

**Západočeská univerzita v Plzni**  
**Fakulta aplikovaných věd**  
**Katedra kybernetiky**

# **BAKALÁŘSKÁ PRÁCE**

**PLZEŇ, 2012**

**Dan Široký**

**Západočeská univerzita v Plzni**

**Fakulta aplikovaných věd**

**Katedra kybernetiky**

**BAKALÁŘSKÁ PRÁCE**

**Monitorování a ohodnocení kvality regulace  
PID regulátoru**

**PLZEŇ, 2012**

**Dan Široký**

**ZADÁNÍ BAKALÁŘSKÉ PRÁCE**  
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Dan ŠIROKÝ**  
Osobní číslo: **A10B0833P**  
Studijní program: **B3918 Aplikované vědy a informatika**  
Studijní obor: **Kybernetika a řídicí technika**  
Název tématu: **Monitorování a ohodnocení kvality regulace PID regulátoru**  
Zadávací katedra: **Katedra kybernetiky**

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

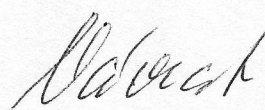
1. Seznamte se s různými metodami experimentálního ohodnocení kvality regulace standardních průmyslových regulátorů (PI, PID,...).
2. Proveďte teoretickou analýzu vybrané metody ohodnocování kvality regulace.
3. Vybranou metodu implementujte v programovacím systému REX jako funkční blok (REXLang).
4. Otestujte funkční vlastnosti vytvořeného bloku na vybraných příkladech průmyslové regulace metodou simulace v programovacím systému Simulink/REX.
5. Diskutujte obdržené výsledky vzhledem k možnosti praktického nasazení vytvořených nástrojů v praxi.

Rozsah grafických prací: **dle potřeby**  
Rozsah pracovní zprávy: **30-40 stránek A4**  
Forma zpracování bakalářské práce: **tištěná**  
Seznam odborné literatury:

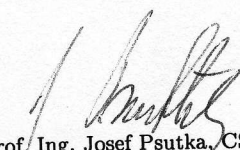
**HARRIS T.J. et. al.: A review of performance monitoring and assessment techniques for univariate and multivariate control systems. Journal of Process Control, 9 (1999).**

Vedoucí bakalářské práce: **Prof. Ing. Miloš Schlegel, CSc.**  
Katedra kybernetiky

Datum zadání bakalářské práce: **13. února 2012**  
Termín odevzdání bakalářské práce: **18. května 2012**



Doc. Ing. František Vávra, CSc.  
děkan



Prof. Ing. Josef Psutka, CSc.  
vedoucí katedry

V Plzni dne 13. února 2012

## PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 21. srpna 2012

.....  
*vlastnoruční podpis*

## PODĚKOVÁNÍ

Děkuji vedoucímu mé práce prof. Ing. M. Schlegelovi, CSc. z Katedry kybernetiky Fakulty aplikovaných věd ZČU Plzeň za odborné vedení práce, trpělivost, ochotu a v neposlední řadě za podporu.

Také bych rád poděkoval členům své rodiny, mé přítelkyni a mým přátelům za jejich trpělivost, toleranci a podporu během mého studia, především v čase tvorby bakalářské práce.

## **Annotation**

In the first chapters, the reader is thoroughly familiarized with the current state of evaluation and monitoring of the quality of control and he is presented with several basic methods. The rest of the work is focused on the design, implementation and verification of functional properties of blocks that would be suitable tools for the practical application of these basic methods.

## **Keywords**

control quality index \* quality of control \* mean value \* variance \* recursive calculation of the estimate \* running discrete Fourier transform \* RDFT \* REXLANG

## **Anotace**

V prvních kapitolách je čtenář zevrubně seznámen s aktuálním stavem hodnocení a sledování kvality regulace a je mu předloženo několik základních metod. Zbytek práce je zaměřen na návrh, implementaci a ověření funkčních vlastností bloků, které by byly vhodnými nástroji při praktickém použití zmíněných základních metod.

## **Klíčová slova**

index kvality regulace \* kvalita regulace \* střední hodnota \* rozptyl \* rekurzivní výpočet odhadu \* běžící diskretní Fourierova transformace \* RDFT \* REXLANG



## **OBSAH**

Seznam obrázků	vii
1 Úvod	9
2 Cíl práce	10
3 Indexy kvality regulace	11
3.1 Index kvality založený na střední hodnotě řízené veličiny	11
3.2 Index kvality založený na varianci řízené veličiny	12
3.3 Index kvality založený na frekvenčních vlastnostech	12
4 Implementace bloků pro ocenění kvality regulace	15
4.1 Blok pro odhad střední hodnoty a variance	15
4.2 Blok pro výpočet běžící diskrétní Fourierovy transformace	17
5 Simulační ověření funkčních vlastností diagnostických bloků	20
5.1 Ověření funkčních vlastností na LS třetího řádu	20
5.1.1 Blok pro odhad střední hodnoty a variance	21
5.1.2 Blok pro výpočet běžící diskrétní Fourierovy transformace	25
5.2 Ověření funkčních vlastností na LS druhého řádu s dopravním zpožděním	31
5.2.1 Blok pro odhad střední hodnoty a variance	33
5.2.2 Blok pro výpočet běžící diskrétní Fourierovy transformace	37
6 Závěr	43
A Literatura	44
B Přílohy	45
B-1 Zdrojový kód algoritmu bloku pro odhad střední hodnoty a variance	45
B-2 Zdrojový kód algoritmu bloku pro výpočet RDFT	47
B-3 Celostránkové verze grafů z kapitoly 5.1	51
B-4 Celostránkové verze grafů z kapitoly 5.2	65

## SEZNAM OBRÁZKŮ

Obr. 3.1: Ilustrace k požadavkům na tvar citlivostní funkce	13
Obr. 3.2: Ilustrace přímkové aproximace citlivostní funkce a jejího rozdělení podle zesílení	14
Obr. 4.1: Blok pro výpočet odhadu stř. hodnoty a variance překrytý subsystémem	16
Obr. 4.2: Blok pro výpočet RDFT překrytý subsystémem	18
Obr. 5.1: Schéma zapojení regulační smyčky	20
Obr. 5.2: Průběh žádané hodnoty a řízené veličiny	21
Obr. 5.3: Schéma zapojení pro ověření funkčních vlastností bloku E_VAR	21
Obr. 5.4: Průběh střední hodnoty $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu	22
Obr. 5.5: Průběh rozptylu a směrodatné odchylky	22
Obr. 5.6: Průběh žádané hodnoty a řízené veličiny při působení poruchy	23
Obr. 5.7: Průběh střední hodnoty $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu	24
Obr. 5.8: Průběh rozptylu a směrodatné odchylky	24
Obr. 5.9: Schéma zapojení pro ověření funkčních vlastností bloku RDFT	25
Obr. 5.10: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$	26
Obr. 5.11: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$	26
Obr. 5.12: Průběh amplitudy a reálné a imaginární rozklad složky $u(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$	27
Obr. 5.13: Průběh amplitudy a reálné a imaginární rozklad složky $u(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$	27
Obr. 5.14: Průběh žádané hodnoty a řízené veličiny	28
Obr. 5.15: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy	29
Obr. 5.16: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy	29
Obr. 5.17: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy	30

Obr. 5.18: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy	30
Obr. 5.19: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy a „rozpojené“ zpětné vazbě	31
Obr. 5.20: Schéma zapojení regulační smyčky 2	32
Obr. 5.21: Průběh žádané hodnoty a řízené veličiny	32
Obr. 5.22: Schéma zapojení pro ověření funkčních vlastností bloku E_VAR	33
Obr. 5.23: Průběh střední hodnoty $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu	34
Obr. 5.24: Průběh rozptylu a směrodatné odchylky	34
Obr. 5.25: Průběh žádané hodnoty a řízené veličiny při působení poruchy	35
Obr. 5.26: Průběh střední hodnoty $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu	36
Obr. 5.27: Průběh rozptylu a směrodatné odchylky	36
Obr. 5.28: Schéma zapojení pro ověření funkčních vlastností bloku RDFT	37
Obr. 5.29: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$	38
Obr. 5.30: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$	38
Obr. 5.31: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$	39
Obr. 5.32: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$	39
Obr. 5.33: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy	40
Obr. 5.34: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy	40
Obr. 5.35: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy	41
Obr. 5.36: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy	41
Obr. 5.37: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy a „rozpojené“ zpětné vazbě	42

## 1 ÚVOD

Drtivá většina ([1] uvádí až 95%) všech v praxi nasazených regulátorů je typu PID. Ačkoliv je tento typ regulátoru starý přes sedmdesát let, není ustálen význam jeho parametrů a neexistuje ani norma pro jeho implementaci. Stejně tak neexistují v praxi obecně přijaté nástroje pro jeho analytický návrh a především není ustálen způsob pro ohodnocení kvality regulace PID regulátorů. Přitom kvalita řízení procesů ovlivňuje významným způsobem jejich užité vlastnosti. Podle odhadů ve je v současnosti až 60% všech v průmyslu nasazených regulátorů buď špatně navrženo či seřízeno anebo pracují v manuálním režimu, kdy vyžadují neustálou pozornost operátor. [2] Tento stav ovlivňuje nejen zmíněnou užítost takto řízených procesů, ale má také obrovský dopad na ekonomiku. Jak napsal B. Liptak (Adjunct Professor, Yale University): „Optimalizace existujících řídicích systémů by přinesla dostatečný profit k eliminaci obchodního deficitu USA.“

Ke zlepšení současného stavu nepřispívá ani existence velkého množství různých variant regulátorů s velmi často nedostatečně obsáhlými uživatelskými příručkami. Z nich nebývá jasný přesný význam zadávaných parametrů a nehovoří ani o chování regulátoru v nestandardních režimech. [1] To vede ke snížení efektivity nasazení regulátorů.

Pokud základní vrstva řídicího systému nepracuje kvalitně, nedojde ke zlepšení ani nasazením pokročilých algoritmů prediktivního řízení nebo aplikací jiných optimalizačních metod. [1] Ani operátoři řídicích systémů nemohou věnovat pozornost kvalitě řízení v každé regulační smyčce a případné nedostatky napravovat, neboť současné řídicí systémy jich obsahují velké množství (desítky až tisíce regulačních smyček).

## 2 CÍL PRÁCE

Z obsahu předchozí kapitoly vyplývá nutnost nějakým způsobem řešit stávající situaci s cílem zlepšit kvalitu regulace. Aby tohoto cíle mohlo být dosaženo musí být nejprve k dispozici příslušné diagnostické nástroje a metody. Je potřeba průběžně automaticky ohodnocovat kvalitu řízení jednotlivých regulátorů a identifikovat ty smyčky, jejichž kvalita regulace neodpovídá optimální. Dále je potřeba určovat zdroje problémů a podle potřeby jednorázově automaticky korigovat nastavení parametrů regulátoru.

Cílem této práce je navrhnout a implementovat bloky využitelné pro ohodnocení a sledování kvality regulace regulační smyčky několika jednoduchými metodami. Část práce je také zaměřena na prosté vysvětlení vybraných metod, pro něž jsou zmíněné bloky primárně navrženy.

### 3 INDEXY KVALITY REGULACE

Základním problémem je ohodnotit kvalitu regulace regulační smyčky. Je třeba ocenit aktuálně dosahovanou kvalitu určitým indexem. Známe-li pak mezní hodnotu tohoto indexu, kterou bychom získali při nasazení ideálního regulátoru nebo optimálním nastavením parametrů stávajícího regulátoru daného typu, obdržíme srovnáním s určeným aktuálním indexem kvality informaci o tom, jak je kvalita právě provozovaného regulátoru blízká zmíněné dosažitelné kvalitě regulace.

Za ideální regulátor lze považovat lineární regulátor s minimální variancí. Ten však má obecně nekonečný řád a velikost a rychlost změny jeho akční veličiny není penalizována. Proto má srovnání s ním spíše teoretickou než praktickou hodnotu. Bylo by potřeba znát spíše mezní hodnotu kvality regulace dosažitelnou stávajícím regulátorem daného typu. [1] Srovnání této hodnoty se zjištěným indexem kvality regulace bude daleko lépe vypovídat o rozdílu stávajícího regulátoru od optimálního prakticky realizovatelného. Pokud bude odchylka mezi zjištěným indexem a jeho mezní hodnotou zanedbatelná, nebude potřeba zásahu do nastavených parametrů. V opačném případě bude žádoucí nastavení stávajícího regulátoru změnit a to nejlépe automaticky bez významného narušení provozu.

Z doposud uvedeného vyplývá, že je potřeba určit vhodný index kvality regulace. Dále nalézt prakticky použitelné metody pro odhad jeho mezní hodnoty a pro zjištění hodnoty tohoto indexu u právě nasazeného regulátoru.

#### 3.1 Index kvality založený na střední hodnotě řízené veličiny

Nejjednodušší hodnocení kvality regulace lze založit na informaci o střední hodnotě řízené veličiny v ustáleném stavu. Při procesu regulace je jedním z požadavků co nejpřesnější sledování požadované hodnoty výstupní veličinou systému. Střední hodnota řízené veličiny se v případě nasazení ideálního regulátoru neliší v ustáleném stavu od požadované hodnoty. Index kvality lze tedy definovat jako

$$I = w - E[y]; I^* = 0,$$

kde  $w$  je požadovaná hodnota,  $y$  řízená veličina,  $E[y]$  její střední hodnota a  $I^*$  index kvality regulace při nasazení ideálního regulátoru. Čím méně kvalitně probíhá proces regulace tím větší je rozdíl  $I$  a  $I^*$ . Jinak řečeno čím lépe regulátor pracuje tím menší je absolutní hodnota  $I$  a blíží se hodnotě  $I^*$ .

Tato metoda však neposkytuje informaci o tom, zda se na odchylce  $E[y]$  od  $w$  podílí skutečně regulátor nebo se jedná o vliv externích chyb. S konstantní či skokově působící externí chybou se regulátor PID vypořádá díky své integrační složce. Poruchy působící náhodně nebo harmonicky se již na střední hodnotě řízené veličiny v ustáleném stavu mohou projevit.

### 3.2 Index kvality založený na varianci řízené veličiny

Další možností jak hodnotit kvalitu regulace je zkoumat rozptyl řízené veličiny. Při regulaci na konstantní hodnotu by ideální regulátor dosáhl v ustáleném stavu nulového rozptylu řízené veličiny

$$I = D[y]; I^* = 0,$$

kde  $D[y]$  je rozptyl řízené veličiny.  $I^*$  je opět hodnota indexu dosažená ideálním regulátorem. Čím lépe regulátor pracuje tím menší je hodnota  $I$  a blíží se hodnotě  $I^*$ .

Tento způsob ocenění kvality regulace opět neposkytuje informaci o původu odchylky  $I$  od ideální hodnoty. Konstantními či skokově působícími poruchami se opět v případě PID regulátoru nemá smysl zabývat. Náhodně působící a harmonické poruchy se ale znovu na hodnotě  $I$  v ustáleném stavu projevit mohou.

Protože při výpočtu rozptylu se hodnoty signálu i střední hodnoty umocní, může index kvality v ustáleném stavu nabývat nevýznamných hodnot v řádech  $10^{-3}$  a menších. V porovnání s hodnotami požadované veličiny a řízené veličiny se však může jednat o významné odchylky. Tomuto je možné v jisté míře zabránit převodem rozptylu na směrodatnou odchylku prostým odmocněním

$$I = \sigma = \sqrt{D[y]}; I^* = 0.$$

Index kvality regulace v tomto tvaru zřetelněji upozorní i na ty odchylky řízené veličiny, které mají relativně malou hodnotu.

### 3.3 Index kvality založený na frekvenčních vlastnostech

Jedním z důležitých přenosů v regulační smyčce je přenos od poruchy na řízenou veličinu

$$d_0 \rightarrow y: S(p) = \frac{1}{1+C(p) \cdot P(p)},$$

kde  $C(p)$  je přenos regulátoru a  $P(p)$  je přenos řízeného procesu.  $S(p)$  se nazývá citlivostní funkce regulační smyčky. Zesílení poruchy  $d_0$  na frekvenci  $\omega_d$  je dáno hodnotou  $|S(j\omega_d)|$

$$y(t) = |S(j\omega_d)| \cdot d_0(t),$$

kde  $y(t)$  je řízená veličina. Na citlivostní funkci je proto kladen požadavek co nejméně zesilovat poruchy na všech frekvencích. V ideálním případě by platilo

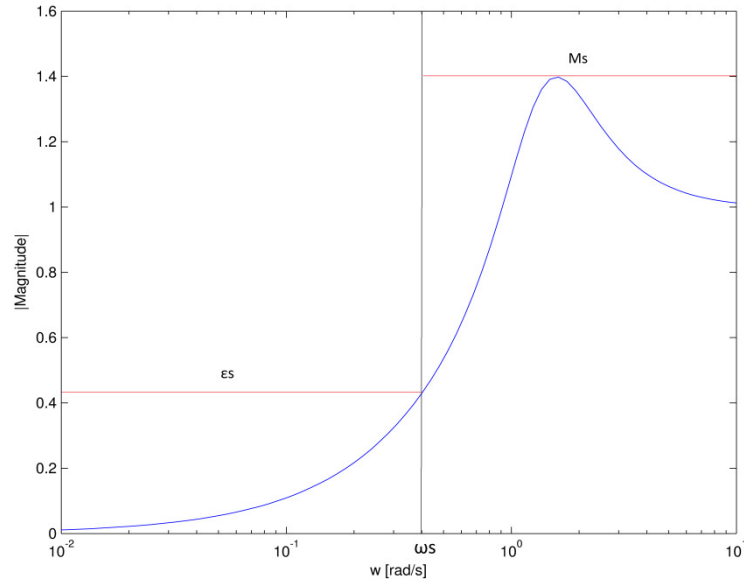
$$|S(j\omega)| = 0, \forall \omega.$$

Bodeho věta však dokazuje, že tohoto požadavku není v reálné situaci možné dosáhnout.

Na místo toho je na citlivostní funkci kladen požadavek, aby co nejlépe tlumila poruchy alespoň na nějakém intervalu frekvencí (viz také Obr. 3.1) [2]:

$$|S(j\omega)| < \varepsilon_S, \forall \omega \in \langle 0, \omega_S \rangle,$$

$$|S(j\omega)| < M_S, \forall \omega.$$



Obr. 3.1: Ilustrace k požadavkům na tvar citlivostní funkce

Další důležitou vlastností reálného procesu je, že na sinusový vstup reaguje s vysokou přesností sinusovým výstupem pouze na omezeném intervalu frekvencí  $\langle 0, \Omega_a \rangle$ . Frekvence  $\Omega_a$  je nazývána efektivní šířkou pásma a omezuje kvalitu regulace bez ohledu na řád lineárního regulátoru. Její hodnotu lze poměrně snadno experimentálně určit. Jsou-li splněny předpoklady Bodeho věty a efektivní šířka pásma je  $\Omega_a$  potom platí [2]:

$$\int_0^{\Omega_a} \ln|S(j\omega)| d\omega \doteq 0. \quad (1)$$

Je tedy možné realizovat citlivostní funkci, jejíž zesílení je na určitém intervalu frekvencí minimální, respektive ne větší než  $\varepsilon_S$ , a do frekvence  $\Omega_a$  dobře potlačuje poruchy. Integrál (1) lze rozdělit na dvě části [2]. První odpovídá intervalu frekvencí, na kterém je zesílení přímkové aproximace citlivostní funkce menší než hodnota  $M_S$ . Druhá odpovídá intervalu frekvencí na kterém je zesílení přímkové aproximace citlivostní funkce rovno hodnotě  $M_S$  (viz Obr. 3.2).

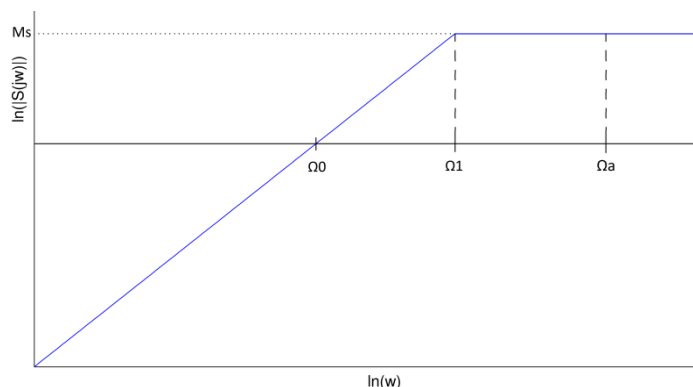
$$\int_0^{\Omega_a} \ln|S(j\omega)| d\omega = \int_0^{\Omega_1} \ln\left(\frac{M_S}{\Omega_1} \cdot \omega\right) d\omega + (\Omega_a - \Omega_1) \cdot \ln M_S \doteq 0,$$

kde  $\Omega_1$  je frekvence, od níž má přímkové aproximace citlivostní funkce zesílení  $M_S$ . Z této rovnice je možné získat vztahy pro  $\Omega_0$  a  $\Omega_1$

$$\Omega_0 \doteq \Omega_a \cdot \ln M_S,$$

$$\Omega_1 \doteq \Omega_0 \cdot \ln M_S.$$





Obr. 3.2: Ilustrace přímkové aproximace citlivostní funkce a jejího rozdělení podle zesílení

Tyto vztahy se ukazují jako vhodné pro ocenění kvality regulace. S jejich využitím je index kvality možné definovat vztahem [2]:

$$I = \frac{1 - |S(j\omega_d)|}{1 - \frac{M_s}{\Omega_1} \omega_d}, I^* = 1,$$

Pro vyčíslení indexu kvality pak stačí znalost  $\Omega_a$ , frekvence významné poruchy  $\omega_d$ , hodnotu  $|S(j\omega_d)|$  a hodnotu  $M_s$  která je známa z požadavků na citlivostní funkci položených při návrhu regulátoru. Výše definovaný index lze ještě upravit do tvaru

$$I = \frac{1 - |S(j\omega_d)|}{1 - \frac{M_s}{\Omega_1} \omega_d} = \frac{\Omega_1 \cdot (1 - |S(j\omega_d)|)}{\Omega_1 - M_s \omega_d} = \frac{\frac{\Omega_1}{\omega_d} (1 - |S(j\omega_d)|)}{\frac{\Omega_1}{\omega_d} - M_s}.$$

Pokud má působící porucha harmonický průběh platí

$$y(t) = |S(j\omega_d)| \cdot d_0(t) = |S(j\omega_d)| \cdot A_d \cdot \sin(\omega_d t + \varphi),$$

$$y(t) = A_y \cdot \sin(\omega_d t + \varphi) \Rightarrow |S(j\omega_d)| = \frac{A_y}{A_d},$$

Amplitudy  $A_y$  a  $A_d$  lze určit spektrální analýzou řízené veličiny  $y(t)$ . Konkrétně amplitudu  $A_d$  lze určit ze složky  $y(t)$  o frekvenci  $\omega_d$  při rozpojené zpětné vazbě a amplitudu  $A_y$  lze určit ze složky  $y(t)$  o frekvenci  $\omega_d$  při nerozpojené zpětné vazbě [2]. Rozpojení zpětné vazby je možné zařazením pásmové zadržky na frekvenci  $\omega_d$  do regulační smyčky.

## 4 IMPLEMENTACE BLOKŮ PRO OCENĚNÍ KVALITY REGULACE

Pro implementaci bloků byl zvolen software REX Control System. Algoritmy jsou realizovány pomocí univerzálních programovatelných bloků REXLANG, do kterých jsou vkládány externím souborem se zdrojovým kódem. Ten je psán skriptovacím jazykem jenž vychází z jazyka C. Pro snadné zapojení a přehlednost a úsporu místa ve schématech pro Simulink byly z bloků REXLANG v obou případech vytvořeny subsystémy opatřené maskou (Obr. 4.1 a Obr. 4.2). Kromě vstupů a výstupů popsaných u obou bloků jsou oba opatřeny ještě vstupem HLD a výstupem Err. Přivedení logické hodnoty „1“ na vstup HLD zastaví provádění skriptu kterým je blok osazen. Tento vstup odpovídá vstupu HLD bloku REXLANG. Na výstupu Err lze číst kód chyby při běhu programu. Tento výstup odpovídá výstupu iE bloku REXLANG.

