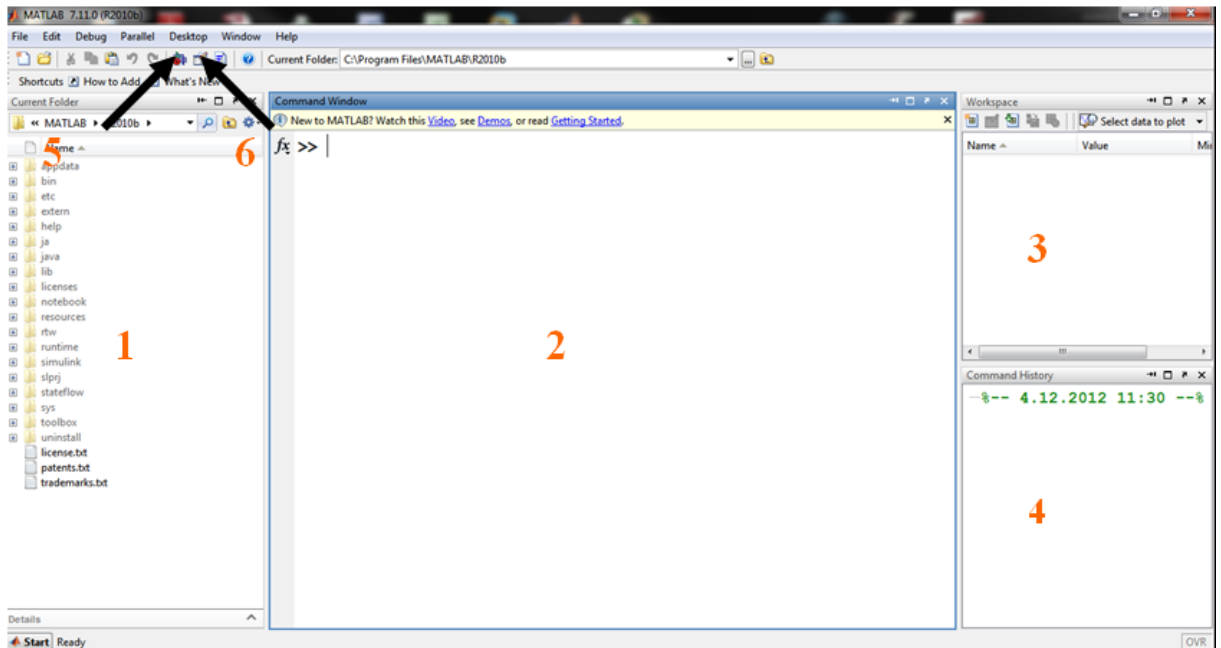
Kde je:

$u(k)$	- aktuální hodnota akčního zásahu
$u(k-1)$	- minulá hodnota akčního zásahu
$e(k)$	- aktuální hodnota regulační odchylky
$e(k-1), e(k-2)$	- minulé hodnoty regulační odchylky
$d_1(k)$	- poruchová veličina před soustavou
q_0, q_1, q_2	- koeficienty PSD regulátoru

Tyto dvě rovnice jsou pak vloženy do konečného cyklu *for* o *n* opakování, kde je postupně počítána regulovaná veličina, regulační odchylka a akční zásah v daných časových okamžicích.

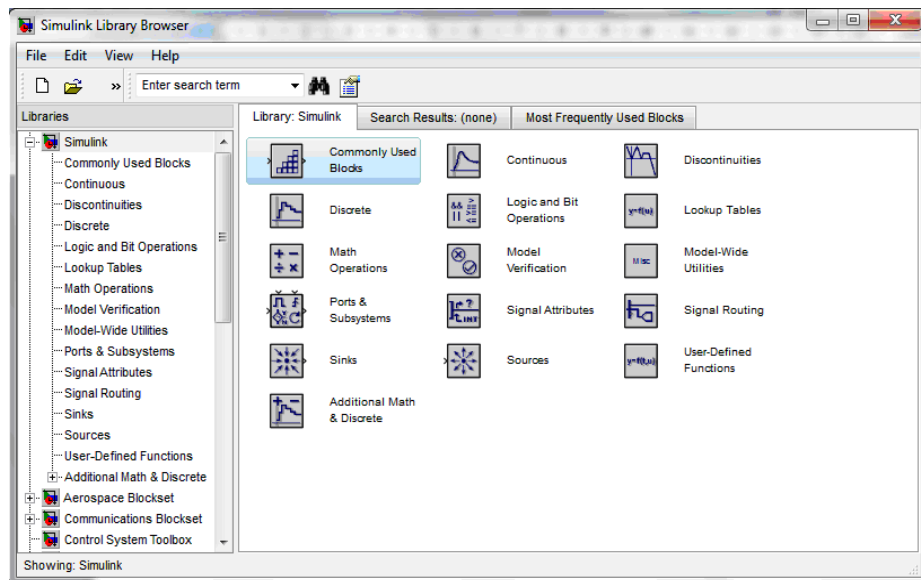
2.3 Základní orientace v Matlabu

Matlab je vyvíjen společností MathWorks. Je dostupný, jak pro operační systémy Windows [3, str. 12-15], tak i pro systémy Linux. Matlab umožňuje různé matematické operace, jako je práce s maticemi, tvorba matematických funkcí, vykreslování 2-D a 3-D grafů a mnoho dalšího. Dále tento program nabízí v tvorbu uživatelských aplikací a možnost počítačové simulace.



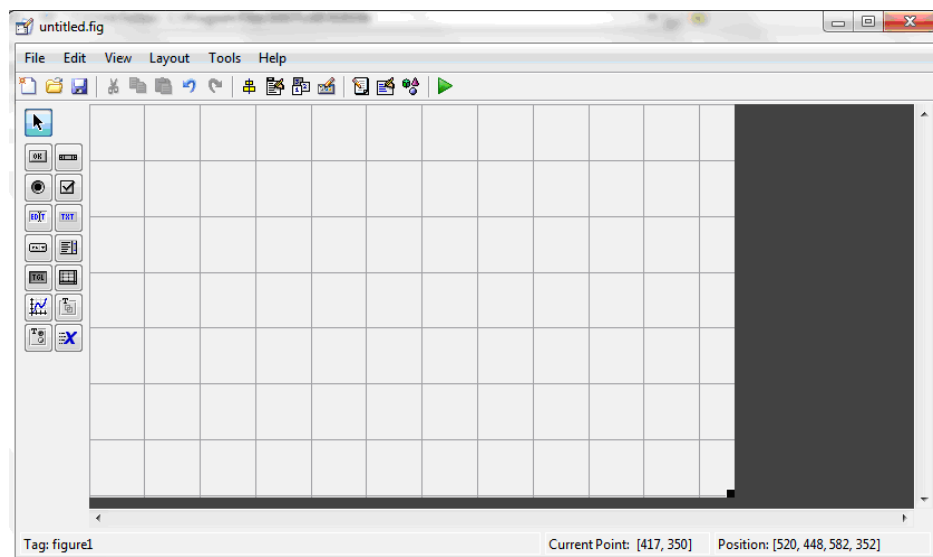
Obrázek 12: Úvodní okno Matlabu

Při otevření programu se zobrazí úvodní okno. Toto okno je možno rozdělit do šesti nepoužívanějších částí. Mezi první část patří *Current Folder* v tomto okně je možno najít všechny doposud vytvořené matematické funkce, skripty popřípadě uživatelské programy *GUI*, které byly vytvořeny a následně je z tohoto adresáře možné opětovně spustit. Druhá část je uživatelem asi nepoužívanější *Command Window* neboli příkazový řádek do tohoto okna zadává uživatel příkazy, které se mají vykonat. Třetí a čtvrtá část souvisí s *Command Windows*. Jestliže uživatel vytvoří proměnou nebo provede matematickou operaci, všechny proměnné a výsledky se uchovávají, respektive ukládají do *Workspace*. *Command History* obsahuje všechny použité příkazy a také umožňuje opětovně spustit dříve zadaný příkaz. Jestliže bude „kliknuto“ levým tlačítkem myši na ikonu pod číslem pět nebo do příkazového řádku bude napsáno *sim*, otevře nastavbové prostředí *Simulink*. Toto prostředí se nejčastěji používá pro modelování dynamických systémů.



Obrázek 13: Nastavbové prostředí Simulink

Mezi další zajímavou funkcí patří tvorba uživatelských aplikací. Tato funkce se skrývá pod ikonou s číslem šest nebo do příkazového řádku napíšeme příkaz *guide*. Toto prostředí nám umožní vkládání grafických objektů, u kterých můžeme měnit jejich vlastnosti jako například volby šířky, výšky, pozice a mnoho dalšího.



Obrázek 14: Rozhraní pro tvorbu uživatelských aplikací

2.4 Princip vytváření GUI aplikací

2.4.1 Předpoklady k vytvoření GUI

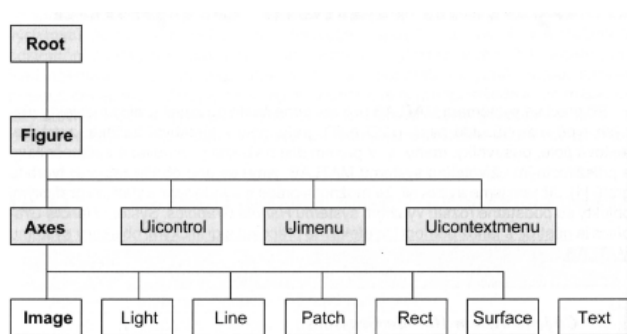
Základním předpokladem pro vytvoření GUI aplikace je, že si uživatel si chce usnadnit práci při vyhodnocování určitých dat, co se týče vykreslování dat do grafů a podobně.

Dále co je nutné si uvědomit, jak se daná aplikace bude chovat, jaké budou její hlavní funkce, jak bude přívětivá pro budoucího uživatele popřípadě její vizuální stránka a její snadné rozšíření o další funkce v případě potřeby.

2.4.2 Co je to systém Handle Graphics

Aby bylo možné v Matlabu vytvářet interaktivní aplikace, je tento výpočetní program doplněn o systém *Handle Graphics*. Tento systém umožňuje programátorovi velice efektivně vytvářet grafické objekty, popřípadě upravovat jejich vlastnosti.

Do tohoto [7, str. 71-73] systému patří veškeré příkazy pro grafické znázornění dat v 2-D a 3-D a grafiku obecně. Jako jsou objekty `figure`, `uicontrol`, `uicontextmenu` a `axes`. Hierarchie základních grafických objektů je znázorněna na obrázku 15.



Obrázek 15: Hierarchie systému Handle

2.4.3 Struktura M-File v GUI

Abych vůbec mohl přistoupit k návrhu, nebo programování jakékoli aplikace je dobré znát strukturu *M-File*. Tento *M-File* můžu rozšířit o uživatelské funkce, skripty a kód, který je vykonám v některých z níže uvedených funkcí.

V případě, že bude rozhodnuto změnit vlastnosti kteréhokoli objektu poslouží nám k tomu příkazy *set()* a *get()*.

M-File je automaticky generován Matlabem a struktura vypadá takto.

Funkce *OpeningFcn*

Tato funkce je vykonána před zviditelněním aplikace uživateli. Uživatel může v této části kódu zadat inicializační parametry jednotlivých proměnných.

Funkce *OutputFcn*

Tato funkce je volána ihned pro provedení funkce *OpeningFcn*.

Funkce *Callback*

Tato funkce je vykonána při jakékoli změně objektu, například při kliknutí na tlačítko či změně textu v editačním poli.

Funkce *CreateFcn*

Tato funkce je vykonána během vytváření objektu. Zde je možnost definovat, jak bude objekt vypadat ve vaší aplikaci i v případě volání funkce *Callback*.

Funkce *WindowButtonMotionFcn*

Je vykonána v případě pohybu kurzoru myši v grafickém objektu *figure*.

Funkce *WindowButtonDownFcn* a *WindowButtonUpFcn*

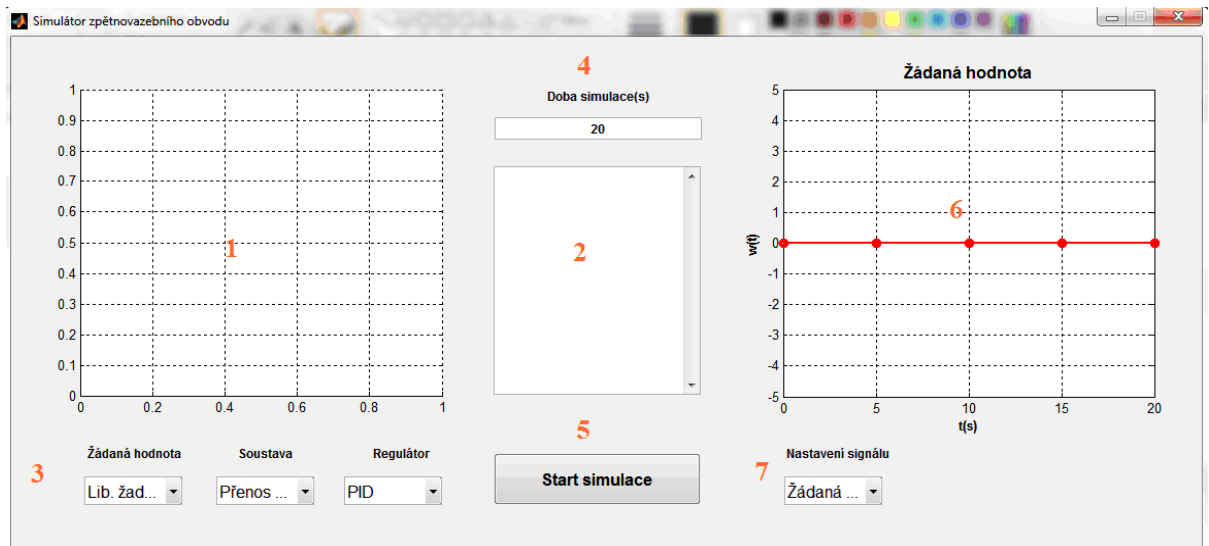
Tato funkce je vykonána, jestliže podržím respektive uvolním pravé tlačítko myši a kurzor myši se musí nacházet v objektu *figure*.

