

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

**SIMULACE ŘÍZENÍ DYNAMICKÝCH SYSTÉMŮ STAVOVÝMI
REGULÁTORY V PROSTŘEDÍ LABVIEW**

Bc. Ladislav Jedlička

Diplomová práce
2014

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Ladislav Jedlička**
Osobní číslo: **I12514**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Řízení procesů**
Název tématu: **Simulace řízení dynamických systémů stavovými regulátory
v prostředí LabVIEW**
Zadávající katedra: **Katedra řízení procesů**

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

Cíl práce:

Cílem práce je využití simulačních nástrojů Control Design and Simulation Module prostředí LabVIEW k modelování a simulaci různých typů dynamických procesů, konstrukci jejich statických i dynamických charakteristik a jejich následnému řízení pomocí stavových regulátorů. Pozornost bude věnována ověření robustnosti navržených regulačních algoritmů, jejichž součástí budou i estimátory úplného či redukovaného řádu.

Teoretická část:

Stručná rešerše problematiky týkající se tvorby aplikací v LabVIEW s využitím nástrojů Control Design and Simulation Module. Statické a dynamické charakteristiky systémů. Řízení dynamických systémů stavovými regulátory s estimátory úplného a redukovaného řádu.

Implementační část:

Tvorba simulačních schémat a souvisejících aplikací v grafickém programovacím jazyce G prostředí LabVIEW.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

- [1] DORF, R. C.; BISHOP, R. H. Modern Control Systems. 10 vyd. New Jersey: Prentice Hall, 2005. 881 s. ISBN 0-13-127765-0.
- [2] BALÁTĚ, J. Automatické řízení. 2 vyd. Praha: BEN - technická literatura, 2004. 663 s. ISBN 80-7300-148-9.
- [3] MODRLÁK, O. Základy analýzy a syntézy ve stavovém prostoru [skriptum]. Liberec: TU, 2004. 82 s.
- [4] SLAPNÍČKA, M. Syntéza regulačního obvodu se stavovým regulátorem s ohledem na jeho robustnost [diplomová práce]. Liberec: TU, 2010. Fakulta mechatroniky, informatiky a mezioborových studií. 74 s.
- [5] KUPKA, L.; JANEČEK, J. Matlab & Simulink: řešené příklady. Lanškroun: TG tisk a SOŠ a SOU, 2007. 224 s. ISBN 978-80-239-9532-9.


Vedoucí diplomové práce: **Ing. Libor Kupka, Ph.D.**
Katedra řízení procesů

Datum zadání diplomové práce: **24. října 2013**
Termín odevzdání diplomové práce: **16. května 2014**


prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.


Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 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 15. 05. 2014

Ladislav Jedlička

Poděkování

Na tomto místě bych rád poděkoval vedoucímu práce panu Ing. Liboru Kupkovi, Ph.D. za ochotu při mých dotazech a na konzultacích a řadu cenných rad při vedení diplomové práce.

V Pardubicích dne 15. 05. 2014

Ladislav Jedlička

Anotace

Cílem práce je vytvořit aplikaci v prostředí LabVIEW s využitím simulačních nástrojů Control Design and Simulation Module. Aplikace bude sloužit k modelování a simulaci různých typů dynamických procesů, konstrukci jejich statických i dynamických charakteristik a jejich následnému řízení pomocí stavových regulátorů. Pozornost bude věnována ověření robustnosti navržených regulačních algoritmů, jejichž součástí budou i estimátory úplného či redukovaného řádu.

Klíčová slova

estimátor, LabVIEW, charakteristiky, stavový popis, stavový regulátor.

Title

Simulation of dynamic systems by state regulators in LabVIEW

Annotation

The aim is to create an application in LabVIEW using simulation tools Control Design and Simulation Module. The application will be used for modeling and simulation of various types of dynamic processes, construction of static and dynamic characteristics and their subsequent control by state regulators. Attention will be given to verify the robustness of the proposed control algorithms, which will also include estimators full or reduced order.

Keywords

estimator, LabVIEW, characteristics, state description, state regulators.

Obsah

Seznam zkratk	9
Seznam symbolů	10
Seznam obrázků	11
Úvod	13
1 LabVIEW	15
1.1 Co je LabVIEW?	15
1.2 Knihovna Control Design and Simulation Module	15
1.2.1 Knihovna Simulation	16
1.2.2 Knihovna Control Design	18
1.2.3 Knihovna System Identification	19
1.2.4 Knihovna PID	19
1.2.5 Knihovny Fuzzy logic a Sim Interface	21
2 Statické a dynamické charakteristiky systémů	23
2.1 Statická charakteristika systému	23
2.2 Dynamické charakteristiky systému	24
2.2.1 Přejchodová charakteristika	24
2.2.2 Impulzní charakteristika	25
2.2.3 Frekvenční charakteristiky	27
2.3 Možnosti zobrazení charakteristik systému v LabVIEW	29
3 Stavový popis dynamických systémů	32
3.1 Estimace stavů	34
3.1.1 Deterministický estimátor	34
3.1.2 Diskrétní estimátor úplného řádu (deterministický Luenbergerův estimátor)	35
3.1.3 Diskrétní estimátor redukovaného řádu	36
3.2 Stavový regulátor	37
3.2.1 Návrh regulátoru v konečném počtu regulačním kroků	38

3.2.2	Návrh regulátoru podle kvadratického kritéria	40
4	Tvorba aplikace v LabVIEW	43
4.1	Vykreslení charakteristik v aplikaci	43
4.2	Matlab skript	44
4.3	Zapojení diskrétního estimátoru	47
4.4	Regulační obvod s diskrétním estimátorem a stavovým regulátorem	48
4.5	Čelní panel aplikace	49
5	Výsledky	51
5.1	Použité systémy	51
5.2	Vypočítané hodnoty	51
5.3	Výsledné průběhy	52
5.3.1	Systém1	52
5.3.2	Systém2	55
6	Závěr	61
	Literatura	62
	Příloha A – Celý Matlab Script	A – 1

Seznam zkratk

LabVIEW	Laboratory Virtual Instruments Engineering Workbench (laboratorní pracoviště virtuálních přístrojů)
LTI	Linear time-invariant (lineární a časově invariantní)
MATLAB	Matrix Laboratory (maticová laboratoř)
MIMO	Multi In Multi Out (více vstupů a více výstupů)
SISO	Single In Single Out (jeden vstup a jeden výstup)
VI	Virtual Instrument (virtuální přístroj)

Seznam symbolů

$w(t)$	žádaná hodnota
$e(t)$	regulační odchylka
$y(t)$	regulovaná veličina
a_0, b_0	koeficienty lineární diferenciální rovnice
r	řád astatismu
s	Laplaceův operátor
$U(s)$	Laplaceův obraz vstupní veličiny
$Y(s)$	Laplaceův obraz výstupní veličiny
$h(t)$	přechodová funkce
$g(t)$	impulsní funkce
n	řád systému
p	počet vstupních (budících veličin)
r	počet výstupních (vybuzených) veličin
A, M	matice systému (n, n)
B, N	matice systému (n, p)
C	matice výstupu (r, n)
D	matice převodu (r, p)
E	jednotková matice
$x(t), x(k)$	stavový vektor
$u(t), u(k)$	vektor buzení
$y(t), y(k)$	vektor výstupu
R	matice regulátoru
H_E, M_E, N_E	matice estimátoru
T_s	vzorkovací perioda

Seznam obrázků

Obr. 1.1 – Podknižovny Control Design & Simulation	16
Obr. 1.2 – Knihovna Simulation	16
Obr. 1.3 – Př. 1 – Blokový diagram	17
Obr. 1.4 – Přejchodová a impulzní charakteristika	18
Obr. 1.5 – Knihovna Control Design	18
Obr. 1.6 – Knihovna System Identification	19
Obr. 1.7 – Knihovna PID	20
Obr. 1.8 – PID a PID autotuning	20
Obr. 1.9 – Př. 2 – blokový diagram	20
Obr. 1.10 – Regulovaná a žádaná hodnota	21
Obr. 1.11 – Knihovny Fuzzy logic a Sim Interface	22
Obr. 2.1 – Přejchodová charakteristika	25
Obr. 2.2 – Impulzní charakteristika	27
Obr. 2.3 – Nyquistova charakteristika	28
Obr. 2.4 – Bodeho charakteristika	29
Obr. 2.5 – Bloky pro tvorbu charakteristik v LabVIEW	29
Obr. 2.6 – Vnitřní struktura bloku pro přejchodovou charakteristiku	30
Obr. 2.7 – Vnitřní struktura bloku pro Bodeho charakteristiku	30
Obr. 2.8 – Vnitřní struktura bloku pro Nyquistovu charakteristiku	31
Obr. 3.1 – Blokové schéma spojitého systému	33
Obr. 3.2 – Blokové schéma diskretního systému	33
Obr. 3.3 – Zapojení s estimátorem	35
Obr. 3.4 – Obvod s diskretním stavovým regulátorem	37
Obr. 3.5 – Ukázka návrhu	40
Obr. 4.1 – Blokové schéma část 1	44
Obr. 4.2 – Zapojení s diskretním estimátorem	47
Obr. 4.3 – Regulační obvod	48
Obr. 4.4 – Čelní panel aplikace	49
Obr. 5.1 – Systém1 – přejchodová a impulzní charakteristika	52
Obr. 5.2 – Systém1 – Bodeho charakteristiky	53
Obr. 5.3 – Systém1 – Nyquistova charakteristika	53
Obr. 5.4 – Systém1 – charakteristiky s diskretním estimátorem	54

Obr. 5.5 – Systém1 – průběhy se stavovým regulátorem	54
Obr. 5.6 – Systém2 – přechodová a impulzní charakteristika	55
Obr. 5.7 – Systém2 – Bodeho charakteristiky	56
Obr. 5.8 – Systém2 – Nyquistova charakteristika	56
Obr. 5.9 – Systém2 – charakteristiky s diskrétním estimátorem	57
Obr. 5.10 – Systém2 – s různými vzorkovacími frekvencemi	58
Obr. 5.11 – Systém2 – Průběhy se stavovým regulátorem	58
Obr. 5.12 – Systém2 – při vzorkovací periodě 0,3 s	60

Úvod

Práce je rozdělena na dvě části – na teoretickou část a na praktickou část. Teoretická část se skládá ze tří kapitol. Na tyto kapitoly je sepsaná krátká rešerše, kde se daný problém čtenáři přiblíží a budou ukázány i příklady.

První kapitola se zabývá tvorbou aplikací v prostředí LabVIEW v případě, že se použije simulační nástroj Control Design and Simulation Module. V této kapitole budou důkladně popsány knihovny, které obsahuje tento nástroj a budou vytvořeny i jednoduché příklady, na kterých je ukázáno uživateli, jak s těmito nástroji pracovat a jak si s nimi usnadnit práci a na co si při práci s nimi dát pozor.

V druhé kapitole je uvedena problematika týkající se statických a dynamických charakteristik systémů. Čtenář se dozví, jak se charakteristiky dělí, co která charakteristika znamená, jakým způsobem se získá. Na příkladech budou dané charakteristiky ukázány, aby bylo jasné, jak mohou tyto charakteristiky vypadat pro uvedené systémy.

Nejobsáhlejší kapitolou je část, která řeší problematiku řízení dynamických systémů stavovými regulátory s estimátory. Nejprve je popsáno, co to vůbec stavový popis je a jak vypadá pro statické a hlavně dynamické systémy. Jsou uvedeny i bloková schéma spojitého a diskrétního systému. Dále je popsáno, co je to estimátor. Estimátory jsou spojité a diskrétní, ale z důvodu velkého využití estimátorů diskrétních se tato práce zaměří na popis právě těchto estimátorů. Diskrétní estimátory se dále dělí na estimátory úplného řádu a estimátory redukovaného řádu. Čtenář se dozví, jaká jsou zapojení pro který estimátor, základní rovnice a kroky nutné pro návrh daného estimátoru. Největší část bude věnována stavovému regulátoru. Bude popsáno, co to stavový regulátor je a především budou důkladně vysvětleny dvě metody pro návrh diskrétního stavového regulátoru. Metody, které budou popsány, jsou jednak návrh regulátoru v konečném počtu regulačních kroků a návrh regulátoru podle kvadratického kritéria. U obou metod bude ukázán princip návrhu regulátoru a na dané soustavě bude ukázán i příklad výpočtu hodnot regulátoru oběma metodami.

Cílem praktické části je vytvořit aplikaci, která využívá simulačních nástrojů Control Design and Simulation Module v prostředí LabVIEW k modelování a simulaci různých dynamických procesů a konstrukci jejich statických i dynamických charakteristik a jejich následnému řízení pomocí stavových regulátorů. Pozornost bude věnována ověření robustnosti navržených regulačních algoritmů, jejichž součástí budou i estimátory úplného a redukovaného řádu. K výpočtům v prostředí LabVIEW musí být použit Matlab skript, který spočítá, jak hodnoty stavového regulátoru, tak i hodnoty diskrétního estimátoru. Ve skriptu se

využijí dvě metody návrhu stavového regulátoru. Jedná se o návrh v konečném počtu regulačních kroků a podle kvadratického kritéria; obě metody jsou popsány v teoretické části práce. Aplikace bude vytvořena tak, aby uživatel pouze zadal matice systému a vzorkovací periodu a vybral způsob výpočtu stavového regulátoru. Jelikož se výpočty budou realizovat pomocí Matlab skriptu, tak si uživatel metodu bude muset vybrat přímo ve skriptu. Na čelním panelu se pak zobrazí všechny charakteristiky systému, charakteristiky s estimátorem a průběhy se stavovým regulátorem. Čelní panel také ukáže, jaké vyšly hodnoty estimátoru a stavového regulátoru. Funkčnost vytvořené aplikace se ověří na několika vybraných systémech.

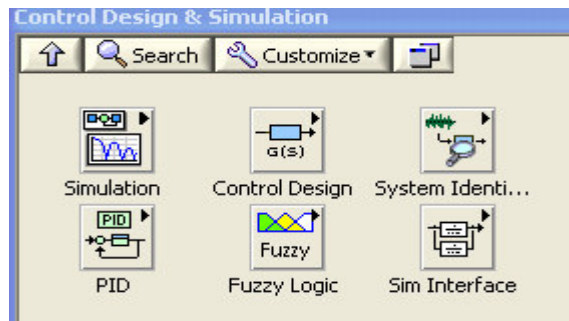
1 LabVIEW

1.1 Co je LabVIEW?

Programovací a vývojové prostředí LabVIEW je produktem americké firmy National Instruments, která je největším výrobcem v oblasti virtuální instrumentace, technické disciplíny, která zažívá veliký rozvoj v oblasti vývoje, výzkumu, školství a průmyslu. Prostředí LabVIEW, které je někdy nazýváno jako „grafický jazyk“, je vhodné k programování systémů pro měření a analýzu signálů, řízení a vizualizaci technologických procesů a také k programování složitých systémů. Hlavním cílem je nahradit dočasně nebo i trvale prostorově, finančně a mnohdy i časově náročné využití technických prostředků řešením virtuálním za přispění programových prostředků a zejména pak grafickými a vizuálními prostředky a zprostředkovat tak uživateli maximální názornost. Výsledný produkt LabVIEW se nazývá **virtuální přístroj** (Virtual Instrument – VI), protože svými projevy a činnostmi připomíná klasický přístroj ve své fyzické podobě. Může prezentovat buď skutečný měřicí přístroj, prvek měřicího nebo řídicího systému případně celý měřicí (řídicí) systém. Vytvoření nového VI probíhá ve dvou oknech, v okně **čelního panelu** (standardně šedé) a v okně **blokového diagramu** (standardně bílé). Každé okno má klasické ovládací prvky dané prostředím Windows, roletové menu, nástrojovou lištu a místo pro ikonu a konektor. Vývoj VI začíná programátor většinou návrhem čelního panelu, po jeho ukončení pak pokračuje v okně blokového diagramu. Čerpáno z (PECHOUŠEK, 2004) a (VLACH, HAVLÍČEK, VLACH, 2008).

1.2 Knihovna Control Design and Simulation Module

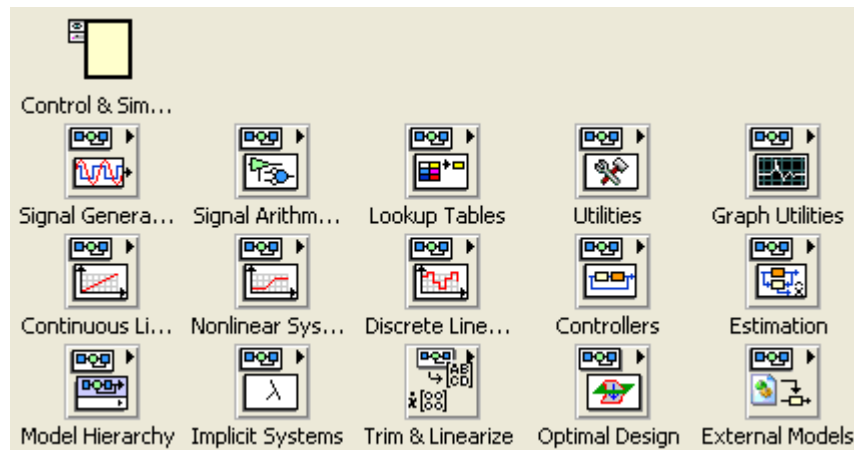
V LabVIEW je mnoho knihoven, ze kterých je možné si vybírat nástroje, které jsou potřebné k řešení požadovaného problému. Jednou z nich jsou nástroje Control Design and Simulation Module, které obsahují další podknihovny, kterými jsou Simulation, Control Design, System Identification, PID, Fuzzy Logic a Simulation Interface; tyto podknihovny je vidět na obr. 1.1.



Obr. 1.1 – Podknihovny Control Design & Simulation

1.2.1 Knihovna Simulation

Jako první je v nabídce knihovna **Simulation**, která je tvořena mnoha podknihovny. Všechny podknihovny jsou vidět na obr. 1.2. Nejdůležitějším blokem v knihovně Simulation je Control & Simulation Loop, bez něj by nešlo nic vytvořit, neboť by nešlo přidávat bloky z knihovny Simulation. Pokud se tedy používají prvky z knihovny Simulation, tak se musí vytvořit prostor, což se udělá právě pomocí Control & Simulation Loop a poté je možné přidávat další bloky z této knihovny a vytvářet svůj program. Control & Simulation Loop umožňuje nastavit parametry simulace a parametry času. Je možné nastavit dobu simulace, minimální a maximální krok, metodu řešení, periodu, fázi a další.



Obr. 1.2 – Knihovna Simulation

Jak je vidět na obr. 1.2, tak si uživatel může vybírat z mnoha knihoven, co potřebuje pro svoji simulaci, ať už z generátorů signálu (např. skok sinusový signál), bloků pro tvorbu diskretních signálů (diskretní přenos, zpoždění), z grafů, kde jsou v nabídce 2 grafy XY graf a Simtime graf, estimátorů, bloků pro optimální návrh nebo nástrojů, které lze využít pro simulaci (paměť, parametry simulace) atd. Díky knihovně Simulation se tvoří v LabVIEW programy velmi podobně jako v programu Matlab & Simulink.