Všechny vytvořené bloky používají pro uchování vzorků zpracovávaného signálu jednoduchý kruhový buffer. Ten se skládá z pole fixní délky, z ukazatele, který odkazuje na první volnou pozici (následující poslední zapsaný vzorek) a z metod pro zápis vzorku do bufferu a pro přečtení vzorku z bufferu.

Zápis probíhá vložením vzorku do pole na pozici ukazatele, který je zároveň posunut na následující pozici v bufferu. Poté dojde ke kontrole, zda tato pozice neleží mimo rozsah pole bufferu. Pokud ano, ukazatel je posunut na počátek pole.

Metoda pro čtení vrací prvek pole, který je vzdálen o  $n$  pozic zpět oproti pozici ukazatele. Nejprve je určen index prvku podle předpisu  $index = ukazatel - n$ , kde  $n$  je parametr metody. Poté je zkontrolováno, zda index neodkazuje mimo rozsah pole bufferu. Pokud ano, znamená to v kontextu použití bufferu, že buffer byl již zaplněn a došlo k přesunutí pozice ukazatele na počátek pole. To znamená, že hledaný prvek je v poli uložen na pozici, jež je vzdálena od konce bufferu o hodnotu  $index$ , která je záporná. Dojde tedy ke korekci hodnoty  $index$  podle vztahu  $index = index + delka\_pole\_bufferu$ . Nakonec je metodou vrácen prvek uložený na určené pozici.

### 4.1 Blok pro odhad střední hodnoty a variance

Subsystém vytvořený nad blokem REXLANG byl označen E\_VAR a prostřednictvím jeho masky je zadávána hodnota  $N$  určující počet vzorků z nichž je odhad počítán. Algoritmus implementovaný blokem realizuje odhad střední hodnoty na základě vztahu

$$e_k = \frac{1}{N} \cdot \sum_{i=0}^{N-1} x_{k-i},$$

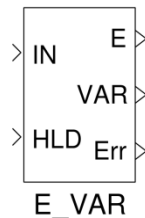
kde  $e_k$  je odhad střední hodnoty v  $k$ -tém kroku,  $N$  je uživatelem definovaný počet vzorků signálu, z nichž je odhad počítán a  $x_{k-i}$  je vzorek signálu z  $k - i$ -tého kroku. Výpočetní vztah je upraven do tvaru vhodného pro rekurzivní výpočet, jímž je odhad počítán:

$$e_k = e_{k-1} - x_{k-(N+1)} + x_k, \quad (2)$$

respektive

$$e_k = e_{k-1} + x_k, \quad (3)$$

kde  $e_k$  je opět odhad střední hodnoty v  $k$ -tém kroku,  $e_{k-1}$  je odhad střední hodnoty v předchozím kroce,  $x_k$  je vzorek signálu (do bloku přiváděn vstupem IN - viz Obr. 4.1), který je v  $k$ -tém kroku na vstupu bloku a  $x_{k-(N+1)}$  je vzorek signálu z  $k-(N+1)$ -tého kroku.



Obr. 4.1: Blok pro výpočet odhadu stř. hodnoty a variance překrytý subsystémem

Algoritmus odhadu střední hodnoty je rozdělen do tří fází. V nulté, startovací, fázi je odhad počítán podle (3) protože v bufferu zatím není uložen dostatečný počet vzorků pro realizaci (2). Necht' je tento výpočet značen jako  $e^1$ . Hodnota odhadu  $e^1/N$  je na výstupu bloku E. Po  $N$  krocích se algoritmus překloupí do fáze 1.

Výpočet odhadu  $e^1$  pokračuje podle (2) a zároveň se začíná počítat odhad  $e^2$  podle (3). Na výstupu E je hodnota odhadu  $e^1/N$ . Po  $N$  krocích dojde k překloupení do fáze 2 a zároveň k vynulování proměnné výpočtu odhadu  $e^1$ .

Ve fázi 2 pokračuje výpočet  $e^2$  podle (2). Nyní je na výstupu E hodnota odhadu  $e^2/N$  a odhad  $e^1$  se začíná znovu počítat podle (3). Fáze 1 je tedy startovací fází pro výpočet odhadu  $e^2$  a fáze 2 je druhou startovací fází pro výpočet odhadu  $e^1$ . Po  $N$  krocích se algoritmus překloupí zpět do fáze 1 a zároveň dojde k vynulování proměnné pro výpočet odhadu  $e^2$ . Dále dochází ke střídání fází 1 a 2 vždy po  $N$  krocích.

Odhad směrodatné odchylky signálu je počítán podle vztahu

$$\delta_k^2 = \frac{1}{N} \cdot \sum_{i=1}^{N-1} (x_{k-i} - e_k)^2,$$

kde  $\delta_k^2$  je odhad rozptylu v  $k$ -tém kroku,  $x_{k-i}$  je vzorek signálu z  $k-i$ -tého kroku,  $e_k$  je odhad střední hodnoty v  $k$ -tém kroku vydělený hodnotou  $N$ . Vztah je opět upraven pro rekurzivní výpočet

$$s_k = s_{k-1} - x_{k-(N+1)}^2 + x_k^2, \quad (4)$$

respektive

$$s_k = s_{k-1} + x_k^2, \quad (5)$$

$$\delta_k^2 = \frac{1}{N} \cdot s_k - \left(\frac{1}{N} \cdot e_k\right)^2,$$

kde  $s_k$  je suma kvadrátů vzorků signálu v  $k$ -tém kroku,  $s_{k-1}$  je suma kvadrátů vzorků signálu v předchozím kroku,  $x_k$  je vzorek signálu, který je v  $k$ -tém kroku na vstupu bloku,

$x_{k-(N+1)}$  je vzorek signálu z  $k-(N+1)$ -tého kroku,  $e_k$  je odhad střední hodnoty vypočtený podle (2), respektive (3) během nulté fáze, a  $\delta_k^2$  je opět odhad rozptylu v  $k$ -tém kroku.

Výpočet odhadu směrodatné odchylky je začleněn do tří fází algoritmu výpočtu odhadu střední hodnoty popsaného výše a je také počítán dvojmo. V nulté fázi je za pomoci odhadu  $e^1$  počítán odhad  $\delta^1$  podle (5), ve fázi 1 je za pomoci odhadu  $e^1$  počítán odhad  $\delta^1$  podle (4) a za pomoci odhadu  $e^2$  odhad  $\delta^2$  podle (5). Ve fázi 2 je za pomoci odhadu  $e^2$  počítán odhad  $\delta^2$  podle (4) a za pomoci odhadu  $e^1$  odhad  $\delta^1$  podle (5). Na konci fáze 1, respektive 2, dochází také k vynulování proměnné výpočtu odhadu  $rms^1$ , respektive  $\delta^2$ . V nulté fázi a ve fázi 1 je na výstupu VAR (viz Obr. 4.1) hodnota  $\delta^1$  a ve fázi 2 je na výstupu VAR hodnota  $\delta^2$ .

Mechanismus střídání a nulování proměnných na přechodu jednotlivých fází eliminuje akumulaci numerických chyb. Odhady jsou samozřejmě ještě zatíženy chybou způsobenou omezením souboru vzorků signálu použitých k určení odhadů na  $N$ . Zvyšováním hodnoty  $N$  dochází ke zpřesnění odhadu, neboť platí

$$\lim_{N \rightarrow +\infty} e_k = e;$$

$$\lim_{N \rightarrow +\infty} \delta_k = \delta,$$

kde  $e$  je skutečná střední hodnota signálu a  $\delta$  skutečná hodnota směrodatné odchylky. Zvýšení hodnoty  $N$  vede logicky k potřebě pojmout do bufferu větší počet vzorků signálu. Velikost bufferu je ovšem omezená, v případě tohoto bloku je délka pole bufferu stanovena na 1000 prvků.

## 4.2 Blok pro výpočet běžící diskretní Fourierovy transformace

Subsystém vytvořený z bloku REXLANG byl nazván RDFT a prostřednictvím jeho masky jsou zadávány parametry  $N$ ,  $\omega_1$  a  $\omega_2$ . Běžící diskretní Fourierova transformace (dále jen RDFT) je zde použita pro identifikaci konkrétní složky zkoumaného signálu o uživatelem definovaných frekvencích  $\omega_1$  a  $\omega_2$ . V případě, že chce uživatel identifikovat pouze jednu složku o frekvenci  $\omega_1$ , stačí zadat druhou frekvenci rovnou nule. Algoritmus je opět rozdělen do tří fází a počítá RDFT dvojmo. Důvod je stejný jako u předchozího bloku, tedy zamezit akumulaci numerických chyb.

Výpočet RDFT je prováděn rekurzivně. Pro odvození rekurzivních vztahů je výchozím vztahem Fourierova transformace signálu ze znalosti konečné časové posloupnosti vzorků signálu [3]:

$$X = \sum_{i=0}^{N-1} x_i \cdot z^{-i},$$

kde  $X$  je Fourierova transformace signálu,  $N$  je uživatelem definovaný počet vzorků signálu,  $x_i$  je  $i$ -tý vzorek signálu a  $z = e^{j \cdot \omega \cdot T}$  je komplexní proměnná diskretní Z-transformace, přičemž  $\omega$  je frekvence zkoumané složky signálu a  $T$  perioda vzorkování signálu.

Z tohoto vztahu je odvozen rekurzivní vztah diskrétní Fourierovy transformace

$$X_k = e^{j\omega T} \cdot [X_{k-1} + x_{k-(N+1)} \cdot e^{-N \cdot j\omega T} - x_k],$$

kde  $X_k$  je Fourierův obraz signálu v  $k$ -tém kroku,  $X_{k-1}$  je Fourierův obraz signálu v předchozím kroku,  $x_k$  je vzorek signálu (do bloku přiváděn vstupem IN - viz Obr. 4.2), který je k dispozici v  $k$ -tém kroku,  $x_{k-(N+1)}$  je vzorek signálu, který byl získán před  $N+1$  kroky. Pokud za komplexní proměnnou  $z$  dosadíme podle Eulerova vzorce

$$e^{j\omega T} = \cos(\omega \cdot T) + j \cdot \sin(\omega \cdot T) = A + j \cdot B,$$

respektive

$$e^{N \cdot j\omega T} = \cos(N \cdot \omega \cdot T) + j \cdot \sin(N \cdot \omega \cdot T) = C + j \cdot D,$$

následně roznásobíme a rozdělíme na reálnou a imaginární část, získáme rekurzivní vztahy pro rozklad signálu na reálnou a imaginární složku

$$Re_k(X) = A \cdot [Re_{k-1}(X) + C \cdot x_k - x_{k-(N+1)}] - B \cdot [Im_{k-1}(X) + D \cdot x_k], \quad (6)$$

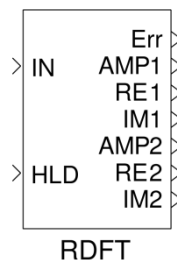
$$Im_k(X) = A \cdot [Im_{k-1}(X) + D \cdot x_k] - B \cdot [Re_{k-1}(X) + C \cdot x_k - x_{k-(N+1)}], \quad (7)$$

respektive

$$Re_k(X) = A \cdot [Re_{k-1}(X) + C \cdot x_k] - B \cdot [Im_{k-1}(X) + D \cdot x_k], \quad (8)$$

$$Im_k(X) = A \cdot [Im_{k-1}(X) + D \cdot x_k] - B \cdot [Re_{k-1}(X) + C \cdot x_k], \quad (9)$$

kde  $Re_k(X)$  je reálná složka signálu vypočtená v  $k$ -tém kroku,  $Re_{k-1}(X)$  je reálná složka signálu vypočtená v předchozím kroku,  $Im_k(X)$  je imaginární složka signálu vypočtená v  $k$ -tém kroku a  $Im_{k-1}(X)$  je imaginární složka signálu vypočtená v předchozím kroku. Hodnoty reálné a imaginární složky signálu o frekvenci  $\omega_1$  jsou na výstupech RE1 a IM1, hodnoty reálné a imaginární složky signálu o frekvenci  $\omega_2$  jsou na výstupech RE2 a IM2. Na výstupech je vždy ten rozklad, který je v právě probíhající fázi počítán podle (6) a (7). Výjimku tvoří nultá fáze, kdy jsou hodnoty na výstupech počítány podle (8) a (9).



Obr. 4.2: Blok pro výpočet RDFT překrytý subsystémem

Blok také počítá amplitudu zkoumaných složek signálu a to podle vztahu [3]:

$$A_X = \sqrt{Re_k^2(X) + Im_k^2(X)}. \quad (10)$$

Amplituda je vždy počítána z těch proměnných výpočtu RDFT, které jsou v právě probíhající fázi počítány podle (6) a (7). Opět platí stejná výjimka pro nultou fázi.

Hodnota amplitudy složky o frekvenci  $\omega_1$  je na výstupu AMP1, amplituda složky o frekvenci  $\omega_2$  je na výstupu AMP2 bloku.

Jak již bylo zmíněno algoritmus je rozdělen do tří fází a pracuje analogicky jako algoritmus předešlého bloku. Nejprve se během inicializace zjistí vzorkovací perioda a předpočítají se hodnoty koeficientů  $A, B, C, D$  pro obě frekvence (případně jen pro  $\omega_1$ ). Vlastní algoritmus pak v nulté fázi začne počítat RDFT<sup>1</sup> podle (8) a (9). Po  $N$  krocích se algoritmus překlopí do fáze 1 a pokračuje ve výpočtu RDFT<sup>1</sup> podle (6) a (7) a začíná se počítat RDFT<sup>2</sup> podle (8) a (9). Po dalších  $N$  krocích se algoritmus překlopí do fáze 2 a zároveň dojde k vynulování proměnných výpočtu RDFT<sup>1</sup>. Ve fázi 2 probíhá stejný proces, pouze proměnné RDFT<sup>1</sup> a RDFT<sup>2</sup> si vymění role. Fáze 1 a 2 se po  $N$  krocích střídají, jak již bylo popsáno u předcházejícího bloku.

Rozklad složky signálu o frekvenci  $\omega_2$  je zařazen za rozklad složky o frekvenci  $\omega_1$ , výpočty neprobíhají paralelně. Důvod oddělení obou výpočtů je přehlednost a snazší vyřazení rozkladu složky o frekvenci  $\omega_2$  v případě, že uživatel má zájem identifikovat pouze jednu složku.

## 5 SIMULAČNÍ OVĚŘENÍ FUNKČNÍCH VLASTNOSTÍ DIAGNOSTICKÝCH BLOKŮ

Ověřování funkčních vlastností bylo provedeno pomocí softwaru MATLAB, jeho nadstavby Simulink a blocksetu REX Industrial Blockset. Pro testování bloků byly vytvořeny dva stabilní lineární systémy (LS). K oběma systémům byl prostřednictvím appletu PID Control Laboratory 3.1 navržen PID regulátor.

### 5.1 Ověření funkčních vlastností na LS třetího řádu

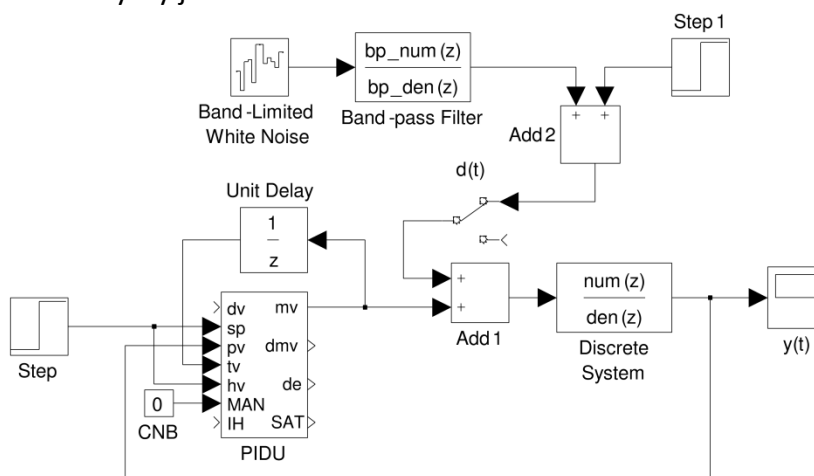
První regulační smyčka na které byly bloky testovány sestává z LS třetího řádu se třemi identickými kořeny a k němu navrženým PID regulátorem. Přenos systému

$$P(p) = \frac{2}{p^3 + 3 \cdot p^2 + 3 \cdot p + 1}$$

je realizován blokem Transfer Function. Regulátor je realizován blokem PIDU ze zmíněného blocksetu systému REX. Obraz akční veličiny regulátoru je

$$U(p) = 1.1597 \cdot \left\{ W(p) - Y(p) + \frac{1}{2.5353 \cdot p} \cdot [W(p) - Y(p)] + \frac{0.6338 \cdot p}{0.6338 \cdot p + 10} \cdot [0 - Y(p)] \right\}.$$

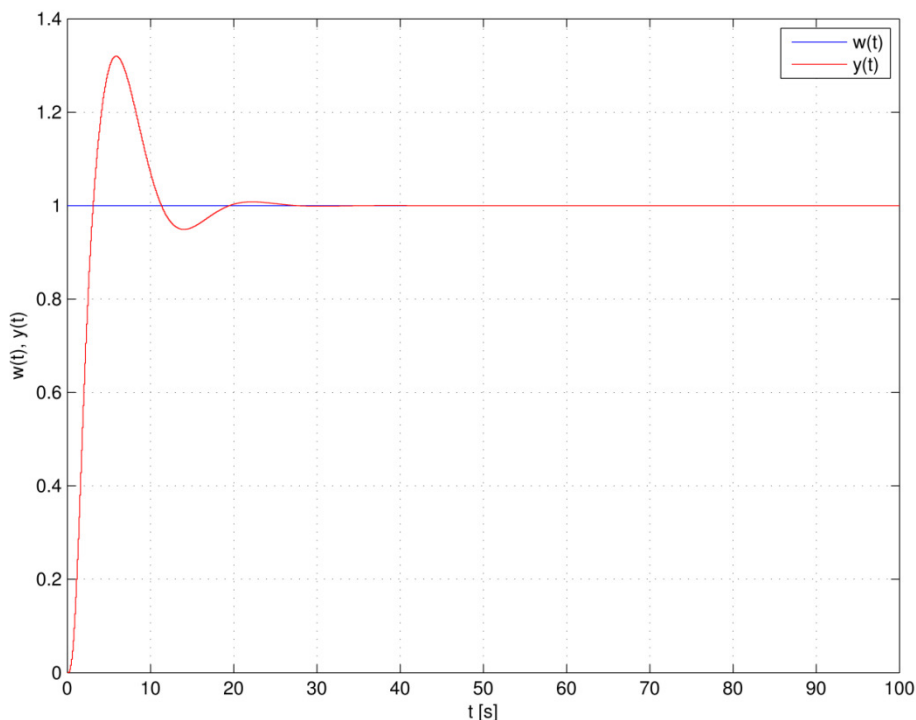
Zapojení regulační smyčky je znázorněno schématem na Obr. 5.1.



Obr. 5.1: Schéma zapojení regulační smyčky

Přepínač  $d(t)$  zavádí do regulační smyčky externí poruchu. Ta je tvořena součtem skokové poruchy a bílého šumu, z něhož je pásmovou propustí vybrána složka o frekvenci  $\omega_d = 5 \text{ rad} \cdot \text{s}^{-1}$ . Tato část schématu je pro následující ověřování funkčních vlastností podstatná neboť právě zavedená porucha ovlivní kvalitu regulace v ustáleném stavu a bude tím zastupovat špatné nastavení regulátoru.

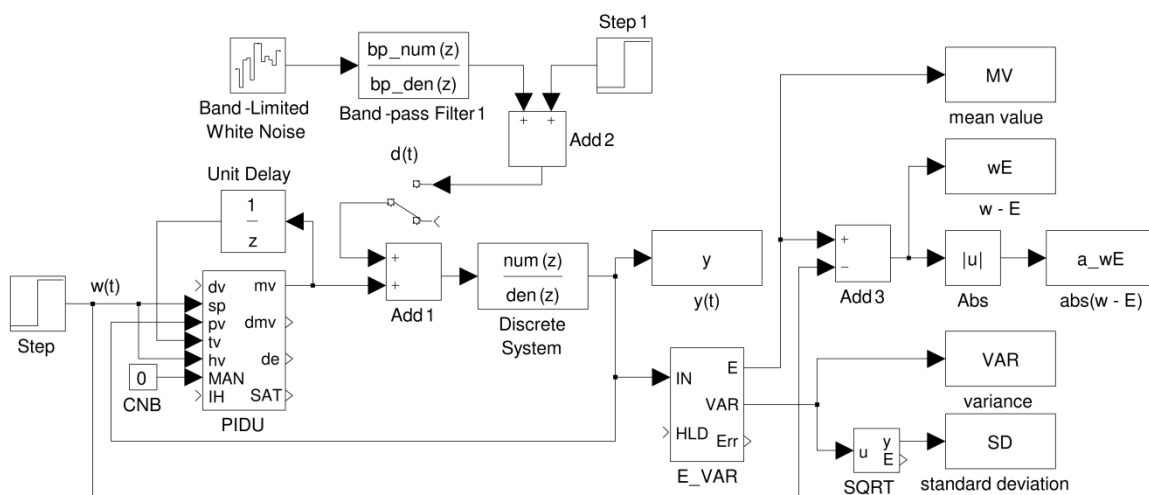
PID regulátor je navržen tak, aby uspokojivě reguloval diskretizovaný systém na hodnotu  $w(t)$  nastavovanou blokem Step (viz Obr. 5.2).



Obr. 5.2: Průběh žádané hodnoty a řízené veličiny

### 5.1.1 Blok pro odhad střední hodnoty a variance

V regulační smyčce přibyl blok E\_VAR který je připojen na řízenou veličinu.