3 Praktická část

3.1 Popis aplikace

3.1.1 Hlavní okno aplikace

Jestliže spustím aplikaci *simulátor chování regulačního obvodu* z obrázku 16. Můžu tuto aplikaci rozdělit na sedm základních částí.

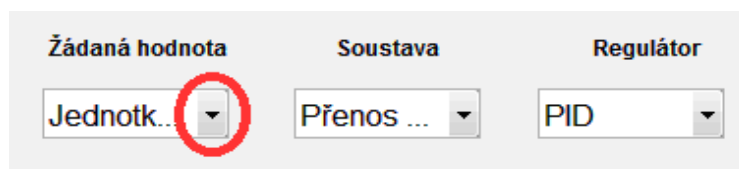


Obrázek 16: Vzhled celé aplikace

Pod číslem jedna se skrývá objekt *axes*, který vyobrazí odezvu systému při daném vstupním signálu. Číslo dva je objekt *edit*, který slouží k zobrazení, případně kontrole nastavených hodnot přenosu, koeficientů PID regulátoru nebo mezních hodnot dvoupolohového regulátoru. Ve třetí části jsou grafické objekty *popupmenu* asi nejdůležitější z celého programu, protože slouží k nastavení žádané hodnoty, typům přenosu, koeficientům PID regulátoru a nastavení dvoupolohového regulátoru. Ve čtvrté části je grafický objekt *edit*, kde je možnost volby, jak dlouho má simulace trvat v sekundách. Pátá část objekt *pushbutton*, kliknutím spustíme simulaci. V šesté části v grafickém objektu *axes*, můžeme nastavit či vymodelovat libovolný vstupní signál včetně poruch a šumu. A nakonec sedmá část slouží k přepínání mezi žádanou hodnotou, poruchami a šumem. Detailní nastavení všech grafických objektu popíšu podrobněji v dalších kapitolách.

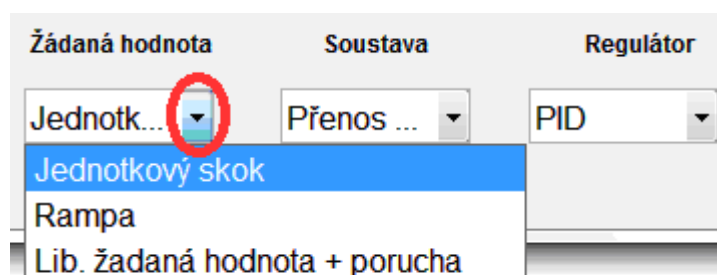
3.1.2 Nastavení žádané hodnoty

V prvé řadě se budu zabývat, jaké jsou možnosti nastavení u žádané hodnoty. Přesunu kurzor myši na *popupmenu* pod názvem žádaná hodnota, vyobrazeno na obrázku 17.



Obrázek 17: Žádaná hodnota

V případě že rozevřu tuto nabídku, mám na výběr ze tří možností volby signálu a to jednotkový skok, rampu a libovolnou žádanou hodnotu včetně poruch a šumu.

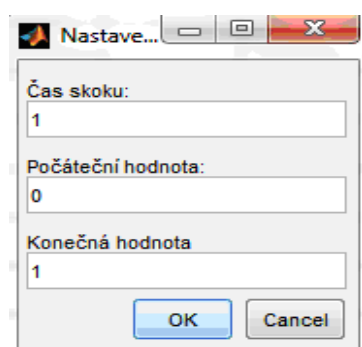


Obrázek 18: Žádaná hodnota - možnosti volby

Jestliže si vyberu jednu z možností, otevře se dialog, kde je nutné zadat potřebné parametry, aby došlo ke správnému vygenerování vstupního signálu pro daný experiment. U každého typu signálu se nastavení od sebe nepatrně liší.

Jednotkový skok

Při otevření dialogového okna je nutné zadat parametry jako je čas skoku, počáteční hodnotu a konečnou hodnotu, kdykoli je může uživatel dle své potřeby změnit podle aktuálních potřeb.

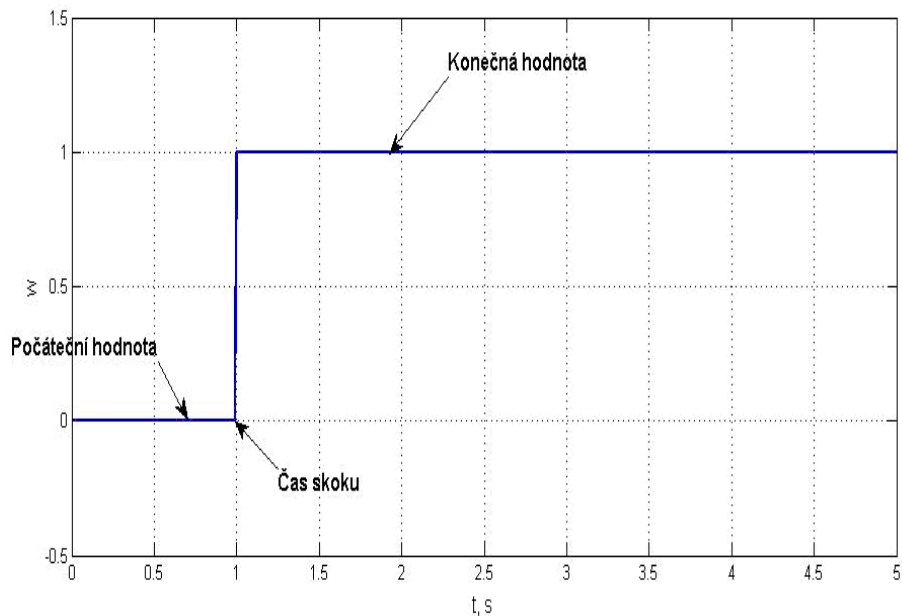


Obrázek 19: Dialogové okno - nastavení jednotkového skoku

Času skoku, uživatel nesmí zadat hodnoty záporné a vyšší jak je celková doba simulace, protože by nebyla splněna vzorkovací perioda.

Počáteční hodnota, je hodnota, na které se má udržovat, dokud nenastane čas, kdy se má signál změnit z počáteční hodnoty na konečnou hodnotu.

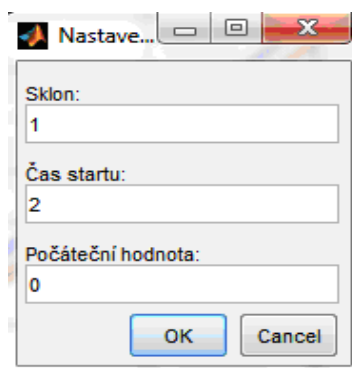
Konečná hodnota, je hodnota, kdy se signál udržuje po zbytek simulace, jestliže již proběhl okamžik změny z počáteční hodnoty na konečnou.



Obrázek 20: Vygenerovaný signál - Jednotkový skok

Rampa

Jako další možnost volby žádané hodnoty je rampa. Při „kliknutí“ na tuto možnost se mě opět otevře dialogové okno s nutností zadat několik parametru. Těmito parametry jsou sklon, čas startu a počáteční hodnota.

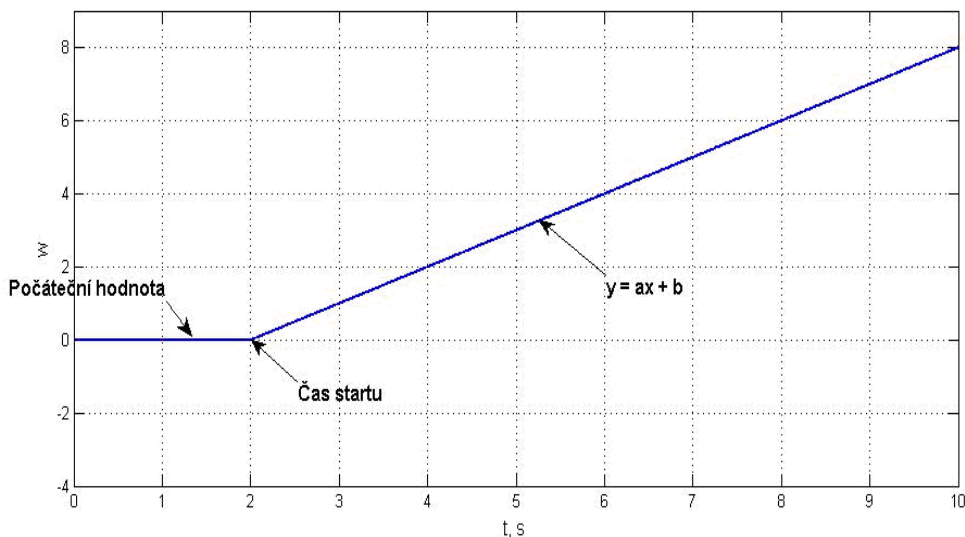


Obrázek 21: Dialogové okno - nastavení rampy

Sklon je hodnota pod jakým úhlem má být signál generován jestliže už nastal okamžik čas startu.

Čas startu je hodnota času, kdy se má začít generovat signál s určitým sklonem. Uživatel tuto hodnotu taktéž nesmí zadat zápornou nebo větší než je celková doba simulace.

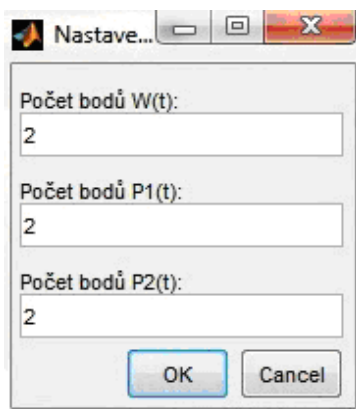
Počáteční hodnota je hodnota, na které se má udržovat, dokud nenastane čas, kdy se má signál začít měnit.



Obrázek 22: Vygenerovaný signál – Rampa

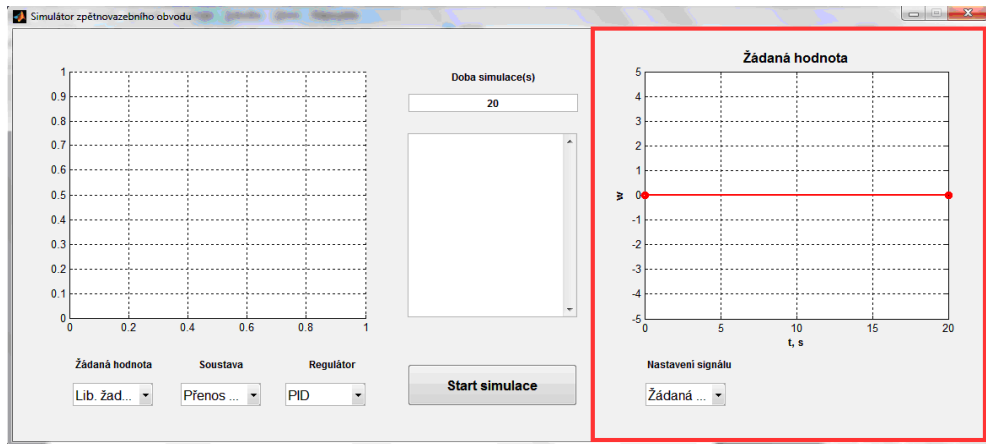
Libovolná žadaná hodnota, porucha

Jako poslední co nám tento objekt nabízí je vlastní vymodelování žadané hodnoty, poruchy na vstupu a výstupu soustavy. Jako v předchozích dvou kapitolách kliknutím na tuto možnost se otevře dialogové okno s nutností zadat počty bodů u žadané hodnoty, poruchy na vstupu a výstupu soustavy.



Obrázek 23: Dialogové okno - Libovolná žadaná hodnota a porucha

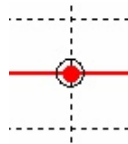
Uživatel si musí dávat pozor, protože pro vytvoření signálu musí zadat minimálně dva body. Jestliže uživatel stiskne tlačítko „OK“ aplikace se rozšíří o další grafický objekt *axes*, který dává možnost si signály vymodelovat. Rozšíření aplikace o možnost modelování signálu je na obrázku 24, vyznačeno červeným obdélníkem.



Obrázek 24: Rozšíření aplikace - modelování signálu

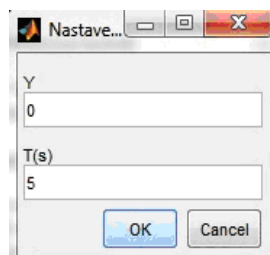
Modelování signálu

Aby bylo možné s jednotlivými body pracovat, musí se kurzor myši nacházet v objektu a velice blízko bodu. Jestliže je kurzor myši dostatečně blízko k bodu, kurzor se změní na kolečko a teprve tehdy lze s bodem manipulovat. Jednotlivé body je možné posouvat jak po vodorovné ose, tak i po svislé ose.