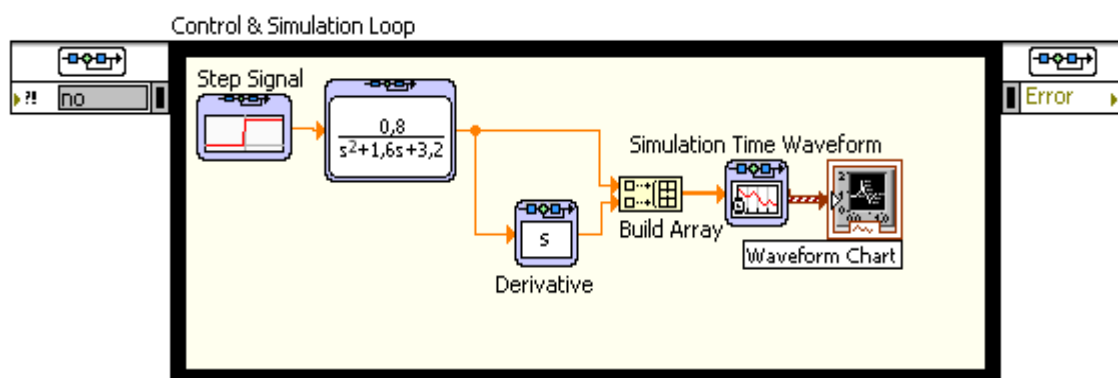
Jako ukázka práce s knihovnou Simulation je vytvořen program, který vykreslí přechodovou charakteristiku systému popsaného přenosem

$$G(s) = \frac{0,8}{s^2 + 1,6s + 3,2},$$

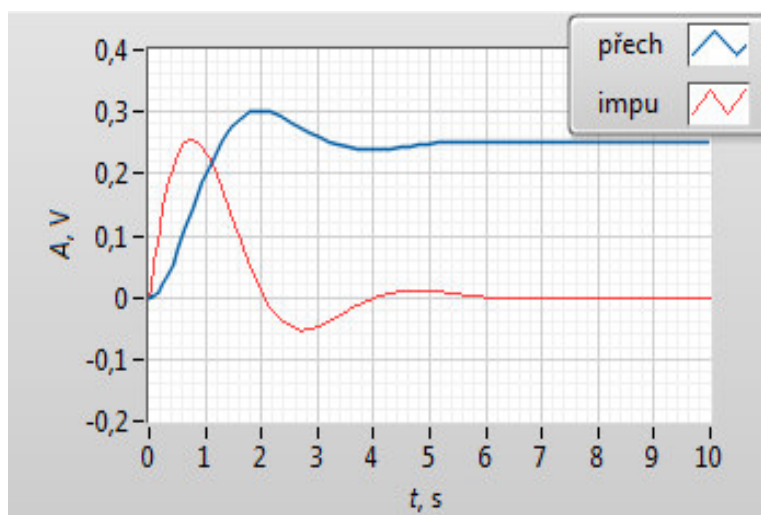
za přenos je dána ještě derivace, čímž získáme z přechodové charakteristiky charakteristiku impulzní. Obě tyto charakteristiky jsou přivedeny do grafu pomocí bloku Build Array. K této realizaci je potřeba Control & Simulation Loop, Step Signal, Transfer Function, Waveform chart, Derivative a výše zmíněný blok BuildArray. Do Transfer Function je zadaný přenos

$$G(s) = \frac{0,8}{s^2 + 1,6s + 3,2}.$$

Step signal má nastavenou finální hodnotu na hodnotu 1. Simulace je v bloku Control & Simulation Loop nastavena na dobu 10 sekund. Nyní stačí propojit blok Step Signal s Transfer Function, výsledkem jsou pořadnice přechodové charakteristiky, které jsou přivedeny BuildArray. Poté se Transfer Function propojí s blokem Derivative a výstup je přiveden také na BuildArray. Výstup Build Array je přiváděn do grafu (Waveform chart). S přidáním bloku Waveform chart se přidá na pracovní plochu i Simulation Time Waveform. Jak program vypadá v okně blokového diagramu je vidět na obr. 1.3. Přechodová a impulzní charakteristika, která se zobrazí na čelním panelu, je na obr. 1.4.



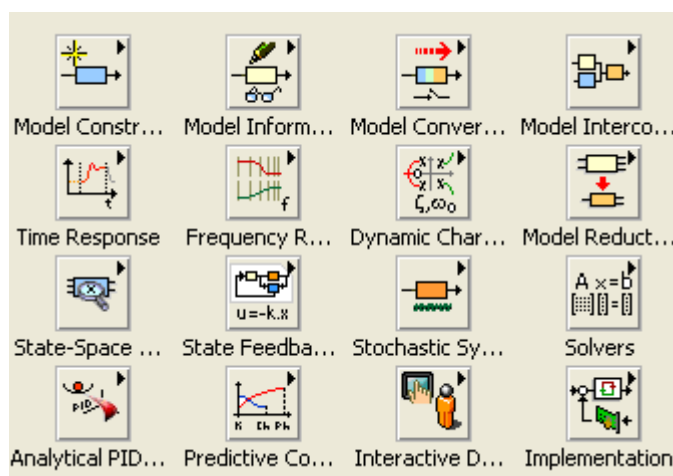
Obr. 1.3 – Př. 1 – Blokový diagram



Obr. 1.4 – Přejchodová a impulsní charakteristika

1.2.2 Knihovna Control Design

Druhou knihovnou v pořadí je **Control Design**, která má také několik podknihoven, ty jsou k vidění na obr. 1.5.



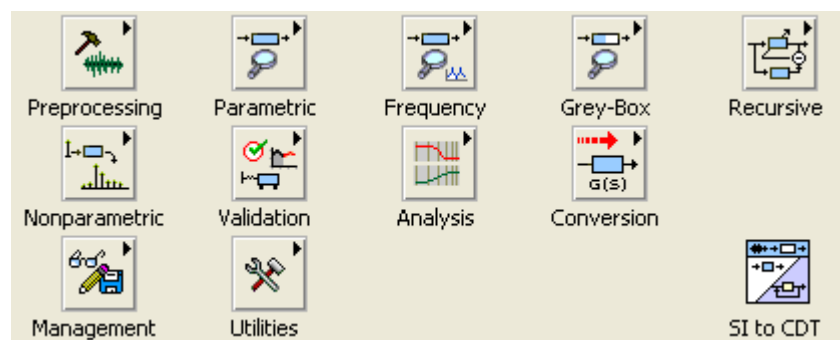
Obr. 1.5 – Knihovna Control Design

V této knihovně není nutné pro přidávání prvků mít na ploše vytvořený prostor pomocí Control & Simulation Loop. Díky této knihovně lze vytvořit model pomocí podknihovny Model Construction, která nabízí několik možností pro vytvoření žádaného modelu. Podknihovna Model Information umožňuje nastavit jméno modelu, typ modelu, typ modelu v diskretním prostoru a další. Dále tu jsou knihovny, ve kterých jsou bloky, pomocí kterých se vykreslují časové odezvy, frekvenční odezvy nebo jiné dynamické charakteristiky. V časových odezvách se nalezne přechodová a impulzní odezva, počáteční odezva nebo

lineární simulace. Ve frekvenčních se nalézají bloky pro charakteristiky Bode, Nyquist, Nichols, zesílení, fáze a šířku pásma. V dynamických charakteristikách je možné si zvolit prvek, který zjistí najednou póly i nuly nebo je možné použít bloky pro zjištění pólů a nul samostatně, dále tu je prvek pro zjištění celkového zpoždění, pro zjištění zesílení, stability. Dalšími podknižovkami jsou redukce modelu, stochastické systémy, řešitel a analýza stavového modelu a další.

1.2.3 Knihovna System Identification

Další knihovnou je *System Identification*, její podknižovny jsou vidět na obr. 1.6.

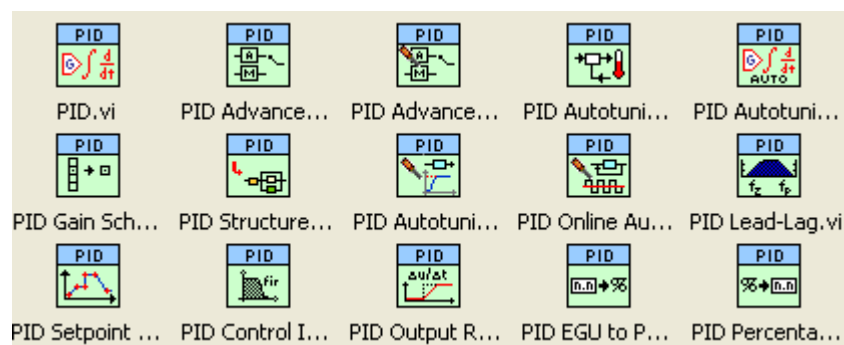


Obr. 1.6 – Knihovna System Identification

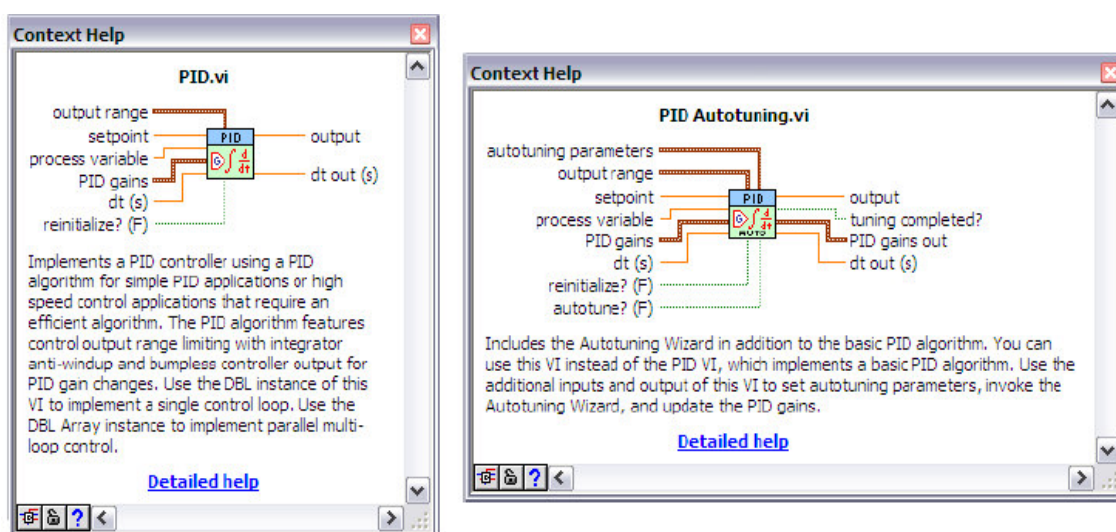
Identifikace systému kombinuje nástroje pro sběr dat s algoritmy identifikačního systému pro přesné modelování. V této knihovně jsou funkce pro běžné úlohy, jako je předzpracování dat, vytváření modelů a systémová analýza. Pomocí dalších nástrojů je možné vykreslovat model s intuitivním grafickým znázorněním nebo také ukládat model. V podknižovně Parametric se vybírá, jaké parametry jsou požadované, v model conversion lze změnit model a v Analysis vykreslit Bodeho charakteristiku, Nyquistovu charakteristiku a póly s nulami.

1.2.4 Knihovna PID

Co vše nabízí tato knihovna je na obr. 1.7. Jak je vidět, je v nabídce mnoho variant PID, ať už klasický PID nebo si uživatel může zvolit PID Advanced, což je rozšířená verze vzhledem ke klasickému PID – možnost přepínání mezi ručním a automatickým řízením. Dále je možné zvolit PID Autotuning, který umožňuje automatické ladění parametrů PID regulátoru. Dále lze nastavit vstupní filtr, PID žádané hodnoty a další.



Obr. 1.7 – Knihovna PID

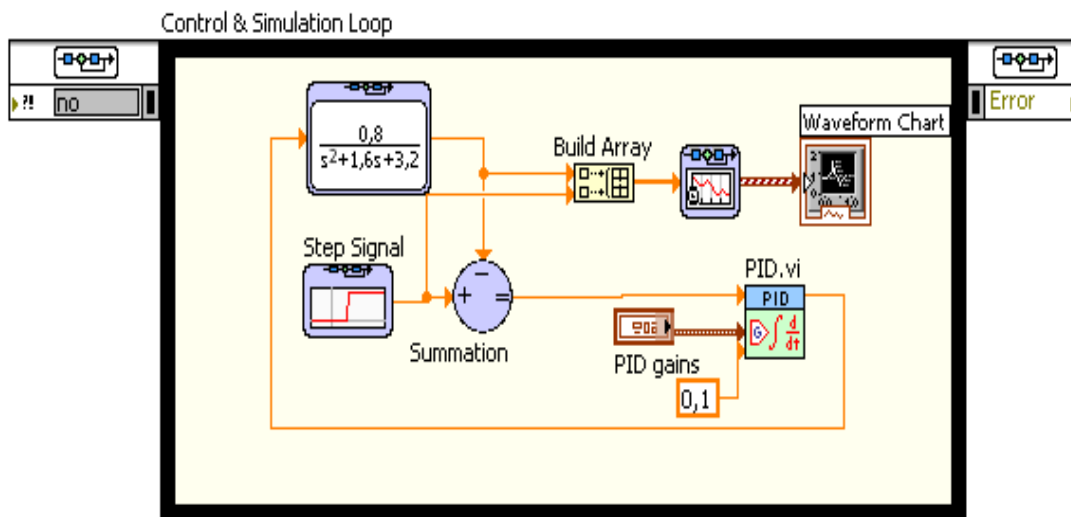


Obr. 1.8 – PID a PID autotuning

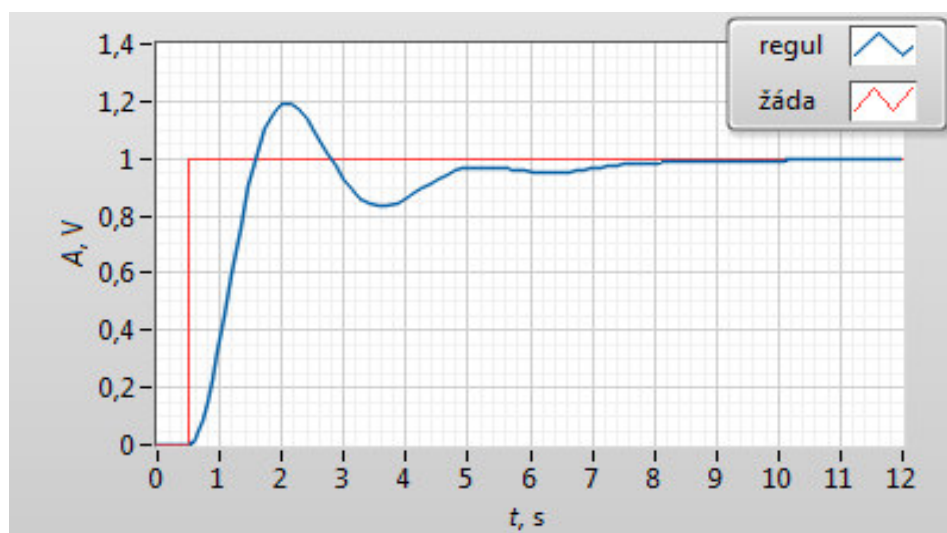
Pro práci s blokem PID je vytvořen ukázkový příklad číslo 2, kde je využitý přenos z minulého ukázkového příkladu

$$G(s) = \frac{0,8}{s^2 + 1,6s + 3,2},$$

sestaví se regulační obvod a najdou se parametry regulátoru PID. Oproti předešlému příkladu je přidám blok Summation a samozřejmě blok PID. Parametry regulátoru jsou díky vstupu PID gains ukázané na čelním panelu, kde se mohou snadno měnit, než se nalezne ideální kombinace. Doba simulace byla nastavena na 12 sekund. V grafu je zobrazena žádaná hodnota, která se v čase 0,5 sekund změní z 0 na 1, a regulovaná veličina. Jak program vypadá v okně blokového diagramu, je vidět na obr. 1.9. Průběh žádané a regulované veličiny je na obr. 1.10.



Obr. 1.9 – Př. 2 – blokový diagram

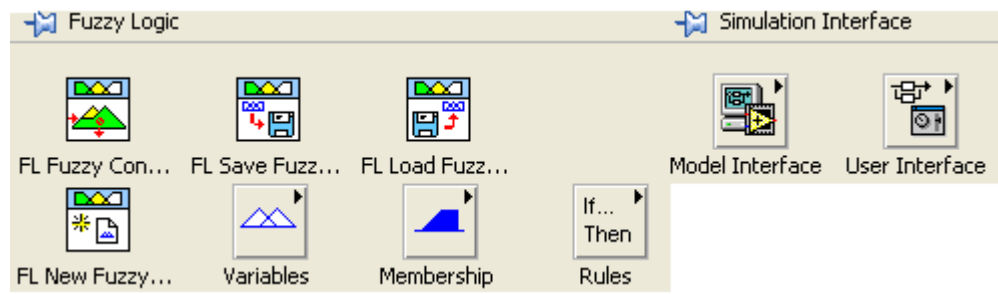


Obr. 1.10 – Regulovaná a žádaná veličina

1.2.5 Knihovny Fuzzy logic a Sim Interface

Knihovna *Fuzzy logic* se zabývá, jak je již z názvu zřejmé, Fuzzy logikou. Fuzzy logika je určitá část matematické logiky, která je odvozena od teorie Fuzzy množin. V nabídce LabVIEW si uživatel může vybrat, jestli chce otevřít již existující Fuzzy systém, vytvořit nový nebo uložit systém. Dále si může vybírat z řady proměnných a podmínek, které se mu ve Fuzzy logice hodí.

Poslední knihovnou je *Sim Interface*, která je tvořena dvěma podknihovnami, kde si uživatel může vybrat, zda chce modelové rozhraní nebo uživatelské rozhraní. Každá podknihovna obsahuje vhodné prvky, které se hodí právě pro takové rozhraní, které si uživatel vybral.



Obr. 1.11 – Knihovny Fuzzy logic a Sim Interface

2 Statické a dynamické charakteristiky systémů

Systém je chápán jako soubor prvků, mezi nimiž jsou vztahy a jako celek má určité vztahy ke svému okolí. Chování systému se popisuje pomocí charakteristik. Charakteristiky se dělí na *statické* a *dynamické* charakteristiky.

Dynamické charakteristiky ještě dále dělíme na:

- přechodovou charakteristiku,
- impulzní charakteristiku (váhovou charakteristiku),
- frekvenční charakteristiky.

Čerpáno z (BALÁTĚ, 2004).

2.1 Statická charakteristika systému

Zvláštním případem vnějšího popisu systému je statická charakteristika. Statická charakteristika je grafická závislost výstupní veličiny systému na vstupní veličině v ustáleném stavu, tj. poté co odezní přechodové děje tj.

$$y(\infty) = \lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sY(s) \quad (2.1)$$

v závislosti

$$u(\infty) = \lim_{t \rightarrow \infty} u(t) = \lim_{s \rightarrow 0} sU(s) \quad (2.2)$$

Z výše uvedené rovnice vyplývá, že platí algebraická rovnice

$$y = f(u) \quad (2.3)$$

$$y = k_0 \cdot u \quad (2.4)$$

kde k_0 je dáno vztahem

$$k_0 = \frac{b_0}{a_0} \quad (2.5)$$

Za předpokladu, že $a_0 \neq 0$ je koeficient přenosu.

($|k_0| > 1$ zesílení systému, $|k_0| < 1$ zeslabení systému). Čerpáno z (BALÁTĚ, 2004).

2.2 Dynamické charakteristiky systému

2.2.1 Přejchodová charakteristika

Přejchodová funkce, která má označení $h(t)$, je odezva na jednotkový skok při nulových počátečních podmínkách. Grafické znázornění přechodové funkce se nazývá přechodová charakteristika.

Jednotkový skok je dán

$$u(t) = \eta(t) \quad (2.6)$$

- pro $t \geq 0$ je rovna 1,
- pro $t < 0$ je rovna 0.