Obr. 5.3: Schéma zapojení pro ověření funkčních vlastností bloku E\_VAR

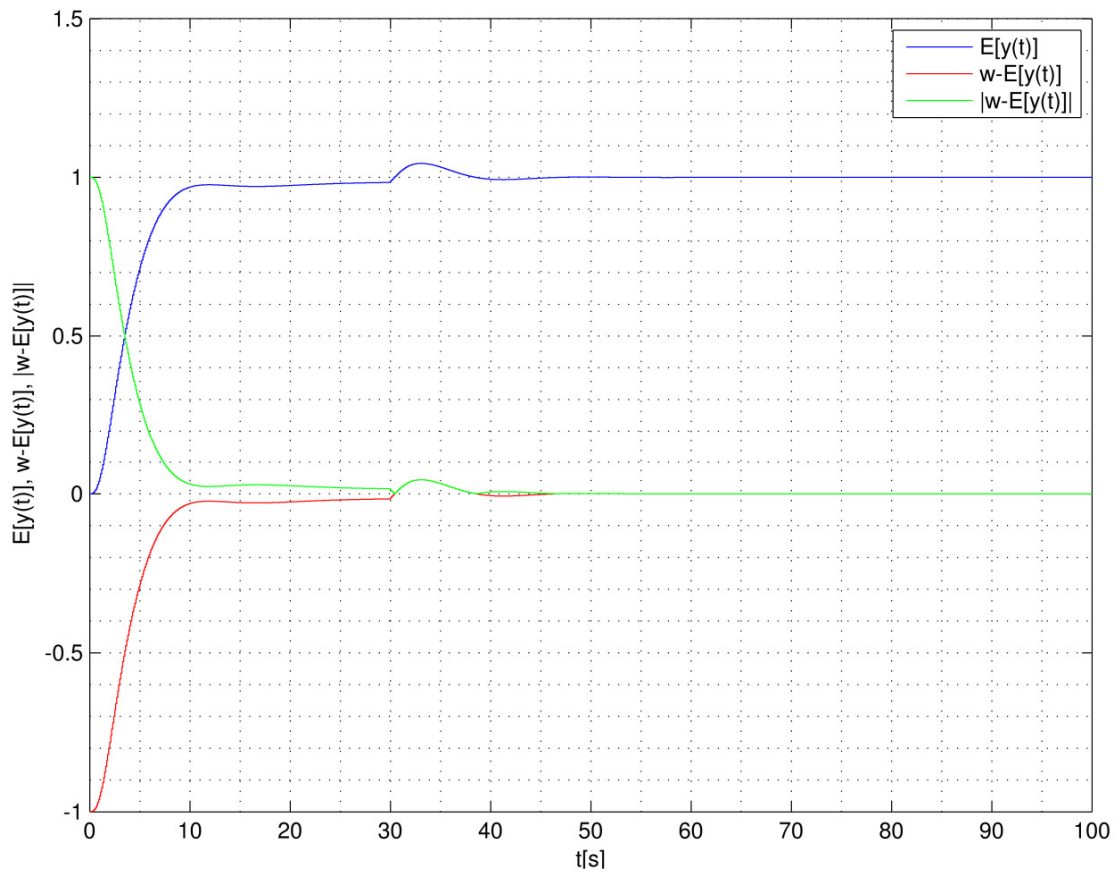
Pokud blok E\_VAR pracuje správně, měla by se jím odhadovaná střední hodnota řízené veličiny v ustáleném stavu bez působení poruchy shodovat s požadovanou hodnotou. Rozptyl (a tedy i směrodatná odchylka) by měla být v ustáleném stavu bez působení poruchy rovna nule. Očekávané výsledky experimentu jsou

$$E[y(t)] = w(t) = 1, w(t) - E[y(t)] = |w(t) - E[y(t)]| = 0,$$

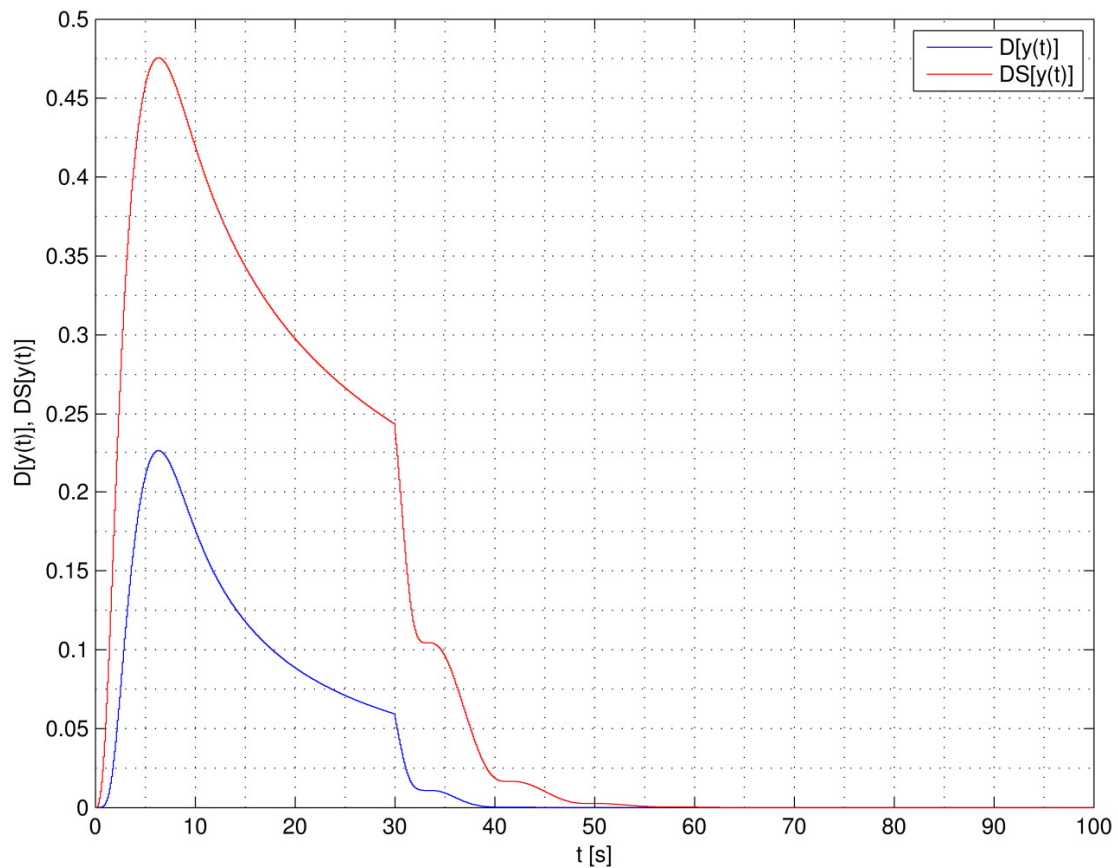
$$VAR[y(t)] = 0, \delta = 0.$$

Na Obr. 5.4 a Obr. 5.5 je vidět, že simulací bylo skutečně dosaženo očekávaných výsledků. Grafy v celostránkové velikosti naleznete v části PŘÍLOHY pod B-3.





Obr. 5.4: Průběh střední hodnoty  $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu



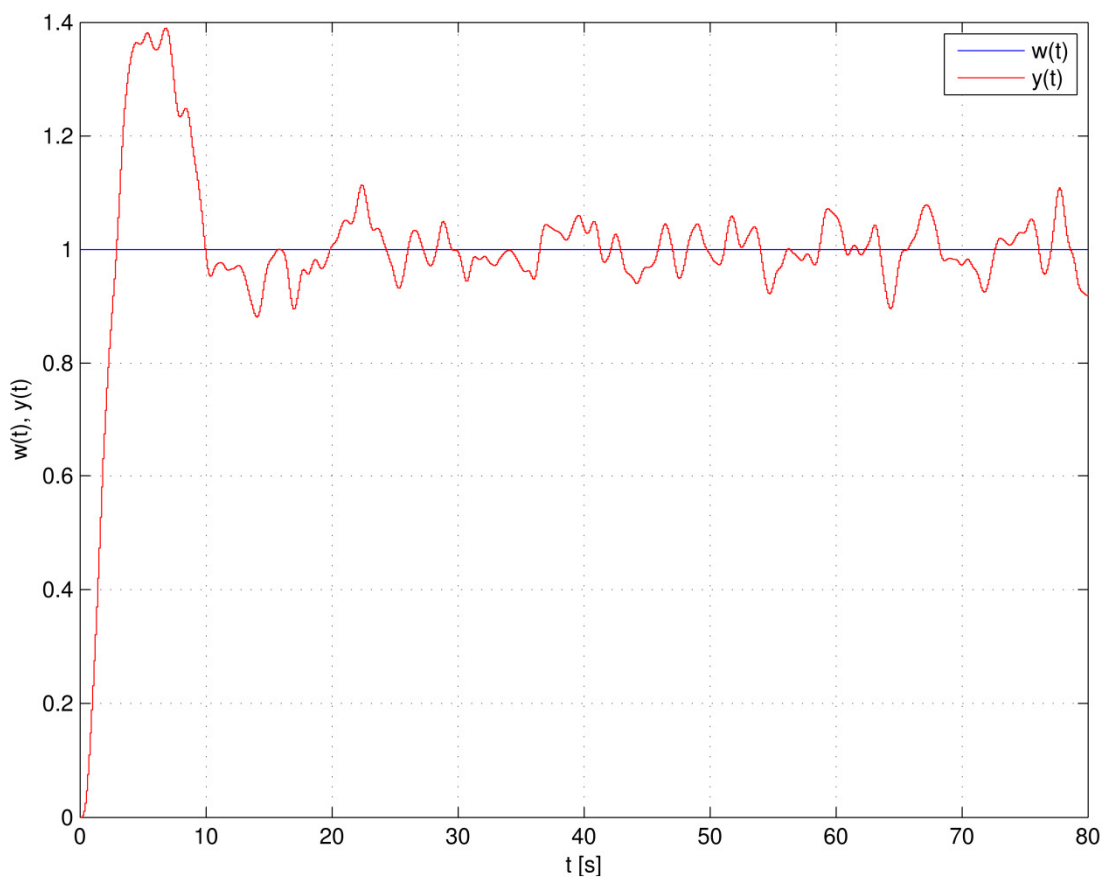
Obr. 5.5: Průběh rozptylu a směrodatné odchyly řízené veličiny

Další experiment prověřuje chování bloku při zavedení externí poruchy.  
Očekávané chování bloku je

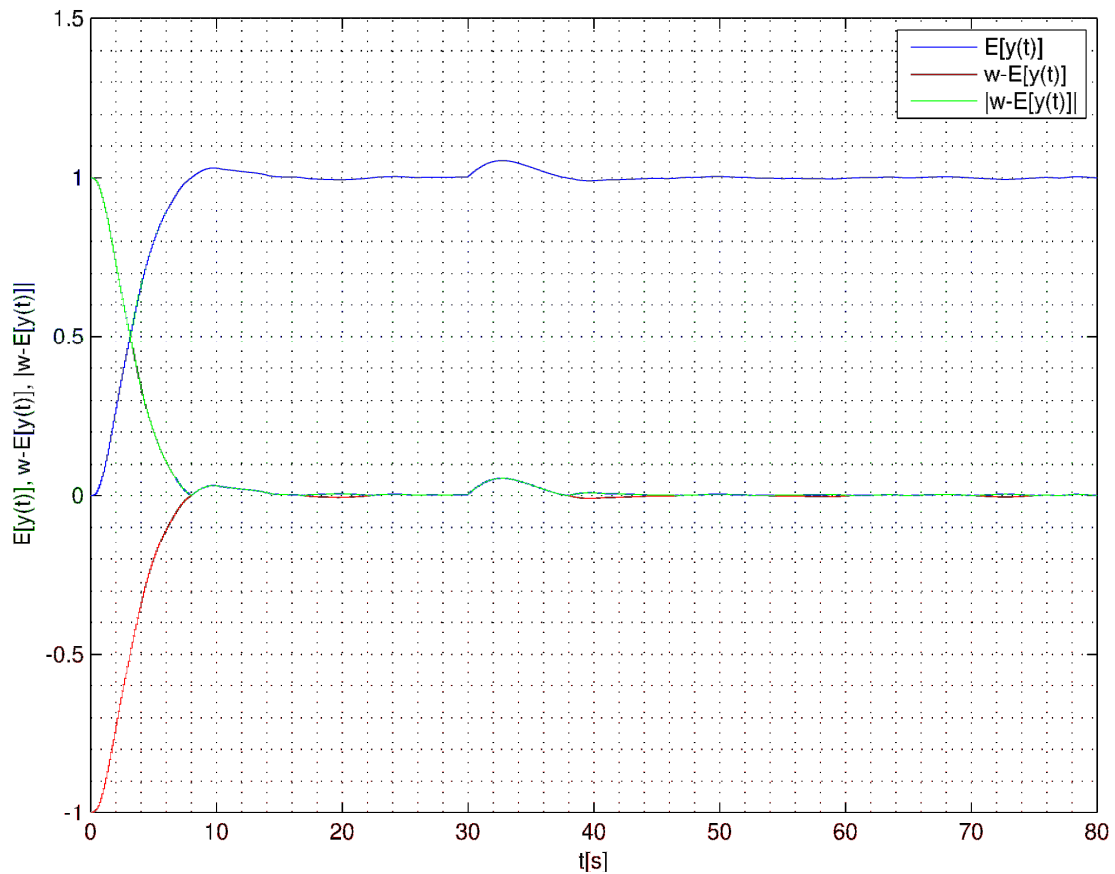
$$E[y(t)] \neq w(t), w(t) - E[y(t)] \neq 0, |w(t) - E[y(t)]| > 0,$$

$$VAR[y(t)] > 0, \delta > 0.$$

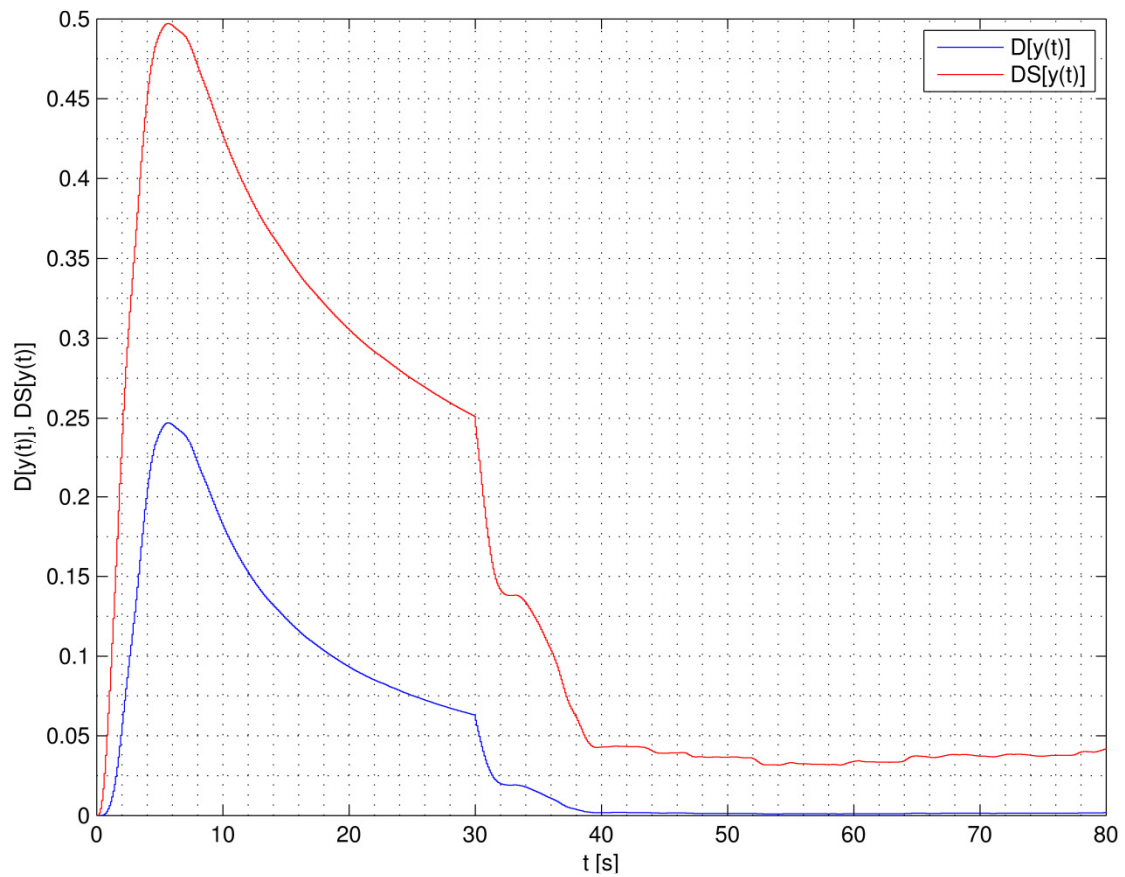
Na Obr. 5.6 je zřetelně vidět změna průběhu řízené veličiny zapříčiněná poruchou. Průběhy zaznamenané na Obr. 5.7 a Obr. 5.8 ukazují, že blokem odhadované hodnoty  $E[y(t)]$  a  $VAR[y(t)]$  opět plní očekávání. Větší formát grafů je opět v části PŘÍLOHA pod B-3.



Obr. 5.6: Průběh žádané hodnoty a řízené veličiny



Obr. 5.7: Průběh střední hodnoty  $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu

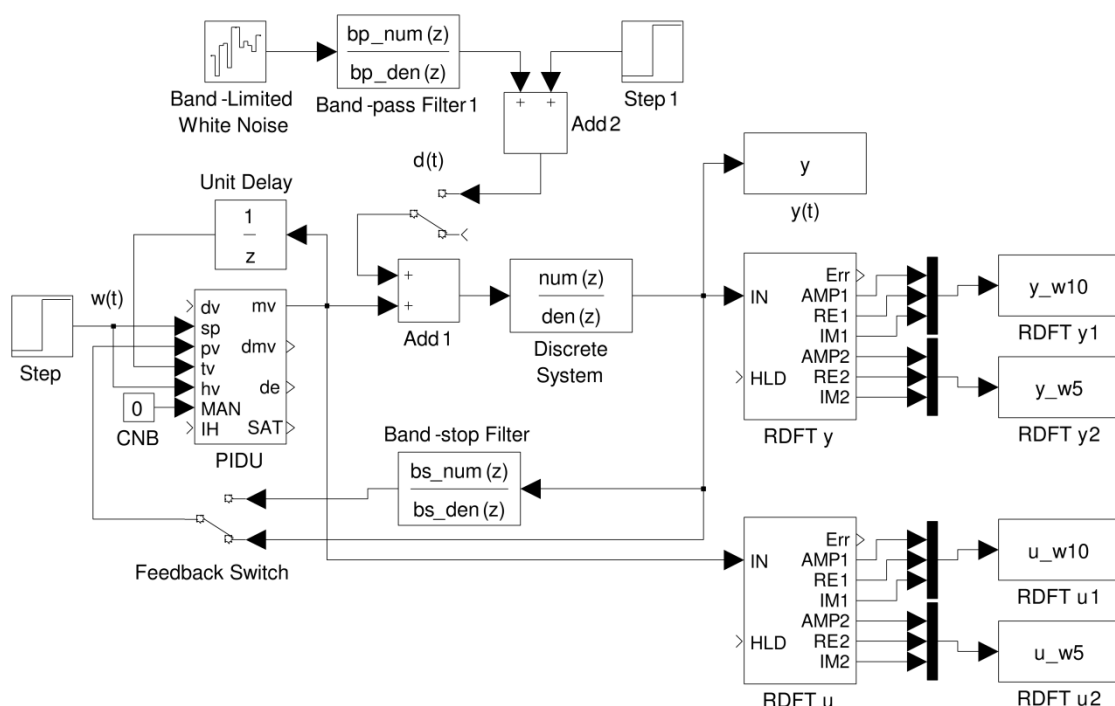


Obr. 5.8: Průběh rozptylu a směrodatné odchylky řízené veličiny

### 5.1.2 Blok pro výpočet běžící diskretní Fourierovy transformace

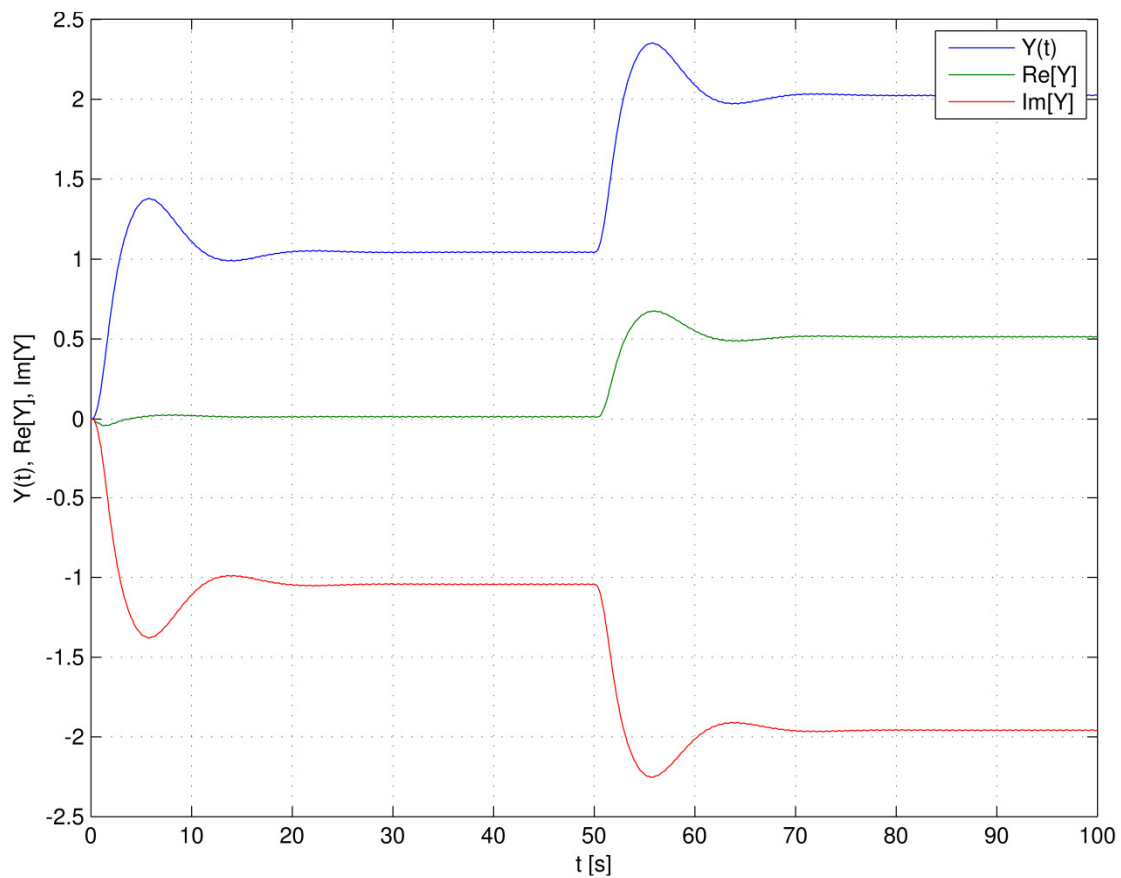
Zapojení použité k experimentům při ověřování funkčních vlastností bloku RDFT je znázorněno schématem na Obr. 5.9. Žádaná hodnota i odezva na ni je stejná jako na Obr. 5.3. Ve schématu se objevují dva bloky RDFT. Parametry obou bloků jsou  $N = 500$ ,  $\omega_1 = 10 \text{ rad} \cdot \text{s}^{-1}$ ,  $\omega_2 = 5 \text{ rad} \cdot \text{s}^{-1}$ . Na vstup prvního, označeného RDFT  $y$ , je přiváděna řízená veličina  $y(t)$ . Na vstup bloku označeného RDFT  $u$  je přiváděna akční veličina  $u(t)$ .

Frekvence  $\omega_2 = 5 \text{ rad} \cdot \text{s}^{-1}$  je totožná s frekvencí, která projde pásmovou propustí Band-pass Filter z generátoru bílého šumu. Frekvence  $\omega_1 = 10 \text{ rad} \cdot \text{s}^{-1}$  je zvolena čistě náhodně pro ověření funkce bloku a demonstraci vlivu poruchy na další složky signálu.