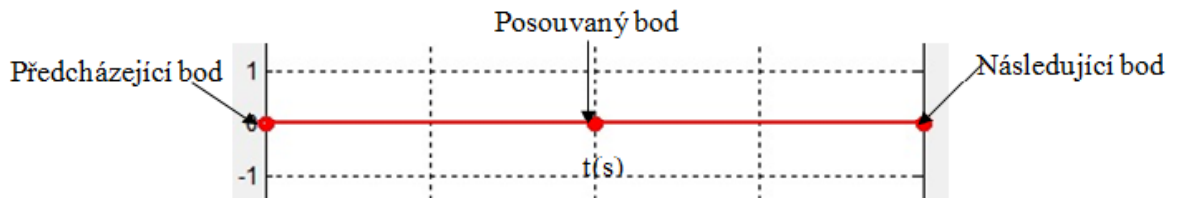
Obrázek 25: Detekce bodu - změna kurzoru

Pokud je nutné provést posuv pouze po svislé ose, stačí podržet levé tlačítko myši. V případě, že je nutné provést posuv po vodorovné a svislé ose, je nutno na bod „kliknout“ pravým tlačítkem myši. Po této operaci se otevře dialogové okno, kde je nutné zadat čas a hodnotu bodu v daném čase.



Obrázek 26: Nastavení bodu - pravé tlačítko myši

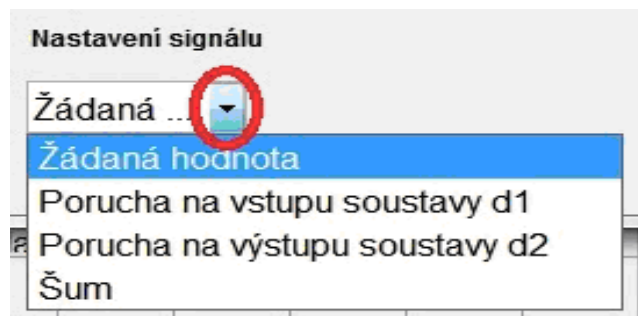
Jestliže jsou nastaveny pouze dva body na signál. Lze s těmito body pracovat pouze ve svislém směru. Pokud je bodů více například tři, s prostředním bodem lze pohybovat, jak ve vodorovném, tak i ve svislém směru. Ale pouze v rozmezí, že bod musí mít hodnotu času větší než předcházející a zároveň hodnotu času menší než bod následující.



Obrázek 27: Pohyb bodu

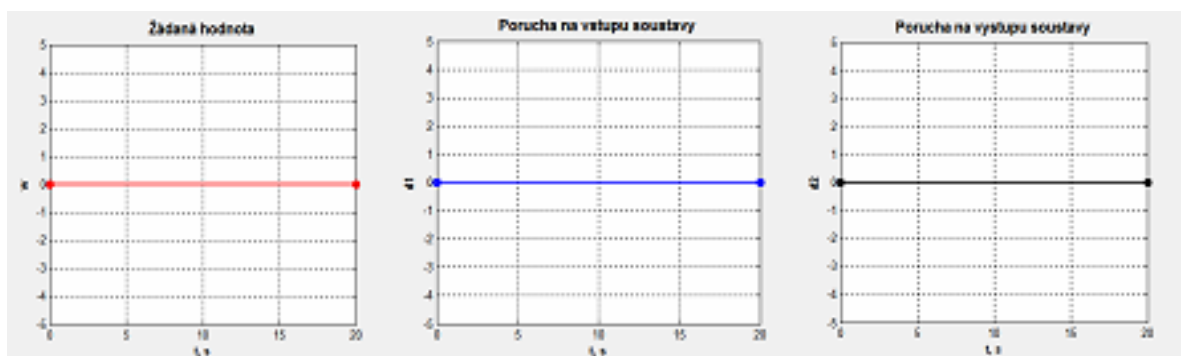
Aby bylo možné pracovat ve regulačním obvodu nejen s žádanou hodnotou, je v této aplikaci možnost vymodelovat si poruchy na vstupu soustavy a na výstupu soustavy, pomocí dalšího objektu *popoupmenu* pod názvem nastavení signálu. Je zobrazeno na obrázku 28.

Aby si uživatel nepletl jednotlivé signály mezi sebou, je provedené barevné rozlišení



Obrázek 28: Nastavení signálů - možnosti volby

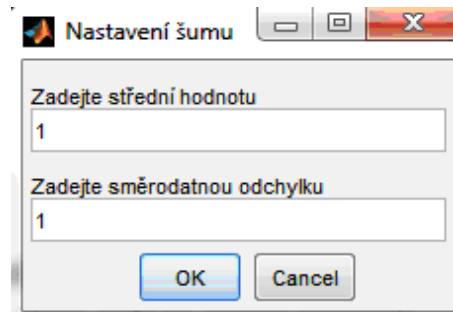
signálů. Žádané hodnotě je přiřazená červená barva, poruše na vstupu soustavy je přiřazena modrá barva a poruše na výstupu soustavy je přiřazena černá barva.



Obrázek 29: Barevné odlišení signálů

Nastavení šumu

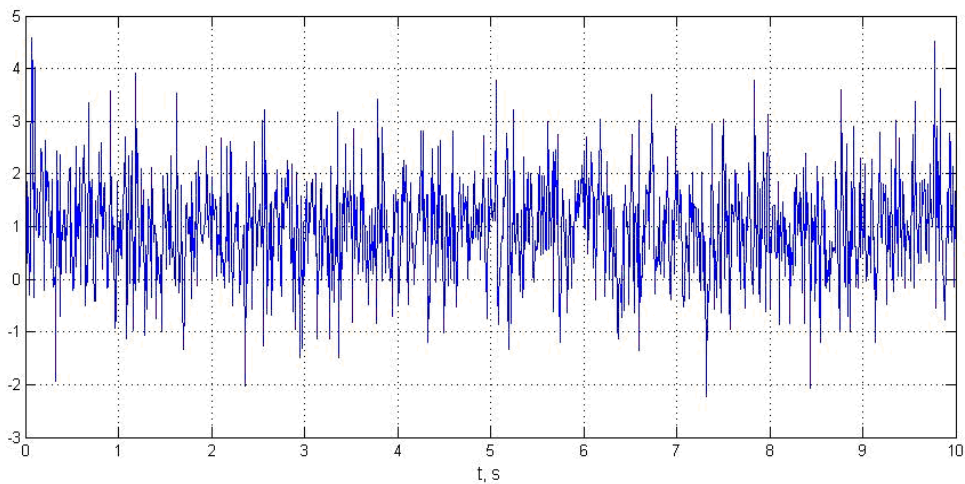
Jestliže se uživatel rozhodne, aby v regulačním obvodu působil i šum. Stačí „kliknout“ na možnost šum. V zápětí se otevře dialogové okno, kde je nutné nastavit dva parametry. Těmito parametry jsou směrodatná odchylka a střední hodnota.



Obrázek 30: Dialogové okno – nastavení šumu

Ke generování šumu byla zvolena funkce *randn* s normálním rozložením. Kde k – střední hodnota, g – směrodatná odchylka a x – počet bodů.

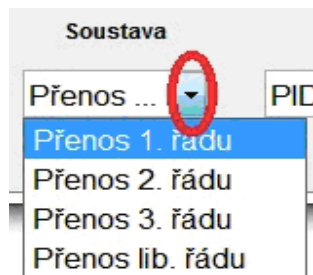
$$y = k + g \cdot \text{randn}(1, x) \quad (40)$$



Obrázek 31: Příklad vygenerovaného šumu

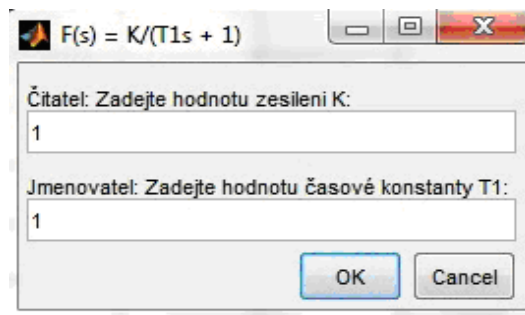
3.1.3 Nastavení soustavy

Tato podkapitola je věnována, jaké jsou možnosti nastavení přenosu soustavy. Přesunu-li kurzor myši na *popupmenu* pod názvem soustava a tuto nabídku rozevřu, tak aplikace nabízí čtyři možnosti výběru přenosu soustavy 1. řadu, 2. řadu, 3. řadu a zadání obecného přenosu soustavy.



Obrázek 32: Soustava – možnosti volby

Jestliže uživatel zvolí jednu ze tří prvních, musí zadat hodnotu zesílení K a potřebný počet časových konstant T podle řádu přenosu.



Obrázek 33: Dialogové okno – přenos 1. řádu

Přenos pro 1. řád, 2. řád a 3. řád.

1. řád

$$F(s) = \frac{K}{(T_1 s + 1)} \quad (41)$$

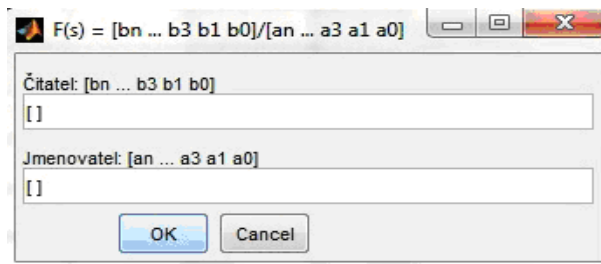
2. řád

$$F(s) = \frac{K}{(T_1 s + 1)(T_2 s + 1)} \quad (42)$$

3. řád

$$F(s) = \frac{K}{(T_1 s + 1)(T_2 s + 1)(T_3 s + 1)} \quad (43)$$

U přenosu libovolného řádu se způsob napatrně liší od předchozích tří možností tím, že uživatel musí zadat přímo koeficienty čitatele a jmenovatele. Při tom musí mít na paměti, aby byla splněna podmínka realizovatelnosti systému a to, že počet koeficientu v čitateli nesmí být větší nebo rovno než je počet koeficientu ve jmenovateli.



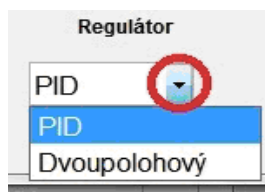
Obrázek 34: Dialogové okno – přenos libovolného řádu

Přenosová funkce pro n-řád.

$$F(s) = \frac{b_n s^{(n)} + b_{n-1} s^{(n-1)} + \dots + b_1 s + b_0}{a_n s^{(n)} + a_{n-1} s^{(n-1)} + \dots + a_1 s + a_0} \quad (44)$$

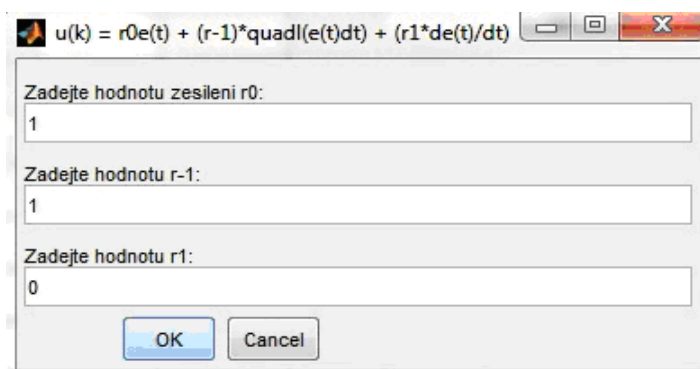
3.1.4 Nastavení PID a dvoupolohového regulátoru

Tato aplikace nabízí možnost volby mezi dvěma typy regulátoru a to PID a dvoupolohovým regulátorem.



Obrázek 35: Regulátor – možnosti volby

PID regulátor



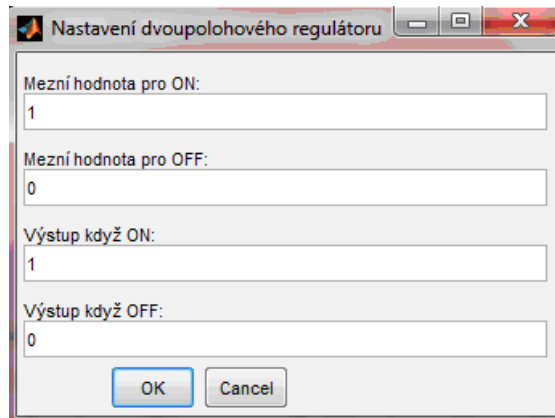
Obrázek 36: Dialogové okno – nastavení PID regulátoru

„Kliknutím“ na možnost PID regulátor se otevře dialogové okno s nutností zadat koeficienty regulátoru r_0 , r_{-1} , r_1 .

Tato aplikace nabízí možnost kombinace jednotlivých složek a to P, I, PD, PI a PID. Jestliže bude rozhodnuto jednu ze složek nepožít, musíme do příslušné kolonky přiřadit číslo nula.

Dvoupolohový regulátor

„Kliknutím“ na možnost dvoupolohový regulátor se taktéž otevře dialogové okno s nutností zadat mezní hodnoty regulátoru. A to mezní hodnoty pro vypnutí a zapnutí regulátoru a hodnoty akční veličiny při „vypnutém“ respektive „zapnutém“ regulátoru.