Laplaceův obraz jednotkového skoku je

$$L\{\eta(t)\} = L\{1(t)\} = \frac{1}{s} \quad (2.7)$$

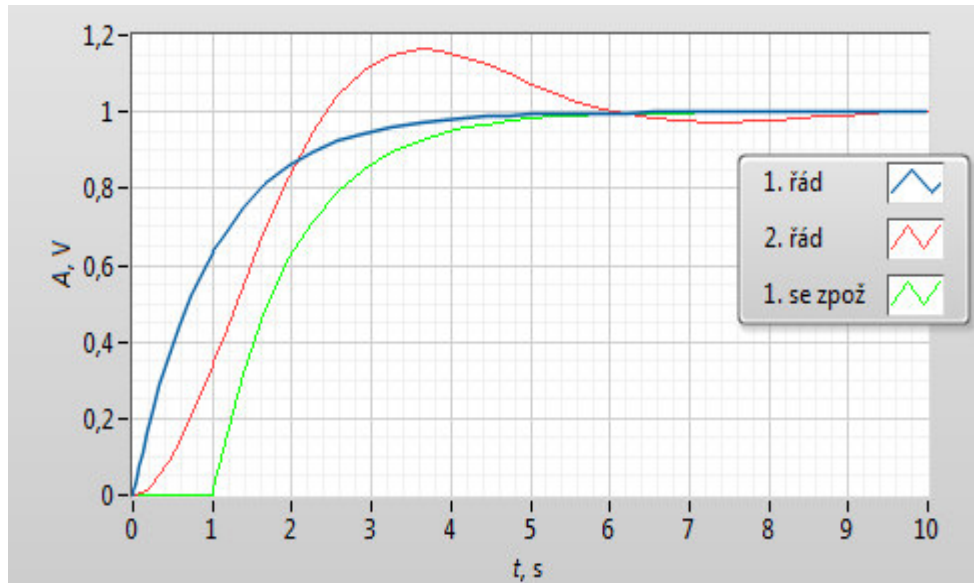
Laplaceův obraz přechodové funkce je

$$L\{h(t)\} = H(s) = \frac{G(s)}{s} \quad (2.8)$$

Hodnota přechodové funkce v čase $t = 0$ je $h(0) = 0$ a to v případě, že v přenosu systému je stupeň jmenovatele aspoň o jednotku větší než stupeň čitatele. Hodnota přechodové funkce v čase $t = \infty$ je rovna

$$h(\infty) = \lim_{s \rightarrow 0} sH(s) = \lim_{s \rightarrow 0} s \cdot \frac{G(s)}{s} = \lim_{s \rightarrow 0} G(s) \quad (2.9)$$

U systému s dopravním zpožděním je přechodová funkce posunuta o čas T_d doprava ve smyslu nárůstu času. Jak vypadá přechodová charakteristika je vidět na obr. 2.1, kde je zobrazen průběh přechodové charakteristiky 1. řádu, 2. řádu a 1. řádu s dopravním zpožděním, které je nastaveno na 2 sekundy.



Obr. 2.1 – Přejchodová charakteristika

Čerpáno z (BALÁTĚ, 2004).

2.2.2 Impulzní charakteristika

Impulzová funkce, která se značí $g(t)$, je odezva na jednotkový impulz, při nulových počátečních podmínkách systému. Grafickému znázornění impulzové funkce se nazývá impulzní charakteristika. Jednotkový impulz (Diracův impulz) – je idealizovaná funkce, kterou nelze fyzikálně realizovat a je definovaná vztahy

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (2.10)$$

$$\delta(t) = 0 \quad (2.11)$$

pro $t \neq 0$

Laplaceův obraz Diracova impulzu je

$$L\{\delta(t)\} = 1 \quad (2.12)$$

Obraz impulzové funkce je

$$L\{g(t)\} = G(s) \quad (2.13)$$

$$L\{\delta(t)\} = G(s) \quad (2.14)$$

a je roven přenosu systému.

Hodnotu impulzové funkce v čase $t = 0$, když je stupeň polynomu jmenovatele i čitatele stejný ($m = n$), se dá určit ze vztahu

$$g(0) = \lim_{s \rightarrow \infty} sG(s) = \begin{cases} \infty \\ \frac{b_{n-1}}{a_n} \\ 0 \end{cases} \quad (2.15)$$

První případ je pro případ, kdy $b_n \neq 0 \rightarrow (m = n)$. Druhý případ $b_n = 0 \rightarrow (m = n - 1)$ a třetí případ je v případě, že $b_n = b_{n-1} = 0 \rightarrow (m = n - 2)$.

Z předešlého vztahu vyplývá, že když je stejný řád polynomu u čitatele i jmenovatele, tak nekonečně vysoký impuls projde systémem až na výstup. Impulzová funkce má v čase $t = 0$ nespojitost, skok z hodnoty $g(t) = 0$ pro $t < 0$ na $g(0) = \frac{b_{n-1}}{a_n}$, je-li o jednotku vyšší stupeň jmenovatele než stupeň čitatele. V případě že začíná impulzová funkce v hodnotě nula, je stupeň jmenovatele minimálně o dvě vyšší než řád čitatele. Systémy s dopravním zpožděním mají impulzovou funkci posunutou doprava právě o čas T_d .

Mezi obrazy přechodové funkce $H(s)$ a impulzní funkce $G(s)$ platí následující vztah

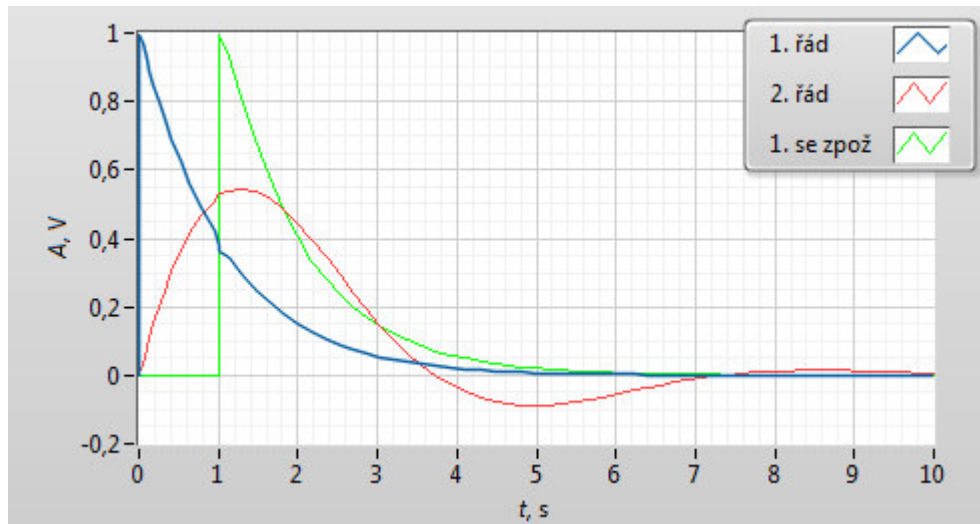
$$H(s) = \frac{G(s)}{s} \quad (2.16)$$

Z výše uvedeného vztahu vyplývá vztah mezi přechodovou funkcí $h(t)$ a impulzní funkcí $g(t)$

$$h(t) = \int_{-\infty}^t g(\tau) d\tau \quad (2.17)$$

$$g(t) = \frac{dh(t)}{dt} \quad (2.18)$$

Na obr. 2.2 je zobrazena impulzní charakteristika 1. řádu, 2. řádu a 1. řádu s dopravním zpožděním.



Obr. 2.2 – Impulzní charakteristika

Čerpáno z (BALÁTĚ, 2004).

2.2.3 Frekvenční charakteristiky

Frekvenční charakteristika je grafické znázornění frekvenčního přenosu $F(j\omega)$ v závislosti na frekvenci ω .

Frekvenční charakteristiky mohou popisovat:

- Jaké frekvence propustí filtr
- Jaké frekvenční pásmo je schopen zpracovat audiozesilovač
- Kolik snímků za sekundu a v jakém rozlišení může zobrazit analogový monitor (souvisí s frekvenčním rozsahem jeho zesilovače)

Frekvenční přenos $F(j\omega)$ je Fourierův obraz váhové funkce $g(t)$, tj. $F(j\omega) = F\{g(t)\}$ a získá se z obrazového přenosu $F(s)$, zavedením substituce $s = j\omega$.

$$F(s) = \frac{Y(s)}{U(s)} = \frac{b_m \cdot s^m + \dots + b_1 \cdot s + b_0}{s^r (a_n \cdot s^{n-r} + \dots + a_{r+1} \cdot s + a_r)} \quad (2.19)$$

$$F(j\omega) = \frac{b_m \cdot (j\omega)^m + \dots + b_1 \cdot (j\omega) + b_0}{(j\omega)^r (a_n \cdot (j\omega)^{n-r} + \dots + a_{r+1} \cdot (j\omega) + a_r)} \quad (2.20)$$

Z důvodu podmínek příčinné souvislosti je nutné, aby stupeň polynomu v čitateli byl nižší než stupeň polynomu ve jmenovateli ($m < n$).

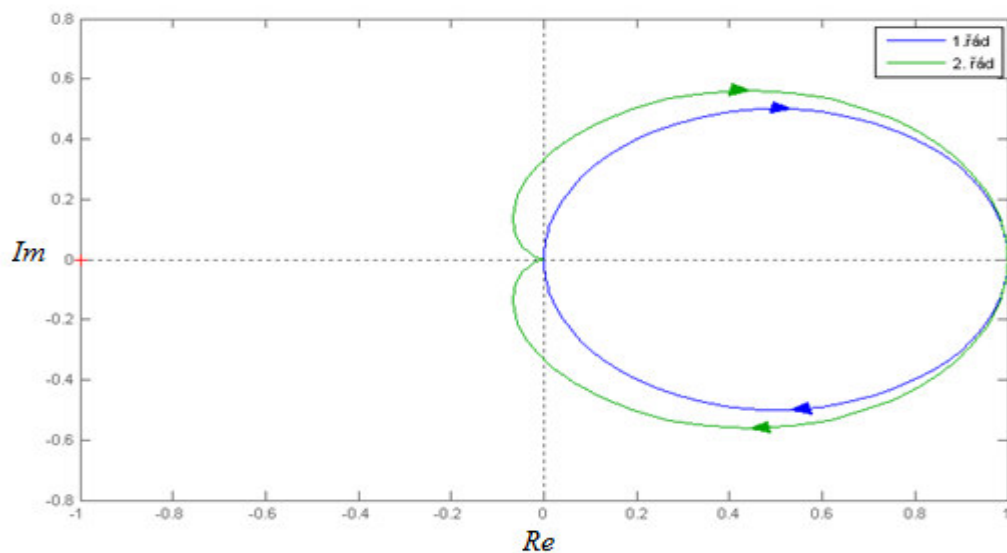
Když se nahradí $s = j\omega$ tak platí, že $F(s)$ je $F(j\omega)$

Dělí se:

- **Frekvenční charakteristiku v Gaussově rovině jako Hodograf (amplitudo-fázová charakteristika resp. Nyquistova charakteristika)**

Hodograf je křivka v komplexní rovině, kde bodům jsou přiřazeny frekvence. Vzdálenost vybraného bodu, který odpovídá určité frekvenci ω , od počátku určuje modul přenosu. Úhel mezi reálnou osou a spojnicí mezi vybraným bodem křivky a počátkem určuje fázi přenosu. Hodograf pasivních obvodů, kromě rezonančního, leží uvnitř jednotkové kružnice.

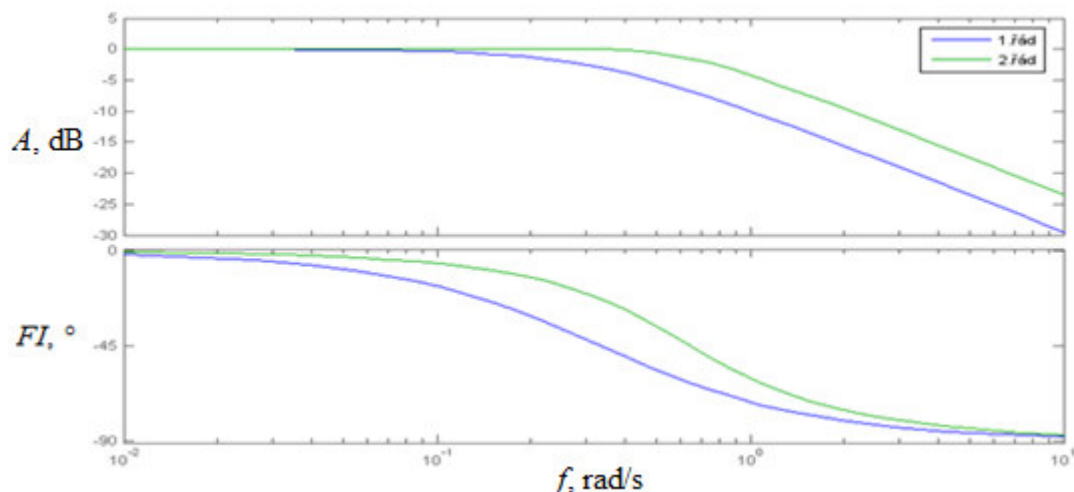
Ukázka Nyquistovy charakteristiky je na obr. 2.3, kde byla vytvořena Nyquistova charakteristika pro 1. řád a 2. řád. U této ukázky jsou zobrazeny obě větve.



Obr. 2.3 – Nyquistova charakteristika

- **Amplitudová a fázová frekvenční charakteristika v logaritmických souřadnicích (Bodeho charakteristika)**

Amplitudová charakteristika se vynáší do grafu jako $20 \cdot \log(F(j\omega))$. Obě osy amplitudové charakteristiky jsou logaritmické a jednotkou osy y je decibel. Fázová charakteristika se vynáší do grafu jako $\arg(F(j\omega))$. Osa x je logaritmická a osa y je lineární. Jednotkou y osy je radián nebo stupeň. Na obr. 2.4 je vytvořena Bodeho charakteristika systému 1. řádu a 2. řádu.



Obr. 2.4 – Bodeho charakteristika

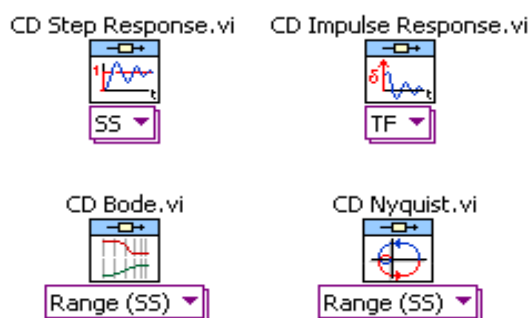
Podstatou Bodeho charakteristik jsou základní vztahy mezi logaritmy

- Logaritmus součinu je součet logaritmů
- Logaritmus podílu je rozdíl logaritmů
- $\log 1 = 0$

Čerpáno z (MÁŠA, 2013).

2.3 Možnosti zobrazení charakteristik systému v LabVIEW

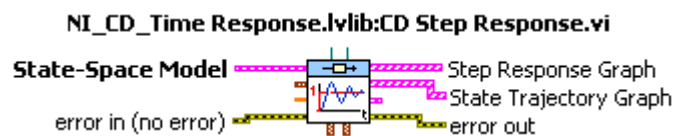
V LabVIEW má uživatel několik možností, jak si vykreslit charakteristiky, které právě potřebuje. Nejvíce možností je v nástrojích Control Design and Simulation Module, tato knihovna má být použita v praktické části, a tak se na možnosti, které nabízí, je potřeba podívat podrobněji. Možnosti bloků, které se dají využít, jsou na obr. 2.5.



Obr. 2.5 – Bloky pro tvorbu charakteristik v LabVIEW

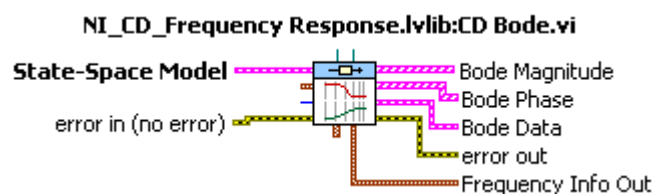
Na obr. 2.6 je vnitřní struktura bloku, díky němuž lze vykreslit přechodovou charakteristiku. Na vstup je možné přivést přenos, nuly-póly-zesílení nebo model stavového prostoru. Na výstupu je přechodová charakteristika a také průběh stavové trajektorie. Pro

správnou funkci bloku je dobré připojit i error signál. Pro případ, že stačí pouze vykreslit přechodovou charakteristiku, tak ani není potřeba připojovat žádný blok na vstup ani na výstup, neboť blok uživateli nabízí možnost vykreslit přechodovou charakteristiku uvnitř bloku. K tomu stačí dvakrát kliknout na blok a uvnitř vyplnit vstupní hodnoty a hned se vykreslí přechodová charakteristika. Stejně možnosti jako má blok pro přechodovou charakteristiku, má i blok pro impulzní charakteristiku pouze s rozdílem, že na výstupu místo přechodové charakteristiky se dostane impulzní charakteristika.



Obr. 2.6 – Vnitřní struktura bloku pro přechodovou charakteristiku

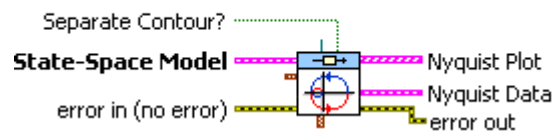
U bloku pro Bodeho charakteristiku má uživatel větší výběr možnosti vstupu, vybrat si může frekvenční rozsah nebo frekvenci vektoru a u každého vstupu může navíc zvolit zadání vstupu. Na výběr je přenos, nuly-póly-zesílení nebo model stavového prostoru. Na výstupu je možné si vybrat výstup pro vykreslení amplitudy nebo fáze a u tohoto bloku může uživatel pracovat i s daty Bodeho charakteristiky. I u tohoto bloku je možné Bodeho charakteristiku vykreslit uvnitř bloku a tím si ulehčit práci. Vnitřní struktura bloku pro Bodeho charakteristiku je na obr. 2.7.



Obr. 2.7 – Vnitřní struktura bloku pro Bodeho charakteristiku

Stejně jako u bloku pro Bodeho charakteristiku, tak i u Nyquistovy jsou stejné možnosti na vstupu. Na výstupu je možnost vykreslení Nyquistovy charakteristiky nebo je možné získat data a s nimi dále v programu pracovat. Stejně jako v předchozích blocích, tak i zde lze charakteristiky vykreslit uvnitř bloku. Vnitřní struktura bloku je ukázaná na obr. 2.8.

NI_CD_Frequency Response.lvlib:CD Nyquist.vi



Obr. 2.8 – Vnitřní struktura bloku pro Nyquistovu charakteristiku

3 Stavový popis dynamických systémů

Řešení diferenciální rovnice, případně chování dynamického systému, který lze popsat touto diferenciální rovnicí, je určeno pomocí vlastní diferenciální rovnice, dále tvarem budící funkce a počátečními podmínkami. Počáteční podmínky určují stav systému na začátku procesu. Souborem počátečních podmínek může být popsán stav systému, nejen na začátku reakce, ale v každém okamžiku. Pomocí dynamického popisu, tvaru budící funkce a stavu systému se dá určit následná reakce v každém okamžiku.

Stavové vyjádření nahrazuje diferenciální či diferenční rovnice soustavou diferenciálních či diferenčních rovnic 1. řádu. Matematický formalismus je vyjádřen maticovými rovnicemi, které popisují vývoj stavového systému. Stavový systém je tvořen n -ticí lineárně nezávislých stavových veličin, které jsou uspořádány do stavového vektoru, a jeho vztahu k výstupním veličinám systému. Čerpáno z (SLAPNIČKA, 2010).