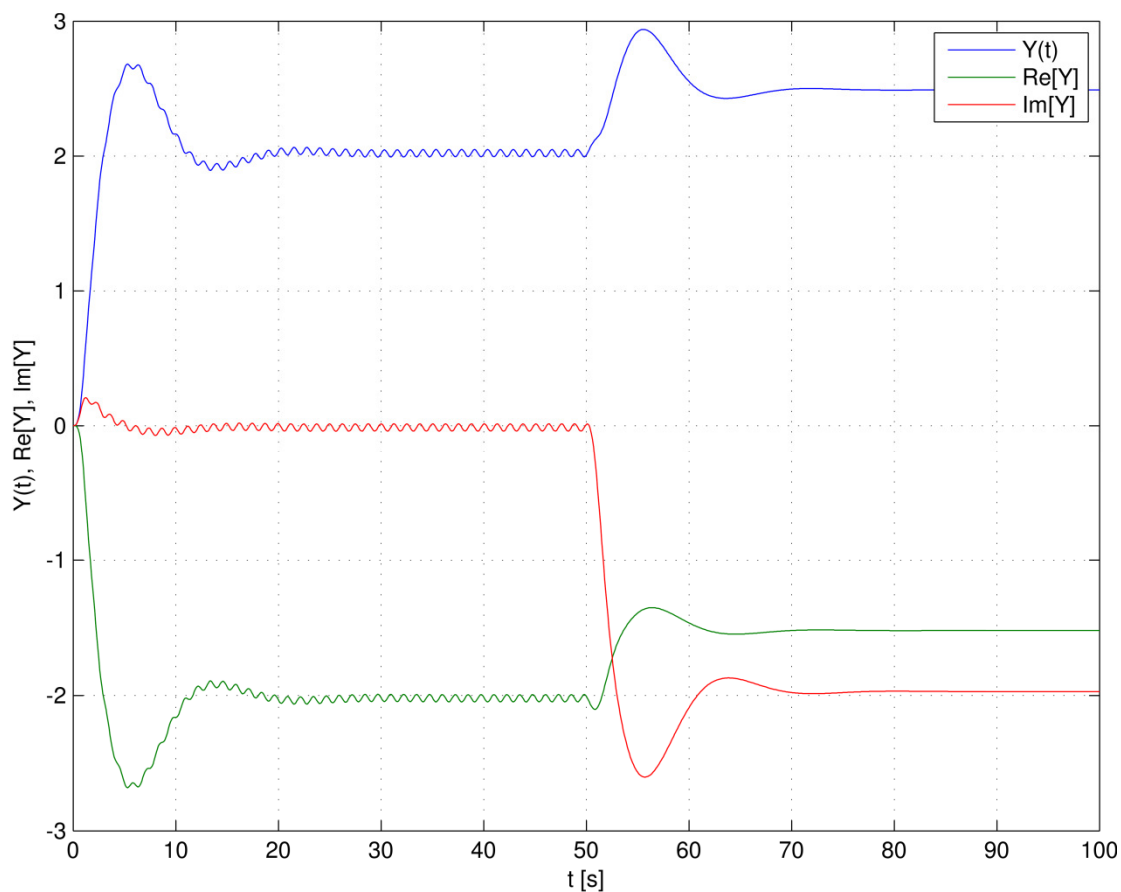


Obr. 5.9: Schéma zapojení pro ověření funkčních vlastností bloku RDFT

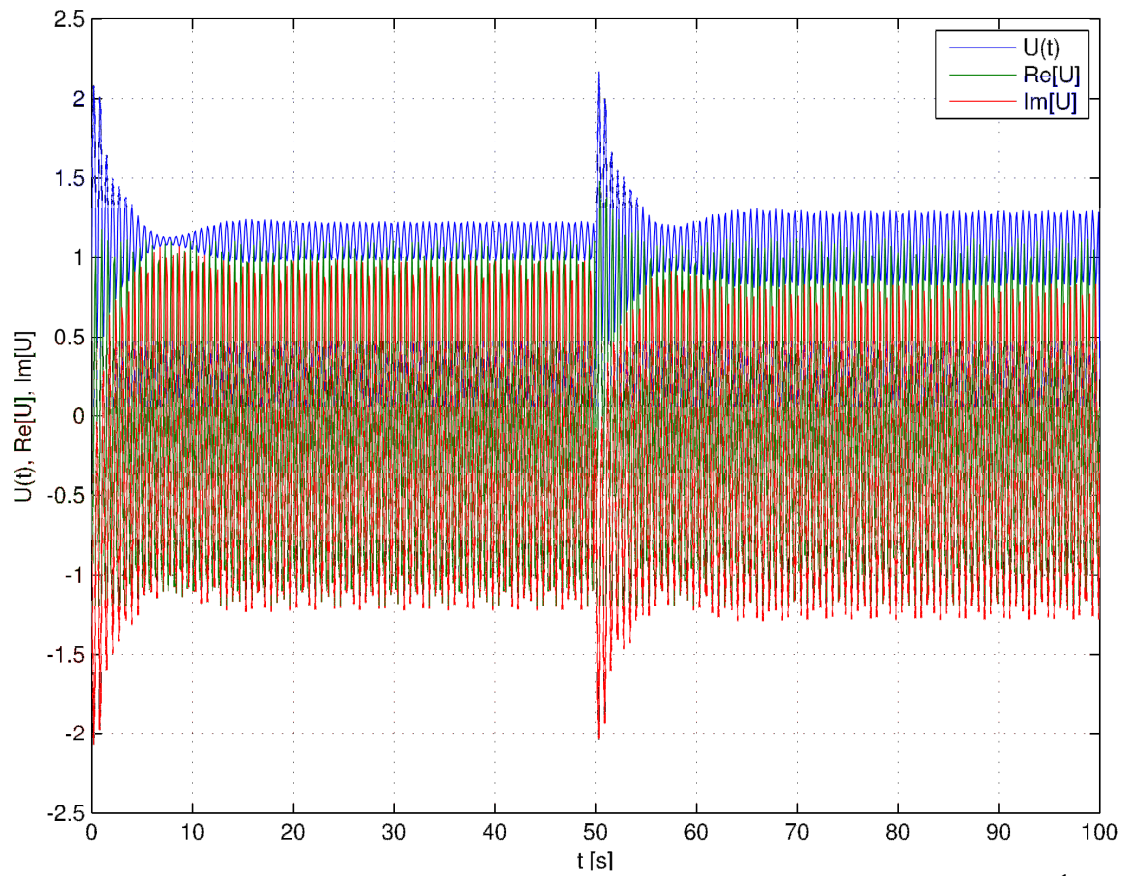
První pokus je proveden bez zavedené poruchy a bez zařazení pásmové zadržky Band-stop Filter ve zpětné vazbě. Průběhy získané bloky RDFT  $y$  a RDFT  $u$  poslouží ke srovnání s výsledky dalších pokusů. Následující grafy na Obr. 5.10 až Obr. 5.13 zachycují zmíněné průběhy. Větší formát grafů je opět v části PŘÍLOHA pod B-3.



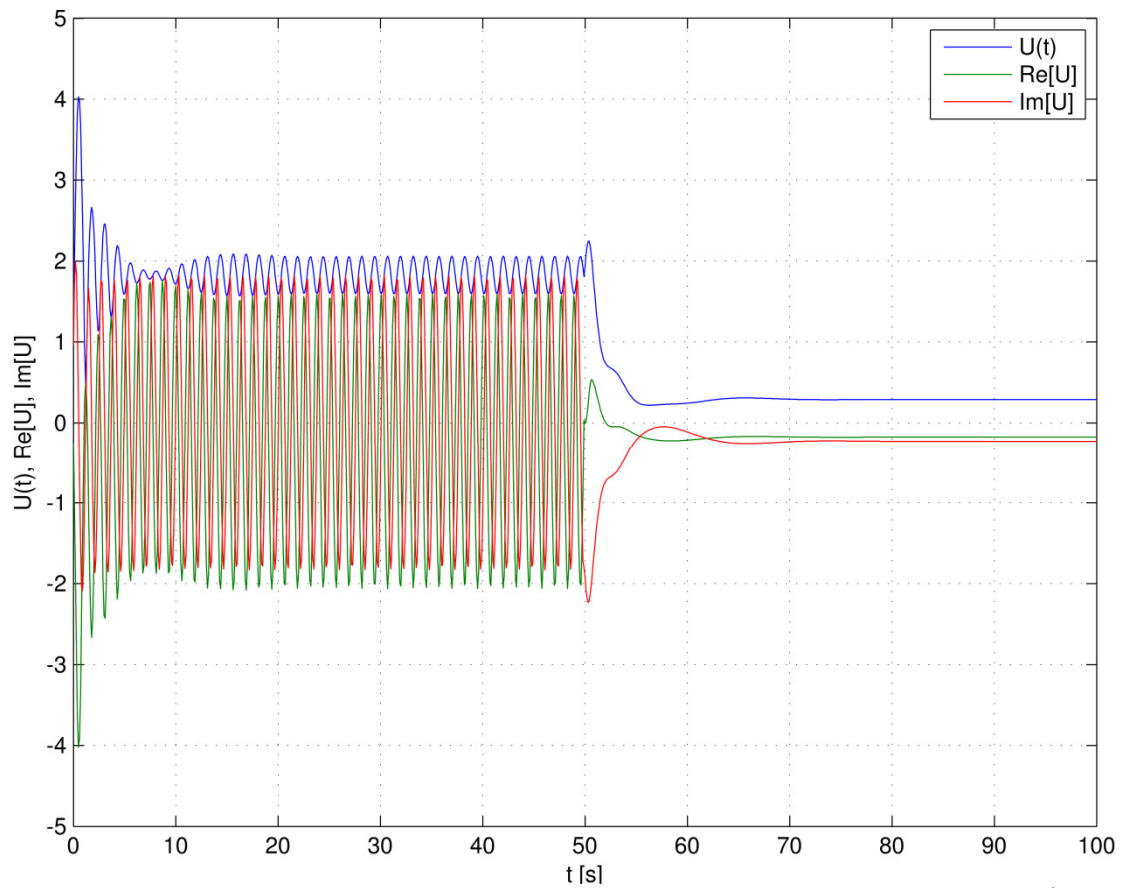
Obr. 5.10: Průběh amplitudy a reálné a imaginární složky  $y(t)$  o frekvenci  $10 \text{ rad} \cdot \text{s}^{-1}$



Obr. 5.11: Průběh amplitudy a reálné a imaginární složky  $y(t)$  o frekvenci  $5 \text{ rad} \cdot \text{s}^{-1}$



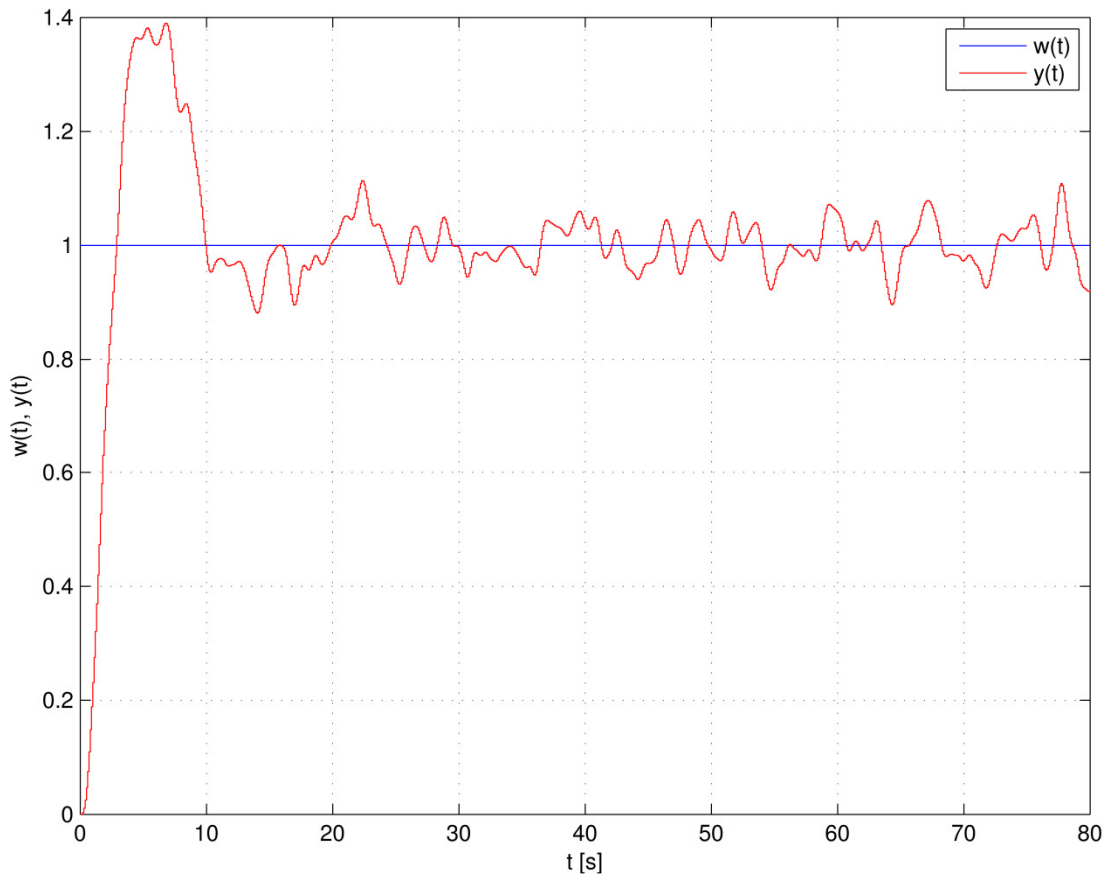
Obr. 5.12: Průběh amplitudy a reálné a imaginární rozklad složky  $u(t)$  o frekvenci  $10 \text{ rad} \cdot \text{s}^{-1}$



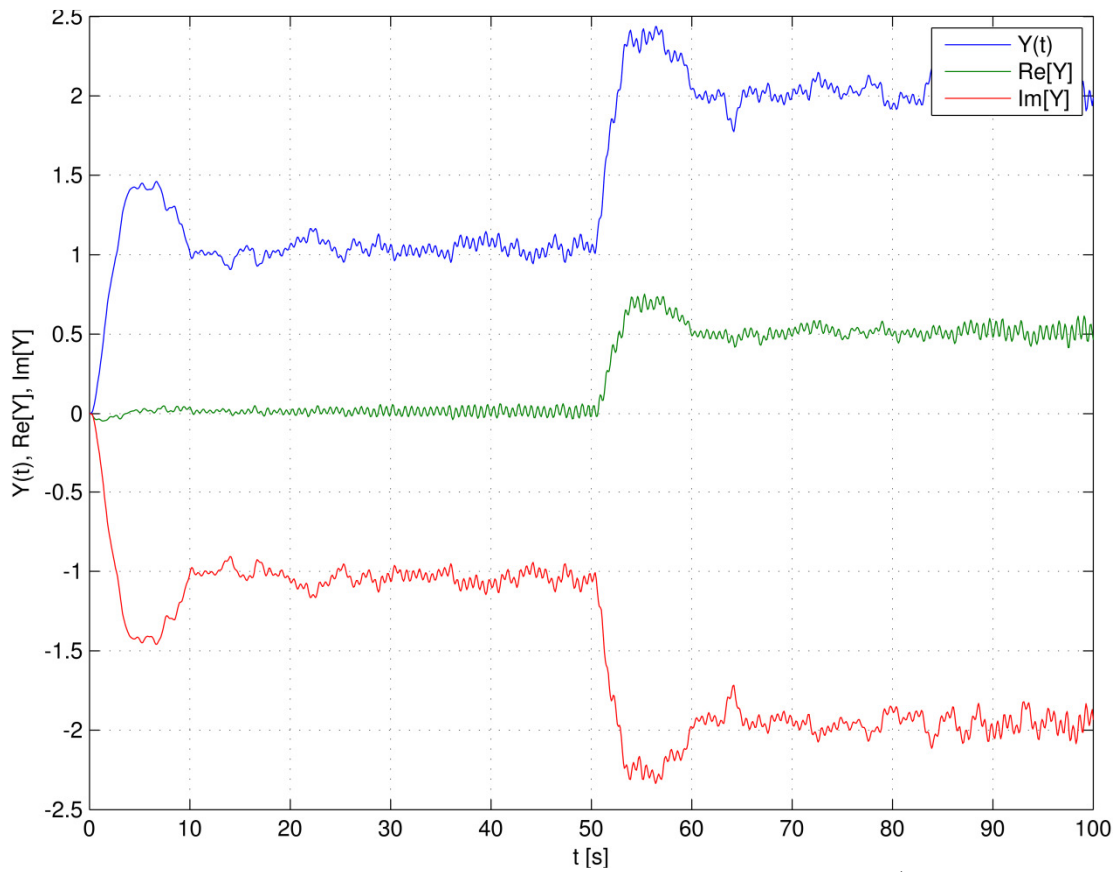
Obr. 5.13: Průběh amplitudy a reálné a imaginární rozklad složky  $u(t)$  o frekvenci  $5 \text{ rad} \cdot \text{s}^{-1}$

Amplitudy a reálné a imaginární rozklady obou zkoumaných složek řízené veličiny  $y(t)$  nevykazují v ustáleném stavu žádné kmity a hodnota amplitudy koresponduje bez znatelných odchylek se vztahem (10) z kapitoly 4.2.

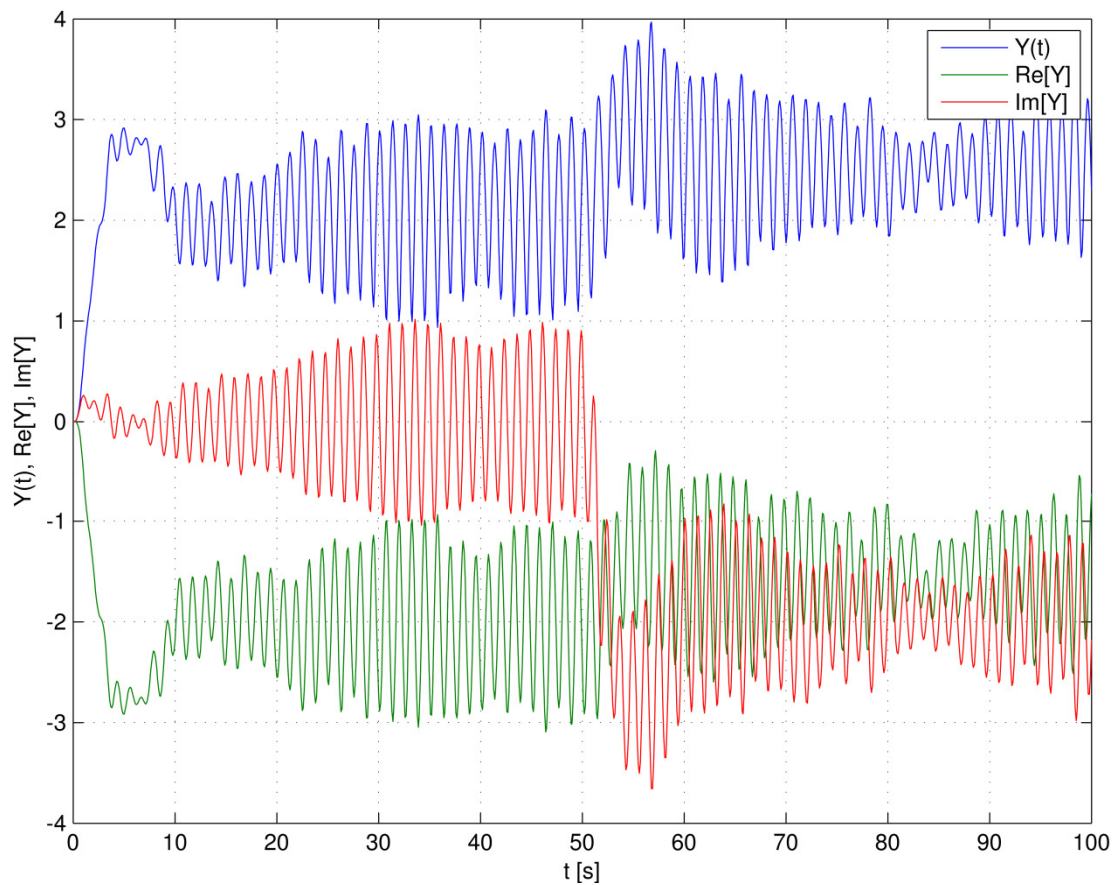
V dalším pokusu bude již do regulační smyčky zavedena externí porucha  $d(t)$ . Zpětná vazba zůstane beze změny. Na Obr. 5.14 je vidět, že řízená veličina je působením poruchy postižena stejně jako v předchozím případě. V RDFT obou zkoumaných složek řízené veličiny  $y(t)$  i akční veličiny  $u(t)$  by se měla v ustáleném stavu porucha projevit v podobě kmitů a změnou amplitudy. Grafy na Obr. 5.15 až Obr. 5.18 zachycují jednotlivé průběhy. Větší formát grafů je opět v části PŘÍLOHA pod B-3.



Obr. 5.14: Průběh žádané hodnoty a řízené veličiny

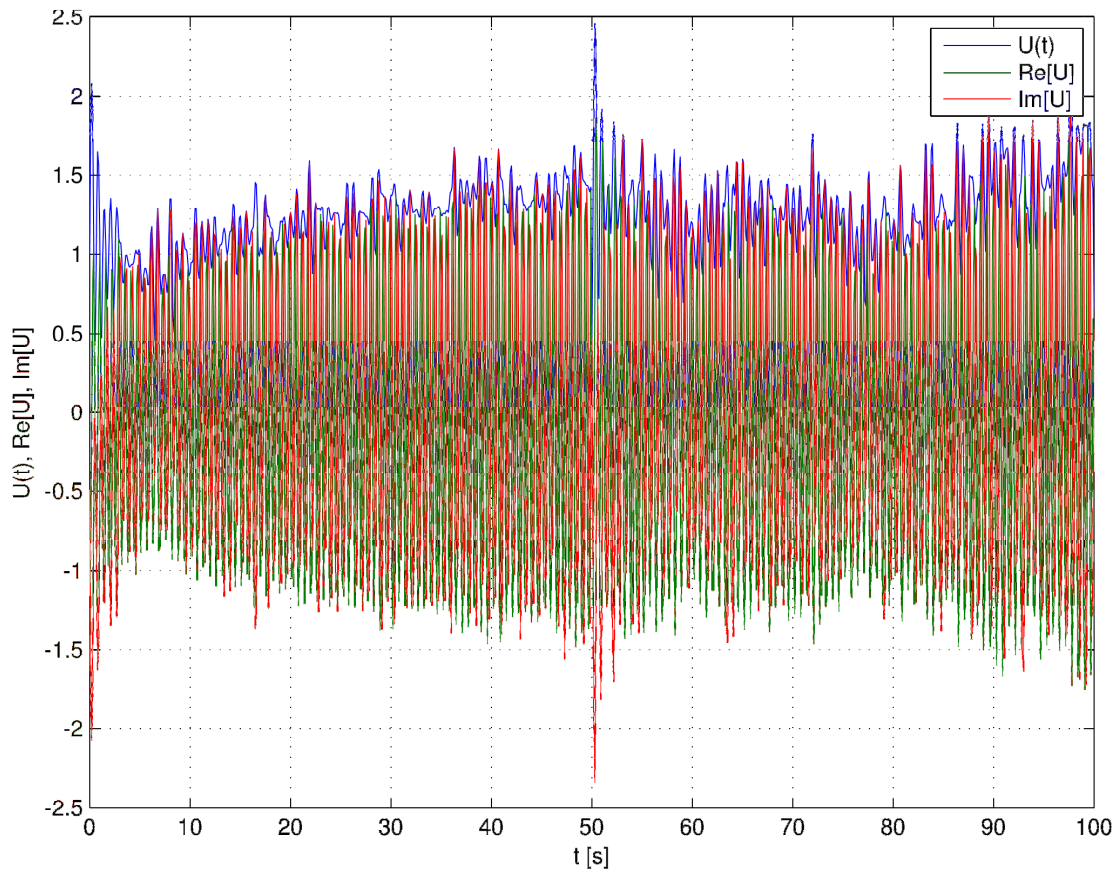


Obr. 5.15: Průběh amplitudy a reálné a imaginární složky  $y(t)$  o frekvenci  $10 \text{ rad} \cdot \text{s}^{-1}$  při působení poruchy