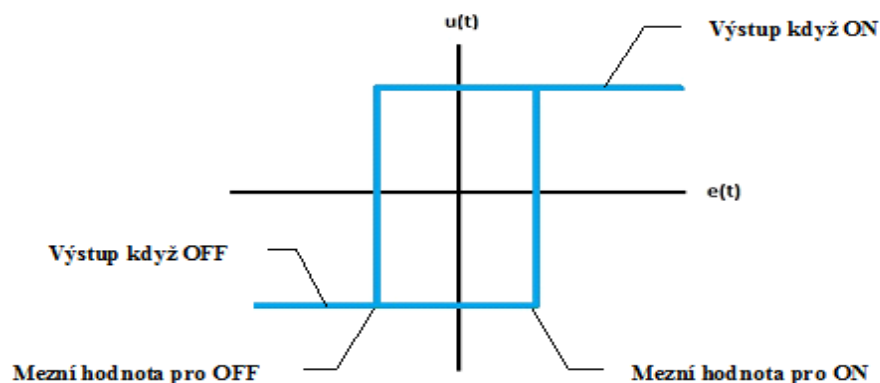
Obrázek 37: Dialogové okno – nastavení Dvoupolohového regulátoru

Mezní hodnota pro ON je hodnota regulační odchylky. Jestliže tato hodnota bude větší nebo rovna, nastav hodnotu akčního zásahu na Výstup když ON.

Mezní hodnota pro OFF je hodnota regulační odchylky. Jestliže tato hodnota bude menší nebo rovna, nastav hodnotu akčního zásahu na Výstup když OFF.

Výstup když ON, je velikost akčního zásahu, pokud je hodnota regulační odchylky větší nebo rovno Mezní hodnota pro ON.

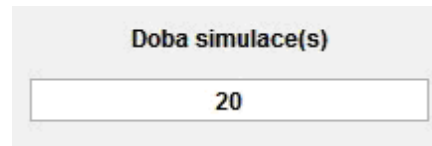
Výstup když OFF, je velikost akčního zásahu, pokud je hodnota regulační odchylky menší nebo rovno Mezní hodnota pro ON.



Obrázek 38: Grafické vyobrazení parametru v přepínací charakteristice u nastavení dvoupolohového regulátoru

3.1.5 Nastavení doby simulace

Jako poslední nedílnou částí, co musí uživatel nastavit je celková doba simulace. Levým tlačítkem myši „kliknu“ do editačního pole doby simulace a v tuto chvíli lze tuto hodnotu měnit. Výchozí hodnota doby simulace je dvacet sekund.

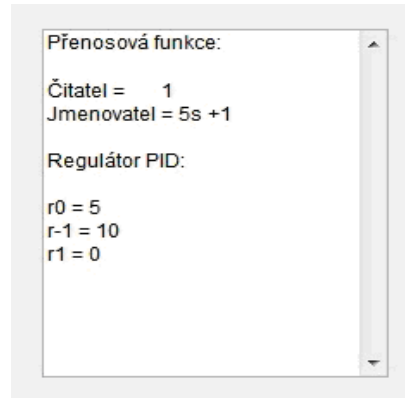


Obrázek 39: Editační pole – doba simulace

V případě, že uživatel udělá překlep, například napíše místo číslice písmeno nebo zápornou hodnotu, vrátí se doba simulace do svoji výchozí hodnoty dvaceti sekund.

3.1.6 Zobrazení nastavených parametru

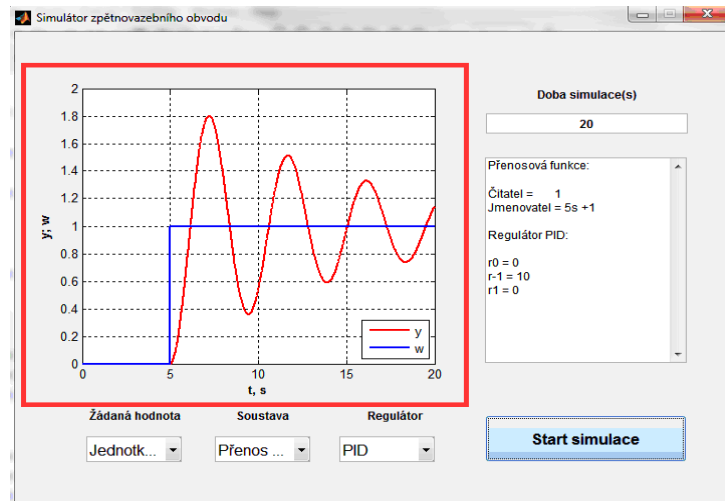
Aby uživatel měl možnost zpětné odezvy, respektive si mohl překontrolovat, jaké nastavil hodnoty přenosu, PID nebo dvoupolohového regulátoru a šumu. Je přidáno do aplikace další editační pole, které nelze měnit, slouží pouze k zobrazení nastavených hodnot.



Obrázek 40: Zobrazení nastavených hodnot

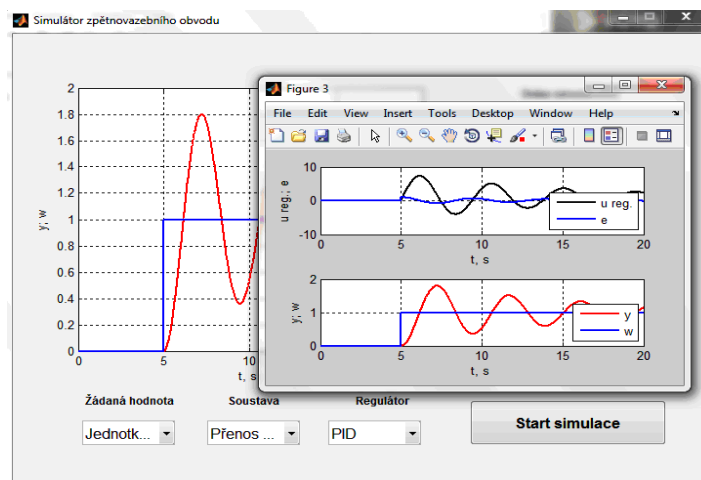
3.1.7 Zobrazení odezvy systému

Pro vyobrazení celkové odezvy zpětnovazebního obvodu slouží grafický objekt *axes*. V základním režimu vyobrazí pouze regulovanou veličinu a žádanou veličinu v závislosti čase.



Obrázek 41: Zobrazení odezvy systému – základní režim

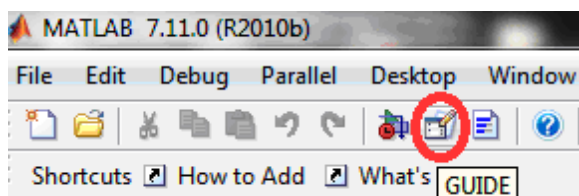
Jestliže na tento objekt bude „kliknuto“ levým tlačítkem myši zobrazí se v novém okně *figure* dvě grafické závislosti. Toto okno umožní uživateli grafické závislosti uložit. V prvním grafu akční veličina a regulační odchylka v závislosti na čase a druhém grafu regulovaná a žádaná veličina v závislosti na čase.



Obrázek 42: Zobrazení odezvy systému – rozšířený režim

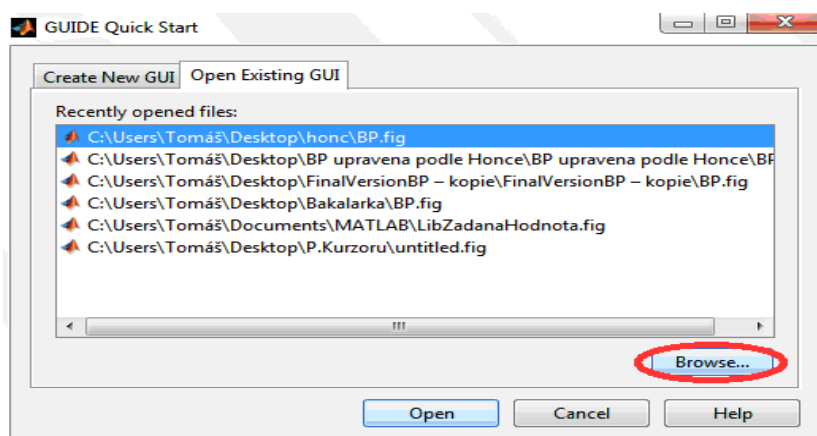
3.1.8 Spuštění aplikace v Matlabu

Jako prvním úkolem co je nutné udělat, je spuštění programu Matlab a následně vyčkám, až se program zcela načte. Existuje mnoho způsobů [3, str. 116-123], jak spustit námi vytvořenou aplikaci. Mně poslouží jednoduchý průvodce, který nabízí program Matlab. Pohybem kurzoru myši se přemístím nad ikonu vyobrazenou na obrázku 43. Jestliže nebudu nad touto ikonou pohybovat kurzorem myši, zobrazí se nám kontextová nápověda *GUIDE*.



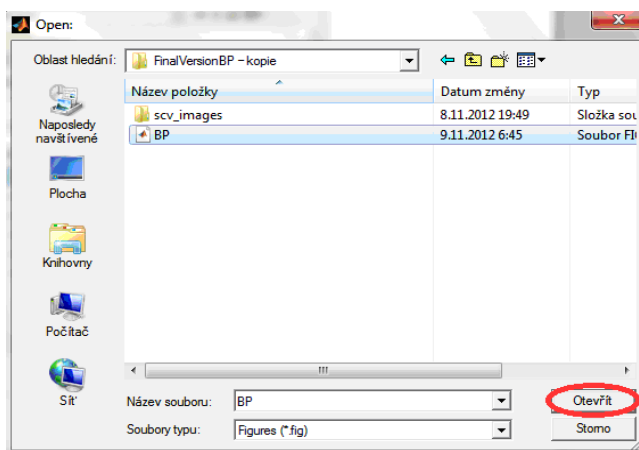
Obrázek 43: Spuštění GUIDE Quick Start

Následně „klepnu“ levým tlačítkem myši na danou ikonu a spustí se mě okno *GUIDE Quick Start*. Jak je viditelné z obrázku 44, v mém případě už mi tento průvodce nabízí nějaké možnosti výběru, dříve vytvořených či spuštěných aplikací. Jestliže se nezobrazí aplikace, kterou právě požaduji, budu si ji muset vyhledat a to přemístěním kurzoru myši na tlačítko *Browse*.



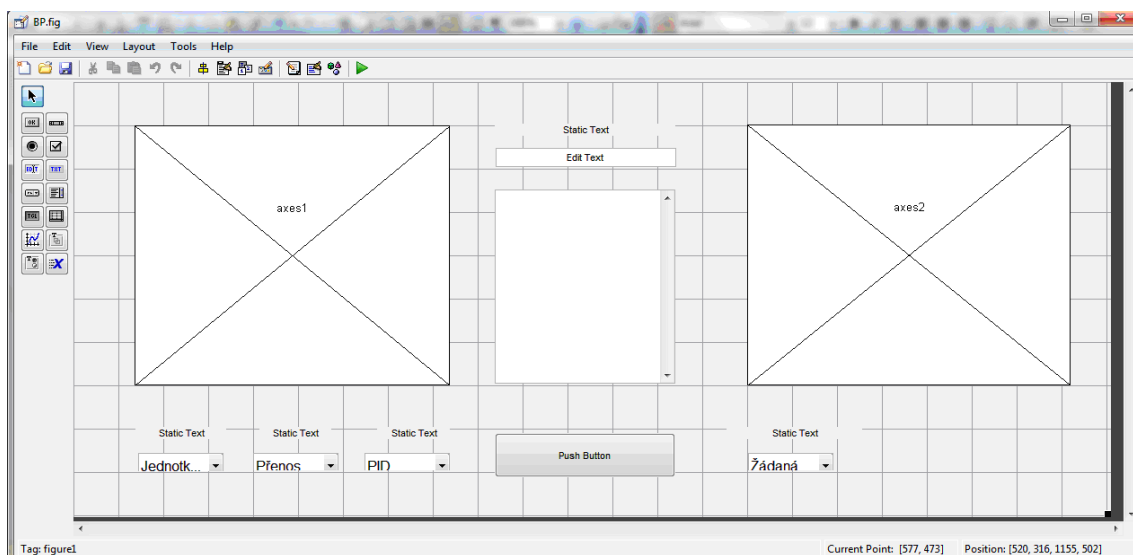
Obrázek 44: Vyhledání aplikace

Po „klepnutí“ na tlačítko *Browse* se otevře dialogové okno na obrázku 45, kde je možné aplikaci vyhledat na příslušném paměťovém mediu. Všechny uživatelské aplikace, které jsou v Matlabu vytvořeny mají příponu *.fig*. Po vyhledání zvoleného souboru otevřu aplikaci „klepnutím“ na tlačítko *Otevřít*.



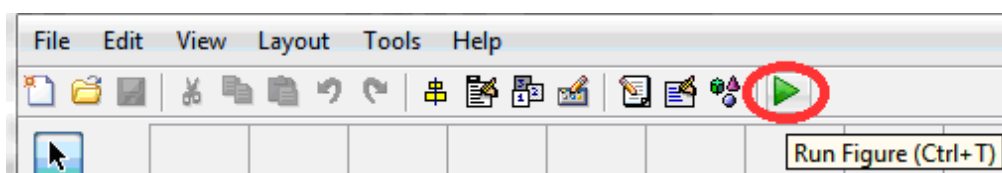
Obrázek 45: Vyhledání aplikace

Po otevření souboru se na ploše obrazovky zobrazí nastavbové prostředí *Grafical User Interface* (GUI) s příslušným rozmístěním všech ovládacích prvků na plátně z obrázku 46. Samozřejmost je, že rozmístění ovládacích prvků může uživatel přizpůsobit podle aktuálních potřeb nebo ponechat původní rozmístění.