Platí:

pro spojitý popis

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{u}, t)$$

$$\mathbf{y}(t) = g(\mathbf{x}, \mathbf{u}, t)$$

f, g jsou obecně nelineární funkce

Pro systémy LTI se získá tvar:

pro spojitý popis

$$\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{D} \cdot \mathbf{u}(t)$$

pro diskrétní popis

$$\mathbf{x}(k+1) = f(\mathbf{x}, \mathbf{u}, k) \quad (3.1 \text{ a, b})$$

$$\mathbf{y}(k) = g(\mathbf{x}, \mathbf{u}, k) \quad (3.2 \text{ a, b})$$

pro diskrétní popis

$$\mathbf{x}(k+1) = \mathbf{M} \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{u}(k) \quad (3.3 \text{ a, b})$$

$$\mathbf{y}(k) = \mathbf{C} \cdot \mathbf{x}(k) + \mathbf{D} \cdot \mathbf{u}(k) \quad (3.4 \text{ a, b})$$

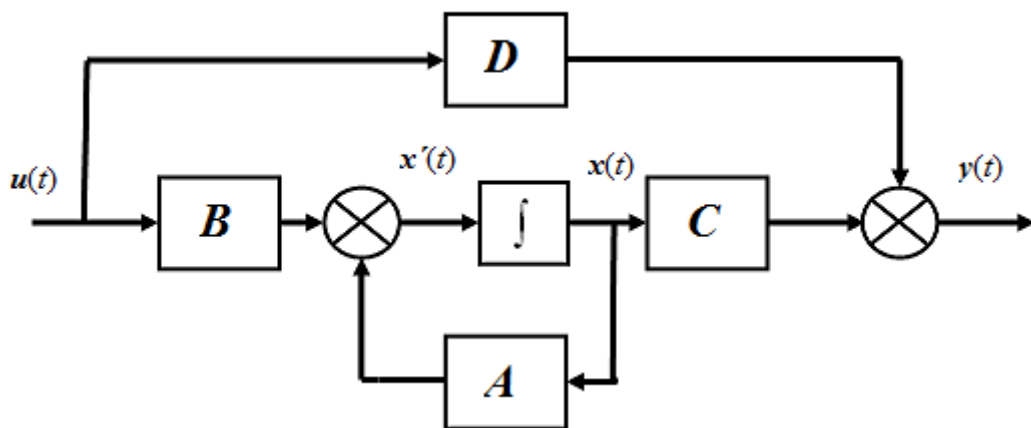
Rovnice $\dot{\mathbf{x}}(t)$ a $\mathbf{x}(k+1)$ se nazývají stavové rovnice a rovnice $\mathbf{y}(t)$ a $\mathbf{y}(k)$ se nazývají výstupní rovnice. Stavový zápis dynamického systému je stejný a to je výhoda stavového vyjádření, že má stejný zápis dynamiky.

Stavový popis dynamického systému je definován pomocí matic $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$, případně pomocí matic $\mathbf{M}, \mathbf{N}, \mathbf{C}, \mathbf{D}$. Vektory $\mathbf{x}, \mathbf{u}, \mathbf{y}$ v obecném případě jsou vektory časových funkcí, resp. diskrétních posloupností měnících v diskrétních časových okamžicích své funkční hodnoty $\mathbf{x}(k) = \mathbf{x}(kT_s), \mathbf{y}(k) = \mathbf{y}(kT_s), \mathbf{u}(k) = \mathbf{u}(kT_s)$ pro $k = 0, 1, 2, \dots$

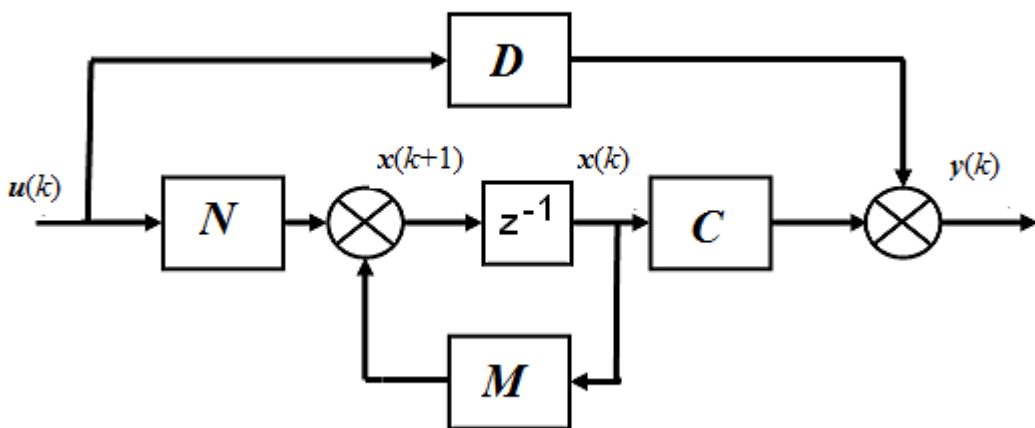
T_s je perioda vzorkování a uvedené matice jsou maticemi časových funkcí.

Jednoprvkové časové funkce $u = u$, $y = y$ by se získaly v případě, že jde o systém s jedním vstupem a jedním výstupem. V tom případě mají matice B a N jeden sloupec, matice C má jeden řádek a matice D je tvořena jedním řádkem a jedním sloupcem (skalár). V případě, že jde o ryze dynamický systém, nedochází k přímé vazbě mezi vstupem a výstupem a matice $D = 0$.

Dynamika fyzikálních procesů většiny regulovaných soustav je popsána nelineárními rovnicemi, které je možné, kvůli jednoduchosti jejich analýz, aproximovat lineárními rovnicemi.



Obr. 3.1 – Blokové schéma spojitého systému



Obr. 3.2 – Blokové schéma diskrétního systému

Blokové schéma spojitého systému je na obr. 3.1. Je vidět, že k zadání spojitého systému stačí pouze matice A, B, C, D a vstup $u(t)$. V obr. 3.1 je integrátor, díky němuž z hodnoty $x'(t)$ dostáváme hodnotu $x(t)$.

Blokové schéma diskretního systému je na obr. 3.2. Diskretní systém je definován pomocí matic N , M , C , D a vstupu $u(k)$. Ve schématu je zpoždění, díky kterému se z budoucí hodnoty $x(k+1)$ získá současná hodnota $x(k)$.

Čerpáno z (SLAPNIČKA, 2010).

3.1 Estimace stavů

Neměřitelné složky se odhadují zařízením, kterému se říká estimátor neboli pozorovatel. Výsledkem jeho činnosti je dynamický odhad x neměřitelných, někdy i měřitelných, složek stavového prostoru, které jsou pak využívány k výpočtu akční veličiny. V případě, že je měřená veličina zatížena šumem nebo nepřesnostmi měření, tak se využívají estimované veličiny místo měřených. Čerpáno z (SLAPNIČKA, 2010).

3.1.1 Deterministický estimátor

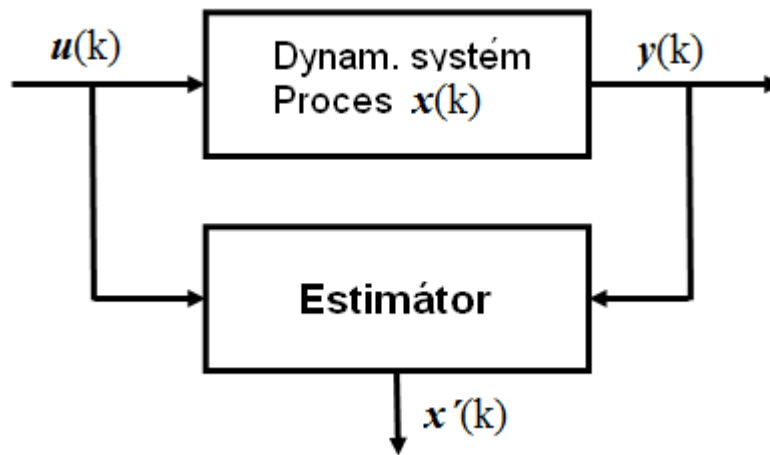
Jestliže měřený signál $y(t)$ (případně $y(k)$) neobsahuje aditivní šumový signál (lze předpokládat, že na stavové veličiny nepůsobí šumové signály), tak mluvíme o deterministické estimaci.

Deterministické i stochastické přístupy vedou ke stejné struktuře estimátorů. Jejich parametry jsou však optimalizovány buď vzhledem k dynamice estimačního procesu (deterministické estimátory) nebo vzhledem k pravděpodobnostním momentům předpokládaných náhodných poruchových signálů (stochastické estimátory).

Deterministický estimátor vychází ze znalosti matic A , B , C , D , případně matic M , N , C a D stavového popisu a ze znalosti vstupu $u(t)$, případně $u(k)$ a výstupu $y(t)$ případně $y(k)$ dynamického systému.

Dále se v práci bude rozebírat již jen diskretní forma estimátoru, neboť se v praxi používá. Čerpáno z (SLAPNIČKA, 2010).

3.1.2 Diskrétní estimátor úplného řadu (deterministický Luenbergerův estimátor)



Obr. 3.3 – Zapojení s estimátorem

Dynamický systém:

$$x(k+1) = M \cdot x(k) + N \cdot u(k) \quad (3.5)$$

$$y(k) = C \cdot x(k) + D \cdot u(k) \quad (3.6)$$

Estimátor:

$$x'(k+1) = M_E \cdot x'(k) + N_E \cdot u(k) + H_E \cdot y(k) \quad (3.7)$$

Podmínka autonomnosti estimace:

$$M - M_E - H_E C = 0 \quad (3.8)$$

$$N - N_E - H_E D = 0 \quad (3.9)$$

Z výše uvedených vztahů je:

$$\Delta x(k+1) = M_E \Delta x(k) \quad (3.10)$$

Podmínka stability estimace:

M_E – stabilní;

$$\det(\lambda E - M_E) = \det(\lambda E - M + H_E C) = (\lambda - \lambda_1) \cdot (\lambda - \lambda_2) \dots \quad (3.11)$$

$$\lambda_i : |\lambda_i| < 1, i = 1, 2, 3, \dots$$

Kroky pro návrh estimátoru:

- Volba \mathbf{H}_E tak, aby λ_i matice \mathbf{M}_E byly uvnitř jednotkového kruhu (speciálně: $\lambda_1 = \lambda_2 = \dots = 0$ bude estimátor konečný a minimální)

- $$\mathbf{M}_E = \mathbf{M} - \mathbf{H}_E \mathbf{C} \quad (3.12)$$

$$\mathbf{N}_E = \mathbf{N} - \mathbf{H}_E \mathbf{D} \quad (3.13)$$

Jestliže po provedení n -té mocniny vznikne nulová matice (vlastní čísla jsou nulová), tak estimační proces končí po n krocích (získán správný odhad).

Pro systémy SISO je volba estimátoru jasně určená, matice \mathbf{H}_E má stejný počet čísel jako je počet vlastních čísel λ_i estimačního pochodu – je dán řádem n dynamického systému. Naopak pro systémy MIMO, které mají více jak jednu výstupní veličinu, je návrh estimátoru nejednoznačný, matice \mathbf{H}_E má q sloupců a n řádků. Proto je více způsobů návrhu estimátoru. Čerpáno z (KUPKA, 2013).

3.1.3 Diskrétní estimátor redukováného řádu

Principem tohoto typu estimátoru je, že do estimace lze nezahrnout ty stavové veličiny, které jsou měřené, jako třeba výstup $\mathbf{y}(k)$.

$$\mathbf{x} = \begin{bmatrix} x_m \\ x_e \end{bmatrix} \quad (3.14)$$

x_m je část x , která je měřená a x_e je část x , která se musí estimovat

Kroky pro návrh estimátoru:

$$\det(\lambda \mathbf{E} - \mathbf{A}_E) = \det(\lambda \mathbf{E} - \mathbf{A}_{22} + \mathbf{Q} \mathbf{A}_{12}) = (\lambda - \lambda_1) \cdot (\lambda - \lambda_2) \quad (3.15)$$

Každé λ_i musí reálnou část větší jak 0, pro $i = 1, 2, 3 \dots$

$$\mathbf{H}_E = \mathbf{A}_{21} + \mathbf{A}_{22} \mathbf{Q} - \mathbf{Q} \mathbf{A}_{11} - \mathbf{Q} \mathbf{A}_{11} \mathbf{Q} \quad (3.16)$$

$$\mathbf{B}_E = \mathbf{B}_2 - \mathbf{Q} \mathbf{B}_1 \quad (3.17)$$

$$\mathbf{A}_E = \mathbf{A}_{22} - \mathbf{Q} \mathbf{A}_{12} \quad (3.18)$$

$$\tilde{x}'_E \sim x'_E \quad (3.19)$$

$$x'_E = \tilde{x}'_E + \mathbf{Q} \cdot x_m \quad (3.20)$$

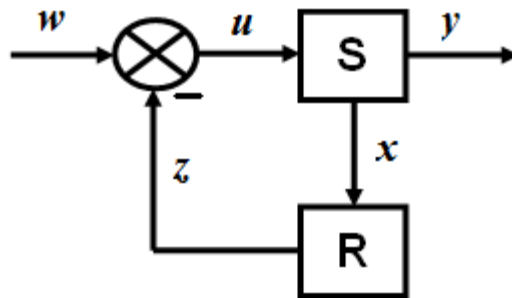
Diskrétní verze je analogická spojité $A_E \rightarrow M_E$ a $B_E \rightarrow N_E$. Čerpáno z (KUPKA, 2013).

3.2 Stavový regulátor

Činnost stavového regulátoru spočívá v tom, že přesune póly regulované soustavy do požadované polohy, kterou definují zadané kořeny jmenovatele přenosu řízení. Činnost korekční matice výstupu spočívá v tom, že přesune nuly regulované soustavy do požadované polohy, kterou definují zadané kořeny čitatele přenosu řízení. Prvky matice stavového regulátoru a prvky matice korekce výstupu jsou konstanty.

Stavové regulátory jsou realizovány nejčastěji jako diskrétní stavové regulátory a to kvůli komplikované struktuře. Čerpáno z (BALÁTEĚ, 2004).

Diskrétní stavový regulátor:



Obr. 3.4 – Obvod s diskrétním stavovým regulátorem

I zde platí vzorce 3.5 a 3.6, které mají tvar

Soustava:

$$\mathbf{x}(k+1) = \mathbf{M} \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{u}(k)$$

$$\mathbf{y}(t) = \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{D} \cdot \mathbf{u}(t)$$

Regulátor:

$$\mathbf{z}(k) = \mathbf{R}\mathbf{x}(k) \tag{3.21}$$

$$\mathbf{u}(k) = \mathbf{w}(k) - \mathbf{z}(k) \tag{3.22}$$

Dále po úpravě se dostává:

$$\mathbf{x}(k+1) = (\mathbf{M} - \mathbf{NR})\mathbf{x}(k) + \mathbf{N}\mathbf{w}(k) \tag{3.23}$$

$$\mathbf{y}(k) = (\mathbf{C} - \mathbf{DR})\mathbf{x}(k) + \mathbf{D}\mathbf{w}(k) \quad (3.24)$$

Čerpáno z (KUPKA, 2013).

3.2.1 Návrh regulátoru v konečném počtu regulačních kroků

Při tomto návrhu regulátoru končí regulační pochody v konečném čase. Jsou obrovské akční zásahy (regulace je velmi rychlá). Obvykle je počet kroků úměrný řádu soustavy. Výhodou návrhu je názornost a jednoduchost.

Princip návrhu:

$$\mathbf{x}(k+1) = \mathbf{M}\mathbf{x}(k) + \mathbf{N}\mathbf{u}(k)$$

$$\mathbf{x}(k+2) = \mathbf{M}\mathbf{x}(k+1) + \mathbf{N}\mathbf{u}(k+1) = \mathbf{M}^2\mathbf{x}(k) + \mathbf{M}\mathbf{N}\mathbf{u}(k) + \mathbf{N}\mathbf{u}(k+1)$$

....

$$\mathbf{x}(k+N) = \mathbf{M}^n\mathbf{x}(k) + \mathbf{M}^{n-1}\mathbf{N}\mathbf{u}(k) + \mathbf{M}^{n-2}\mathbf{N}\mathbf{u}(k+1) + \dots + \mathbf{M}\mathbf{N}\mathbf{u}(k+n-2) + \mathbf{M}^0\mathbf{N}\mathbf{u}(k+n-1)$$

$$\mathbf{x}(k+N) = \mathbf{M}^n\mathbf{x}(k) + \begin{bmatrix} \mathbf{N}, \mathbf{M}\mathbf{N}, \dots, \mathbf{M}^{n-1}\mathbf{N} \end{bmatrix} \cdot \begin{bmatrix} u(k+n-1) \\ u(k+n-2) \\ \dots \\ u(k) \end{bmatrix} \quad (3.25)$$

$\begin{bmatrix} \mathbf{N}, \mathbf{M}\mathbf{N}, \dots, \mathbf{M}^{n-1}\mathbf{N} \end{bmatrix}$ se nazývá matice říditelnosti a označuje se písmenem \mathbf{G} .

Jestliže se jedná o systém SISO a návrh řízení v konečném počtu regulačních kroků pak se dostává

$$\mathbf{M}^n\mathbf{x}(k) + \mathbf{G} \begin{bmatrix} u(k+n-1) \\ u(k) \end{bmatrix} = 0 \quad (3.26)$$

Po úpravě předešlé rovnice

$$\begin{bmatrix} u(k+n-1) \\ \dots \\ u(k) \end{bmatrix} = -\mathbf{G}^{-1}\mathbf{M}^n\mathbf{x}(k) \quad (3.27)$$

Z předešlé rovnice se vyjádří $u(k)$

$$\mathbf{u}(k) = -\begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix} \mathbf{G}^{-1}\mathbf{M}^n\mathbf{x}(k) \quad (3.28)$$

Nyní se již získala matice regulátoru \mathbf{R}

$$\mathbf{R} = -[0 \ 0 \ \dots \ 0 \ 1]\mathbf{G}^{-1}\mathbf{M}^n \quad (3.29)$$

Pro systémy SISO se nazývá **Ackermannova formule**

$$\mathbf{R} = -[0 \ 0 \ \dots \ 0 \ 1]\mathbf{G}^{-1}\Delta(\mathbf{M}) \quad (3.30)$$

$$\Delta\lambda = (\lambda - \lambda_1)(\lambda - \lambda_2)\dots(\lambda - \lambda_n) = \lambda^n + a_{n-1}\lambda^{n-1} + a_1\lambda + a_0 \quad (3.31)$$

$$\Delta\mathbf{M} = \mathbf{M}^n + a_{n-1}\mathbf{M}^{n-1} + \dots + a_1\mathbf{M} + a_0\mathbf{E} \quad (3.32)$$

Pro systémy MIMO se získá

$$\mathbf{R} = -[0 \ 0 \ \dots \ 0 \ \mathbf{E}]\mathbf{G}^{-1}\Delta(\mathbf{M}) \quad (3.33)$$

\mathbf{E} je jednotková matice

Pomocí \mathbf{R} lze ovlivnit všechna vlastní čísla. Čerpáno z (KUPKA, 2013).