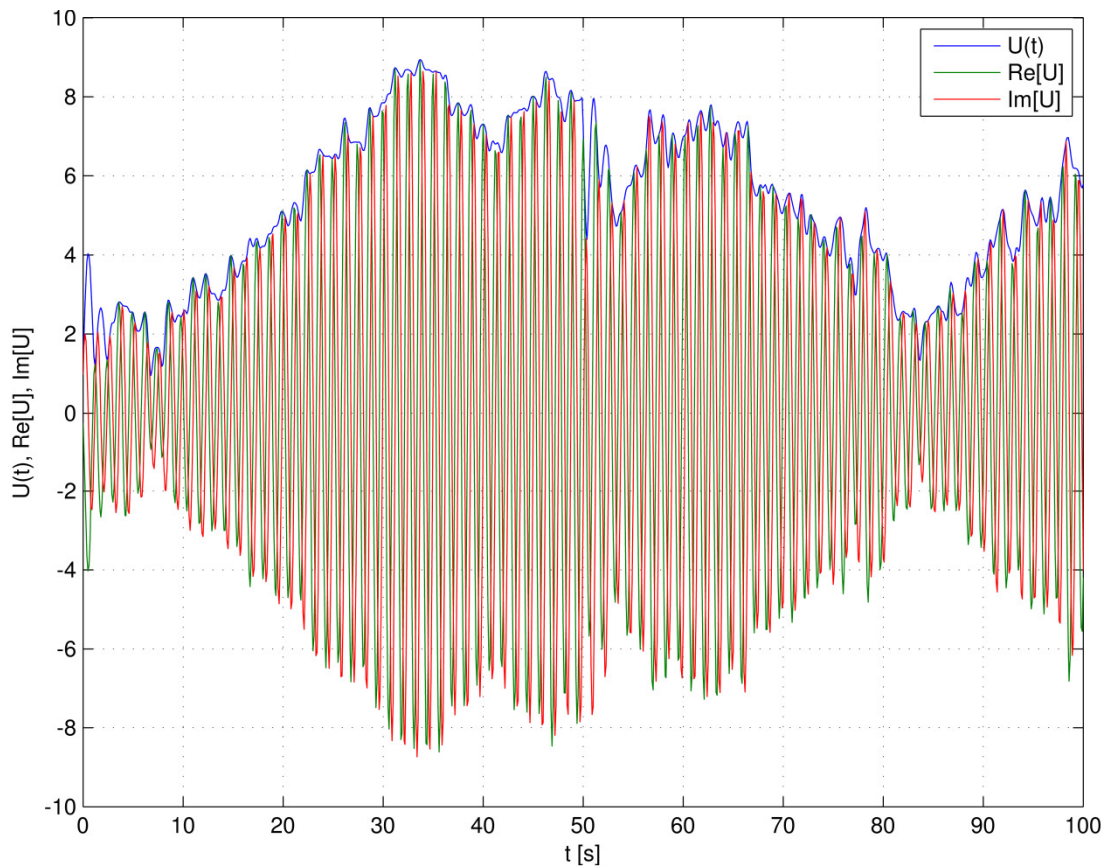


Obr. 5.16: Průběh amplitudy a reálné a imaginární složky  $y(t)$  o frekvenci  $5 \text{ rad} \cdot \text{s}^{-1}$  při působení poruchy



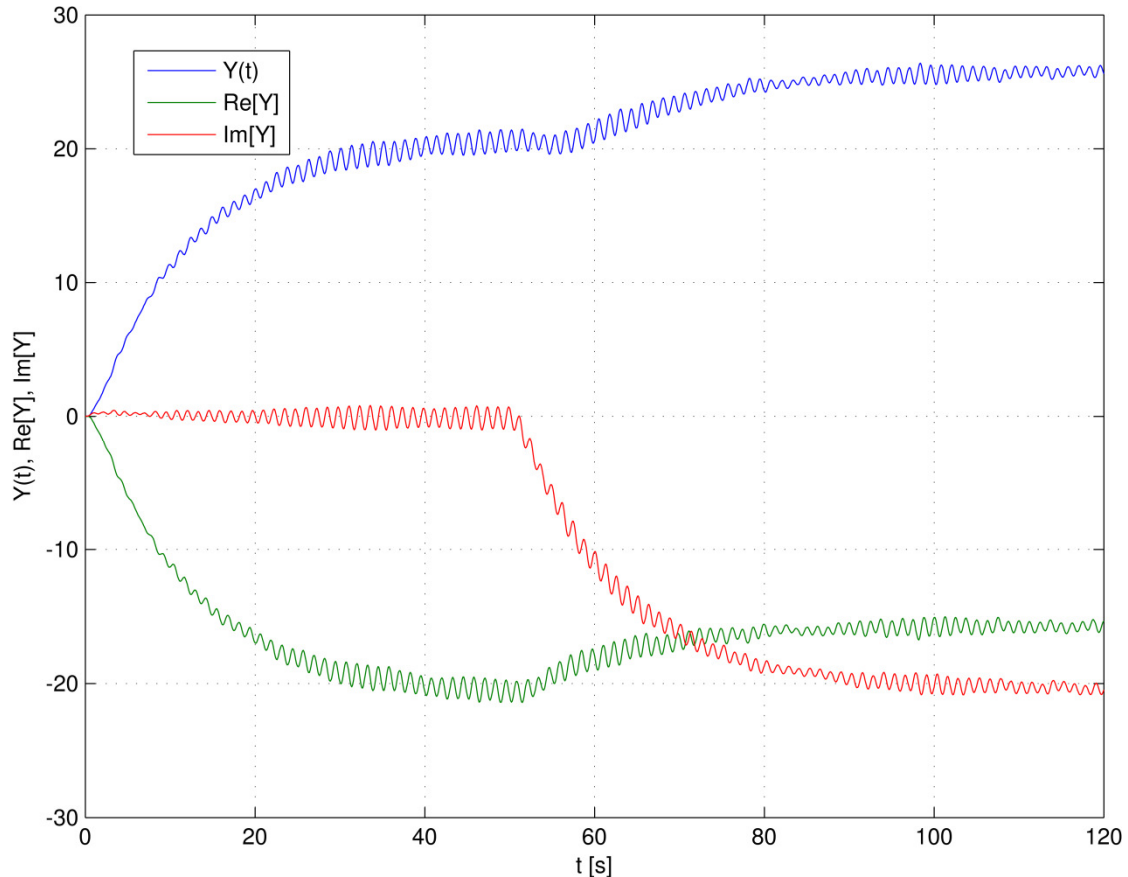


Obr. 5.17: Průběh amplitudy a reálné a imaginární složky  $u(t)$  o frekvenci  $10 \text{ rad} \cdot \text{s}^{-1}$  při působení poruchy



Obr. 5.18: Průběh amplitudy a reálné a imaginární složky  $u(t)$  o frekvenci  $5 \text{ rad} \cdot \text{s}^{-1}$  při působení poruchy

Při následujícím experimentu bude využit blok *RDFT* y k určení amplitudy složky řízené veličiny o frekvenci  $\omega_d = 5 \text{ rad} \cdot \text{s}^{-1}$  při působení externí poruchy a se zařazenou pásmovou zádrží ve zpětné vazbě. Experiment demonstruje postup měření  $A_y$  uvedený v kapitole 3.3 u vyčíslení indexu kvality. Výsledek je na Obr. 5.19, větší formát grafu je v části PŘÍLOHY pod B-3.



Obr. 5.19: Průběh amplitudy a reálné a imaginární složky  $y(t)$  o frekvenci  $5 \text{ rad} \cdot \text{s}^{-1}$  při působení poruchy a „rozpojené“ zpětné vazbě

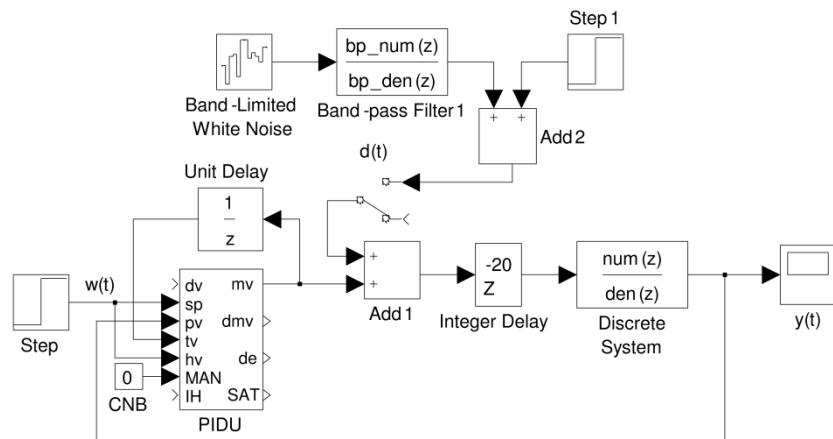
## 5.2 Ověření funkčních vlastností na LS druhého řádu s dopravním zpožděním

Druhá regulační smyčka na které byly bloky testovány sestává z LS druhého řádu s dominantním dopravním zpožděním a k němu navrženým PID regulátorem. Přenos systému

$$P(p) = \frac{3 \cdot e^{-2 \cdot p}}{p^2 + 5 \cdot p + 6}$$

je realizován blokem Transfer Function. Regulátor je opět realizován blokem PIDU z blocksetu REX Industrial Blockset. Obraz akční veličiny regulátoru je

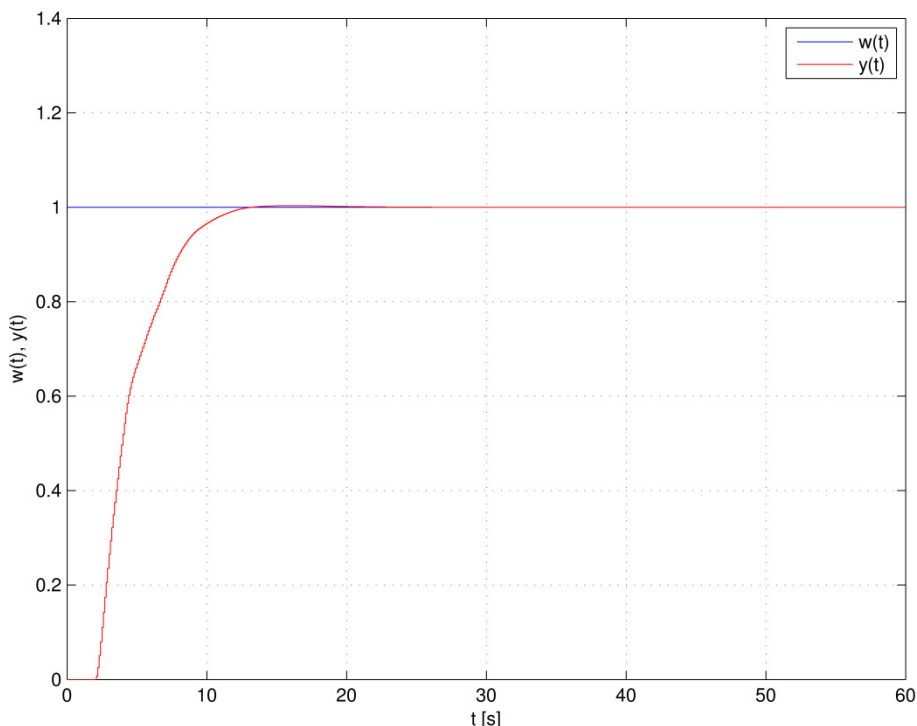
$$U(p) = 0.5579 \cdot \left\{ W(p) - Y(p) + \frac{1}{1.3233 \cdot p} \cdot [W(p) - Y(p)] + \frac{0.3308 \cdot p}{0.3308 \cdot p + 1} \cdot [0 - Y(p)] \right\}.$$



Obr. 5.20: Schéma zapojení regulační smyčky 2

Jediný rozdíl ve schématu zapojení této regulační smyčky od schématu na Obr. 5.1 je blok Integer Delay realizující dopravní zpoždění (viz Obr. 5.20). Přepínač  $d(t)$  opět umožňuje zavedení externí poruchy do regulační smyčky. Ta je jako v předchozích případech tvořena součtem skokové poruchy a bílého šumu, z něhož je pásmovou propustí vybrána složka o frekvenci  $\omega_d = 5 \text{ rad} \cdot \text{s}^{-1}$ .

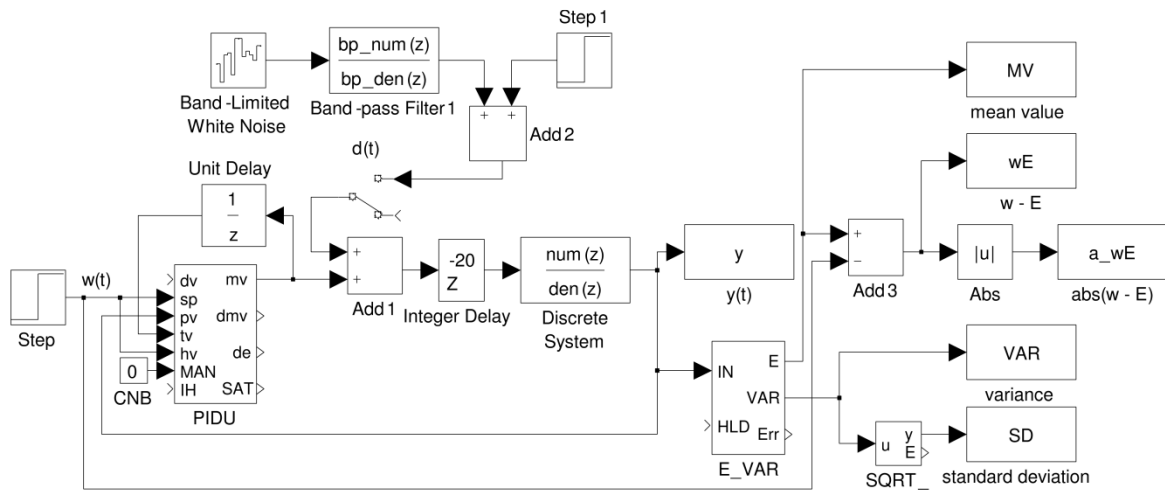
PID regulátor je navržen tak, aby uspokojivě reguloval diskretizovaný systém na hodnotu  $w(t)$  nastavovanou blokem Step (viz Obr. 5.21).



Obr. 5.21: Průběh žádané hodnoty a řízené veličiny

### 5.2.1 Blok pro odhad střední hodnoty a variance

Na Obr. 5.22 je schéma zapojení pro ověření funkčních vlastností bloku E\_VAR.



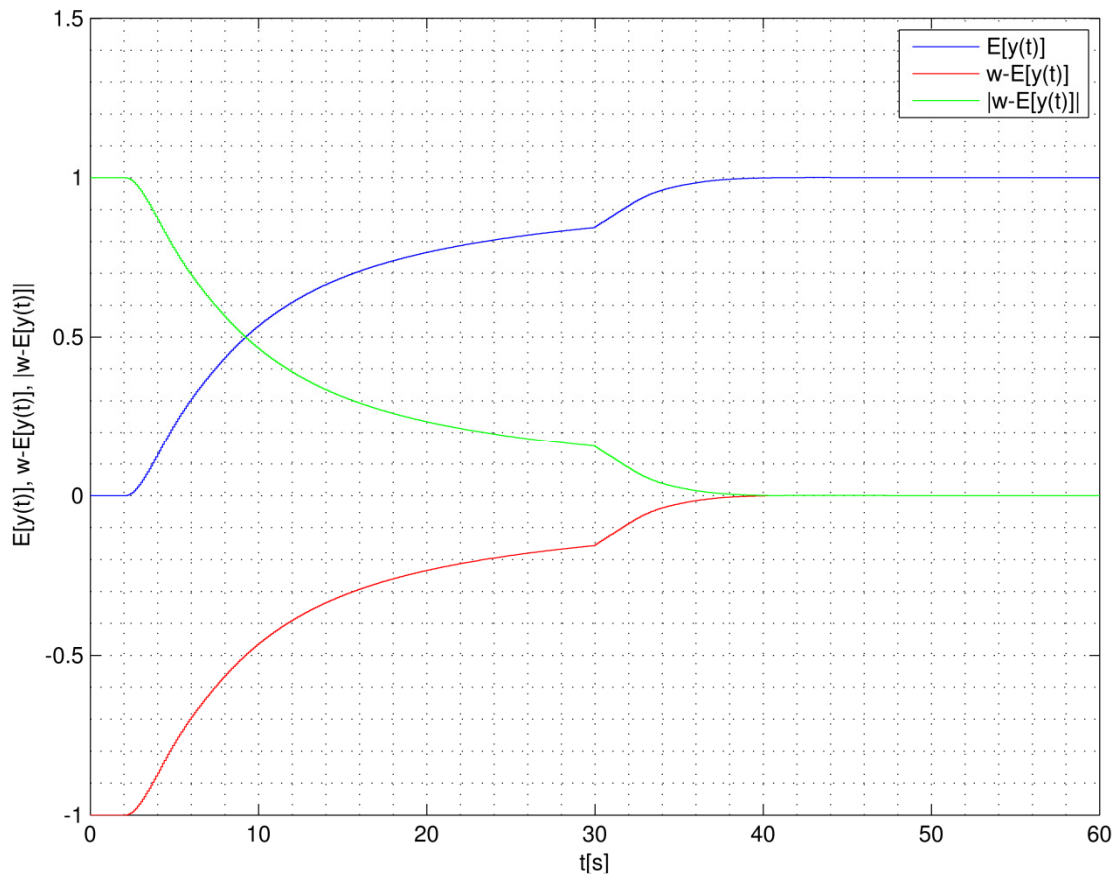
Obr. 5.22: Schéma zapojení pro ověření funkčních vlastností bloku E\_VAR

Nejprve bude simulace spuštěna bez zavedení poruchy do regulační myčky. Očekávané výsledky experimentu jsou

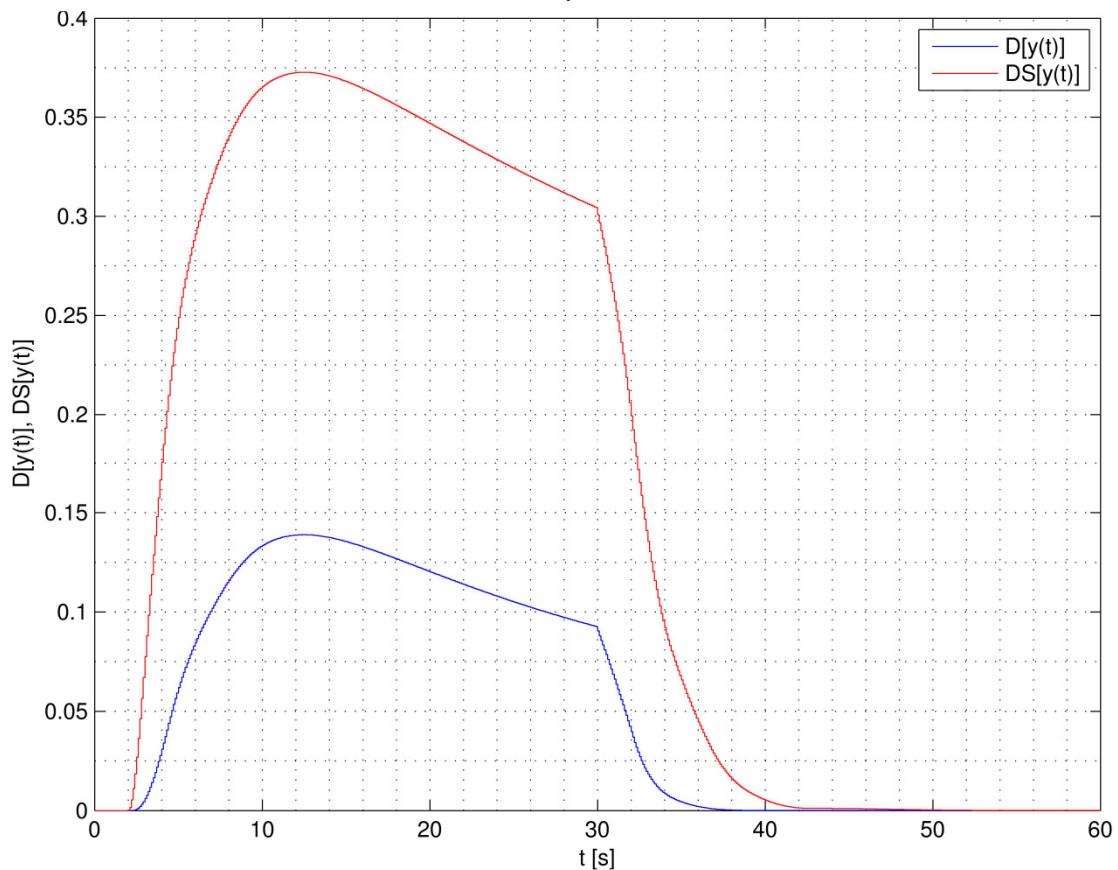
$$E[y(t)] = w(t) = 1, w(t) - E[y(t)] = |w(t) - E[y(t)]| = 0,$$

$$VAR[y(t)] = 0, \delta = 0.$$

Průběhy získané simulací jsou na Obr. 5.23 a Obr. 5.24. Hodnoty v ustáleném stavu odpovídají očekávaným. Celostránkové verze obrázků jsou v části PŘÍLOHA pod B-4.