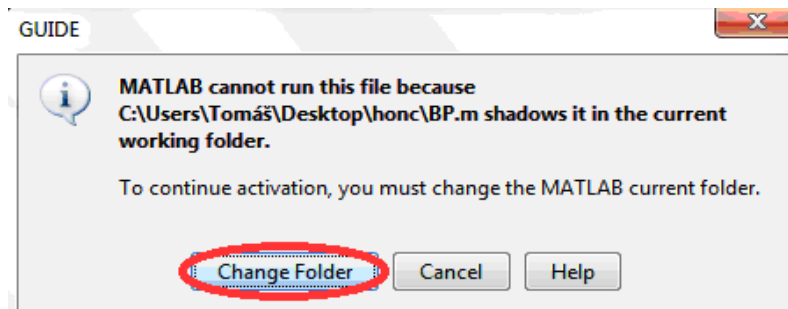
Obrázek 46: Rozmístění ovládacích prvků v GUI

Pro spuštění uživatelské aplikace existuje dvojitý způsob. První způsob je, že použiji klávesovou zkratku *CTRL + T* nebo „kliknutím“ na ikonku vyobrazenou na obrázku 47.

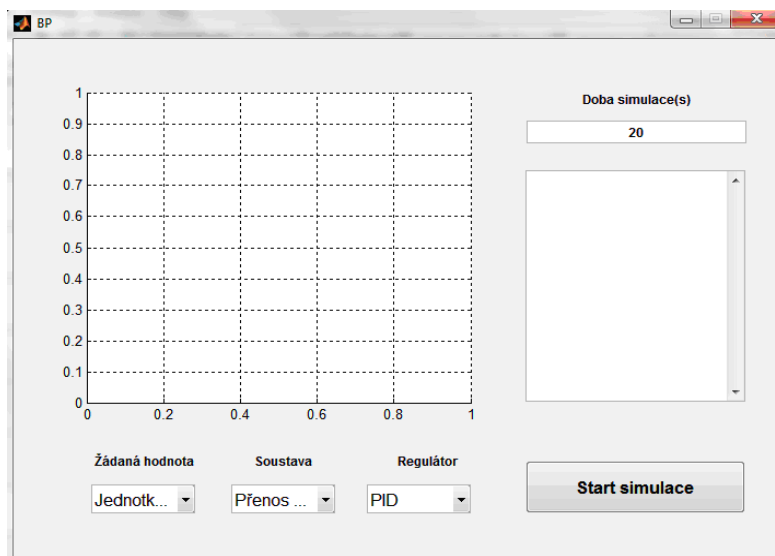


Obrázek 47: Dvojitý způsob spuštění aplikace

Při prvním spuštění aplikace se může stát, že se nám zobrazí dialogové okno na obrázku 48. Je to způsobeno tím, že se daná aplikace nachází v odlišném místě než je nastavený *Current Folder*. Abych zajistil správný chod aplikace, musím tedy „kliknout“ na tlačítko *Change Folder*, a tím změnil původní *Current Folder* na aktuální, tam kde se nacházejí aplikační soubory spouštěné aplikace se všemi funkcemi a skripty a vyvaroval se zbytečným chybám. A nakonec uvidím spuštěnou aplikaci na ploše na obrázku 49.



Obrázek 48: Dialogové okno změna Current Folder



Obrázek 49: Spuštěná aplikace

3.2 Příklady

3.2.1 Příklad 1 – Soustava 1. řádu s P regulátorem

Je zadána diferenciální rovnice systému $5y'+y=u$, tento systém regulujte P regulátorem s koeficientem $r_0 = 5$. Určete přenos soustavy, regulátoru. Určete a vykreslete odezvu na jednotkový skok žadané hodnoty pro veličiny $y(t)$, $u(t)$, $e(t)$ v délce trvání $t(s) = 20$. Správnost výpočtu ověřte pomocí simulačního programu.

Přenos soustavy je.

$$5y' + y = u$$

$$5sY(s) + Y(s) = U(s)$$

$$Y(s)(5s + 1) = U(s)$$

$$G_s = \frac{Y(s)}{U(s)} = \frac{1}{5s + 1}$$

Přenos regulátoru je.

$$G_R = \frac{U(s)}{E(s)} = r_0 + \frac{r_{-1}}{s} + r_1 s = 5$$

Výpočet přenosu a přechodové funkce pro regulovanou veličinu z žádané hodnoty.

Přenos regulované veličiny z žádané hodnoty je.

$$G_{YW} = \frac{Y(s)}{W(s)} = \frac{G_R G_s}{1 + G_R G_s} = \frac{5 \frac{1}{5s+1}}{1 + 5 \frac{1}{5s+1}} = \left(\frac{5}{5s+1} \right) \left(\frac{5s+1}{5s+6} \right) = \frac{5}{\underline{\underline{5s+6}}}$$

$$Y(s) = G_{YW} W(s) = \left(\frac{5}{5s+6} \right) \left(\frac{1}{s} \right) = \frac{5}{s(5s+6)} = \frac{5}{5s \left(s + \frac{6}{5} \right)} = \frac{1}{\underline{\underline{s \left(s + \frac{6}{5} \right)}}}$$

Pomocí slovníku zpětné Laplaceovy transformace vyhledám vhodný vzorec a určím originál pro přechodovou funkci regulované veličiny $y(t) = h(t)$.

$$y(t) = L^{-1} \left\{ \frac{1}{s \left(s + \frac{6}{5} \right)} \right\} = \underline{\underline{\frac{5}{6} \left(1 - e^{-\frac{6}{5}t} \right)}}$$

Výpočet přenosu a přechodové funkce pro akční veličinu z žádané hodnoty. Přenos akční veličiny z žádané hodnoty je.

$$G_{UW} = \frac{U(s)}{W(s)} = \frac{G_R}{1 + G_R G_s} = \frac{5}{1 + 5 \frac{1}{5s+1}} = \left(\frac{5}{5s+1} \right) \left(\frac{5s+1}{5s+6} \right) = \frac{25s+5}{\underline{\underline{5s+6}}}$$

$$U(s) = G_{UW} W(s) = \left(\frac{25s+5}{5s+6} \right) \left(\frac{1}{s} \right)$$

Výše odvozený vztah rozložím na součet parciálních zlomků a určím neznámé A, B .

$$\left(\frac{25s+5}{5s+6}\right)\left(\frac{1}{s}\right) = \frac{A}{s} + \frac{B}{5s+6}$$

$$25s+5 = A(5s+6) + Bs$$

$$25s+5 = 5sA+6A+Bs$$

$$25s+5 = s(5A+B)+6A$$

$$s^0: 6A = 5 \Rightarrow A = \underline{\underline{\frac{5}{6}}}$$

$$s^1: 5A+B = 25 \Rightarrow B = 25 - \frac{25}{6} = \frac{150-25}{6} = \underline{\underline{\frac{125}{6}}}$$

$$U_{(s)} = G_{\text{UW}} W_{(s)} = \left(\frac{25s+5}{5s+6}\right)\left(\frac{1}{s}\right) = \left(\frac{5}{6}\right)\frac{1}{s} + \left(\frac{125}{6}\right)\frac{1}{5s+6} = \left(\frac{5}{6}\right)\frac{1}{s} + \left(\frac{125}{30}\right)\frac{1}{s+\frac{6}{5}}$$

Pomocí slovníku zpětné Laplaceovy transformace je nutno vyhledat vhodný vzorec a určit originál pro přechodovou funkci akční veličiny $u(t) = h(t)$.

$$u(t) = L^{-1}\left\{\left(\frac{5}{6}\right)\frac{1}{s} + \left(\frac{125}{30}\right)\frac{1}{s+\frac{6}{5}}\right\} = \underline{\underline{\frac{5}{6} + \frac{125}{30}e^{-\frac{5}{6}t}}}$$

Vypočet přenosu a přechodové funkce pro regulační odchylku z žádané hodnoty. Přenos regulační odchylky z žádné hodnoty je.

$$G_{\text{EW}} = \frac{E_{(s)}}{W_{(s)}} = \frac{1}{1+G_{\text{R}}G_{\text{S}}} = \frac{1}{1+5\frac{1}{5s+1}} = \underline{\underline{\frac{5s+1}{5s+6}}}$$

$$E_{(s)} = G_{\text{EW}} W_{(s)} = \left(\frac{5s+1}{5s+6}\right)\left(\frac{1}{s}\right)$$

Výše odvozený vztah rozložím na součet parciálních zlomků a určím neznámé A, B .

$$\left(\frac{5s+1}{5s+6}\right)\left(\frac{1}{s}\right) = \frac{A}{s} + \frac{B}{5s+6}$$

$$5s+1 = A(5s+6) + Bs$$

$$5s+1 = 5sA + 6A + Bs$$

$$5s+1 = s(5A+B) + 6A$$

$$s^0: 6A = 1 \Rightarrow A = \frac{1}{6}$$

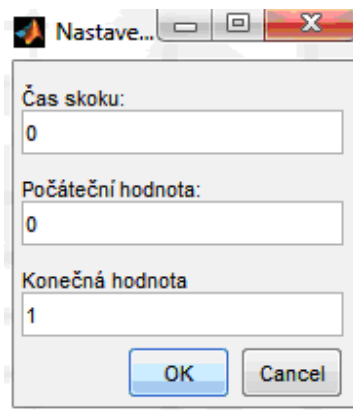
$$s^1: 5A + B = 5 \Rightarrow B = 5 - \frac{5}{6} = \frac{30-5}{6} = \frac{25}{6}$$

$$E_{(s)} = G_{EW} W_{(s)} = \left(\frac{5s+1}{5s+6}\right)\left(\frac{1}{s}\right) = \left(\frac{1}{6}\right)\frac{1}{s} + \left(\frac{25}{6}\right)\frac{1}{5s+6} = \left(\frac{1}{6}\right)\frac{1}{s} + \left(\frac{25}{30}\right)\frac{1}{s+\frac{6}{5}}$$

Pomocí slovníku zpětné Laplaceovy transformace je nutno vyhledat vhodný vzorec a určit originál pro přechodovou funkci regulační odchylky $e(t) = h(t)$.

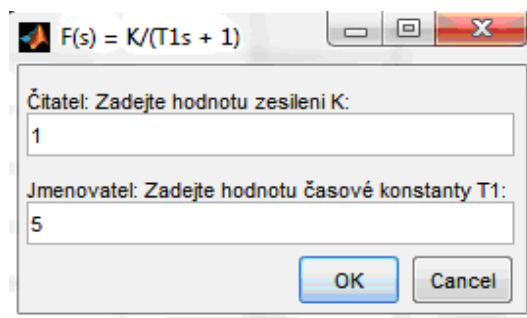
$$e(t) = L^{-1}\left\{\left(\frac{5}{6}\right)\frac{1}{s} + \left(\frac{125}{30}\right)\frac{1}{s+\frac{6}{5}}\right\} = \frac{1}{6} + \frac{25}{30}e^{-\frac{5}{6}t}$$

Ted'ko je nutné přistoupit k samotnému nastavení potřebných parametrů v simulátoru, abychom si mohli zkontrolovat vypočtené vztahy.



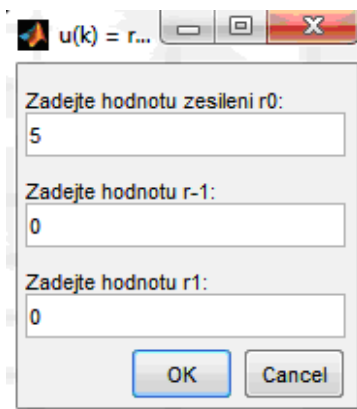
Obrázek 50: Nastavení jednotkového skoku - Příklad 1

Na obrázku 50 je vidět celkové nastavení žádané hodnoty pro jednotkový skok. Za parametr čas skoku, je nastavena hodnota na nulu, počáteční hodnota je taktéž rovna nule a konečnou hodnotu rovno jedné.



Obrázek 51: Nastavení přenosu soustavy – Příklad 1

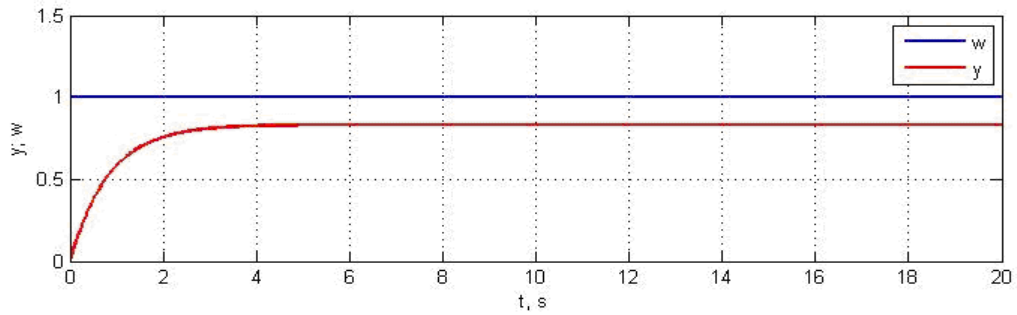
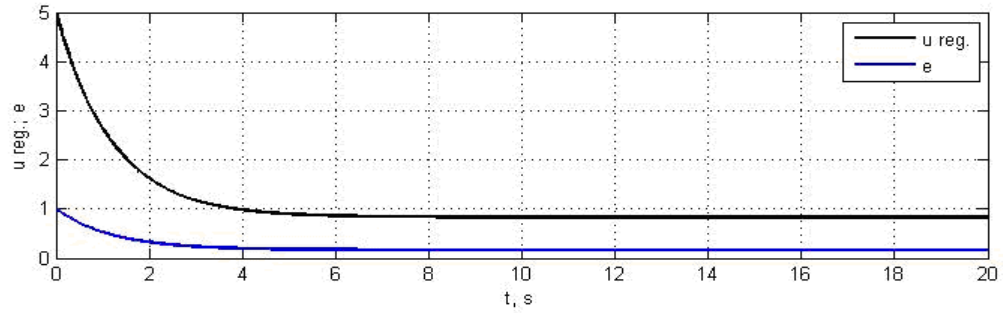
Další co je nutné nastavit, je přenos soustavy pro 1. řád na obrázku 51. Hodnotu zesílení nastavme na hodnotu jedna a časovou konstantu na hodnotu pět.