Příklad: Navrhněme stavový regulátor v konečném počtu kroků, je-li řízený systém popsán diferenciální rovnicí

$$y' + 3y' + 2y = u \text{ a } T_s = 0,5\text{s}$$

Díky tomu nám vychází matice

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \mathbf{C} = [1 \ 0]; \mathbf{D} = 0$$

Protože známe matici \mathbf{A} , \mathbf{B} a vzorkovací periodu T_s , tak dopočítáme matice \mathbf{M} a \mathbf{N} , které vychází takto

$$\mathbf{M} = \begin{bmatrix} 0,845 & 0,239 \\ -0,477 & 0,129 \end{bmatrix}; \mathbf{N} = \begin{bmatrix} 0,077 \\ 0,239 \end{bmatrix}$$

Jelikož chceme regulaci v konečném počtu kroků, tak se $\lambda = \lambda_1 = \lambda_2 = 0$

$$\Delta\mathbf{M} = \mathbf{M}^2 = \begin{bmatrix} 0,6 & 0,233 \\ -0,465 & 0,097 \end{bmatrix}; \mathbf{G} = [\mathbf{N} \ \mathbf{NM}] = \begin{bmatrix} 0,077 & 0,122 \\ 0,239 & -0,006 \end{bmatrix}$$

$$\mathbf{G}^{-1} = \begin{bmatrix} 0,206 & 4,118 \\ 8,069 & -2,559 \end{bmatrix}$$

Nyní máme všechno pro výpočet regulátoru R

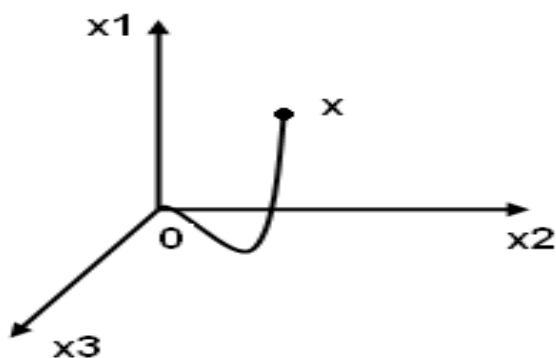
$$R = -[0 \ 1]G^{-1}M^2 = -[8,069 \ 2,599] \begin{bmatrix} 0,6 & 0,233 \\ -0,465 & 0,097 \end{bmatrix} = \begin{bmatrix} -6,05 \\ -1,627 \end{bmatrix}$$

Výsledná rovnice má tedy tvar

$$u(k) = Rx(k) = -6,05x_1(k) - 1,627x_2(k)$$

3.2.2 Návrh regulátoru podle kvadratického kritéria

Využívá se Riccatiho rovnice, která se hodí pro systémy s jedním vstupem a jedním výstupem. Cílem je najít posloupnost regulačních (akčních) zásahů, pomocí kterých se z libovolného stavu stavového prostoru dostaneme do počátku v minimálním počtu kroků.



Obr. 3.5 – Ukázka návrhu

Formálně upravené kvadratické kritérium

$$J = \sum_{j=1}^{N-1} (x^T(j)Qx(j) + u^T(j)Lu(j) + x^T(N)Px(N)) \quad (3.34)$$

Tato rovnice se musí minimalizovat.

Q , L a P jsou symetrické, pozitivně definitní matice (vlastní čísla jsou kladná). Matice L je pro systémy s jedním vstupem a jedním výstupem skalárem. Pomocí matic Q , L a P ovlivňujeme rychlost a kvalitu regulačních pochodů.

Pro návrh regulátoru touto metodou lze využít prostředí Matlab, kde tuto metodu lze řešit pomocí příkazu *dlqr*. Tento příkaz využívá k řešení toto kritérium

$$J = \sum_{j=0}^{\infty} (x^T(j)Qx(j) + u^T(j)Lu(j) + 2 \cdot x^T(j)Pu(j)) \quad (3.35)$$

Tato rovnice se musí minimalizovat. Čerpáno z (KUPKA, 2013).

Příklad: Navrhněte diskretní stavový regulátor minimalizací kvadratické regulační plochy zajišťující $e(\infty) = 0$, je-li řízený systém popsán diferenciální rovnicí

$$y'' + 3y' + 2y = u \text{ a } T_s = 0,5\text{s}$$

Díky tomu nám vychází matice

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \mathbf{C} = [1 \quad 0]; \mathbf{D} = 0$$

Máme spojitý stavový popis, ale my se potřebujeme dostat na diskretní stavový popis. Protože známe matici \mathbf{A} , \mathbf{B} a T_s , tak dopočítáme matice \mathbf{M} a \mathbf{N} , které vychází takto

$$\mathbf{M} = \begin{bmatrix} 0,845 & 0,239 \\ -0,477 & 0,129 \end{bmatrix}; \mathbf{N} = \begin{bmatrix} 0,077 \\ 0,239 \end{bmatrix}; \mathbf{C} = [1 \quad 0]$$

\mathbf{D} je nulové a tak s ním dále ve výpočtech nepočítáme. Nyní se musíme dostat na rozšířený stavový popis, v něm jsou matice \mathbf{M}_R , \mathbf{N}_R a \mathbf{C}_R . Rozšířený stavový popis má tento tvar

$$\mathbf{x}_R(k+1) = \mathbf{M}_R \mathbf{x}_R(k) + \mathbf{N}_R \mathbf{u}(k)$$

$$\mathbf{y}(k) = \mathbf{C}_R \mathbf{x}_R(k)$$

Hodnoty \mathbf{M}_R , \mathbf{N}_R a \mathbf{C}_R spočítáme následovně

$$\mathbf{M}_R = \begin{bmatrix} \mathbf{M} & \mathbf{s} \\ \mathbf{C}\mathbf{M} & 1 \end{bmatrix} = \begin{bmatrix} 0,845 & 0,239 & 0 \\ -0,497 & 0,129 & 0 \\ 0,845 & 0,239 & 1 \end{bmatrix}$$

$$\mathbf{N}_R = \begin{bmatrix} \mathbf{N} \\ \mathbf{C}\mathbf{N} \end{bmatrix} = \begin{bmatrix} 0,077 \\ 0,239 \\ 0,077 \end{bmatrix}; \mathbf{C}_R = [\mathbf{C} \quad 0] = [1 \quad 0 \quad 0]$$

\mathbf{s} – dvou složkový sloupcový vektor nul

Tvrký regulátor – regulátor bez penalizace $u \rightarrow L = 0$

Matice \mathbf{QI} a \mathbf{P} volíme takto

$$\mathbf{QI} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \mathbf{P} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

V \mathbf{P} volíme hodnotu 100 k rychlému zklidnění regulačního pochodu

$$\mathbf{J} = \sum_{j=1}^{N-1} (x_3^2(j) + 100x_2^2(N)) \rightarrow \text{minimalizujeme}$$

$$\mathbf{R} = [10,92 \quad 3,08 \quad 12,92]$$

$$\mathbf{u}(k) = -10,92x_1(k) - 3,08x_2(k) - 12,92x_3(k)$$

Měkký regulátor – regulátor s výraznou penalizací $u \rightarrow L = 1$

Hodnoty matic \mathbf{QI} a \mathbf{P} jsou stejné jako u tvrdého regulátoru

$$\mathbf{J} = \sum_{j=1}^{N-1} (x_3^2(j) + u^2(j)) + 100x_2^2(N) \rightarrow \text{minimalizujeme}$$

$$\mathbf{R} = [1,607 \quad 0,603 \quad 0,840]$$

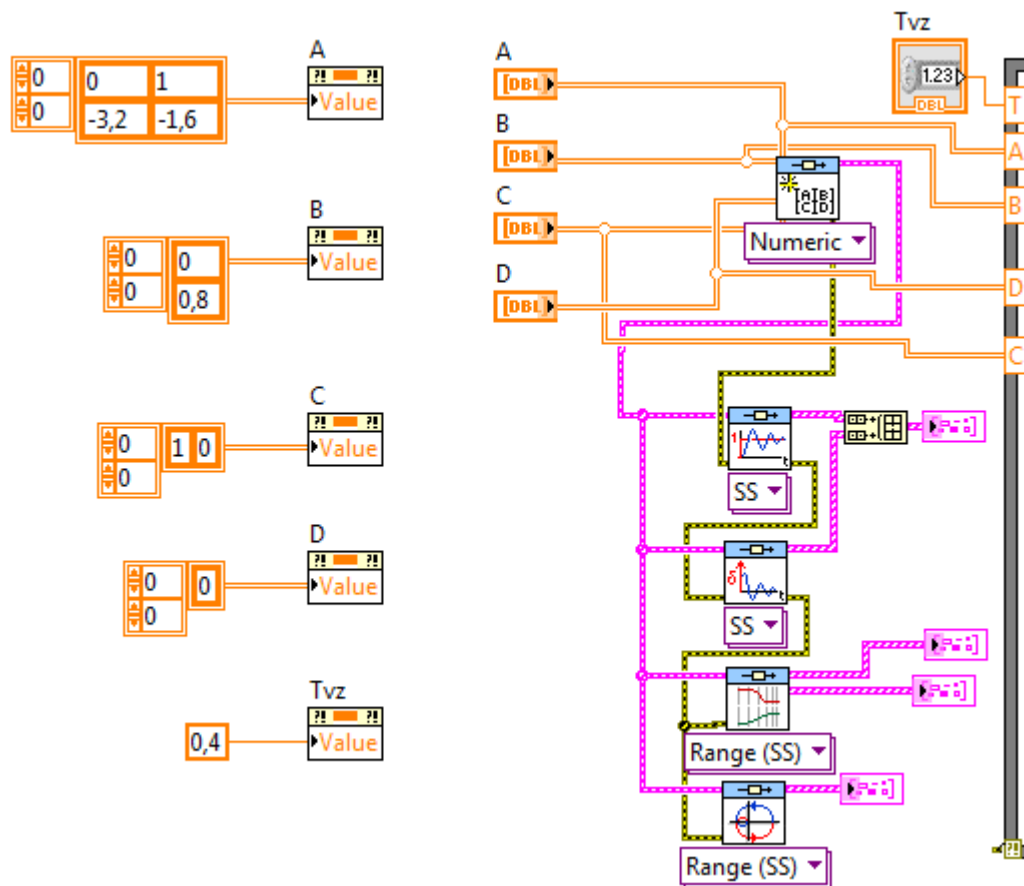
$$\mathbf{u}(k) = -1,607x_1(k) - 0,603x_2(k) - 0,840x_3(k)$$

4 Tvorba aplikace v LabVIEW

4.1 Vykreslení charakteristik v aplikaci

Dynamické procesy budou zadávány do aplikace v LabVIEW formou matic A , B , C a D , které jsou přiváděny do bloku CD Construct State-Space Model, který vytvoří potřebný model pro další práci s procesem. Pokud se přidá matice do blokového diagramu, tak se přidá i na čelní panel a tak uživatel bude mít rychlý přístup k zadání těchto matic. Malým problémem je, že si program LabVIEW nedokáže uložit hodnoty těchto matic, takže vždy, když se znova spouští LabVIEW a pak tato aplikace, tak se musí zadávat znovu i hodnoty matic A , B , C a D . Tento problém byl vyřešen, přidáním ke každé matici a také ke vzorkovací frekvenci prvku Property Node, který je nastavený na hodnotu. Když se do schématu Property Node přidá, tak je brán jako vstup a tak je potřeba ho otočit na výstup. Poté již stačí na jeho vstupu vytvořit konstantu, kam se zadají hodnoty matic a vzorkovací periody. Dále tu je jeden problém s Matlab skriptem, u kterého LabVIEW předpokládá, že z něj půjdou vždy pouze výstupy, ale v případě této aplikace má i vstupy a s tím je v LabVIEW problém. Při prvním spuštění programu se plní velikosti matic a vzorkovací periody, ale zároveň probíhá i zbytek programu, a jelikož se ještě před plněním provádí výpočty v Matlab skriptu, tak program vypisuje chybu, že nesouhlasí velikosti matic. Při druhém spuštění jsou již matice naplněny a tak program bez problému proběhne a vše je jak má. Tento problém byl řešen pomocí Flat Sequence i pomocí Stacked Sequence, což jsou cykly, pomocí kterých uživatel zadá, co chce, aby se v programu řešilo nejdříve. Tyto možnosti byly nalezeny na internetu i v ukázkových příkladech LabVIEW, ale i s těmito prvky se stále Matlab skript spouští jako první a vypisuje uvedenou chybu. Tento problém nebyl tedy, i přes velkou snahu, vyřešen a aplikace pracuje až na druhé spuštění, při prvním hlásí chybu, že nesedí velikosti matic. Pro výpočty estimátoru a stavového regulátoru jsou potřeba matice A , B , C a D i vzorkovací perioda, tak jsou tyto hodnoty rovnou přivedeny i na vstup MATLAB skriptu, který bude právě pro výpočet diskrétního estimátoru a stavového regulátoru využit. V LabVIEW, když je vytvořený model, tak již lehce je možné vykreslit charakteristiky, které jsou žádané, neboť bloky pro charakteristiky mají stejný datový typ, stejně jako vytvořený model. Bloky pro tvorbu charakteristik jsou vybrány z knihovny Control Design a poté z podknihoven Time Response a Frequency Response. V aplikaci jsou bloky pro vykreslení přechodové, impulzní, amplitudo-frekvenční a amplitudo-fázové

charakteristiky. Frekvenční a impulzní charakteristika jsou dány do jednoho grafu. Pro správnou funkci jsou bloky propojeny i chybovým signálem.



Obr. 4.1 – Blokové schéma část 1

4.2 Matlab skript

Nejdůležitější část celé aplikace je právě Matlab skript. Jak je popsáno výše, tato aplikace se neobejde bez výpočtového programu a nejjednodušší je využít Matlab skript, neboť LabVIEW podporuje spolupráci s Matlabem a tak je lepší skript přidat přímo do aplikace. Pracovat s výsledky, které skript vypočte přímo v LabVIEW a nemusí se výsledky složitě přepisovat z jiného programu do LabVIEW. Vstupy skriptu jsou matice A , B , C a D a také vzorkovací perioda T_s , všechny tyto hodnoty má uživatel možnost nastavit na čelním panelu. Ve skriptu se musí převést spojité stavový popis do diskretního stavového popisu, který má tvar zápisu

$$\mathbf{x}(k+1) = \mathbf{M} \cdot \mathbf{x}(k) + \mathbf{N} \cdot \mathbf{u}(k)$$

$$\mathbf{y}(t) = \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{D} \cdot \mathbf{u}(t)$$

Z toho je vidět, že jsou neznámé matice M a N a ty je potřeba spočítat. Když jsou známé matice A , B a vzorkovací perioda T_s , tak se M a N spočítá pomocí vztahů

$$M = e^{AT}; N = A^{-1}(M - E)B$$

Těmto vzorcům odpovídají v Matlabu příkazy

M=expm(A*T);

N=inv(A)*(M-[1 0; 0 1])*B;

Pro diskrétní estimátor je rovnice

$$\hat{x}(k+1) = M_E \cdot \hat{x}(k) + N_E \cdot u(k) + H_E \cdot y(k)$$

Z této rovnice jsou neznámé M_E , N_E a H_E . K jejich určení je nutné nejprve určit Q . Poté, co se získá Q , tak lze dopočítat M_E , N_E a H_E z těchto vzorců

$$M_E = M_{22} - QM_{12}; N_E = N_2 - QN_1$$

$$H_E = M_{21} + M_{22}Q - QM_1 - QM_{12}Q$$

Těmto vzorcům v Matlabu odpovídají tyto zápisy

He=M(2,1)+M(2,2)*Q-Q*M(1,1)-Q*M(1,2)*Q;

Me=M(2,2)-Q*M(1,2);

Ne=N(2)-Q*N(1);

Pro výpočet hodnot stavového regulátoru je převeden diskrétní stavový popis na rozšířený diskrétní stavový popis.

Hodnoty M_r , N_r a C_r jsou počítány ve skriptu následovně

Mr=[M E;C*M 1];

Nr=[N;C*N];

Cr=[C 0];

Když jsou spočítané tyto hodnoty, tak lze spočítat hodnotu vektoru G_r , kde jeho první složka je rovna N_r , druhá se spočítá $M_r N_r$ a třetí hodnota se spočítá $M_r^2 N_r$. Ve skriptu je tento výpočet zapsán následovně

Gr=[NrMr*Nr Mr^2*Nr].

Ve skriptu jsou také napsány hodnoty pro měkký (tvrdý) regulátor $L = 0$ ($L = 1$) a hodnoty matic QI a P , které jsou připravené pro výpočet stavového regulátoru pomocí kvadratického kritéria. Matice QI je definována

$$QI = [0 \ 0 \ 0; 0 \ 0 \ 0; 0 \ 0 \ 1]$$

a matice \mathbf{P} takto

$$\mathbf{P} = [0 \ 0 \ 0; 0 \ 100 \ 0; 0 \ 0 \ 0].$$

Ve skriptu jsou vypracovány dvě metody pro výpočet stavového regulátoru. Jedná se o metodu v konečném počtu regulačních kroků a taky o metodu výpočtu hodnot stavového regulátoru podle kvadratického kritéria.

Výpočet hodnot regulátoru v konečném počtu regulačních kroků je definován

$$\mathbf{R} = - [0 \ 0 \ 1] g_r^{-1} M_r^3.$$

První vektor byl ve skriptu zadán v proměnné \mathbf{EI} a tak zápis ve skriptu vypadá

$$\mathbf{Rr} = - (\mathbf{EI}) * (\mathbf{G}r^{-1}) * (\mathbf{M}r^3).$$

Pro metodu podle kvadratického kritéria se využije pro řešení cyklus for

for i=1:1:10

$$\mathbf{P}_i = \mathbf{Q}1 + \mathbf{M}r' * \mathbf{P} * \mathbf{M}r - \mathbf{M}r' * \mathbf{P} * \mathbf{N}r * \text{inv}(\mathbf{L} + \mathbf{N}r' * \mathbf{P} * \mathbf{N}r) * \mathbf{N}r' * \mathbf{P} * \mathbf{M}r;$$

$$\mathbf{R} = \text{inv}(\mathbf{L} + \mathbf{N}r' * \mathbf{P} * \mathbf{N}r) * \mathbf{N}r' * \mathbf{P} * \mathbf{M}r$$

$$\mathbf{P} = \mathbf{P}_i;$$

end

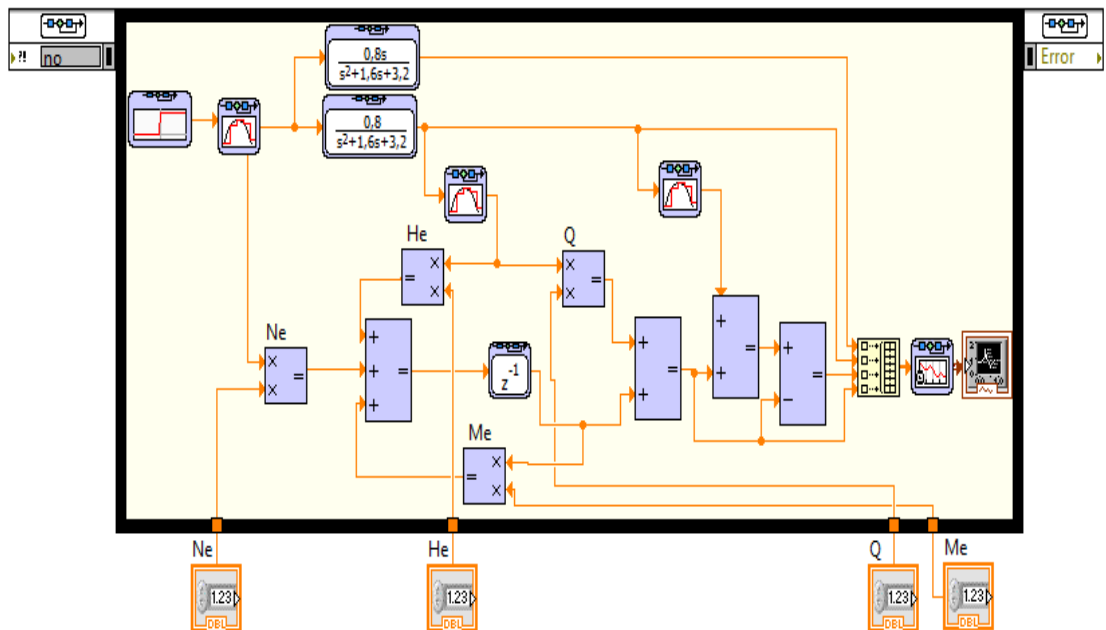
Poté co proběhne cyklus, jsou hodnoty stavového regulátoru v proměnné \mathbf{R} , ale bylo by dobré je mít ve stejné proměnné jako v předchozí metodě, tedy v proměnné \mathbf{Rr} . Při dávání hodnoty do \mathbf{Rr} se nesmí zapomenout změnit znaménka. Nyní vždy, když jsou spočítány hodnoty stavového regulátoru jednou z metod, jsou výsledné hodnoty v proměnné \mathbf{Rr} s kterou se bude dále pracovat.