Obr. 5.23: Průběh střední hodnoty  $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu



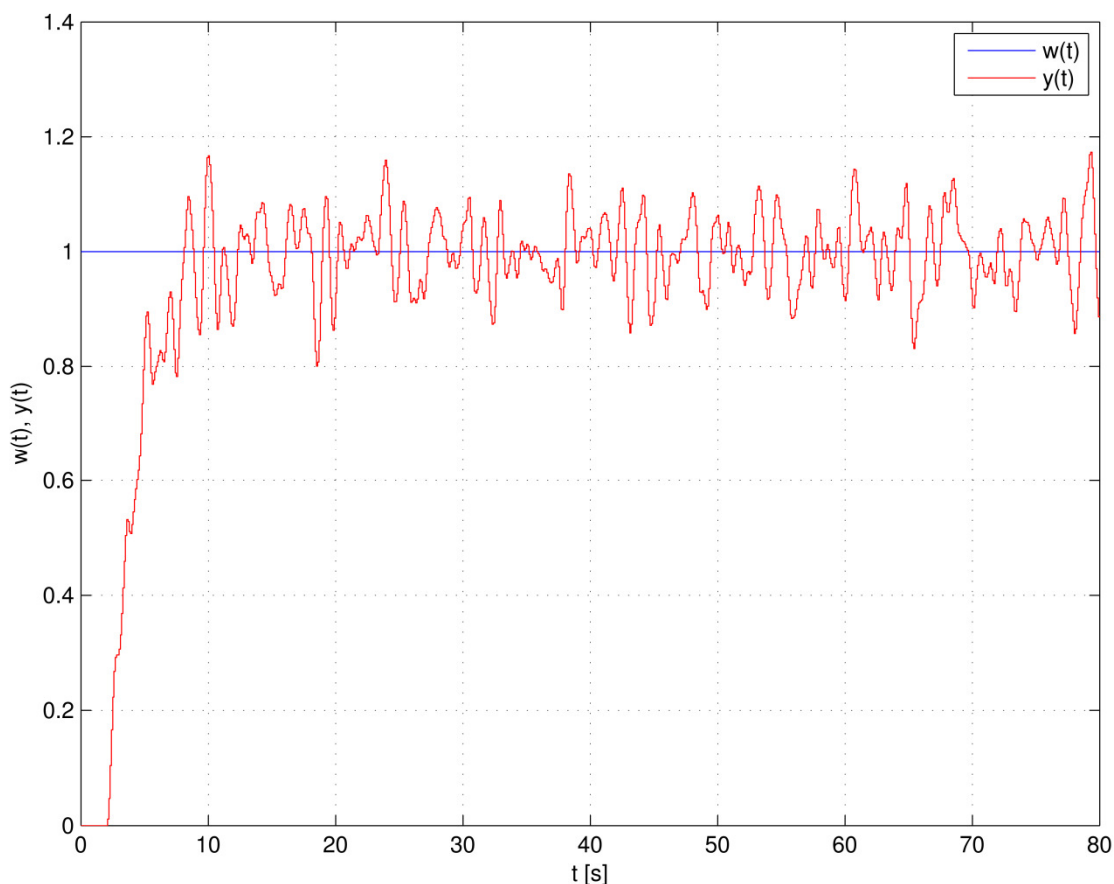
Obr. 5.24: Průběh rozptylu a směrodatné odchylky

V druhém experimentu bude simulace provedena se zavedenou externí poruchou. Očekávané chování bloku je

$$E[y(t)] \neq w(t), w(t) - E[y(t)] \neq 0, |w(t) - E[y(t)]| > 0,$$

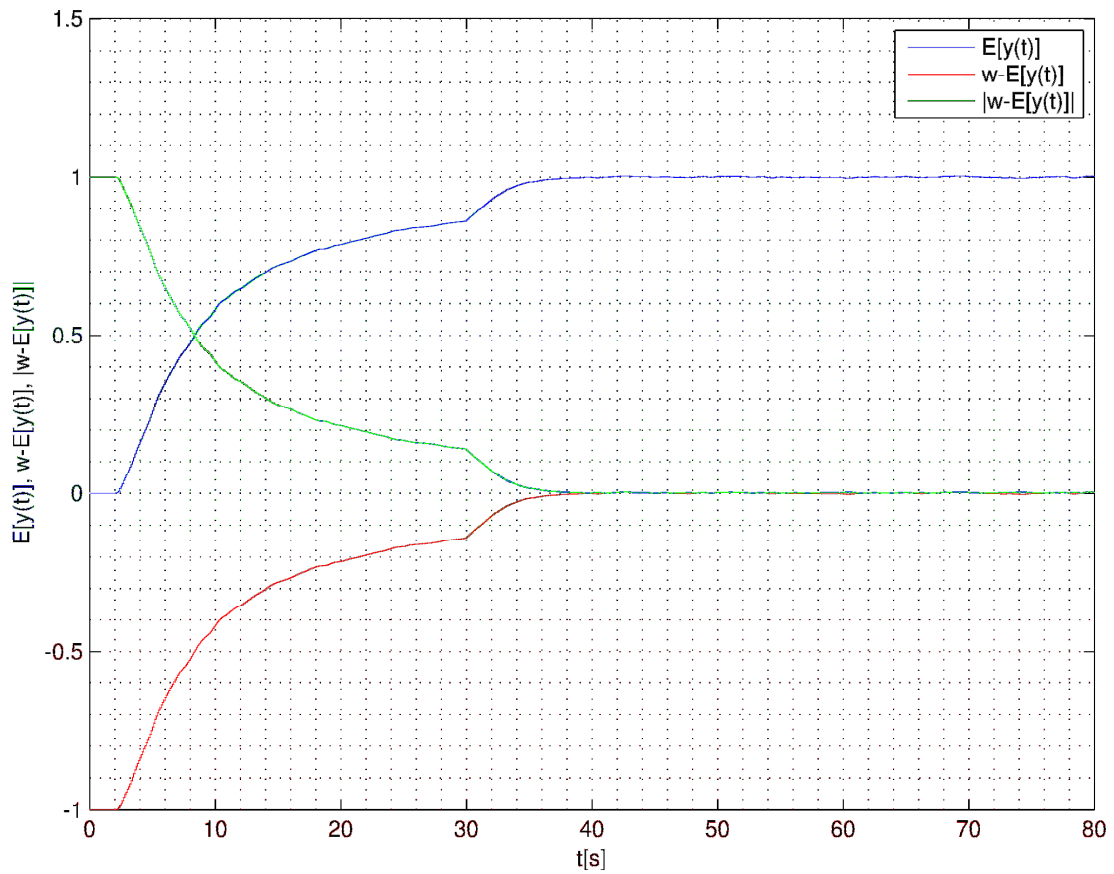
$$VAR[y(t)] > 0, \delta > 0.$$

Na Obr. 5.25 je zřetelně vidět změna průběhu řízené veličiny zapříčiněná poruchou.

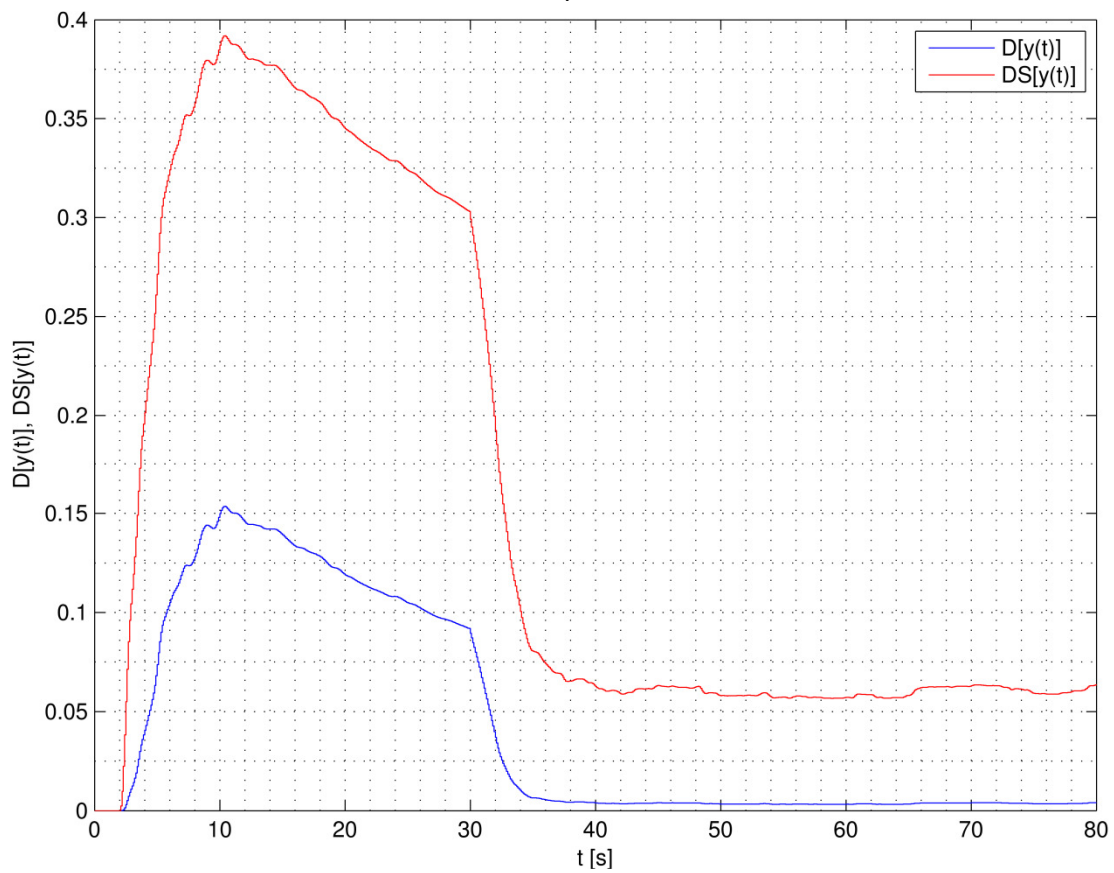


Obr. 5.25: Průběh žádané hodnoty a řízené veličiny při působení poruchy

Průběhy  $E[y(t)]$ ,  $w(t) - E[y(t)]$ ,  $|w(t) - E[y(t)]|$  na Obr. 5.26 se téměř nezměnily, v ustáleném stavu vykazují jen sotva znatelné odchylky korespondující s očekáváním. Na průběh  $VAR[y(t)]$  (Obr. 5.27) není v ustáleném stavu poznat změna oproti předchozímu stavu, ale při pohledu na průběh směrodatné odchylky (Obr. 5.27) je jasné, že k očekávaným změnám skutečně došlo. Kmity řízené veličiny mají v ustáleném stavu amplitudu dostatečně malou na to, aby její vliv nebyl na průběhu  $VAR[y(t)]$  znát. Větší formát grafů je opět v části PŘÍLOHA pod B-4.



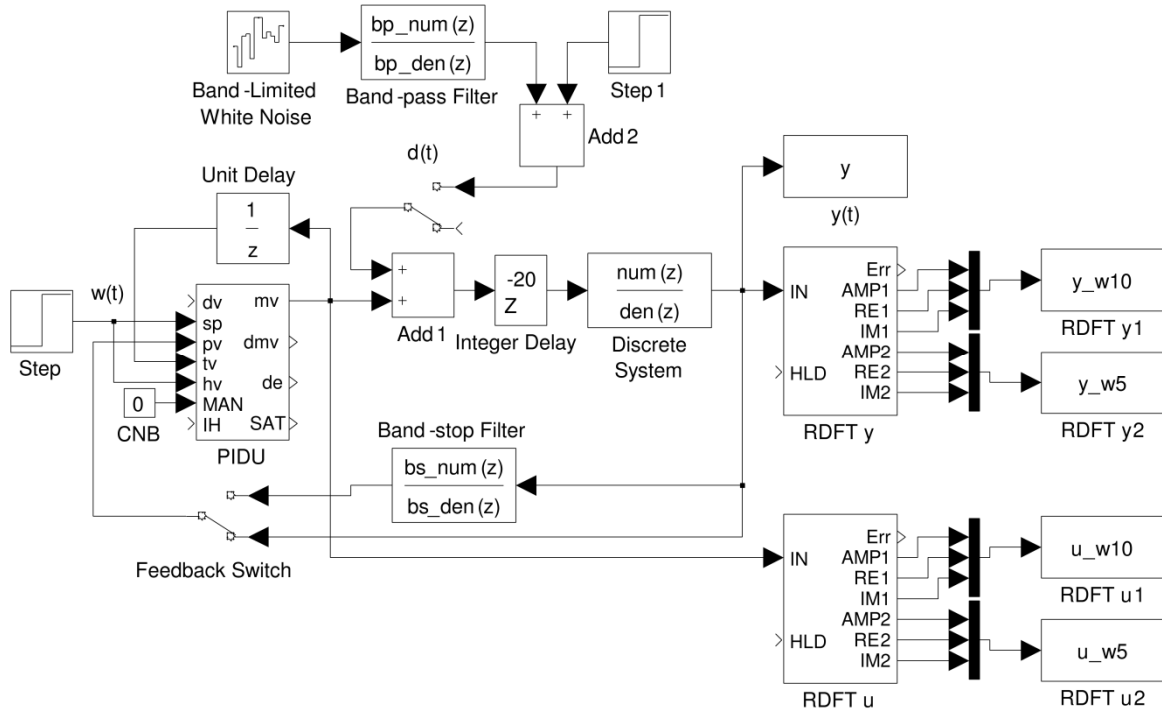
Obr. 5.26: Průběh střední hodnoty  $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu



Obr. 5.27: Průběh rozptylu a směrodatné odchylky

## 5.2.2 Blok pro výpočet běžící diskrétní Fourierovy transformace

Zapojení použité k experimentům při ověřování funkčních vlastností bloku RDFT na druhé regulační smyčce je znázorněno schématem na Obr. 5.28. Žádaná hodnota  $i$  odezva na ni je stejná jako na Obr. 5.21. Umístění a využití bloků RDFT je stejné jako v kapitole 5.1.2.

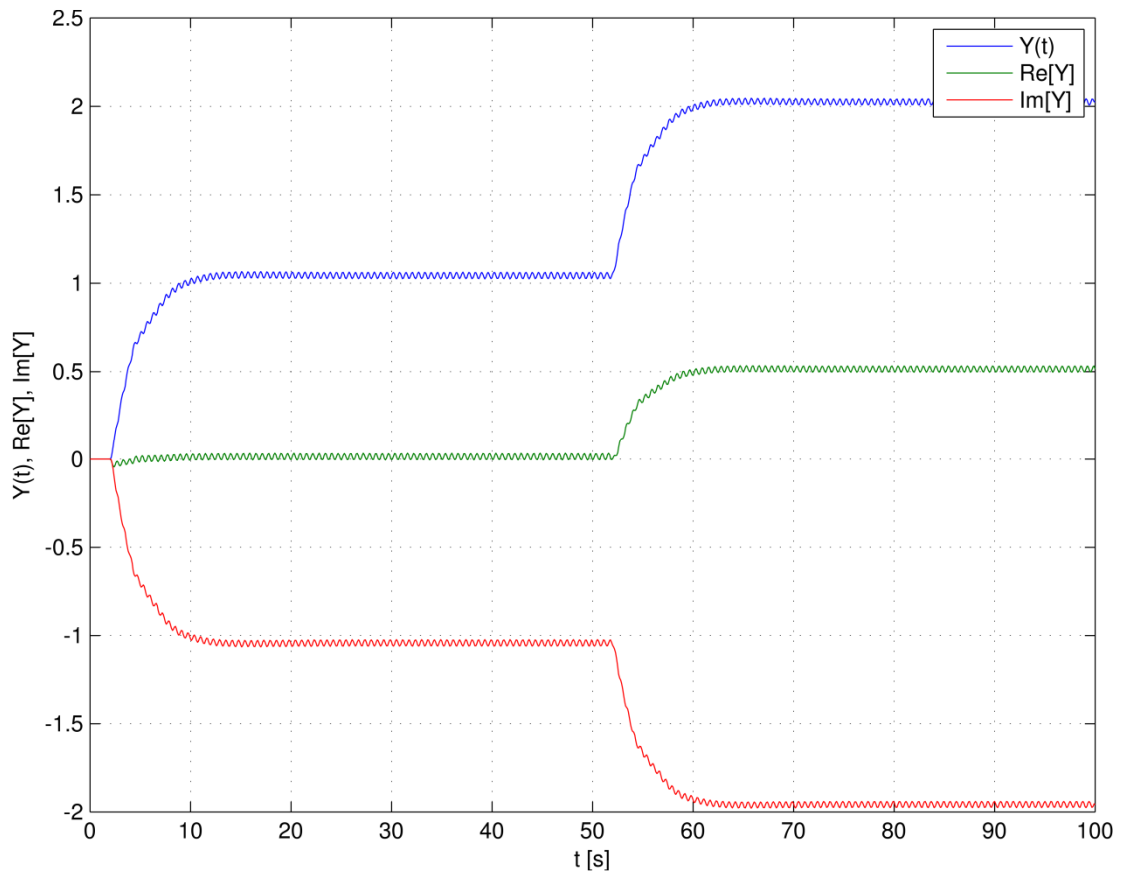


Obr. 5.28: Schéma zapojení pro ověření funkčních vlastností bloku RDFT

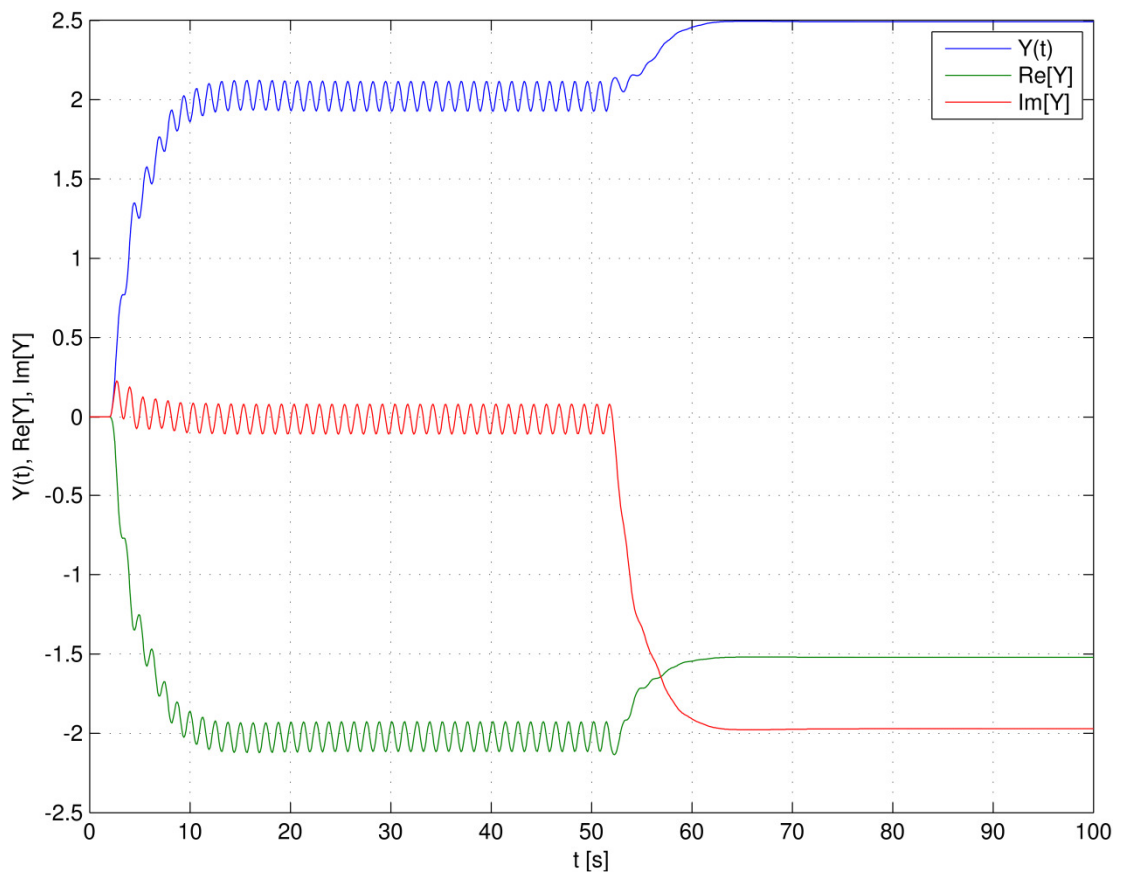
První pokus je opět proveden bez zavedené poruchy a bez zařazení pásmové zádrže ve zpětné vazbě. Průběhy získané bloky *RDFT y* a *RDFT u* poslouží ke srovnání s výsledky dalších pokusů. Následující grafy na Obr. 5.29 až Obr. 5.32 zachycují zmíněné průběhy. Větší formát grafů je opět v části PŘÍLOHA pod B-4. Amplituda a reálný a imaginární rozklad jedné ze zkoumaných složek řízené veličiny  $y(t)$  vykazují v ustáleném stavu kmitu způsobené zřejmě dopravním zpožděním systému.

V dalším pokusu bude již do regulační smyčky zavedena externí porucha  $d(t)$ . Zpětná vazba zůstane beze změny. V RDFT obou zkoumaných složek řízené veličiny  $y(t)$  i akční veličiny  $u(t)$  by se měla v ustáleném stavu porucha projevit změnou amplitudy. Grafy na Obr. 5.33 až Obr. 5.36 zachycují jednotlivé průběhy. Větší formát grafů je opět v části PŘÍLOHA pod B-4.

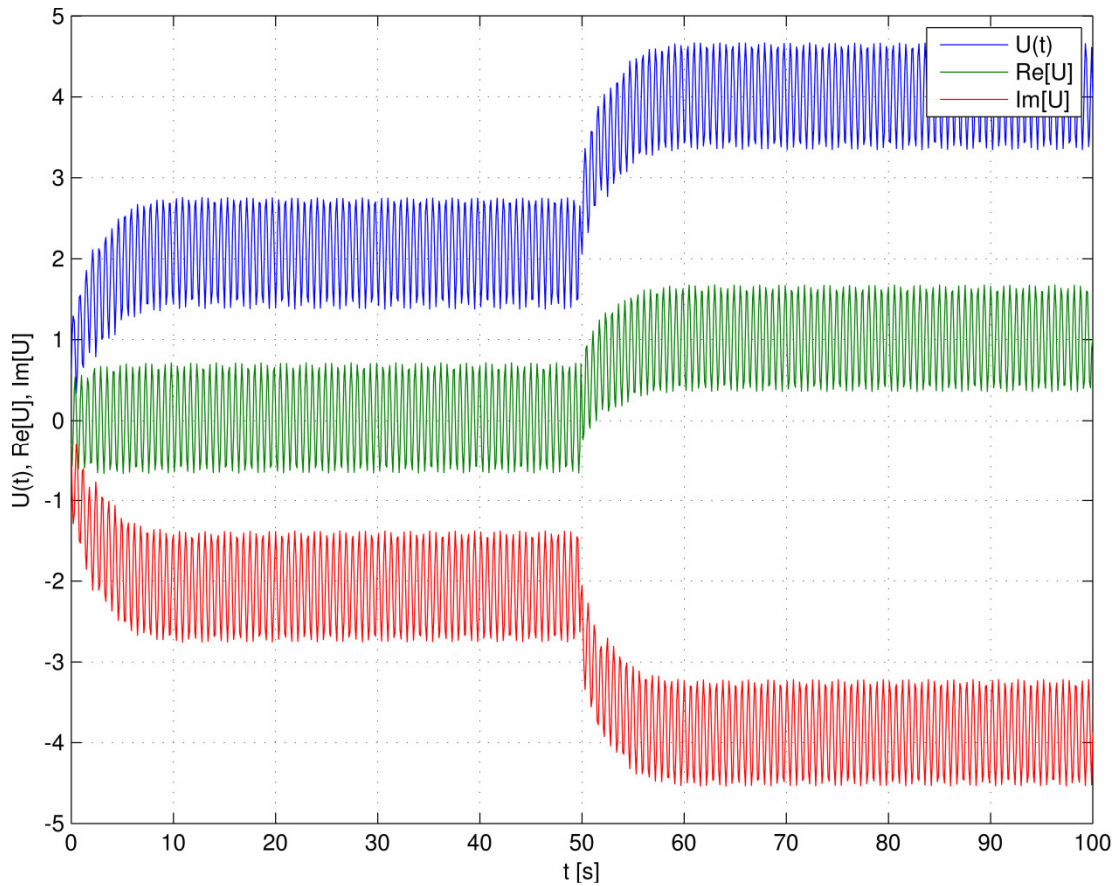




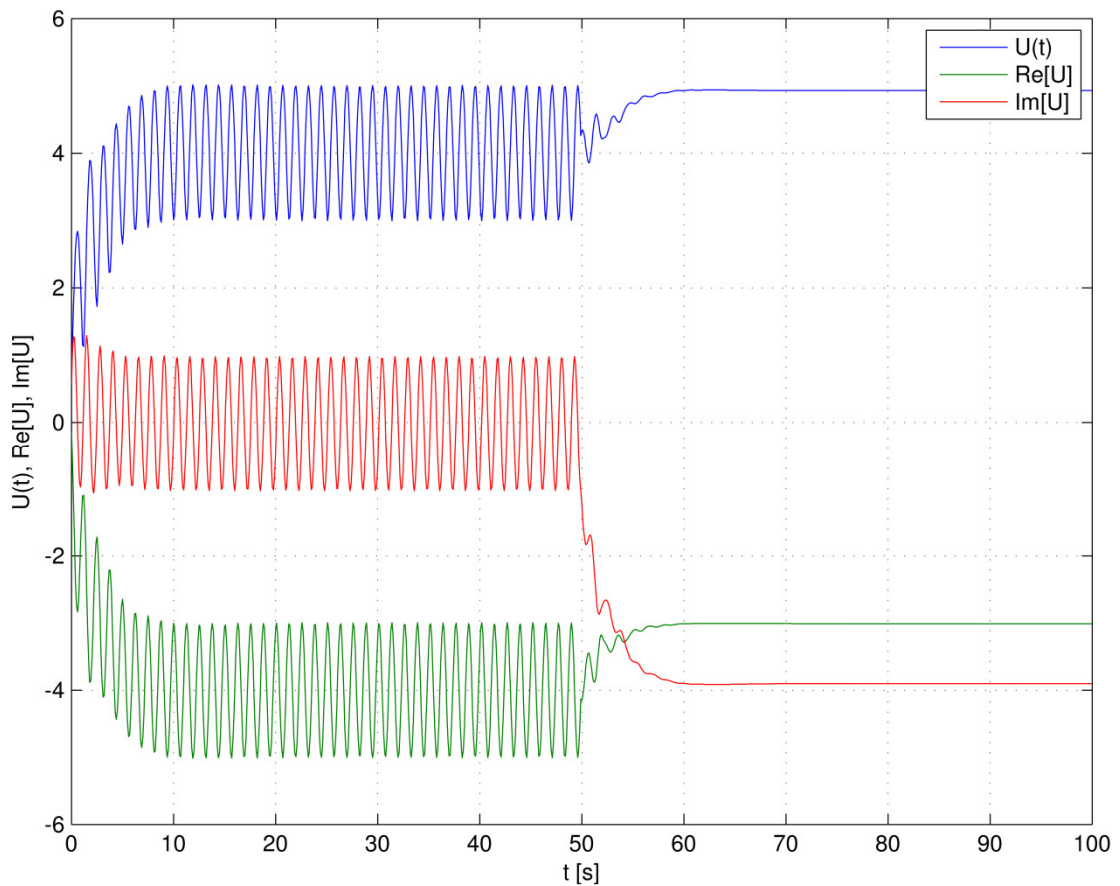
Obr. 5.29: Průběh amplitudy a reálné a imaginární složky  $y(t)$  o frekvenci  $10 \text{ rad} \cdot \text{s}^{-1}$