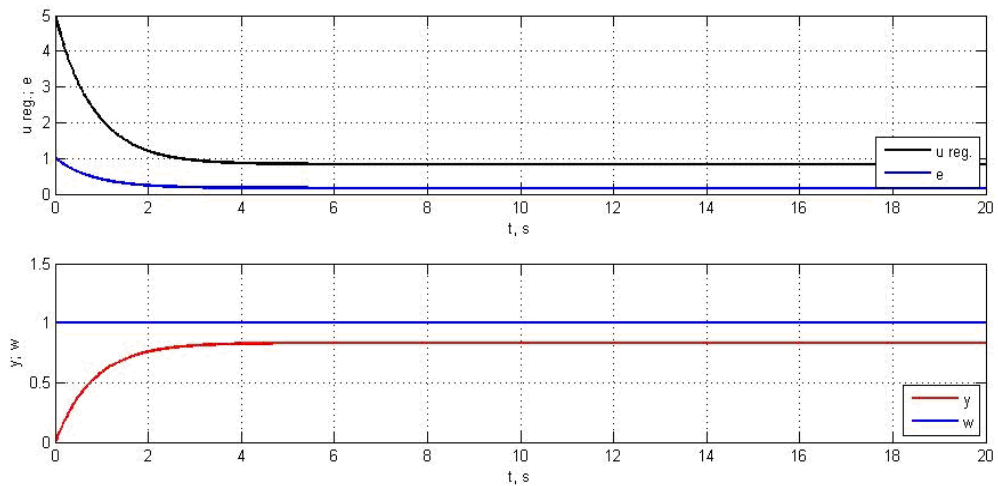
Obrázek 52: Nastavení PID regulátoru – Příklad 1

Jako poslední nedílnou součástí regulačního obvodu je nastavení PID regulátoru na obrázku 52. Nastavme koeficient r_0 na hodnotu pět dle zadání. Ostatní parametry nastavíme na nulu.

Posledním úkolem pro dokončení této úlohy, je vykreslení grafických závislostí. Na obrázku 53 můžu pozorovat výsledné průběhy pro odvozené vztahy a na obrázku 54 výsledné průběhy vypočítané pomocí simulátoru regulačního obvodu.



Obrázek 53: Odvozené vztahy – Příklad 1



Obrázek 54: Simulátor zpětnovazebního obvodu – Příklad 1

3.2.2 Příklad 2 – Soustava 1. řádu s PI regulátorem

Je zadána diferenciální rovnice systému $5y'+y=u$, tento systém regulujte PI regulátorem s koeficientem $r_0 = 5$ a koeficientem $r_{-1} = 1$. Určete přenos soustavy, regulátoru. Určete a vykreslete odezvu na jednotkový skok žádané hodnoty pro veličiny $y(t)$, $u(t)$, $e(t)$ v délce trvání $t(s) = 20$. Správnost výpočtu ověřte pomocí simulačního programu.

Přenos soustavy je.

$$5y' + y = u$$

$$5sY(s) + Y(s) = U(s)$$

$$Y(s)(5s + 1) = U(s)$$

$$G_s = \frac{Y(s)}{U(s)} = \frac{1}{5s + 1}$$

Přenos regulátoru je.

$$G_R = \frac{U(s)}{E(s)} = r_0 + \frac{r_{-1}}{s} + r_1 s = 5 + \frac{1}{s} = \frac{5s + 1}{s}$$

Vypočet přenosu a přechodové funkce pro regulovanou veličinu z žádané hodnoty.

Přenos regulované veličiny z žádané hodnoty.

$$G_{YW} = \frac{Y(s)}{W(s)} = \frac{G_R G_S}{1 + G_R G_S} = \frac{\left(\frac{5s+1}{s}\right)\left(\frac{1}{5s+1}\right)}{1 + \left(\frac{5s+1}{s}\right)\left(\frac{1}{5s+1}\right)} = \left(\frac{5s+1}{5s^2+s}\right)\left(\frac{5s^2+s}{5s^2+6s+1}\right) = \frac{5s+1}{\underline{\underline{(s+1)\left(s+\frac{1}{5}\right)}}}$$

$$Y(s) = G_{YW} W(s) = \left(\frac{5s+1}{(s+1)\left(s+\frac{1}{5}\right)}\right)\left(\frac{1}{s}\right)$$

Výše odvozený vztah rozložím na součet parciálních zlomků a určím neznámé A, B, C .

$$\left(\frac{5s+1}{(s+1)\left(s+\frac{1}{5}\right)}\right)\left(\frac{1}{s}\right) = \frac{A}{s} + \frac{B}{s+1} + \frac{C}{s+\frac{1}{5}}$$

$$5s+1 = A(s+1)\left(s+\frac{1}{5}\right) + Bs\left(s+\frac{1}{5}\right) + Cs(s+1)$$

$$5s+1 = A\left(s^2+1,2s+\frac{1}{5}\right) + Bs^2 + \frac{1}{5}sB + s^2C + sC$$

$$5s+1 = s^2(A+B+C) + s\left(1,2A+\frac{1}{5}B+C\right) + \frac{1}{5}A$$

$$s^0: \frac{1}{5}A = 1 \Rightarrow \underline{\underline{A=5}}$$

$$s^1: 1,2A + \frac{1}{5}B + C = 5 \Rightarrow 1,2 \cdot 5 + \frac{1}{5}(-5 - C) + C = 5 \Rightarrow \underline{\underline{C=0}}$$

$$s^2: A + B + C = 0 \Rightarrow B = -5 - C \Rightarrow \underline{\underline{B=-5}}$$

$$Y_{(s)} = G_{YW}W_{(s)} = \left(\frac{5s+1}{(s+1)\left(s+\frac{1}{5}\right)} \right) \left(\frac{1}{s} \right) = 5\frac{1}{s} - 5\left(\frac{1}{s+1}\right)$$

Pomocí slovníku zpětné Laplaceovy transformace je nutno vyhledat vhodný vzorec a určit originál pro přechodovou funkci regulované veličiny $y(t) = h(t)$.

$$y(t) = L^{-1}\left\{5\frac{1}{s} - 5\left(\frac{1}{s+1}\right)\right\} = \underline{\underline{5-5e^{-t}}} = \underline{\underline{1-e^{-t}}}$$

Vypočet přenosu a přechodové funkce pro akční veličinu z žádané hodnoty. Přenos akční veličiny z žádané hodnoty.

$$G_{UW} = \frac{U_{(s)}}{W_{(s)}} = \frac{G_R}{1+G_R G_S} = \frac{\frac{5s+1}{s}}{1+\left(\frac{5s+1}{s}\right)\left(\frac{1}{5s+1}\right)} = \left(\frac{5s+1}{s}\right)\left(\frac{5s^2+s}{5s^2+6s+1}\right) = \frac{25s^2+10s+1}{(s+1)\left(s+\frac{1}{5}\right)}$$

$$U_{(s)} = G_{UW}W_{(s)} = \left(\frac{25s^2+10s+1}{(s+1)\left(s+\frac{1}{5}\right)} \right) \left(\frac{1}{s} \right)$$

Výše odvozený vztah rozložím na součet parciálních zlomků a určím neznámé A, B, C .

$$\left(\frac{25s^2+10s+1}{(s+1)\left(s+\frac{1}{5}\right)} \right) \left(\frac{1}{s} \right) = \frac{A}{s} + \frac{B}{s+1} + \frac{C}{s+\frac{1}{5}}$$

$$25s^2 + 10s + 1 = A(s+1)\left(s + \frac{1}{5}\right) + Bs\left(s + \frac{1}{5}\right) + Cs(s+1)$$

$$25s^2 + 10s + 1 = A\left(s^2 + 1,2s + \frac{1}{5}\right) + Bs^2 + \frac{1}{5}sB + s^2C + sC$$

$$25s^2 + 10s + 1 = s^2(A+B+C) + s\left(1,2A + \frac{1}{5}B + C\right) + \frac{1}{5}A$$

$$s^0 : \frac{1}{5}A = 1 \Rightarrow \underline{\underline{A=5}}$$

$$s^1 : 1,2A + \frac{1}{5}B + C = 5 \Rightarrow 1,2(5) + \frac{1}{5}(20 - C) + C = 10 \Rightarrow \underline{\underline{C=0}}$$

$$s^2 : A + B + C = 25 \Rightarrow B = 20 - C \Rightarrow \underline{\underline{B=20}}$$

$$U_{(s)} = G_{\text{UW}}W_{(s)} = \frac{25s^2 + 10s + 1}{(s+1)\left(s + \frac{1}{5}\right)} \left(\frac{1}{s}\right) = \underline{\underline{5\frac{1}{s} - 20\left(\frac{1}{s+1}\right)}}$$

Pomocí slovníku zpětné Laplaceovy transformace je nutno vyhledat vhodný vzorec a určit originál pro přechodovou funkci akční veličiny $u(t) = h(t)$.

$$u(t) = L^{-1}\left\{5\frac{1}{s} - 20\left(\frac{1}{s+1}\right)\right\} = \underline{\underline{1 + 4e^{-t}}}$$

Vypočet přenosu a přechodové funkce pro regulační odchylku z žádané hodnoty. Přenos regulační odchylky z žádané hodnoty.

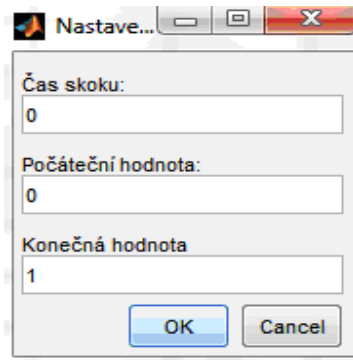
$$G_{\text{EW}} = \frac{E_{(s)}}{W_{(s)}} = \frac{1}{1 + G_{\text{R}}G_{\text{S}}} = \frac{1}{1 + \frac{5s+1}{s} \frac{1}{5s+1}} = \frac{5s^2 + s}{5s^2 + 6s + 1} = \frac{5s^2 + s}{(s+1)\left(s + \frac{1}{5}\right)} = \frac{5s\left(s + \frac{1}{5}\right)}{(s+1)\left(s + \frac{1}{5}\right)} = \underline{\underline{\frac{5s}{s+1}}}$$

$$E_{(s)} = G_{\text{EW}}W_{(s)} = \left(\frac{5s}{s+1}\right)\left(\frac{1}{s}\right) = \underline{\underline{5\frac{1}{s+1}}}$$

Pomocí slovníku zpětné Laplaceovy transformace je nutno vyhledat vhodný vzorec a určit originál pro přechodovou funkci regulační odchylky $e(t) = h(t)$.

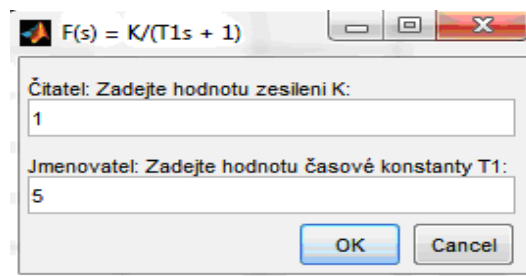
$$e(t) = L^{-1}\left\{5\frac{1}{s+1}\right\} = \underline{\underline{5e^{-t}}} = \underline{\underline{e^{-t}}}$$

Teďko je nutné přistoupit k samotnému nastavení potřebných parametrů v simulátoru, abychom si mohli zkontrolovat vypočtené vztahy.



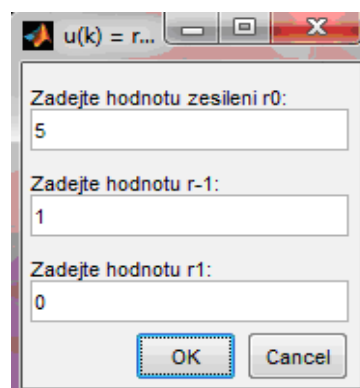
Obrázek 55: Nastavení jednotkového skoku - Příklad 2

Na obrázku 55 můžeme vidět celkové nastavení žádané hodnoty pro jednotkový skok. Za parametr čas skoku jsme nastavili nulu, počáteční hodnotu rovno nule a konečnou hodnotu rovno jedné.