Hodnoty regulátoru jsou poté samostatně dány do nové konstanty pro přehlednější připojení do schématu stavového regulátoru. Zde je potřeba si dát pozor, aby se nezapomnělo změnit znaménko u třetí hodnoty stavového regulátoru. Tyto nové konstanty jsou poté dány na výstup skriptu a jsou přiváděny do obvodu se stavovým regulátorem, který bude popsán níže.

Výstupem ze skriptu je i vektor vypočítaných hodnot stavového regulátoru, který je vyvedený tak, aby se zobrazoval uživateli při spuštění programu na hlavní stránce a on mohl hned vidět, jakých hodnot nabývá stavový regulátor.

Výstupem jsou také hodnoty, které jsou potřebné pro realizaci diskrétního estimátoru. Jedná se o hodnoty \mathbf{Q} , \mathbf{H}_E , \mathbf{M}_E a \mathbf{N}_E .

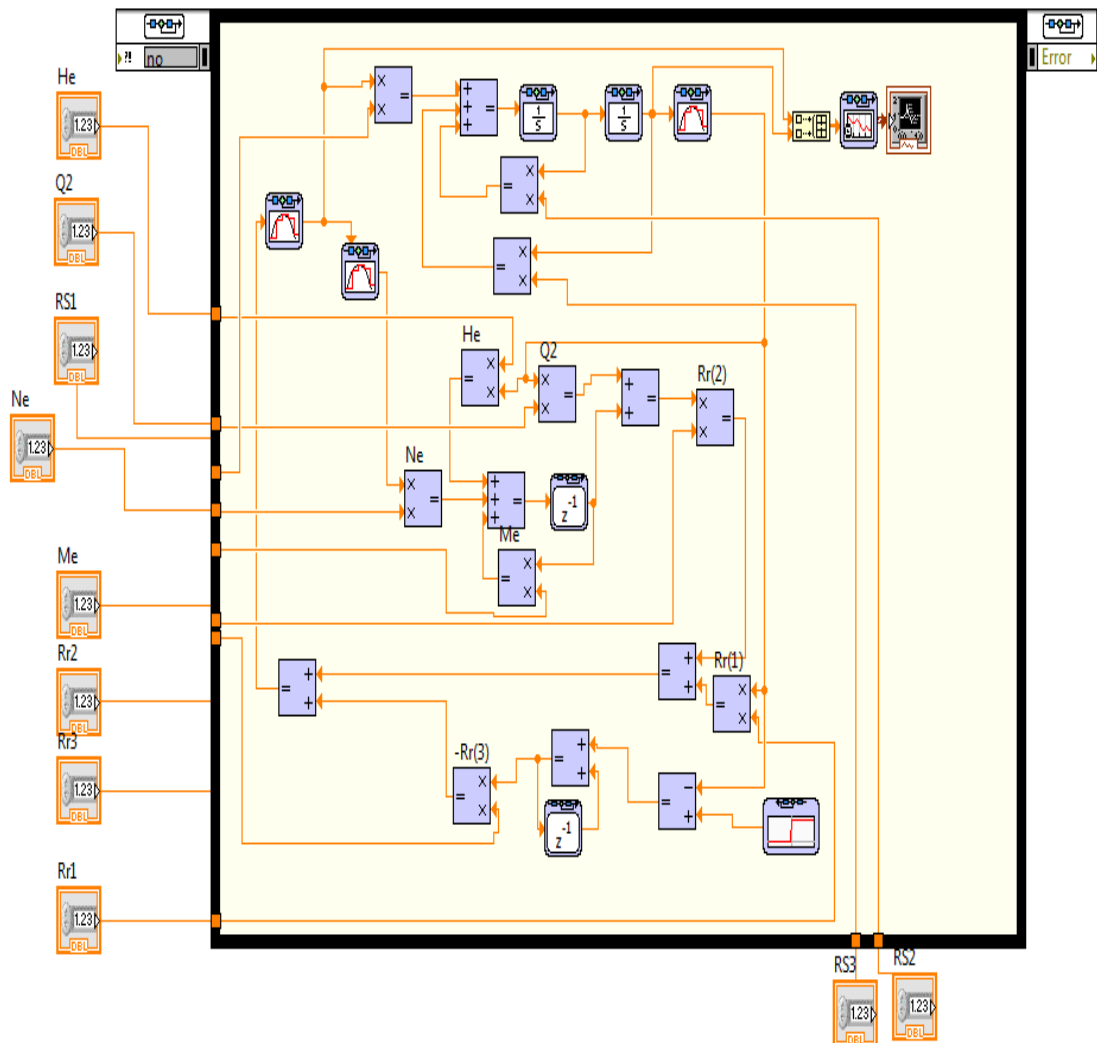
4.3 Zapojení diskretního estimátoru



Obr. 4.2 – Zapojení s diskretním estimátorem

Jak je realizované zapojení diskretního estimátoru je vidět na obr. 4.2. Hodnoty Q , H_E , M_E a N_E , které jsou pod smyčkou Control & Simulation Loop 2, jsou ty hodnoty, které jsou vyvedeny z Matlabu a pro větší přehled jsou zde dány jako konstanty, jinak v celkové aplikaci jsou přímo vedeny z Matlab skriptu, což bude vidět na obrázku, kde bude zobrazena celá aplikace. Uživatel si zde musí měnit přenosy neboť v LabVIEW není možné zadávat přenosy automaticky, neboť tento datový typ není od LabVIEW podporovaný. Výstupem tohoto schéma je přechodová a impulsní charakteristika s diskretním estimátorem. V případě, že budou dobře zvolené parametry, tak by průběh s diskretním estimátorem měl být hodně podobný průběhu přechodové a impulsní charakteristiky. Jak je vidět z obr. 4.2, zapojení je v LabVIEW tvořeno hodně podobně jako v programu Matlab. Malá nevýhoda LabVIEW je, že při výpočtu v Matlab skriptu si prostředí nepamatuje, jaké hodnoty vypočítané proměnné nabývají a tak musí být všechny proměnné ze skriptu vyvedeny do tohoto zapojení a to je řešeno pomocí násobení, kdy aktuální signál je násobený právě hodnotou, kterou vypočítal skript a tak se dostaneme na požadovanou hodnotu.

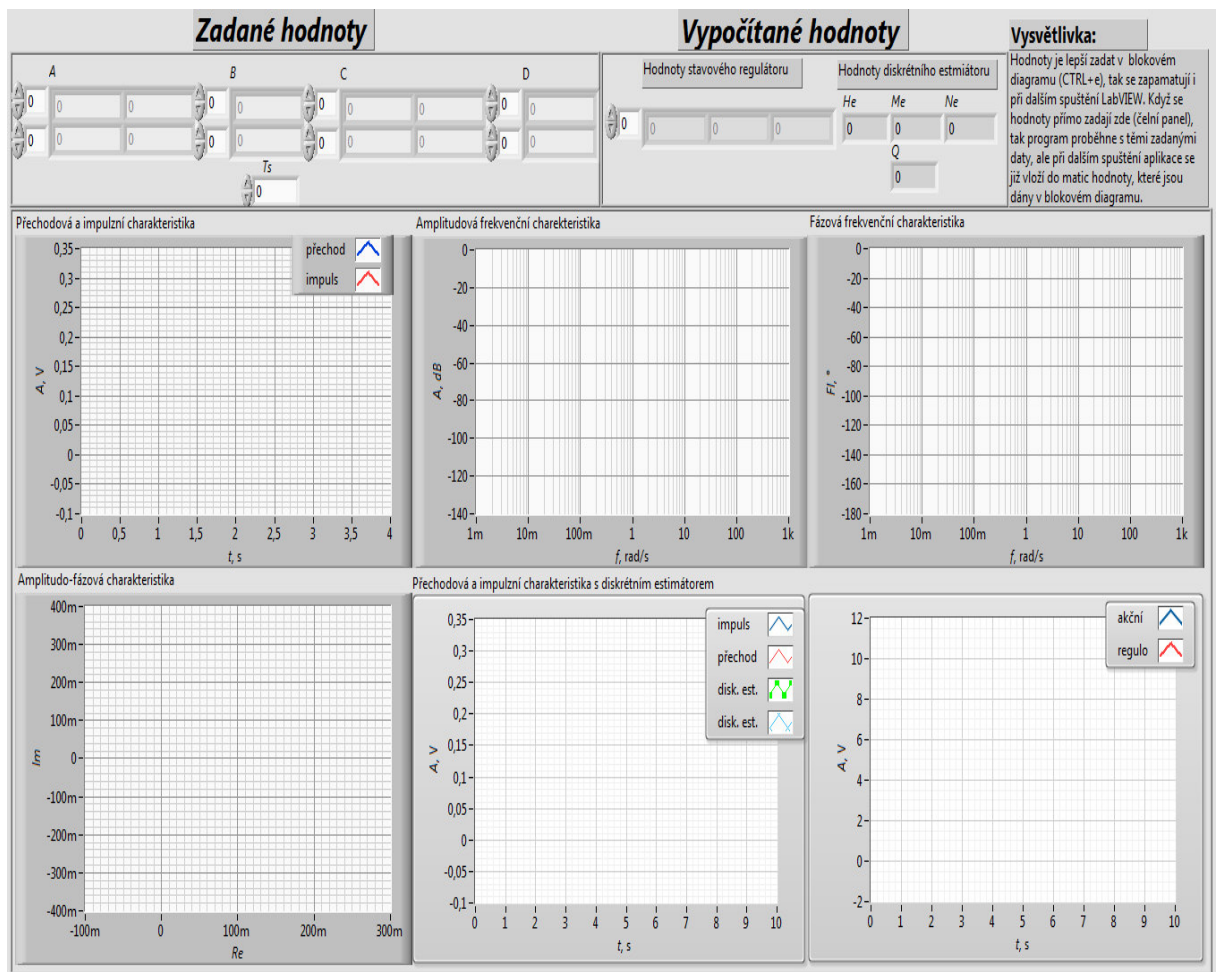
4.4 Regulační obvod s diskretním estimátorem a stavovým regulátorem



Obr. 4.3 – Regulační obvod

Stejně jako u zapojení s diskretním estimátorem i zde jsou hodnoty, které jsou přiváděny přímo z Matlabu zadány pro větší přehlednost jako konstanty. U tohoto zapojení se jedná opět o Q , H_E , M_E a N_E a pak také o vypočítané hodnoty stavového regulátoru, které jsou pojmenovány $Rr1$, $Rr2$ a $Rr3$. Regulační obvod je složený ze spojitě regulované soustavy, diskretního estimátoru a stavového regulátoru. U tohoto zapojení není potřeba při změně přenosu nic měnit, vše je řešeno stejně jako v zapojení výše. Tímto způsobem je řešení i zadávání spojitě regulované soustavy, takže když uživatel změní přenos, tak po spuštění programu se vše přepočítá a na výstupu dostaneme průběh stavového regulátoru s novými parametry.

4.5 Čelní panel aplikace



Obr. 4.4 – Čelní panel aplikace

Vzhled čelního panelu aplikace je na obr. 4.4. Čelní panel se skládá ze tří částí. Jedna část je zadávací, druhá ukazuje vypočítané hodnoty a ve třetí jsou průběhy zadaného systému. Na čelním panelu zadáváme matice A , B , C , D a vzorkovací periodu T_s . Neboť by se muselo při každém dalším spuštění LabVIEW zadávat hodnoty A , B , C , D a T_s , tak to je ošetřeno v blokovém diagramu, kde je možné matice naplnit tak, že si LabVIEW bude při spuštění programu hodnoty pamatovat, což dělá problém při zadání matic na čelním panelu. Při prvním spuštění se program provede s hodnotami, které byly zadány na čelním panelu, takže všechny průběhy jsou získány, ale při dalším spuštění se hodnoty zadané na čelním panelu přepíše hodnotami, které jsou zadány v blokovém diagramu a proto je lepší matice zadat v blokovém diagramu. Ve výpočtové části jsou po spuštění programu vypočteny hodnoty stavového regulátoru, velikost parametru Q – přidání stupně volnosti, pomocí kterého jsou dopočítány hodnoty diskrétního estimátoru (H_e , M_e , N_e). Všechny dříve uvedené hodnoty se

zobrazí na čelním panelu a tak má uživatel hned přehled o tom jak hodnoty vyšly. Poslední částí je část, kde jsou zobrazeny průběhy zadaného systému. Čelní panel obsahuje šest grafů, v kterých se zobrazují následující tyto charakteristiky. V prvním grafu se zobrazuje přechodová a impulzní charakteristika. V druhém grafu je průběh amplitudové – frekvenční charakteristiky. Třetí graf zobrazuje fázovou frekvenční charakteristiku. Ve čtvrtém grafu je průběh amplitudo – fázové charakteristiky. Ke všem předešlým grafům není potřeba Matlab skript a tak se průběhy vykreslí i v případě, že uživatel nemá v počítači nainstalovaný program Matlab. Poslední dva grafy se již bez programu Matlab nezobrazí, neboť ve skriptu jsou vypočítávány hodnoty, které se zadávají do schémat pro příslušné grafy. V pátém grafu se zobrazí průběh přechodové a impulzní charakteristiky, a průběh těchto charakteristik získaných pomocí diskrétního estimátoru. V posledním grafu se zobrazuje průběh regulace se stavovým regulátorem.

5 Výsledky

5.1 Použité systémy

V této práci je použit systém, který je nazván systém1, s přenosem

$$Gs1(s) = \frac{0,8}{s^2 + 1,6s + 3,2}.$$

Matice u tohoto systému vychází takto

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -3,2 & -1,6 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0 \\ 0,8 \end{bmatrix}; \mathbf{C} = [1 \quad 0]; \mathbf{D} = 0.$$

Vzorkovací perioda u tohoto systému bude 0,4 sekund.

Dalším systémem je systém2 s přenosem

$$Gs2(s) = \frac{8}{s^2 + 4s + 16,3}.$$

Matice tohoto systému jsou

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -16,3 & -4 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0 \\ 8 \end{bmatrix}; \mathbf{C} = [1 \quad 0]; \mathbf{D} = 0.$$

Vzorkovací perioda tohoto systému bude stejně jako u prvního systému 0,4 sekund.

5.2 Vypočítané hodnoty

Pro systém1 aplikace vypočítala hodnotu parametru $Q = -1,349$. Když se dosadí tato hodnota do rovnic pro výpočet parametrů diskrétního estimátoru, tak hodnoty He , Me a Ne vycházejí takto $He = -0,775$, $Me = 0,731$ a $Ne = 0,284$.

Hodnoty stavového regulátoru vycházejí při použití kvadratického regulátoru s tvrdým regulátorem ($L = 0$)

$$\mathbf{R} = [-15,927 \quad -5,401 \quad -19,927].$$

Pro měkký regulátor ($L = 1$) vychází hodnota regulátoru

$$\mathbf{R} = [-0,823 \quad -0,675 \quad -0,865].$$

Když použijeme kritérium v konečném počtu regulačních kroků, tak se hodnoty změní

$$\mathbf{R} = [-11,961 \quad -4,662 \quad -11,037].$$

Pro systém2 se získala hodnota parametru $Q = 1,405$. Diskrétní estimátor má potom své hodnoty rovny $He = -3,019$, $Me = -0,355$ a $Ne = 0,547$.

Hodnoty stavového regulátoru při použití kvadratického regulátoru s tvrdým regulátorem ($L = 0$) vycházejí

$$\mathbf{R} = [-0,993 \ -0,383 \ -3,081].$$

Pro měkký regulátor ($L = 1$) vychází hodnota regulátoru

$$\mathbf{R} = [-0,146 \ -0,103 \ -0,788].$$

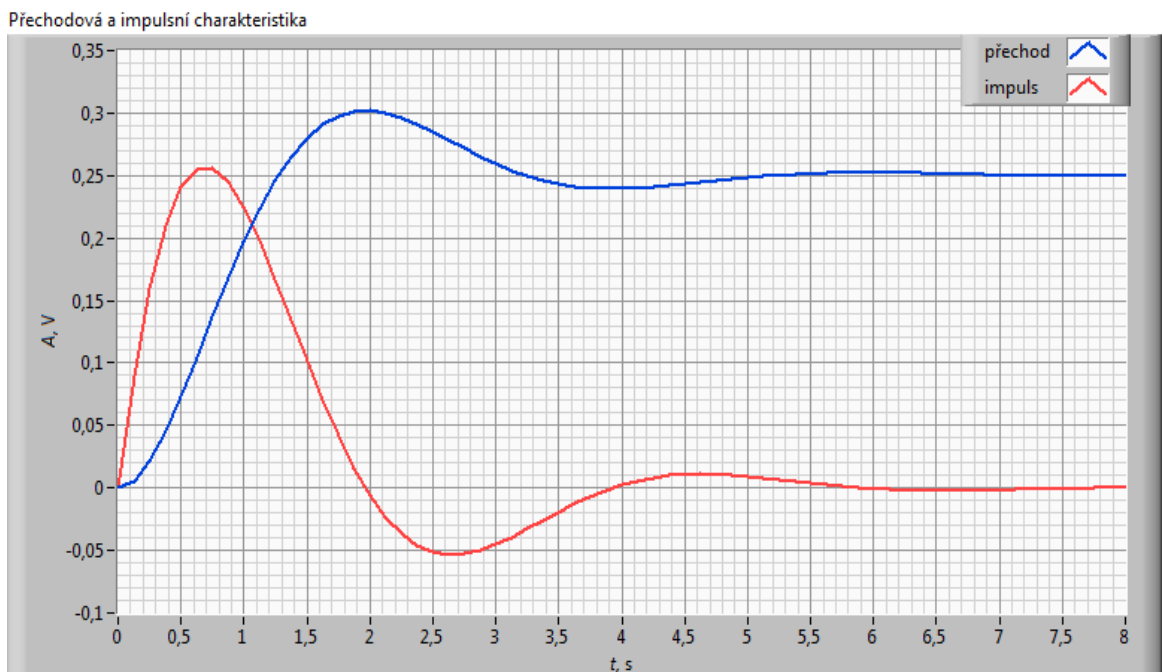
Když použijeme pro výpočet stavového regulátoru kritérium v konečném počtu kroků regulace, tak hodnoty vychází pro regulátor

$$\mathbf{R} = [-0,599 \ -0,310 \ -1,937].$$

5.3 Výsledné průběhy

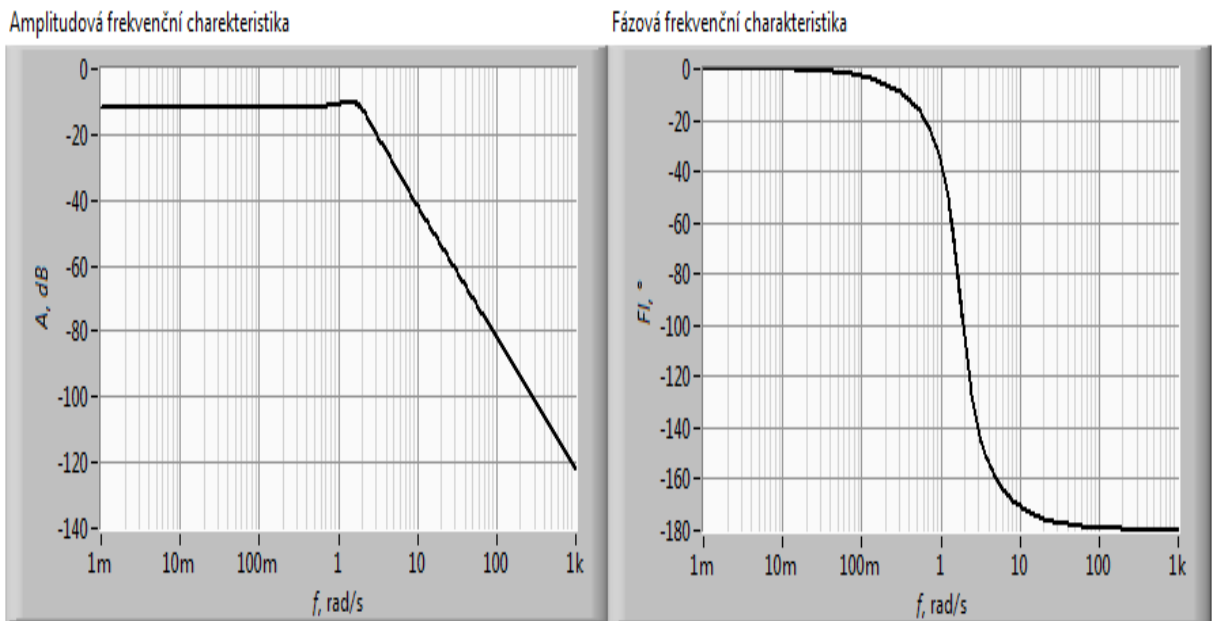
5.3.1 Systém1

Na obr. 5.1 je zobrazena přechodová a impulzní charakteristika. Je vidět, že systém se ustálí za 5,5 sekundy.