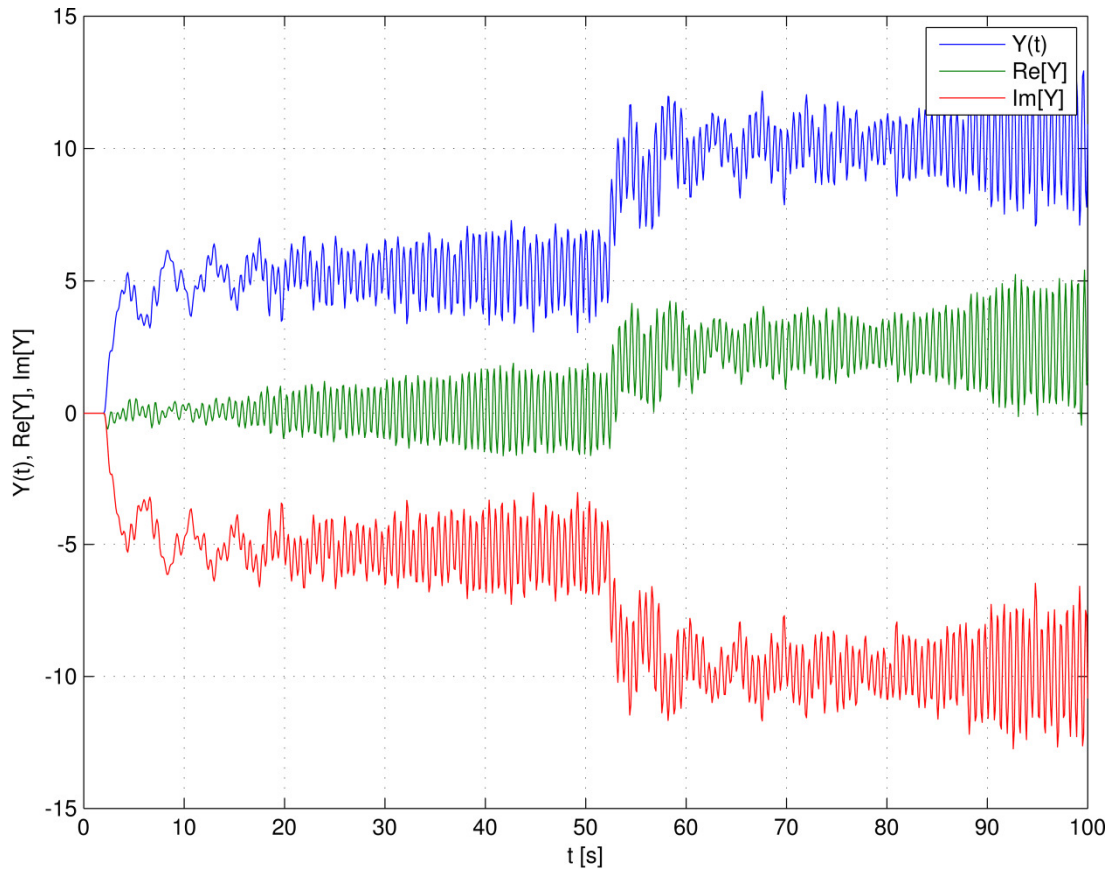
Obr. 5.30: Průběh amplitudy a reálné a imaginární složky  $y(t)$  o frekvenci  $5 \text{ rad} \cdot \text{s}^{-1}$



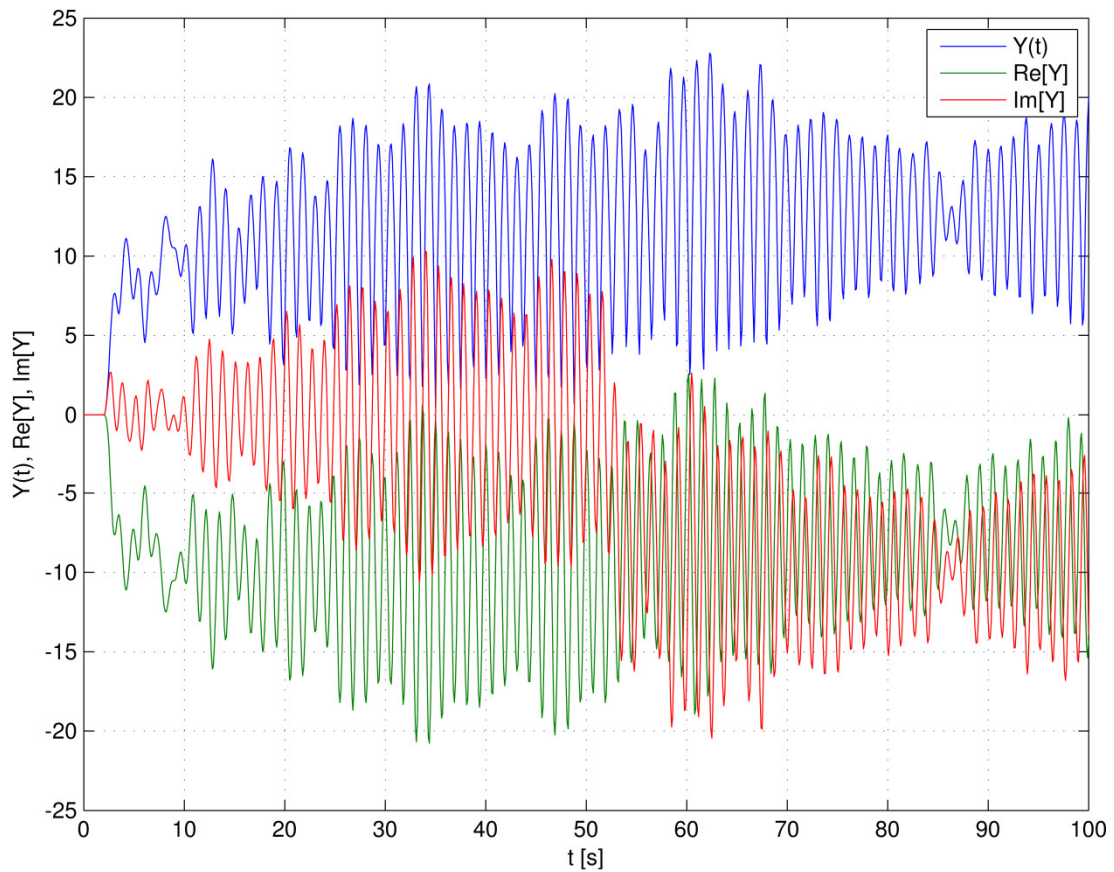
Obr. 5.31: Průběh amplitudy a reálné a imaginární složky  $u(t)$  o frekvenci  $10 \text{ rad} \cdot \text{s}^{-1}$



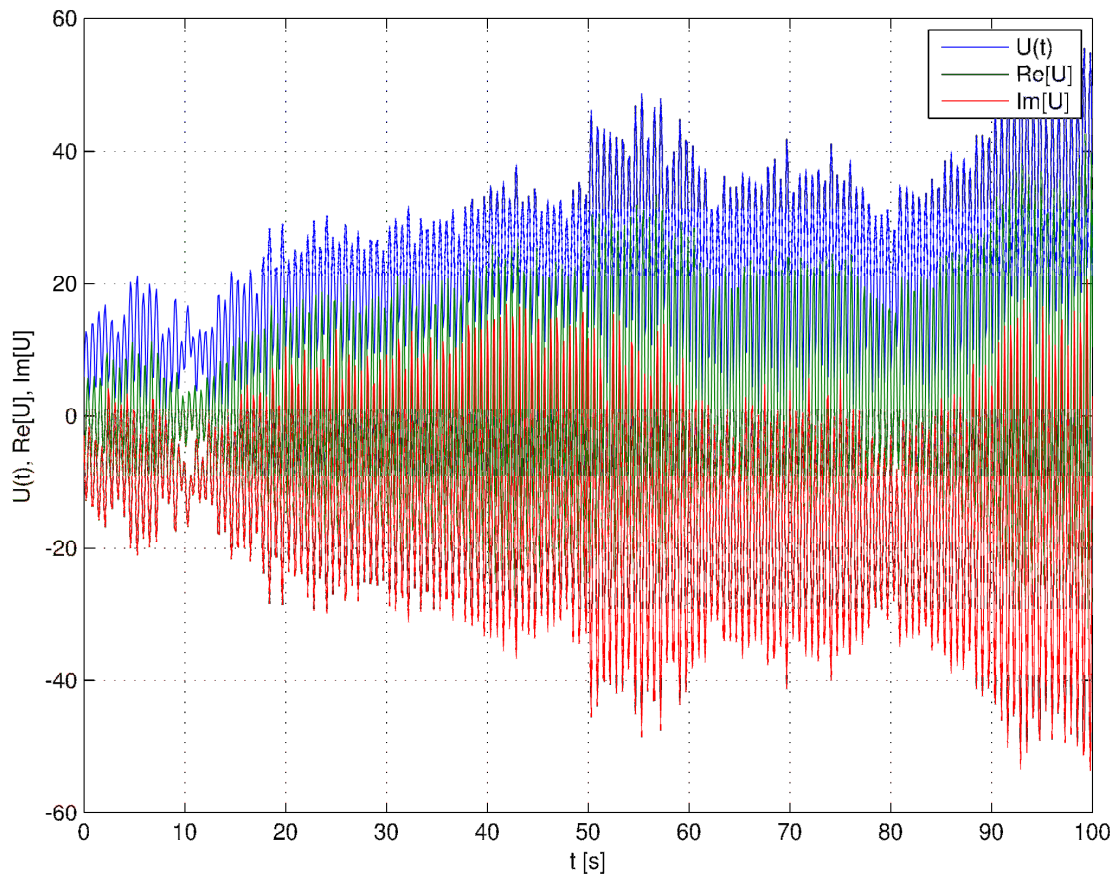
Obr. 5.32: Průběh amplitudy a reálné a imaginární složky  $u(t)$  o frekvenci  $5 \text{ rad} \cdot \text{s}^{-1}$



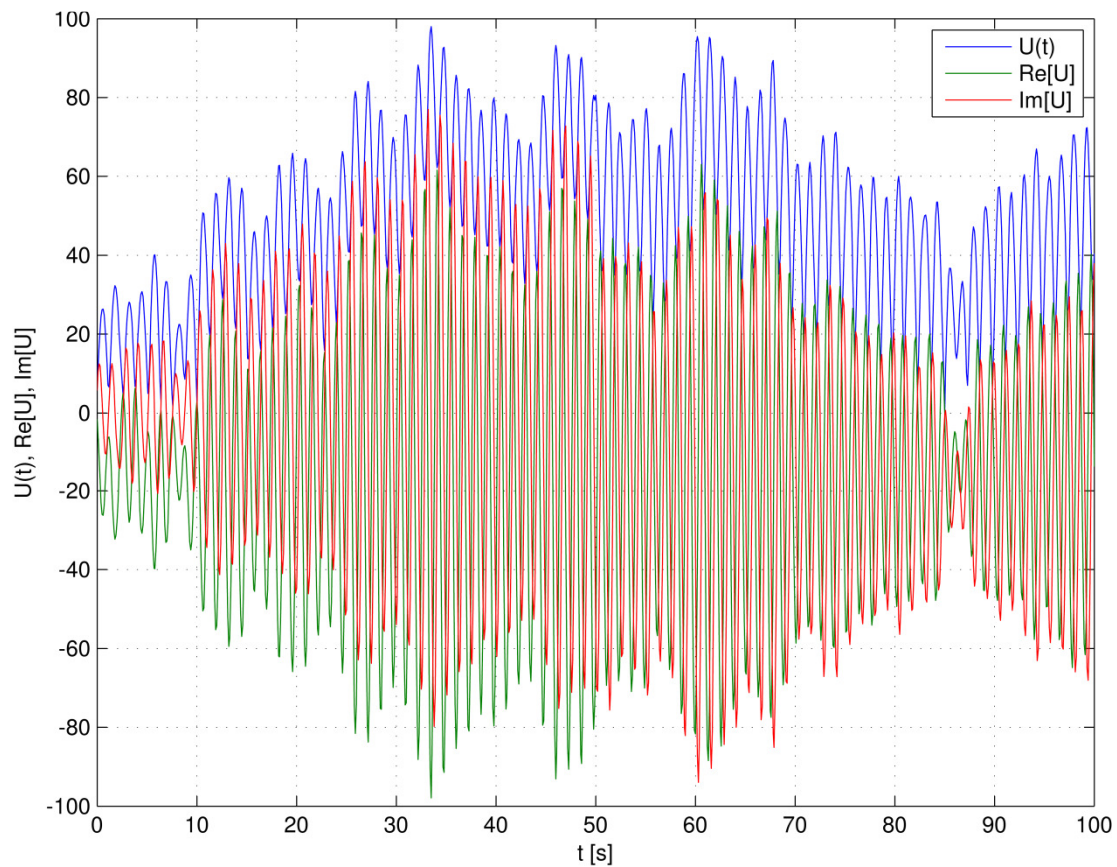
Obr. 5.33: Průběh amplitudy a reálné a imaginární složky  $y(t)$  o frekvenci  $10 \text{ rad} \cdot \text{s}^{-1}$  při působení poruchy



Obr. 5.34: Průběh amplitudy a reálné a imaginární složky  $y(t)$  o frekvenci  $5 \text{ rad} \cdot \text{s}^{-1}$  při působení poruchy

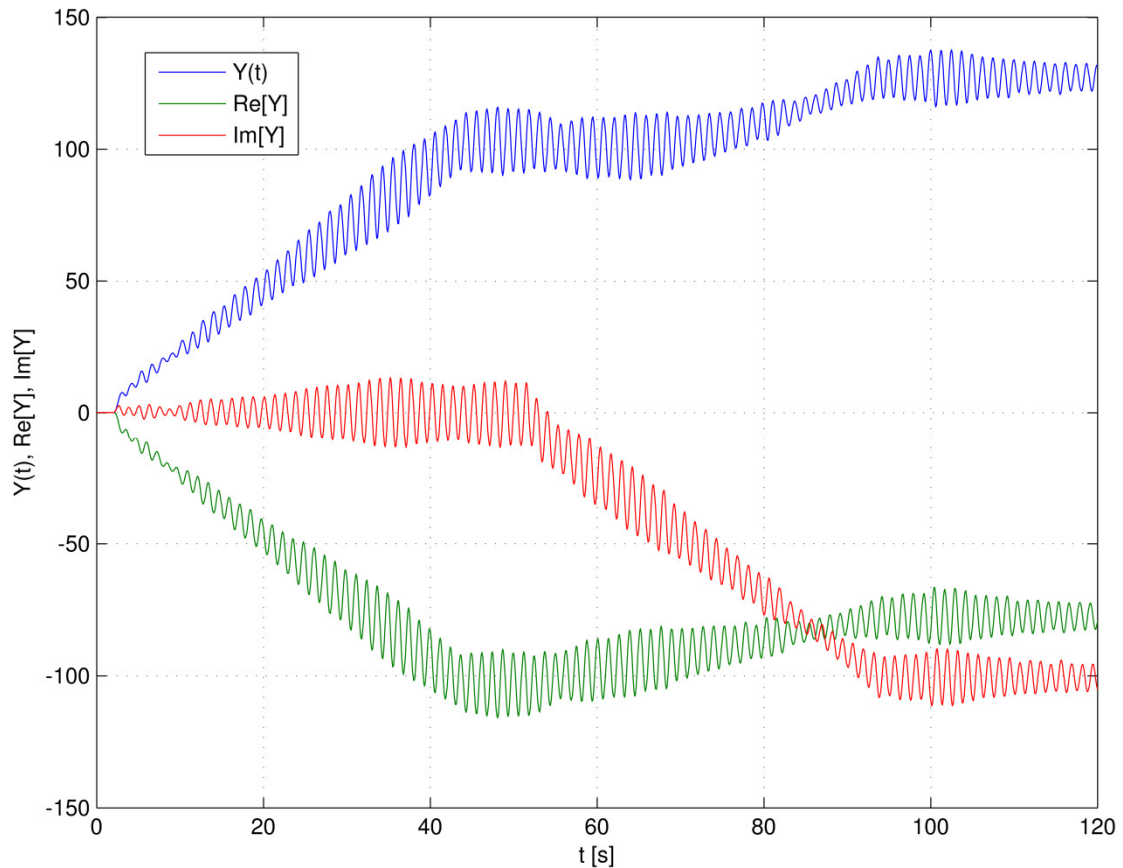


Obr. 5.35: Průběh amplitudy a reálné a imaginární složky  $u(t)$  o frekvenci  $10 \text{ rad} \cdot \text{s}^{-1}$  při působení poruchy



Obr. 5.36: Průběh amplitudy a reálné a imaginární složky  $u(t)$  o frekvenci  $5 \text{ rad} \cdot \text{s}^{-1}$  při působení poruchy

Graf na Obr. 5.37 znázorňuje amplitudu a reálnou a imaginární složku řízené veličiny o frekvenci  $\omega_d = 5 \text{ rad} \cdot \text{s}^{-1}$ . Jedná se opět o demonstraci postupu měření  $A_y$  uvedeného v kapitole 3.3 u vyčíslení indexu kvality. Výsledek je na Obr. 5.19, větší formát grafu je v části PŘÍLOHY pod B-4.



Obr. 5.37: Průběh amplitudy a reálné a imaginární složky  $y(t)$  o frekvenci  $5 \text{ rad} \cdot \text{s}^{-1}$  při působení poruchy a „rozpojené“ zpětné vazbě

## 6 ZÁVĚR

Podařilo se úspěšně navrhnout a implementovat dva bloky použitelné v nadstavbě Simulink softwaru MATLAB a v řídicím systému REX Control System. Oba bloky se ukázaly jako vhodné diagnostické nástroje při sledování a oceňování kvality regulace pomocí metod uvedených v této práci.

V kontextu s výsledky simulačního ověření funkčních vlastností se blok E\_VAR ukázal jako vhodný pro jednoduché sledování kvality regulace. K jeho provozu stačí pouze měřit řízenou veličinu (případně ještě požadovanou hodnotu). Z metod sledování kvality, pro které byl blok navrhován, se jako nejméně vhodné jeví určování indexu kvality na základě střední hodnoty neboť nezahrnuje informaci o požadované hodnotě. Identifikuje kmity řízené veličiny ale pokud je v ustáleném stavu řízená veličiny konstantní není tato metoda schopna odhalit jakékoliv nedostatky. Navíc nedokáže identifikovat harmonické kmity řízené veličiny, neboť ty by nezpůsobily změnu střední hodnoty. Pokud by ovšem stačilo sledovat a hodnotit kvalitu procesu bez srovnání s požadovanou hodnotou (např. v případě, že je správné nastavení regulátoru jisté a zajímá nás přítomnost poruchy) je stejně lepší volbou metoda sledování kvality rozptylem. Ta však nemusí podat informaci o kmitech nízké amplitudy. Ještě vhodnější pro tento účel by proto bylo sledování kvality regulace směrodatnou odchylkou.

Blok RDFT a metoda, pro níž byl navržen, je aplikovatelná jen v případě, že známe frekvenci externí poruchy vůči které chceme index kvality vymezit. Navíc vyžaduje určení amplitud spektrální analýzou řízené veličiny. K tomu je potřeba „rozpojit“ zpětnou vazbu pásmovou zádrží.

## A LITERATURA

- [1] SCHLEGEL, M. *Průmyslové PID regulátory : Tutorial*. REX Controls [online] 2009. [cit. 9.7.2012]. Dostupný z WWW: <[http://www.rexcontrols.cz/downloads/clanky/PIDTutor\\_CZ.pdf](http://www.rexcontrols.cz/downloads/clanky/PIDTutor_CZ.pdf)>.
- [2] SCHLEGEL, M. Kvalita řízení procesů – Teoretická a praktická omezení. In *Sborník přednášek z technické konference Automatizace, regulace a procesy*. Praha: DIMART, 2010.
- [3] SCHLEGEL, M. a kol. *Frekvenční identifikace monotónních procesů*. Výzkumná zpráva. Plzeň: Západočeská univerzita, Fakulta aplikovaných věd, 1999.