Obrázek 56: Nastavení přenosu soustavy – Příklad 2

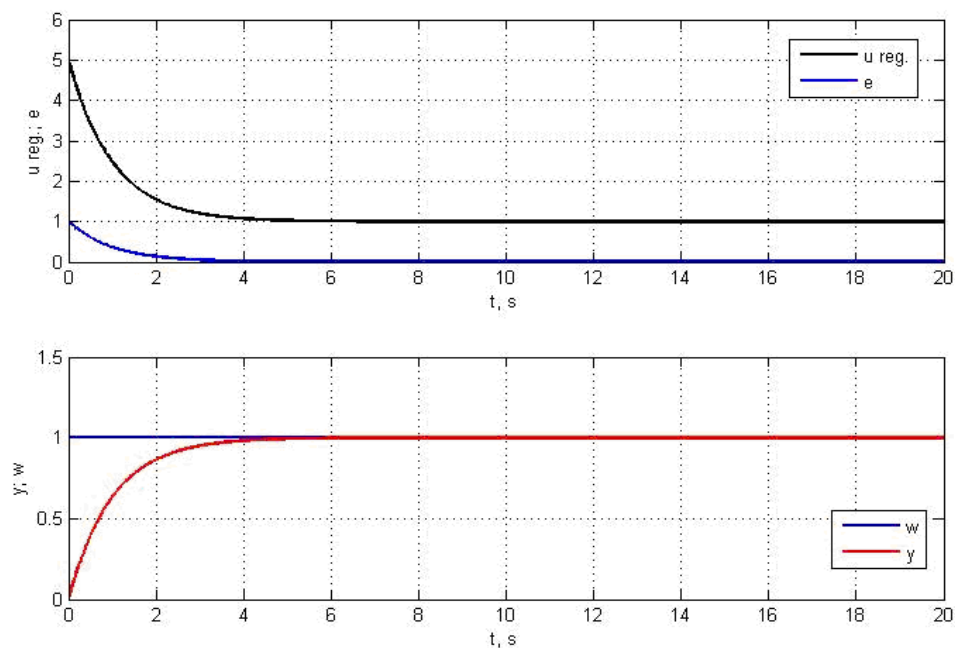
Další co je nutné nastavit je přenos soustavy pro 1. řád na obrázku 56. Hodnotu zesílení nastavme na jedničku a časovou konstantu na hodnotu pět.



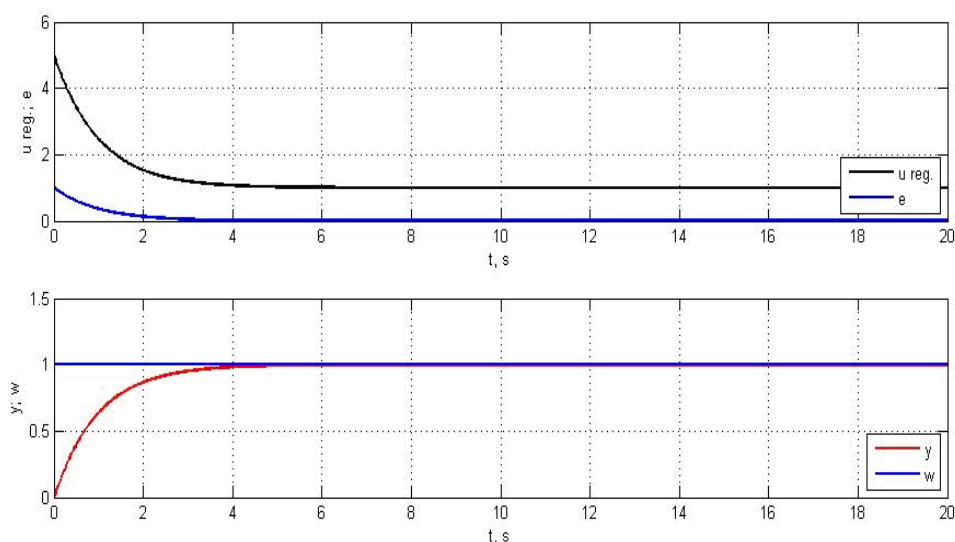
Obrázek 57: Nastavení PID regulátoru – Příklad 2

Jako poslední nedílnou součástí zpětnovazebního obvodu je nastavení PID regulátoru na obrázku 57. Nastavme koeficient r_0 na hodnotu pět a koeficient r_{-1} na hodnotu jedna. Ostatní parametry nastavíme na nulu.

Posledním úkolem pro dokončení této úlohy je vykreslení grafických závislostí. Na obrázku 58 můžu pozorovat výsledné průběhy pro odvozené vztahy a na obrázku 59 výsledné průběhy vypočítané pomocí simulátoru regulačního obvodu.



Obrázek 58: Odvozené vztahy – Příklad 2



Obrázek 59: Simulátor zpětnovazebního obvodu – Příklad 2

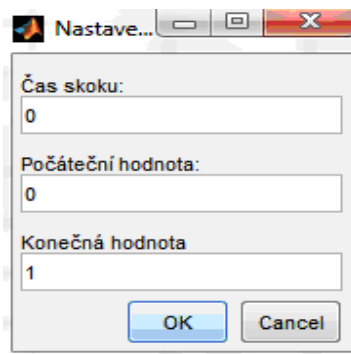
3.2.3 Příklad 3 – Soustava 1. řádu s Dvoupolohovým regulátorem

Je zadán přenos soustavy $G_s = \frac{1}{5s + 1}$. Danou soustavu regulujte dvoupolohovým regulátorem. Dvoupolohový regulátor nastavte podle tabulky 2. Vykreslete odezvu systému na jednotkový skok žádané hodnoty pro celkovou dobu simulace $t(s) = 20$. Následně můžete měnit jednotlivé parametry a sledovat jak se systém bude chovat.

Tabulka 2: Nastavení dvoupolohového regulátoru Příklad 3

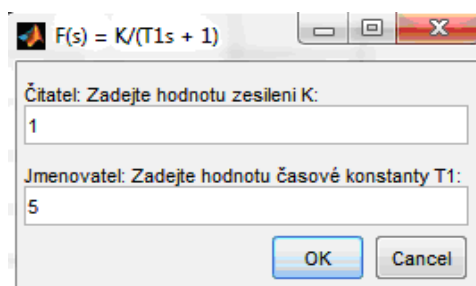
Mezní hodnota pro ON	Mezní hodnota pro ON	Výstup když ON	Výstup když OFF
1	0	1	0

Nyní můžu přistoupit k samotnému nastavení potřebných parametrů v simulátoru, abychom si mohli zkontrolovat vypočtené vztahy.



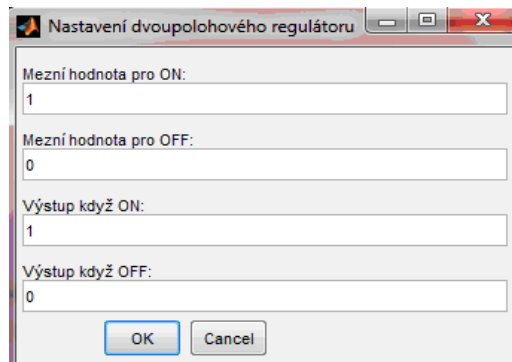
Obrázek 60: Nastavení jednotkového skoku - Příklad 3

Na obrázku 60 je vidět celkové nastavení žádané hodnoty pro jednotkový skok. Za parametr čas skoku jsme nastavili nulu, počáteční hodnotu rovno nule a konečnou hodnotu rovno jedné.



Obrázek 61: Nastavení přenosu soustavy – Příklad 3

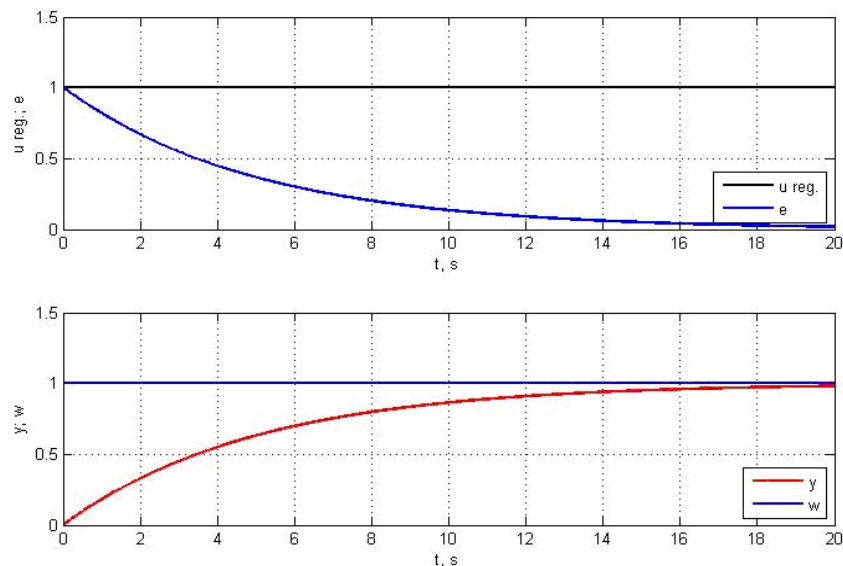
Další co je nutné nastavit, je přenos soustavy pro 1. řád na obrázku 61. Hodnotu zesílení nastavme na jedničku a časovou konstantu na hodnotu pět.



Obrázek 62: Nastavení Dvupolohového regulátoru – Příklad 3

Jako poslední nedílnou součástí regulačního obvodu je nastavení dvupolohového regulátoru na obrázku 63. Nastavme parametry mezní hodnotu pro ON na jedničku, mezní hodnotu pro OFF na nulu, výstup když ON na jedničku a výstup když OFF na nulu.

Posledním úkolem pro dokončení této úlohy je vykreslení grafických závislostí. Průběhy vypočítané pomocí simulátoru regulačního obvodu pro dvupolohový regulátor jsou na obrázku 63.



Obrázek 63: Simulátor zpětnovazebního obvodu – Příklad 3

3.2.4 Příklad 4 – Soustava 3. řádu s P a PI regulátorem

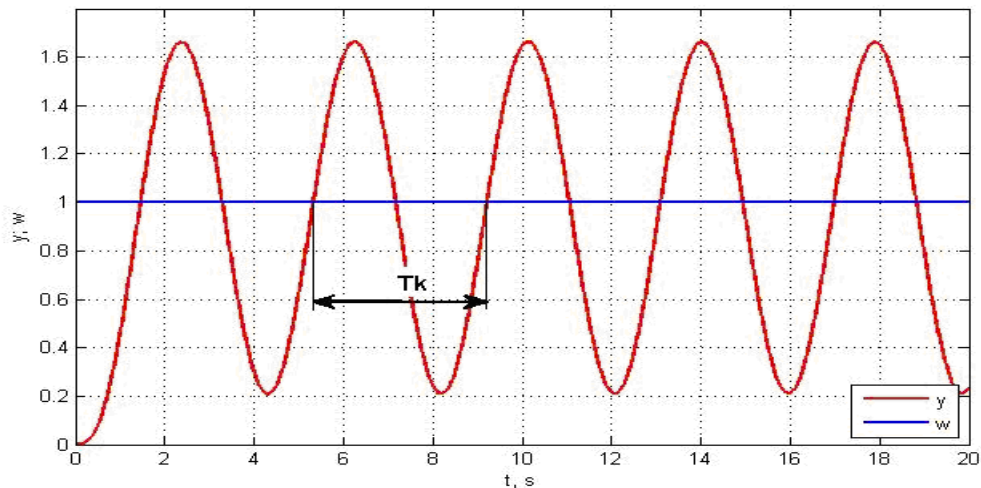
Mějme přenos soustavy $G_s = \frac{1}{3s^3 + 6s^2 + 8s + 1}$. Vypočtete kritické zesílení r_k a periodu kmitů T_k na hranici stability. Pomocí pravidel Ziegler-Nikolsovi metody, nalezněte nastavení P a PI regulátoru. Pro jednotlivé regulátory v simulačním programu nastavte žádanou hodnotu podle následující tabulky a vykreslete odezvy pro celkovou dobu simulace $t(s) = 200$.

Tabulka 3: Průběh žádané hodnoty Příklad 4

$t(s)$	0	50	50,1	100	150	200
$w(s)$	2	2	4	4	2	2

Určení kritického zesílení

Pro nalezení kritického zesílení musíme položit integrační a derivační složku rovno nule. A snažíme se najít takové zesílení, aby se v uzavřeném obvodu objevily netlumené kmity. Netlumené kmity jsou takové kmity, které v čase mají stejnou amplitudu a neztrácejí energii.



Obrázek 64: Zpětnovazební obvod na hranici stability – netlumené kmity

Přenos soustavy je.

$$G_s = \frac{1}{3s^3 + 6s^2 + 8s + 1}$$

Přenos regulátoru je.

$$G_R = \frac{U(s)}{E(s)} = r_0 = r_K$$

Nejdříve určíme přenos regulačního obvodu, například poruchy. Čítel na vyšetření stability nemá vliv.

$$G_{YV_2} = \frac{1}{1 + G_R G_S} = \frac{1}{1 + \frac{r_0}{3s^3 + 6s^2 + 8s + 1}} = \frac{1}{\frac{3s^3 + 6s^2 + 8s + 1 + r_K}{3s^3 + 6s^2 + 8s + 1}} = \frac{3s^3 + 6s^2 + 8s + 1}{3s^3 + 6s^2 + 8s + 1 + r_K}$$

Pomocí Hurwitzova kritéria stability určíme r_K . Aby obvod byl stabilní, musí být všechny koeficienty charakteristické rovnice kladné. A všechny subdeterminanty Hurwitzovi matice H_{N-1} až H_2 větší jak nula.