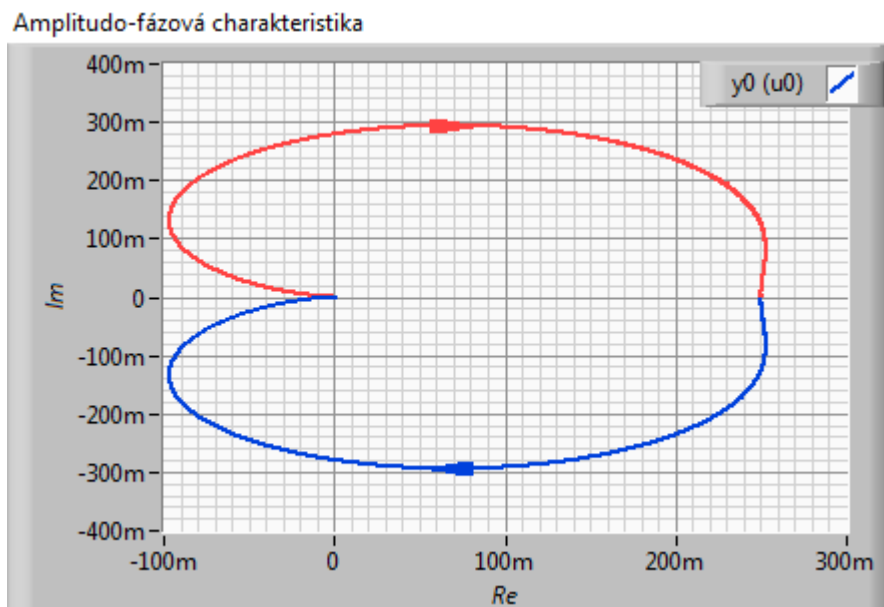


Obr. 5.1 – Systém1 – přechodová a impulzní charakteristika

Na obr. 5.2 jsou zobrazeny Bodeho charakteristiky uvedeného systému a na obr. 5.3 je zobrazena Nyquistova charakteristika.

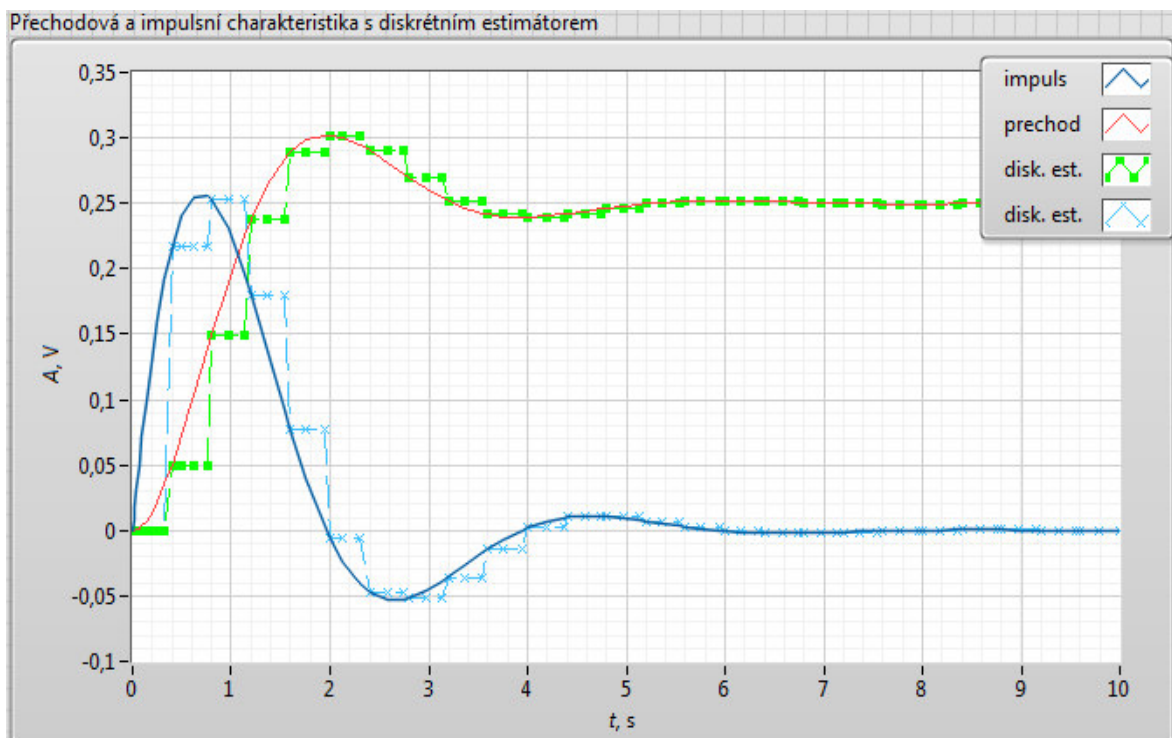


Obr. 5.2 – Systém1 – Bodeho charakteristiky

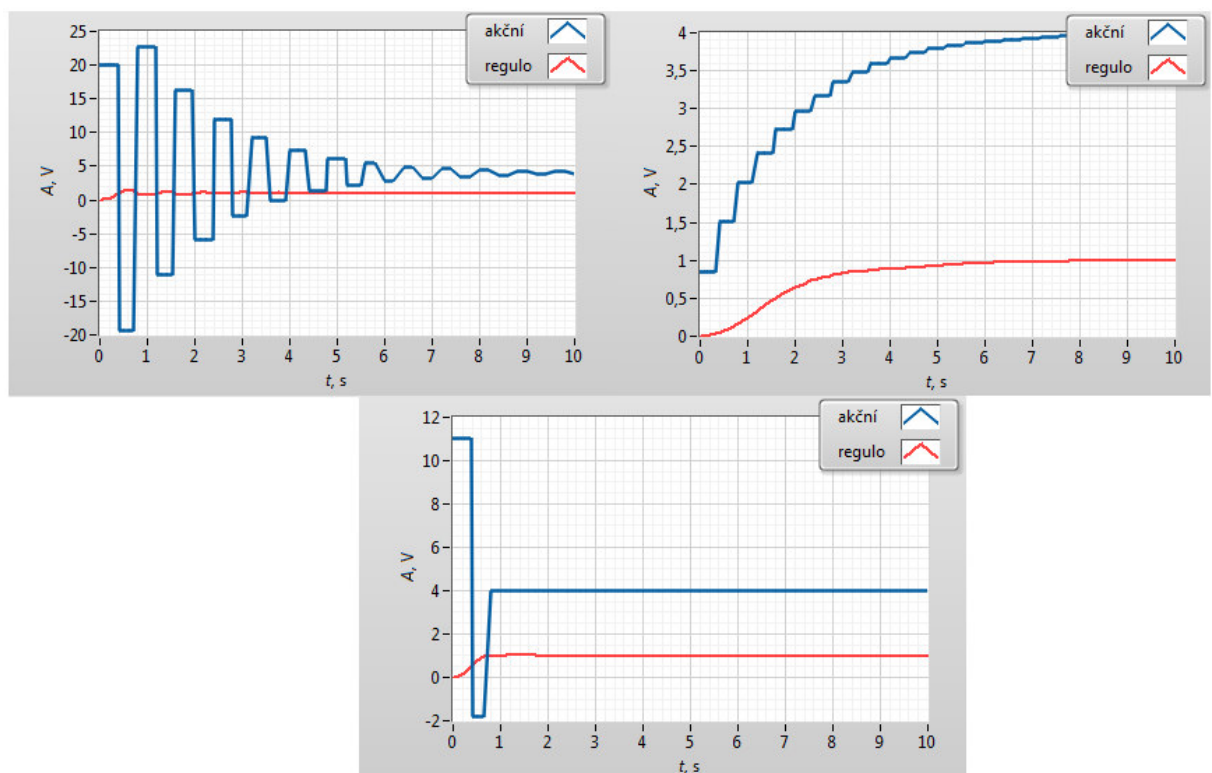


Obr. 5.3 – Systém1 – Nyquistova charakteristika

Na obr. 5.4 jsou zobrazeny průběhy přechodové a impulzní charakteristiky a také průběhy těchto charakteristik s diskretním estimátorem a z obrázku je patrné, že vypočítané hodnoty diskretního estimátoru jsou velice dobré a průběh diskretního estimátoru kopíruje průběh charakteristik.



Obr. 5.4 – Systém1 – charakteristiky s diskrétním estimátorem



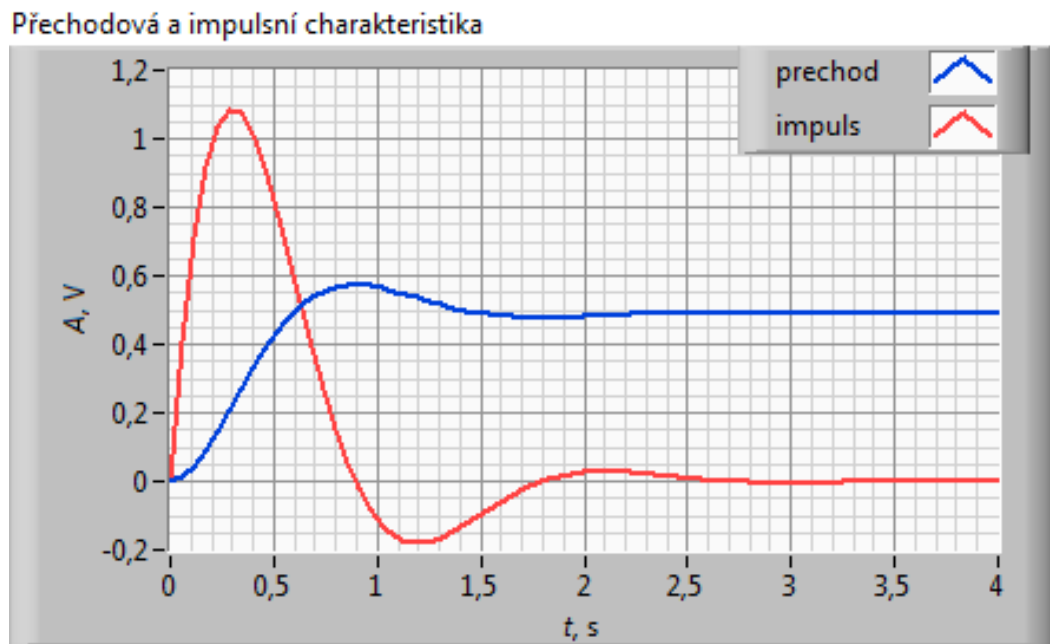
Obr. 5.5 – Systém1 – průběhy se stavovým regulátorem

Na obr. 5.5 jsou průběhy stavového regulátoru – pro všechny případy, byla shodně zvolena reakce na žádanou hodnotu $w = 1$ v čase 0. Regulátory jsou navrženy tak, aby eliminovaly vliv trvalé regulační odchylky. Na prvním obrázku je průběh stavového regulátoru, který je

vypočítán pomocí kvadratického kritéria; nejprve byl zvolen tvrdý regulátor. Jak je vidět z průběhu, tato regulace nebude vhodná pro použití, neboť průběh je kmitavý. Druhý průběh je pro měkký regulátor, kde se regulace sice ustálí, ale za dlouhou dobu. Třetí průběh je podle kritéria v konečném počtu regulačních kroků. Regulátor v tomto případě splňuje svou funkci, protože po konečném počtu kroků skutečně drží regulovanou hodnotu na hodnotě žádané s nulovou regulační odchylkou. V tomto případě by bylo nejlepší zvolit regulátor, který je vypočítán pomocí kritéria v konečném počtu regulačních kroků.

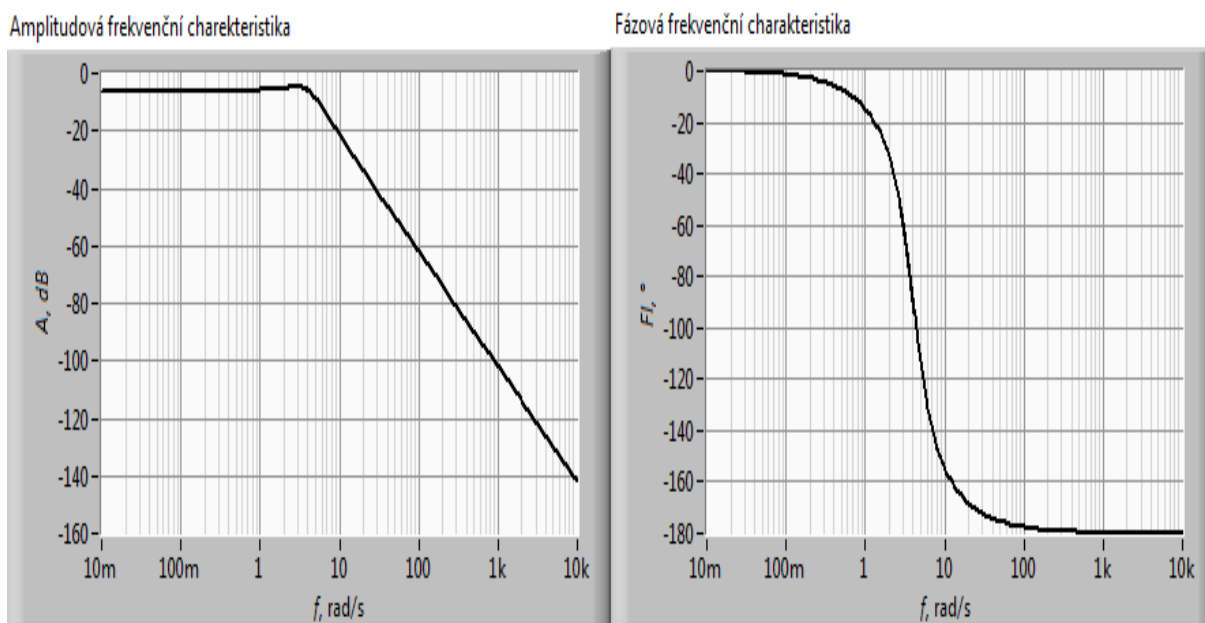
5.3.2 Systém2

Stejně jako u prvního systému bude nejprve vykreslena přechodová a impulzní charakteristika, tyto charakteristiky jsou zobrazeny na obr. 5.6. Z obrázku je patrné, že se systém ustálí okolo 2,5 sekund.

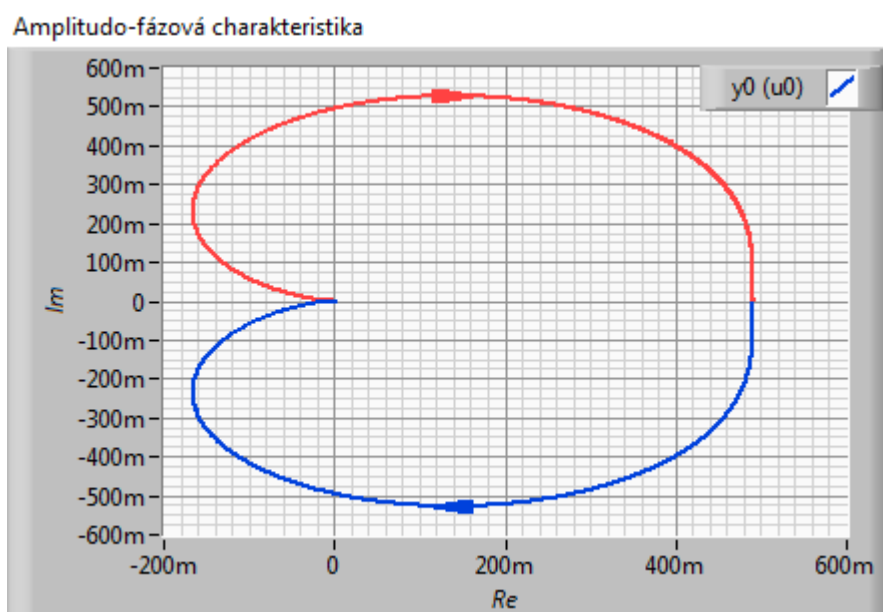


Obr. 5.6 – Systém2 – přechodová a impulzní charakteristika

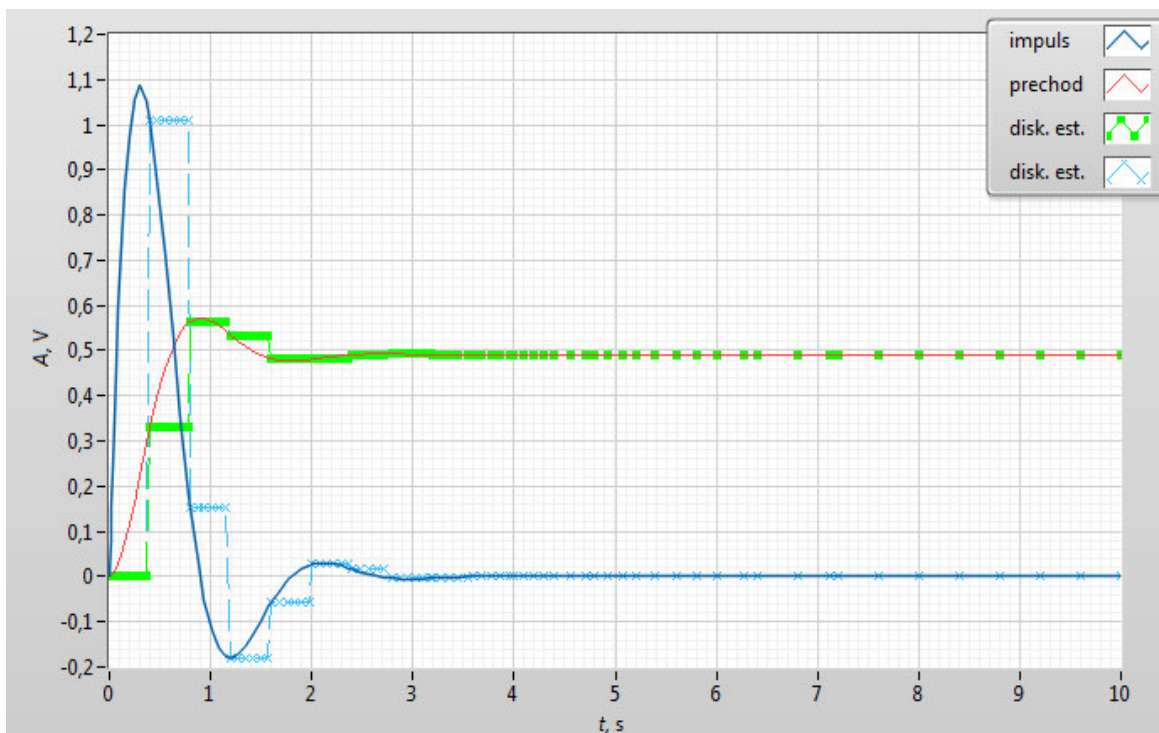
Na obr. 5.7 jsou zobrazeny Bodeho charakteristiky systému a na obr. 5.8 je charakteristika Nyquistova.