Charakteristická rovnice 3. řádu

$$\underbrace{3s^3}_{a_3} + \underbrace{6s^2}_{a_2} + \underbrace{8s}_{a_1} + \underbrace{1 + r_K}_{a_0} = 0$$

$$H_{N-1} = H_{3-1} = H_2 = \begin{pmatrix} a_{N-1} & a_{N-3} \\ a_N & a_{N-2} \end{pmatrix} = \begin{pmatrix} a_{3-1} & a_{3-3} \\ a_3 & a_{3-2} \end{pmatrix} = \begin{pmatrix} a_2 & a_0 \\ a_3 & a_1 \end{pmatrix} = \begin{pmatrix} 6 & (1 + r_K) \\ 3 & 8 \end{pmatrix} = 48 - 3(1 + r_K)$$

Pro nalezení kritického zesílení je potřeba tuto rovnici položit rovno nule.

$$48 - 3(1 + r_K) = 0$$

$$48 - 3 - 3r_K = 0$$

$$-3r_K = -45 \Rightarrow \underline{\underline{r_K = 15}}$$

Výpočet periody kmitu na hranici stability. Do charakteristické rovnice je nutno dosadit vypočítanou hodnotu kritického zesílení.

$$3s^3 + 6s^2 + 8s + 1 + r_K = 0$$

$$3s^3 + 6s^2 + 8s + 1 + 15 = 0$$

$$3s^3 + 6s^2 + 8s + 16 = 0$$

Na hranici stability bude mít charakteristická rovnice dvojici imaginárních kořenu, proto za každé s dosadím $j\omega$.

$$3(j\omega)^3 + 6(j\omega)^2 + 8(j\omega) + 16 = 0$$

Následně je nutné rozdělit charakteristickou rovnici na reálnou a imaginární část a určit ω . Jak pro reálnou, tak i pro imaginární část charakteristické rovnice musí vyjít ω stejné.

$$-3\omega^3 + 8\omega^2 = 0 \Rightarrow \omega = \sqrt{\frac{8}{3}}$$

$$-6\omega^2 + 16 = 0 \Rightarrow \omega = \sqrt{\frac{16}{6}} = \sqrt{\frac{8}{3}}$$

Pomocí vzorce určíme periodu kmitu na hranici stability.

$$T_K = \frac{2\tau}{\omega} = \frac{2\tau}{\sqrt{\frac{8}{3}}} = \frac{2\tau}{\sqrt{\frac{8}{3}}} \frac{\sqrt{3}}{\sqrt{3}} = \frac{6\tau\sqrt{3}}{8} = \underline{\underline{3,85s}}$$

Výpočet nastavení P a PI regulátoru

Tabulka 4: Seřízení PID regulátoru - Ziegler-Nichols

Typ regulátoru	r_0	r_{-1}	r_1
P	$0,5 r_K$	-	-
PI	$0,45 r_K$	$0,54 r_K / T_K$	-
PID	$0,6 r_K$	$1,2 r_K / T_K$	$0,075 r_K / T_K$

P regulátor

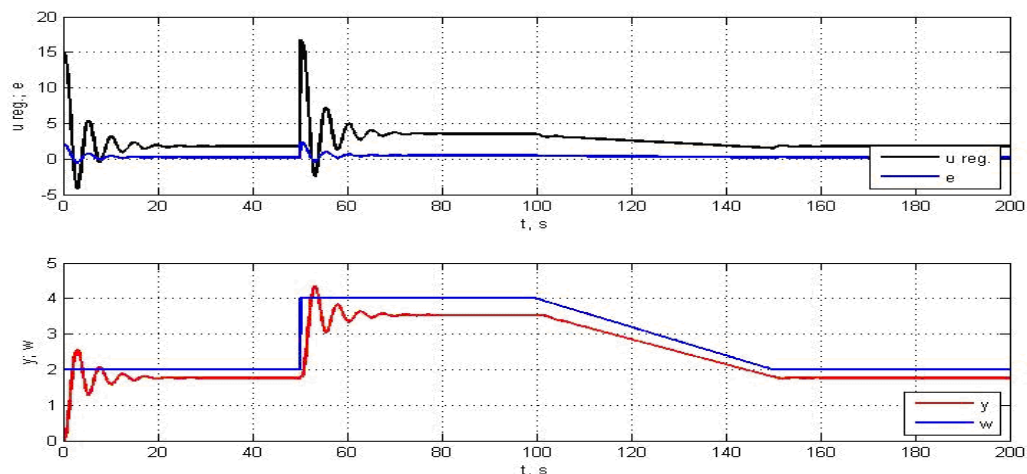
$$r_0 = 0,5r_K = 0,5 \cdot 15 = 7,5$$

PI regulátor

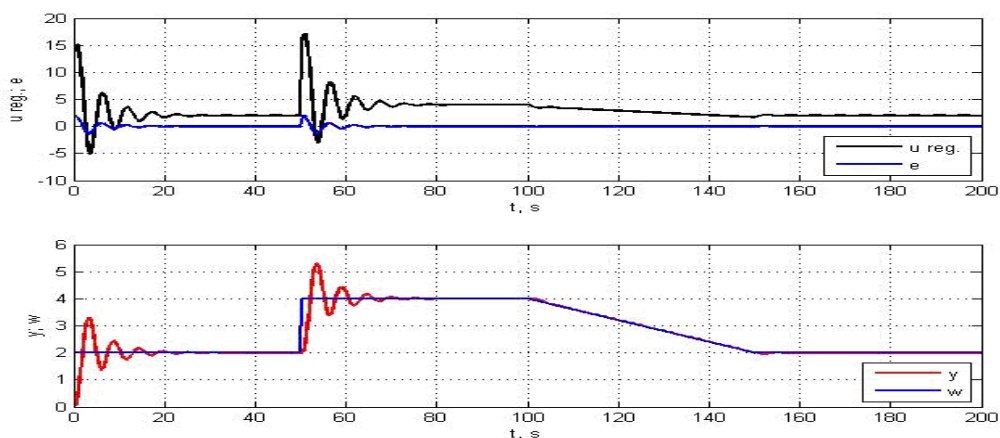
$$r_0 = 0,45r_K = 0,45 \cdot 15 = 6,3$$

$$r_{-1} = \frac{0,54r_K}{T_K} = 15 \frac{0,54}{3,85} = 2,104$$

Posledním úkolem pro dokončení této úlohy je vykreslení grafických závislostí seřízení P a PI regulátoru pomocí Ziegler-Nicholsovi metody. Na obrázku 65 je možné pozorovat odezvu systému pro seřízený P regulátor a na obrázku 66 odezvu systému pro seřízený PI regulátor.



Obrázek 66: Soustava 3. řádu – seřízení P regulátorem



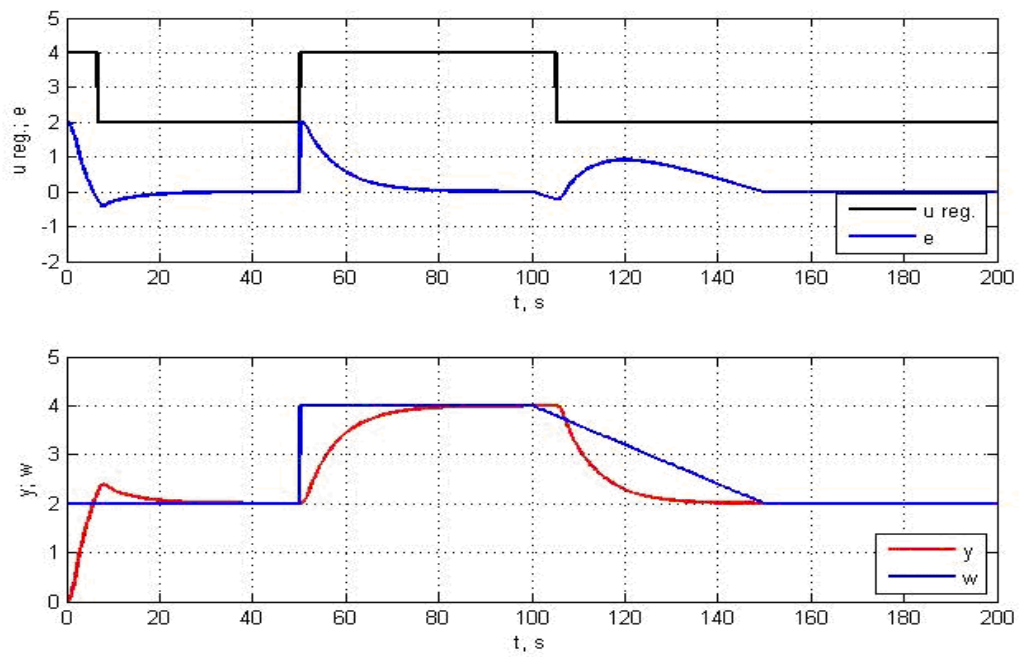
Obrázek 65: Soustava 3. řádu – seřízení PI regulátorem

3.2.5 Příklad 5 – Soustava 3. řádu s Dvupolohovým regulátorem

Je zadán přenos soustavy $G_s = \frac{1}{3s^3 + 6s^2 + 8s + 1}$. Danou soustavu regulujte dvupolohovým regulátorem. Dvupolohový regulátor nastavte podle tabulky 5. V simulačním programu nastavte žádanou hodnotu podle tabulky 3. Vykreslete odezvu systému pro celkovou dobu simulace $t(s) = 200$. Následně můžete měnit jednotlivé parametry dvupolohového regulátoru a sledovat jak se systém bude chovat.

Tabulka 5: Nastavení dvupolohového regulátoru Příklad 5

Mezní hodnota pro ON	Mezní hodnota pro ON	Výstup když ON	Výstup když OFF
1	-0,2	4	2



Obrázek 67: Soustava 3. řádu – Dvoupolohový regulátor

4 Závěr

Mezi první úkoly patřilo naučit se pracovat s GUI ve výpočetním systému Matlab. Dále bylo nutné navrhnout celý vzhled aplikace včetně rozmístění všech ovládacích prvků a jejich programových funkcí.

Aplikace dává možnost definovat vlastnosti experimentu. Uživatel může volit ze tří typů žádaných hodnot. Mezi které patří jednotkový skok, rampa a libovolná žádaná hodnota. Jestliže uživatel zvolí libovolnou žádanou hodnotu, může si vytvořit vstupní signál včetně poruch a šumu. Dále aplikace nabízí volbu přenosu soustavy, v které jsou čtyři možnosti výběru a to mezi 1. řádem, 2. řádem, 3. řádem a libovolným řádem soustavy. Poslední volbou aplikace je typ regulátoru. Zde je možnost volby mezi dvěma typy regulátoru a to PID regulátorem a dvupolohovým regulátorem. Hlavní nevýhodou mnou navržené aplikace je, že nelze zadat počáteční podmínky simulace.

Algoritmus spočívá v diskretizaci spojitého systému a regulátoru, kde je převedena diferenciální rovnice systému a regulátoru na diferenční rovnici. Dále bylo nutné navrhnout dostatečnou vzorkovací periodu, která byla zvolena jako podíl celkové doby simulace ku deseti tisícům, aby výsledky simulace odpovídaly řešení původním diferenciálním rovnicím.

Dále jsem navrhl několik příkladů včetně jejich analytických řešení. Ve většině příkladu je i uvedený postup, jak nastavit parametry do jednotlivých editačních polí. Tyto příklady mohou společně s vytvořenou aplikací sloužit studentům předmětu automatizace jako laboratorní úloha.

Navržená aplikace má širší oblast použití než aplikace *Regulátory TARka*. Na internetových stránkách je dostupný zdrojový kód aplikace jinými slovy „skript“. Abychom tuto aplikaci mohli spustit [4], musíme tento kód zkopírovat do Matlabu. Hlavní nevýhodou zmiňované aplikace je, že uživateli není umožněno vytvořit si vstupní signál, respektive, aby v regulačním obvodu působila porucha a šum. Aplikace nabízí pouze vykreslení odezvy systému na jednotkový skok. Regulovanou soustavu můžeme regulovat pouze PID regulátorem.

Literatura

- [1] ŠVARC, Ivan a Miluše VÍTEČKOVÁ. *Automatické řízení*. 1. vyd. Brno: Akademické nakladatelství CERM, 2007, s. 42-43. ISBN 978-80-214-3491-2.
- [2] BALÁTĚ, Jaroslav. *Automatické řízení*. 2. vyd. Praha: BEN - technická literatura, 2004, 663 s. ISBN 80-730-0148-9.
- [3] ZAPLATÍLEK, Karel. *MATLAB pro začátečníky*. 2. vyd. Praha: BEN - technická literatura, 2005, 151 s. ISBN 80-730-0175-6.
- [4] LINDA, Miloslav a Ilja MAŠÍK. *Materiály ke cvičením: Technická fakulta České zemědělské univerzity v Praze, Katedra elektrotechniky a automatizace* [online]. Praha, 2009, 3.1.2011 [cit. 2013-04-19]. Dostupné na: <<http://www.tf.imasik.cz/tar/regulshow.m>>
- [5] DORF, Richard C a Robert H BISHOP. *Modern control systems*. 10. vyd. New Jersey: Prentice Hall, 2005, xxii, 1082 s. ISBN 0-13-127765-0.
- [6] DUŠEK, František a Daniel HONC. *Matlab a Simulink: úvod do používání*. Vyd. 1. Pardubice: Univerzita Pardubice, 2005, 172 s. ISBN 80-719-4776-8.
- [7] ZAPLATÍLEK, Karel a Bohuslav DOŇAR. *MATLAB: tvorba uživatelských aplikací*. 1. vyd. Praha: BEN - technická literatura, 2004, 215 s. ISBN 80-730-0133-0.
- [8] JANČÍKOVÁ, Z. *Modelování a simulace* [online]; Ostrava, 2008; s. 18–25. [cit. 2013-04-19]. Dostupné na: <http://www.fmmi.vsb.cz/miranda2/export/sites-root/fmmi/cs/okruhy/urceno-pro/studenty/podklady-ke-studiu/studijni-opory/638-Jancikova-Modelovani-a-simulace.pdf>