Obr. 5.7 – Systém2 – Bodeho charakteristiky

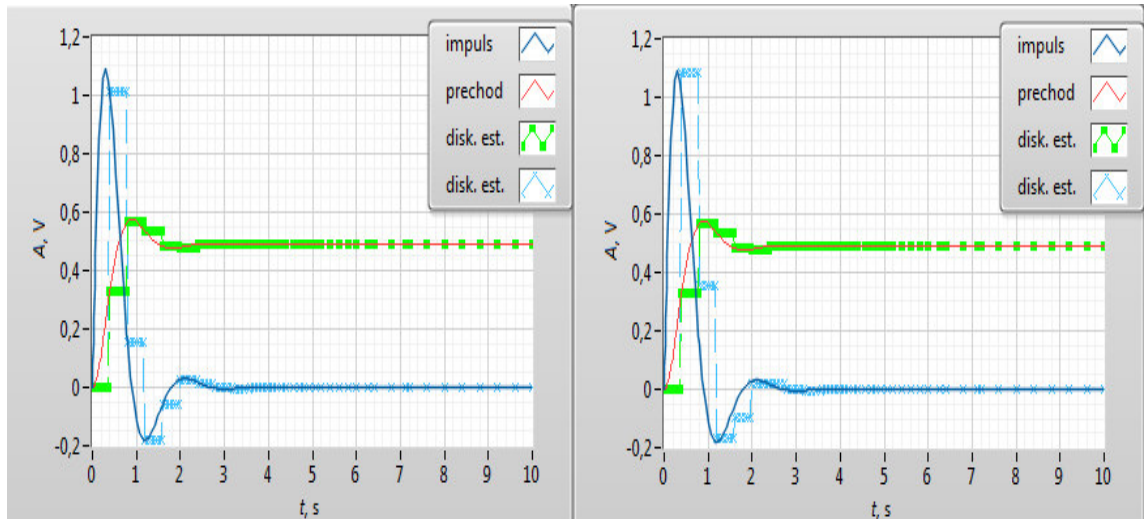


Obr. 5.8 – Systém2 – Nyquistova charakteristika

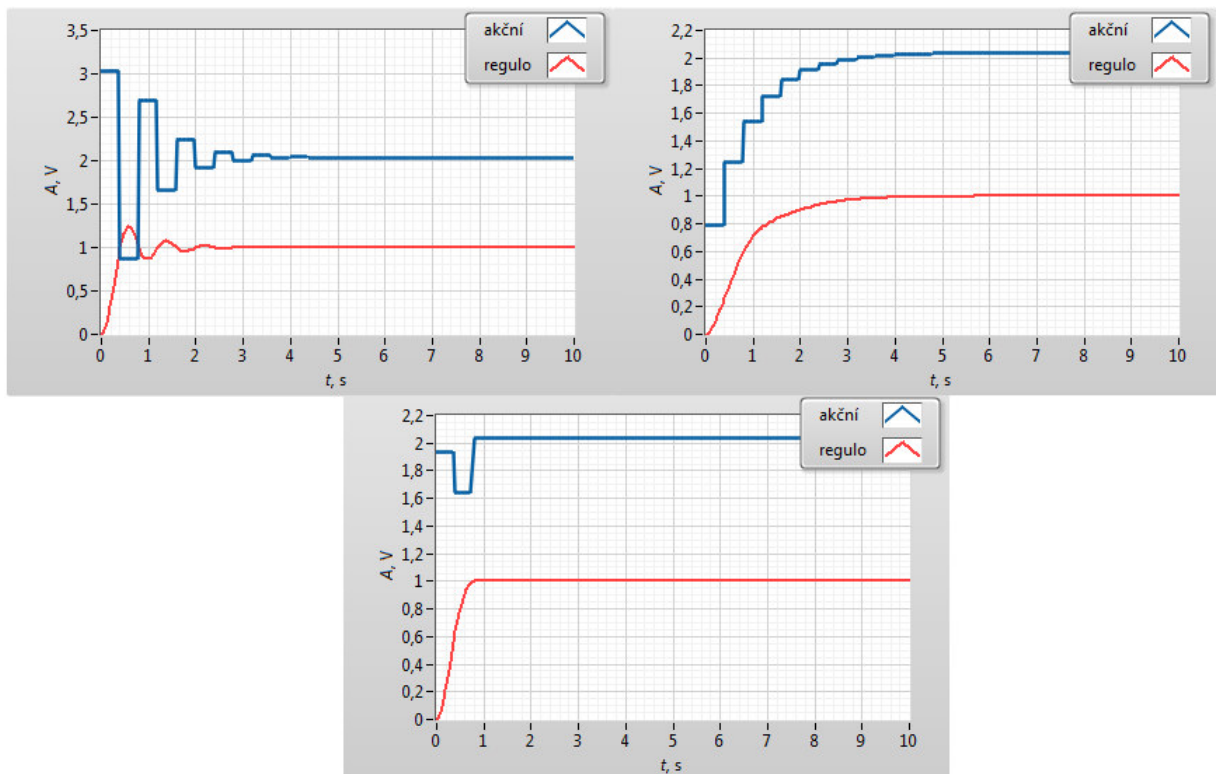


Obr. 5.9 – Systém2 – charakteristiky s diskrétním estimátorem

Na obr. 5.9 je průběh přechodové a impulzní charakteristiky a průběh těchto charakteristik s diskrétním estimátorem. I u tohoto systému byly vypočítány hodnoty H_e , M_e a N_e . Hodnoty skutečné a hodnoty vypočtené diskrétním estimátorem se pro přechodovou charakteristiku shodují. Problém je u impulzní charakteristiky, kde se hodnoty spočítané diskrétním estimátorem nedostanou až na maximální hodnotu (1,1), ale pouze na hodnotu 1. Zde by bylo lepší použít menší vzorkovací periodu. Na obr. 5.10 je ukázán rozdíl mezi průběhy, kdy se použije vzorkovací perioda 0,4 – 1. průběh a vzorkovací perioda 0,3 – 2. průběh. Je vidět, že u vzorkovací periody 0,3 s se již vypočítané hodnoty pomocí diskrétního estimátoru dostanou na maximální hodnotu a průběhy se nyní shodují i pro impulzní charakteristiku.



Obr. 5.10 – Systém2 – s různými vzorkovacími frekvencemi



Obr. 5.11 – Systém2 – Průběhy se stavovým regulátorem

Na obr. 5.11 jsou vykresleny průběhy pro stavový regulátor. Regulátory jsou navrženy tak, aby eliminovaly vliv trvalé regulační odchylky. Pro všechny regulátory byla shodně nastavena reakce na žádanou hodnotu $w = 1$ v čase 0. První průběh odpovídá kvadratickému kritériu při použití tvrdého regulátoru. Z obrázku je vidět, že je sice regulace stabilní, ale kmitavá. Druhý zobrazený průběh je pro stejné kritérium, jen je použit měkký regulátor. Regulace v tomto případě je velmi pomalá. Na třetím průběhu jsou vypočítány hodnoty stavového regulátoru

pomocí kritéria v konečném počtu regulačních kroků. Regulátor v tomto případě splňuje svou funkci, protože po konečném počtu kroků skutečně drží regulovanou hodnotu na hodnotě žádané s nulovou regulační odchylkou. Když se zhodnotí všechny tři regulace, tak by bylo nejlepší využít regulátor, který je vypočítán pomocí kritéria v konečném počtu regulačních kroků. Regulační proces je stabilní a proběhne v daný časový interval.

Charakteristiky s diskretním estimátorem byly lepší při použití vzorkovací periody 0,3 s, na obr. 5.12 jsou ukázány průběhy při této vzorkovací periodě i pro stavový regulátor. Je jasné, že při použití jiné vzorkovací periody vyjdou jinak i hodnoty stavového regulátoru. Pro vzorkovací periodu 0,3 s vychází hodnoty stavového regulátoru takto pro metodu v konečném počtu regulačních kroků

$$\mathbf{R} = [-1,723 \ -0,511 \ -2,619].$$

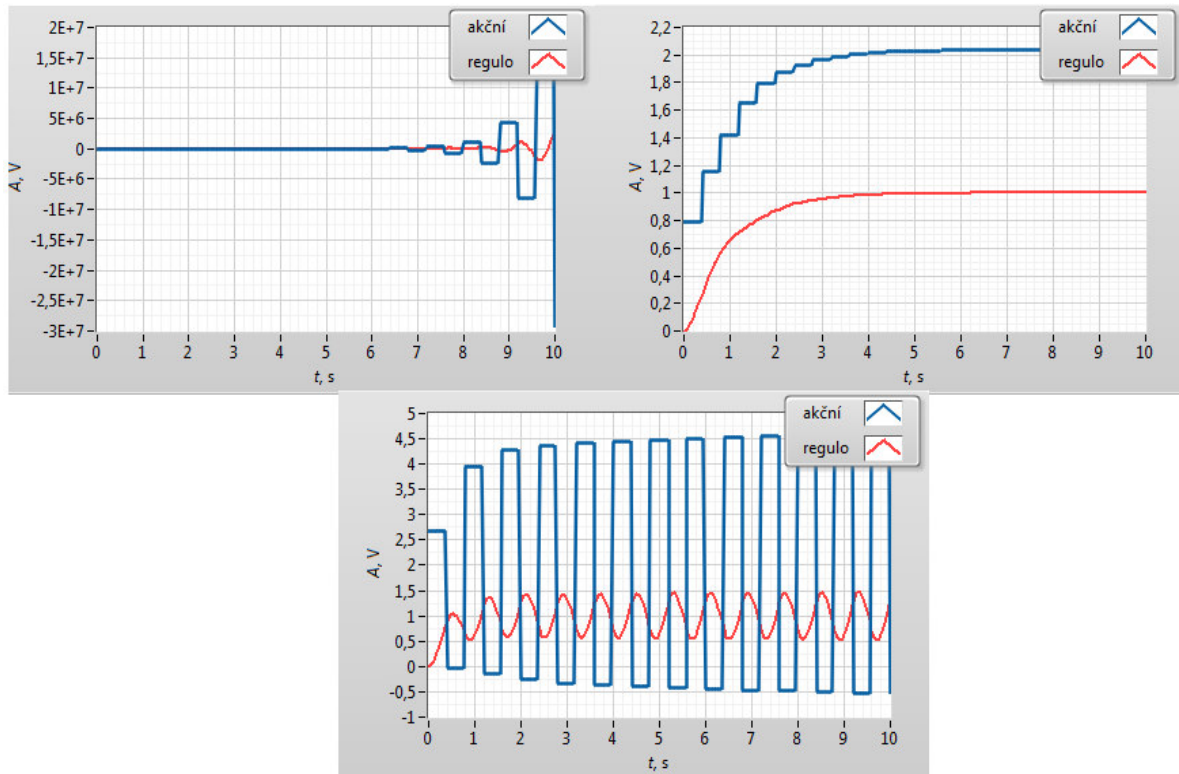
Pro kvadratické kritérium při použití měkkého regulátoru vychází

$$\mathbf{R} = [-0,823 \ -0,675 \ -0,865]$$

Pro tvrdý regulátor se dostanou hodnoty

$$\mathbf{R} = [-2,492 \ -0,607 \ -4,467].$$

Jak je vidět z obr. 5.12, tak se průběhy hodně změnily a nyní již není nejlepší použít regulátor, který spočítáme pomocí metody v konečném počtu regulačních kroků, neboť se tento průběh regulace ani neustálí. Nejlepší je použít kvadratické kritérium s měkkým regulátorem, který se jako jediný z uvedených příkladů ustálí a to za 5 sekund. U tohoto typu regulátoru se nejméně projevila změna vzorkovací periody.



Obr. 5.12 – Systém2 – při vzorkovací periodě 0,3 s

6 Závěr

V práci se postupovalo podle zadání, nejprve byly vypracovány požadované řešerše, které přiblížily prostředí LabVIEW a především nástroj Control Design and Simulation Modul. V dalších kapitolách byly popsány druhy charakteristik, stavový regulátor a estimátory. Tyto poznatky jsou využity v praktické části.

V praktické části bylo cílem vypracovat v prostředí LabVIEW aplikaci, za pomoci simulačních nástrojů Control Design and Simulation Module, která bude simulovat různé typy dynamických procesů. Tento cíl byl splněn a vytvořená aplikace vykresluje přechodovou charakteristiku, impulzní charakteristiku, Bodeho charakteristiku a Nyquistovu charakteristiku. Dále je v aplikaci navržen diskrétní estimátor, jehož hodnoty se zobrazují na čelním panelu a průběhy jsou zobrazeny v grafu. Dále je v aplikaci vypočítáván stavový regulátor. Pro návrh stavového regulátoru jsou využity dvě metody, metoda konečném počtu regulačních kroků a návrh regulátoru podle kvadratického kritéria, kde je možné si vybrat mezi „měkkým“ a „tvrdým“ regulátorem. Mezi metodami se volí pomocí Matlab Scriptu, kde se nepoužívaná metoda dá jako komentář. Na čelním panelu se zobrazují výsledné hodnoty stavového regulátoru a také průběh se stavovým regulátorem, který eliminuje vliv trvalé regulační odchylky.

Tato aplikace musí mít výpočetní část, neboť je potřeba spočítat hodnoty estimátoru a stavového regulátoru. V této práci byl využit Matlab Script, který je přímo v prostředí LabVIEW. Zařazením Matlab Script vznikly potíže, které se v dané verzi LabVIEW nedají odstranit a tak, program pracuje správně až na druhé spuštění aplikace, neboť při prvním spuštění objeví chybové hlášení, že nesouhlasí velikosti matic. Tento problém je právě díky Matlab Script, který se vykonává jako první a v té době ještě nejsou matice A , B , C a D naplněny. Ve starších verzích LabVIEW tento problém bylo možné vyřešit pomocí Flat Sequence, ale v nové verzi to již není možné: tento problém snad půjde vyřešit v novějších verzích.

Na této aplikaci se do budoucna dá vylepšit již zmíněné spuštění aplikace, dále by se mohl rozšířit výpočet i pro větší systémy. V novějších verzích prostředí LabVIEW by také mohl být přehlednější výpočtový prvek, který by lépe zpřehlednil metody pro výpočet stavového regulátoru.

Aplikace může sloužit jako ukázka studentům, jak vypadají základní charakteristiky pro dané systémy a také jak se navrhují stavové regulátory a diskrétní estimátory a jak vypadají průběhy regulace s nimi.

Literatura

- BALÁTĚ, J. 2004. *Automatické řízení, 2.vydání*. Praha: BEN – Technická literatura. 663 s. ISBN 80-7300-148-9.
- KUPKA, L. 2007. *Frekvenční charakteristiky*. [online]. [cit. 30.11.2013]. Dostupné na: http://www.fm.tul.cz/~libor.kupka/Frekvencni_charakteristiky.pdf
- KUPKA, L., JANEČEK, J. 2007. *Matlab&Simulink: řešené příklady*. Lanškroun: TG tisk a SOŠ a SOU. 224 s. ISBN 978-80-239-9532-9.
- KUPKA, L., 2013. *Teorie automatického řízení II*. Pardubice: Osobní sdělení.
- MÁŠA, P. 2013. *Frekvenční charakteristiky EO2 – Přednáška 11* [online]. [cit. 30.11.2013]. Dostupné na: <http://amber.feld.cvut.cz/vyu/eo2/files/lectures/P11.pdf>
- PECHOUŠEK, J. 2004. *Základy programování v prostředí LabVIEW, 1. Vydání*. Olomouc: Univerzita Palackého v Olomouci. ISBN 80-244-0800-7.
- SLAPNÍČKA, M. 2010. *Syntéza regulačního obvodu se stavovým regulátorem s ohledem na jeho robustnost*. Diplomová práce. Liberec: Technická Univerzita v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií, Katedra elektrotechniky a informatiky. 82s.
- VLACH, J., HAVLÍČEK, J., VLACH, M. 2008. *Začínáme s LabVIEW*. Praha: BEN – technická literatura. 247 s. ISBN 978-80-7300-245-9.

Příloha A – Celý Matlab Script

%Výpočet matic diskrétního popisu

M=expm(A*T);

N=inv(A)*(M-[1 0; 0 1])*B;

%Výpočet hodnot diskrétního estimátoru

Q=-M(2,2)/M(1,2);

Qu=Q

He=M(2,1)+(M(2,2)*Q)-(Q*M(1,1))-(Q*M(1,2)*Q);

Me=M(2,2)-(Q*M(1,2));

Ne=N(2)-(Q*N(1));

Heu=He;

Meu=Me;

Neu=Ne;

E=[0;0]

Mr=[M E;C*M 1];

Nr=[N;C*N];

Cr=[C 0];

Gr=[NrMr*Nr Mr^2*Nr]

%Návrh stavového regulátoru

% v konečném počtu regulačním kroků

E1=[0 0 1];

Rr=-(E1)*(Gr^-1)*(Mr^3)

%pomocí kvadratického kritéria

Q1=[0 0 0;0 0 0;0 0 1]

%L=0; %tvrdý regulátor

%L=1; %měkký regulátor

P=[0 0 0; 0 100 0; 0 0 0]

%for i=1:1:10

Pi=Q1+Mr'*P*Mr-Mr'*P*Nr*inv(L+Nr'*P*Nr)*Nr'*P*Mr;

R=inv(L+Nr'*P*Nr)*Nr'*P*Mr

```
P=Pi;  
%end  
%Rr=-R;  
% příprava hodnot stavového regulátoru na vyvedení ze skriptu  
Rr1=Rr(1)  
Rr2=Rr(2)  
Rr3=-Rr(3)  
%konstanty přenosu dány do nové proměnné, aby se mohly vyvést ze scriptu  
A1=A(2,1)  
A2=A(2,2)  
B2=B(2)
```