## B PŘÍLOHY

### B-1 Zdrojový kód algoritmu bloku pro odhad střední hodnoty a variance

```
#define BUFFER_L 1000          //velikost archivu vstup. hodnot
double input(0) in;           //vstup - signal
double output(0) outMV;       //vystup - str. hodnota
double output(1) outRMS;      //vystup - variance
long parameter(0) N;          //parametr - pocet vzorku
double buffer[BUFFER_L];      //archiv vstupnich hodnot
long head = 0, count = 0, phase = 0; //hlava zapisu do bufferu, citac
                                //pruchodu alg., prepinač faze alg.
double s1 = 0.0, s2 = 0.0, rms1 = 0.0, rms2 = 0.0; //promenne vypoctu

/*Operace vlozeni prvku do cyklickeho bufferu. Koriguje hodnotu pointeru
head tak, aby neodkazoval mimo rozsah bufferu.
*/
int push(double value) {
    buffer[head++] = value;
    if(head >= BUFFER_L) {
        head = 0;
    }
    return 0;
}

/*Vraci prvek, který je v bufferu o n pozic v minulosti. Index prvku, který
se ma precist, je ziskan odectenim parametru n od pozice zapisove hlavy
head. Ziskany index je korigovan tak, aby neodkazoval mimo rozsah bufferu.
*/
double getAt(int n) {
    int index = head - n;
    if(index < 0) {
        index += BUFFER_L;
    }
    return buffer[index];
}
```



/\*Algoritmus v nulte fazi nasbira prvnych N vzorku vstupniho signalu, jehož stredni hodnota a rozptyl se ma vypocitat. Nasledne alg. prepina mezi fazemi 1 a 2, které obe rekurzivne odhaduji z N vzorku stredni hodnotu a rozptyl signalu. Tento mechanismus zabranuje akumulaci numerickych chyb.  
\*/

```
int main(void) {
    ++count;
    push(in);
    switch(phase) {
    case 0: //sbira prvnych N vzorku signalu potrebných pro
            //zacatek odhadu stredni hodnoty a rozptylu
            s1 = s1 + in;
            outMV = s1/count;
            rms1 += pow(in,2);
            outRMS = (rms1/count) - pow((s1/count),2);
            if(count >= N) { //po nasbirani prvnych N vzorku
                phase = 1; //prepne na fazi 1
                count = 0; //a vynuluje citac pruchodu
            }
            break;

    case 1: //rekurzivne pocita odhad stredni hodnoty a rozptylu
            //zaroven sbira startovaci hodnoty pro fazi 2
            s1 = s1 - getAt(N+1) + in;
            s2 = s2 + in;
            outMV = s1/N;
            rms1 += pow(in,2) - pow(getAt(N+1),2);
            rms2 += pow(in,2);
            outRMS = (rms1/N) - pow(s1/N,2);
            if(count >= N) { //po N krocich
                phase = 2; //prepne na fazi 2
                count = 0; //vynuluje citac pruchodu
                s1 = 0.0; //a obe promenne vypoctu ve fazi 1
                rms1 = 0.0;
            }
            break;

    case 2: //rekurzivne pocita odhad stredni hodnoty a rozptylu
            //zaroven sbira startovaci hodnoty pro zopakovani faze 1
            s1 = s1 + in;
            s2 = s2 - getAt(N+1) + in;
            outMV = s2/N;
            rms1 += pow(in,2);
            rms2 += pow(in,2) - pow(getAt(N+1),2);
            outRMS = (rms2/N) - pow(s2/N,2);
            if(count >= N) { //po N krocich
                phase = 1; //prepne zpet na fazi 1
                count = 0; //vynuluje citac pruchodu
                s2 = 0.0; //a obe promenne vypoctu ve fazi 2
                rms2 = 0.0;
            }
            break;
    }
    return 0; }

int exit(void) { return 0; }
```

## B-2 Zdrojový kód algoritmu bloku pro výpočet RDFT

```
#define BUFFER_L 2000          //velikost archivu vstupnich hodnot
double input(0) in;          //vstup
double output(0) outAmp;     //vystup - amplituda slozky signalu o frek. w1
double output(1) outRe;      //vystup - Re cast slozky signalu o frek. w1
double output(2) outIm;      //vystup - Im cast slozky signalu o frek. w1
double output(3) outAmp2;    //vystup - amplituda slozky signalu o frek. w2
double output(4) outRe2;     //vystup - Re cast slozky signalu o frek. w2
double output(5) outIm2;     //vystup - Im cast slozky signalu o frek. w2
long parameter(0) N;         //parametr - pocet vzorku
double parameter(1) w1;      //parametr - frekvence 1
double parameter(2) w2;      //parametr - frekvence 2
double buffer[BUFFER_L];     //archiv hodnot vstupniho signalu
long head = 0, tail = 0, count = 0, phase = 0; //pointery archivu,
//citac pruchodu, prepinač faze
bool bufferFull = false;     //indikator zaplneni bufferu
double Re1 = 0.0, Im1 = 0.0, Re2 = 0.0, Im2 = 0.0; //promenne vypoctu
(w1)
double Re12 = 0.0, Im12 = 0.0, Re22 = 0.0, Im22 = 0.0; //promenne
vypoctu (w2)
double hlpR1 = 0.0, hlpI1 = 0.0, hlpR2 = 0.0, hlpI2 = 0.0;
//pomocne promenne vypoctu (w1)
double hlpR12 = 0.0, hlpI12 = 0.0, hlpR22 = 0.0, hlpI22 = 0.0;
//pomocne promenne vypoctu (w2)
double Ts = 0.0;             //perioda vzorkovanni
double A = 0.0, B = 0.0, C = 0.0, D = 0.0;
//koeficienty rozkladu na Re a Im cast (w1)
double A2 = 0.0, B2 = 0.0, C2 = 0.0, D2 = 0.0;
//koeficienty rozkladu na Re a Im cast (w2)

/*Vypocet koeficientu rozkladu signalu na Re a Im cast.
*/
int init(void) {
    Ts = GetPeriod();
    A = cos(w1*Ts);
    B = sin(w1*Ts);
    C = cos(-N*w1*Ts);
    D = sin(-N*w1*Ts);
    if(w2 > 0) {
        A2 = cos(w2*Ts);
        B2 = sin(w2*Ts);
        C2 = cos(-N*w2*Ts);
        D2 = sin(-N*w2*Ts);
    }
}
return 0;
}

/*Operace vlozeni prvku do cyklickeho bufferu. Koriguje hodnotu pointeru
head tak, aby neodkazoval mimo rozsah bufferu.
*/
int push(double value) {
    if(head >= BUFFER_L) {
        head = 0;
    }
    buffer[head++] = value;
return 0; }
```

```

/*Vraci prvek, který je v bufferu o n pozic v minulosti. Index prvku, který
se ma precist, je ziskan odectenim parametru n od pozice zapisove hlavy
head. Ziskany index je korigovan tak, aby neodkazoval mimo rozsah bufferu.
*/
double getAt(long n) {
    long index = head - n;
    if(index < 0) {
        index += BUFFER_L;
    }
    return buffer[index]; }

/*Algoritmus nejprve ve fazi 0 nacte N vzorku signalu jako startovaci
hodnoty pro fazi 1. Nasleduje prepnuti do teto faze, vypocet RDFT a
nacistani N vzorku pro fazi 2. Ve fazi 2 opet probiha sber vzorku pro fazi 1
a vypocet RDFT. Faze 1 a 2 se opakují po N krocich. Ve vsehch fazich se
navic pocita amplituda slozky signalu o zadane frekvenci. Algoritmus dokaze
pracovat se dvema zadanymi frekvencemi jednoho signalu zaroven. Zpracovani
signalu podle druhe frekvence (w2) se "vypne" zadaním parametru w2 = 0.
*/
int main(void) {
    ++count;
    push(in);
    switch(phase) {
    case 0: //nacisti N vzorku signalu pro vypocet faze 1
        //hodnoty na vystupech nejsou k uzitku
        hlpR1 = Re1 + (C * in);
        hlpI1 = Im1 + (D * in);
        Re1 = (A * hlpR1) - (B * hlpI1);
        Im1 = (A * hlpI1) + (B * hlpR1);
        //vystupy
        outRe = Re1;
        outIm = Im1;
        outAmp = sqrt(pow(Re1,2) + pow(Im1,2));/**(2.0/N);
        if(count >= N) { //po N krocich
            phase = 1; //prepne do faze 1
            count = 0; //a vynuluje citac pruchodu alg.
            hlpR1 = Re1 + (C * in) - getAt(N+1);
            //getAt(N+1) -> buffer[head - (N+1) = head - N - 1]
            hlpI1 = Im1 + (D * in);
            Re1 = (A * hlpR1) - (B * hlpI1);
            Im1 = (A * hlpI1) + (B * hlpR1);
            outRe = Re1;
            outIm = Im1;
            outAmp = sqrt(pow(Re1,2) + pow(Im1,2));/**(2.0/N);
        }
        break;

    case 1: //nacisti N vzorku signalu pro vypocet faze 2
        //rekrzivne resi RDFT a pocita amplitudu slozky zadane frek.
        hlpR1 = Re1 + (C * in) - getAt(N+1);
        hlpI1 = Im1 + (D * in);
        hlpR2 = Re2 + (C * in);
        hlpI2 = Im2 + (D * in);
        Re1 = (A * hlpR1) - (B * hlpI1);
        Im1 = (A * hlpI1) + (B * hlpR1);
        Re2 = (A * hlpR2) - (B * hlpI2);
        Im2 = (A * hlpI2) + (B * hlpR2);
        //vystupy
        outRe = Re1;
        outIm = Im1;

```

```

outAmp = sqrt(pow(Re1,2) + pow(Im1,2));/*(2.0/N);
if(count >= N) { //po N krocich
    phase = 2; //prepne do faze 2
    count = 0; //vynuluje citac pruchodu alg.
    hlpR1 = 0.0; //a vynuluje promenne vypoctu f. 1
    hlpI1 = 0.0;
    Re1 = 0.0;
    Im1 = 0.0;
}
break;

case 2: //nacita N vzorku signalu pro vypocet faze 1
//rekrzivne resi RDFT a pocita amplitudu slozky zadane frek.
hlpR1 = Re1 + (C * in);
hlpI1 = Im1 + (D * in);
hlpR2 = Re2 + (C * in) - getAt(N+1);
hlpI2 = Im2 + (D * in);
Re1 = (A * hlpR1) - (B * hlpI1);
Im1 = (A * hlpI1) + (B * hlpR1);
Re2 = (A * hlpR2) - (B * hlpI2);
Im2 = (A * hlpI2) + (B * hlpR2);
//vystupy
outRe = Re2;
outIm = Im2;
outAmp = sqrt(pow(Re2,2) + pow(Im2,2));/*(2.0/N);
if(count >= N) { //po N krocich
    phase = 1; //prepne zpět do faze 1
    count = 0; //vynuluje citac pruchodu alg.
    hlpR2 = 0.0; //a vynuluje promenne vypoctu f. 2
    hlpI2 = 0.0;
    Re2 = 0.0;
    Im2 = 0.0;
}
break;
}

if(w2 > 0) {
switch(phase) {
case 0: //nacita N vzorku signalu pro vypocet faze 1
//hodnoty na vystupech nejsou k uzitku
hlpR12 = Re12 + (C2 * in);
hlpI12 = Im12 + (D2 * in);
Re12 = (A2 * hlpR12) - (B2 * hlpI12);
Im12 = (A2 * hlpI12) + (B2 * hlpR12);
//vystupy
outRe2 = Re12;
outIm2 = Im12;
outAmp2 = sqrt(pow(Re12,2) + pow(Im12,2));/*(2.0/N);
if(count >= N) { //po N krocich
    phase = 1; //prepne do faze 1
    count = 0; //a vynuluje citac pruchodu alg.
    hlpR12 = Re12 + (C2 * in) - getAt(N+1);
    hlpI12 = Im12 + (D2 * in);
    Re12 = (A2 * hlpR12) - (B2 * hlpI12);
    Im12 = (A2 * hlpI12) + (B2 * hlpR12);
    outRe2 = Re12;
    outIm2 = Im12;
    outAmp2 = sqrt(pow(Re12,2) + pow(Im12,2));/*(2.0/N);
}
break;
}
}

```

```

case 1:      //nacita N vzorku signalu pro vypocet faze 2
            //rekrzivne resi RDFT a pocita amplitudu slozky zadane frek.
            hlpR12 = Re12 + (C2 * in) - getAt(N+1);
            hlpI12 = Im12 + (D2 * in);
            hlpR22 = Re22 + (C2 * in);
            hlpI22 = Im22 + (D2 * in);
            Re12 = (A2 * hlpR12) - (B2 * hlpI12);
            Im12 = (A2 * hlpI12) + (B2 * hlpR12);
            Re22 = (A2 * hlpR22) - (B2 * hlpI22);
            Im22 = (A2 * hlpI22) + (B2 * hlpR22);
            //vystupy
            outRe2 = Re12;
            outIm2 = Im12;
            outAmp2 = sqrt(pow(Re12,2) + pow(Im12,2));/*(2.0/N);
            if(count >= N) { //po N krocich
                phase = 2; //prepne do faze 2
                count = 0; //vynuluje citac pruchodu alg.
                hlpR12 = 0.0; //a vynuluje promenne vypoctu f. 1
                hlpI12 = 0.0;
                Re12 = 0.0;
                Im12 = 0.0;
            }
            break;

case 2:      //nacita N vzorku signalu pro vypocet faze 1
            //rekrzivne resi RDFT a pocita amplitudu slozky zadane frek.
            hlpR12 = Re12 + (C2 * in);
            hlpI12 = Im12 + (D2 * in);
            hlpR22 = Re22 + (C2 * in) - getAt(N+1);
            hlpI22 = Im22 + (D2 * in);
            Re12 = (A2 * hlpR12) - (B2 * hlpI12);
            Im12 = (A2 * hlpI12) + (B2 * hlpR12);
            Re22 = (A2 * hlpR22) - (B2 * hlpI22);
            Im22 = (A2 * hlpI22) + (B2 * hlpR22);
            //vystupy
            outRe2 = Re22;
            outIm2 = Im22;
            outAmp2 = sqrt(pow(Re22,2) + pow(Im22,2));/*(2.0/N);
            if(count >= N) { //po N krocich
                phase = 1; //prepne zpět do faze 1
                count = 0; //vynuluje citac pruchodu alg.
                hlpR22 = 0.0; //a vynuluje promenne vypoctu f. 2
                hlpI22 = 0.0;
                Re22 = 0.0;
                Im22 = 0.0;
            }
            break;
        }
    }
return 0; }

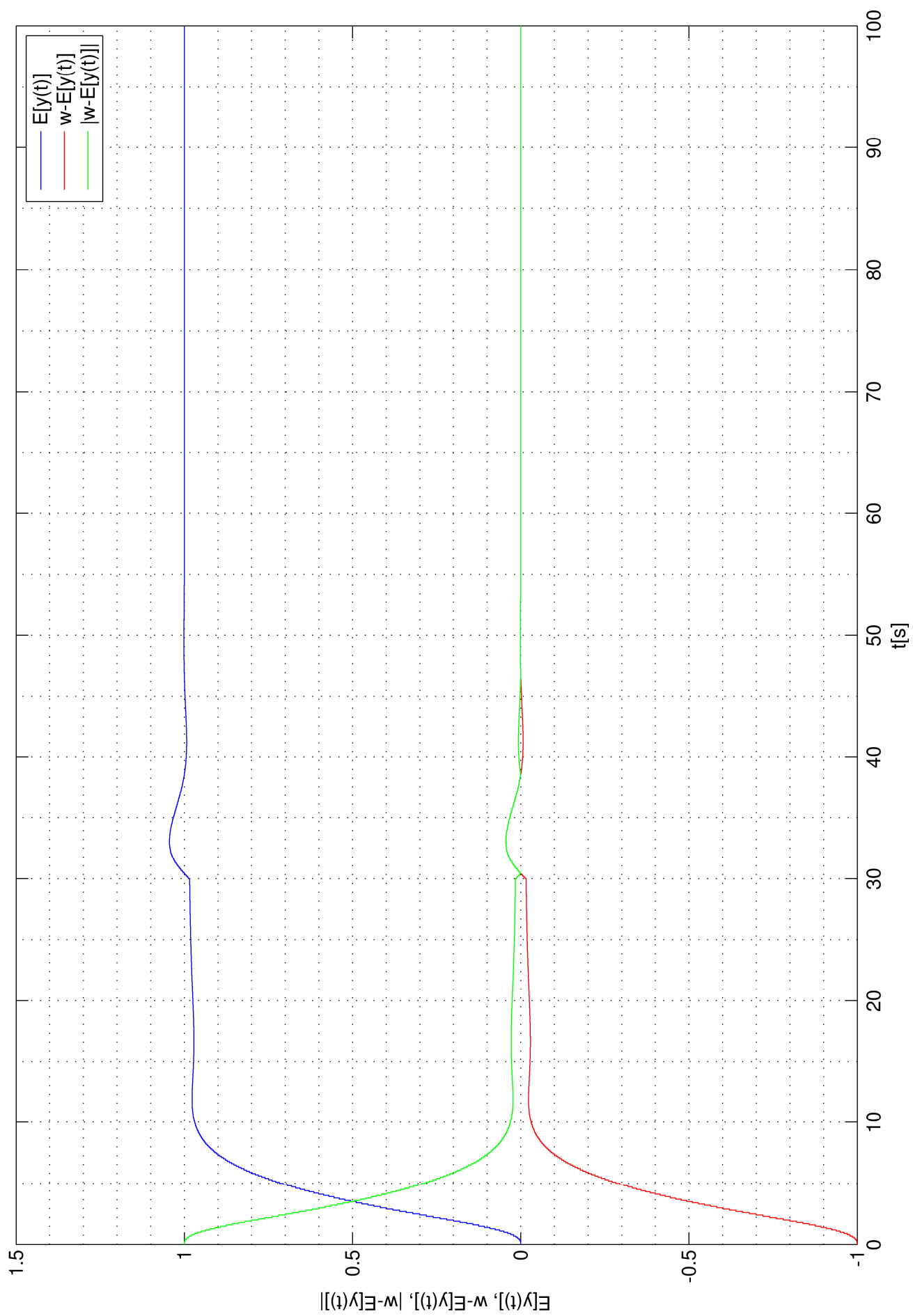
int exit(void) { return 0; }

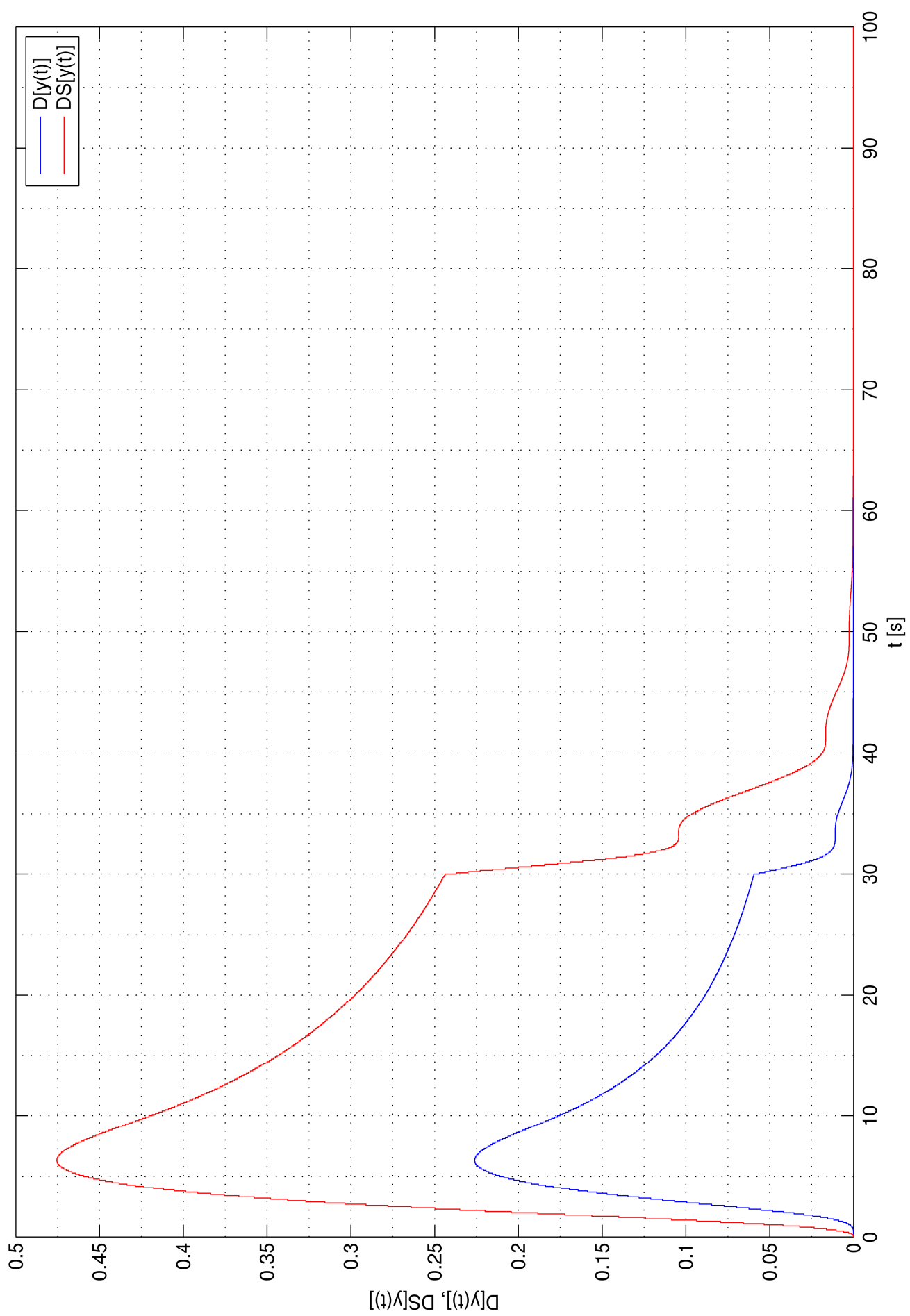
```

### B-3 Celostránkové verze grafů z kapitoly 5.1

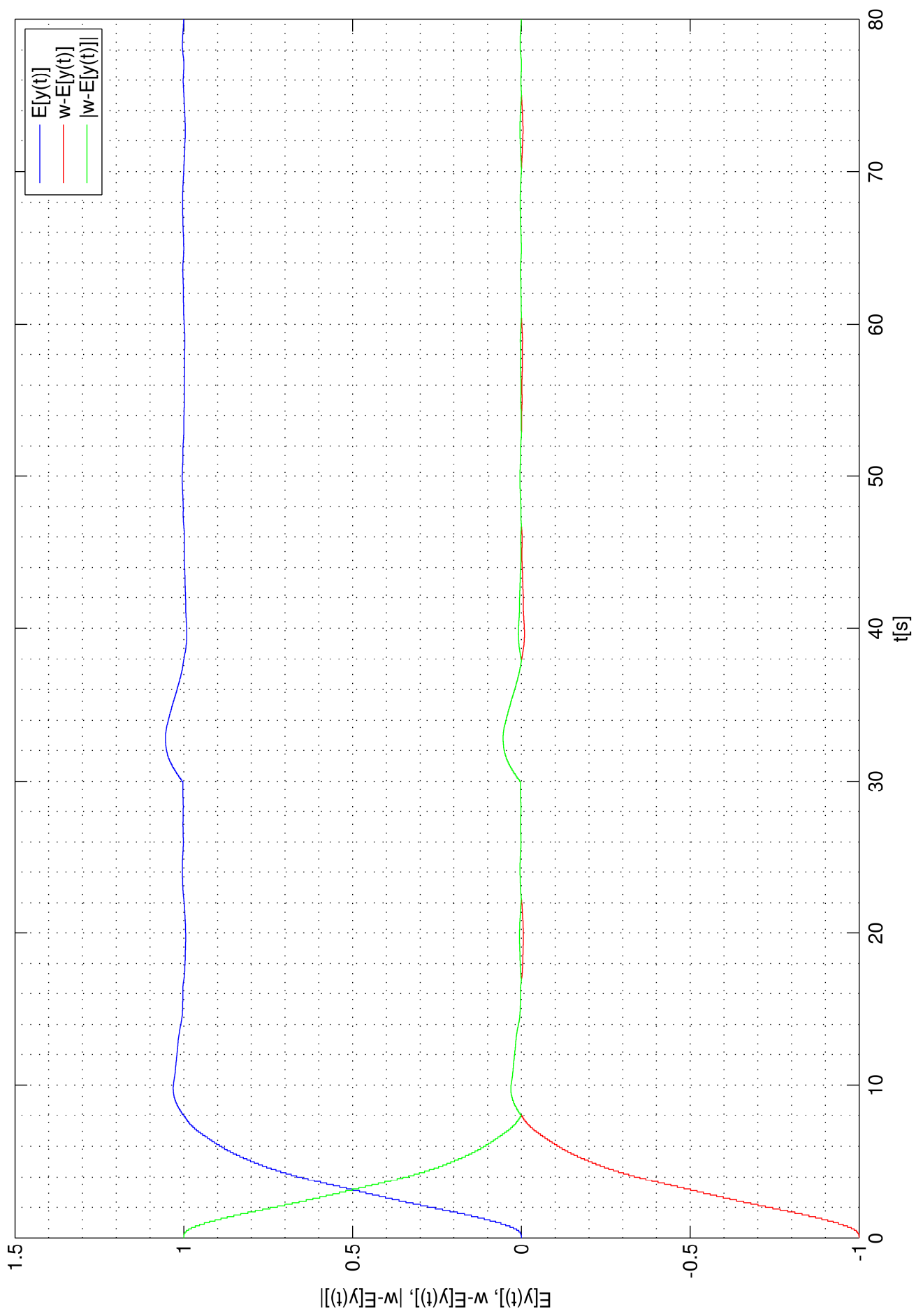
#### Obsah

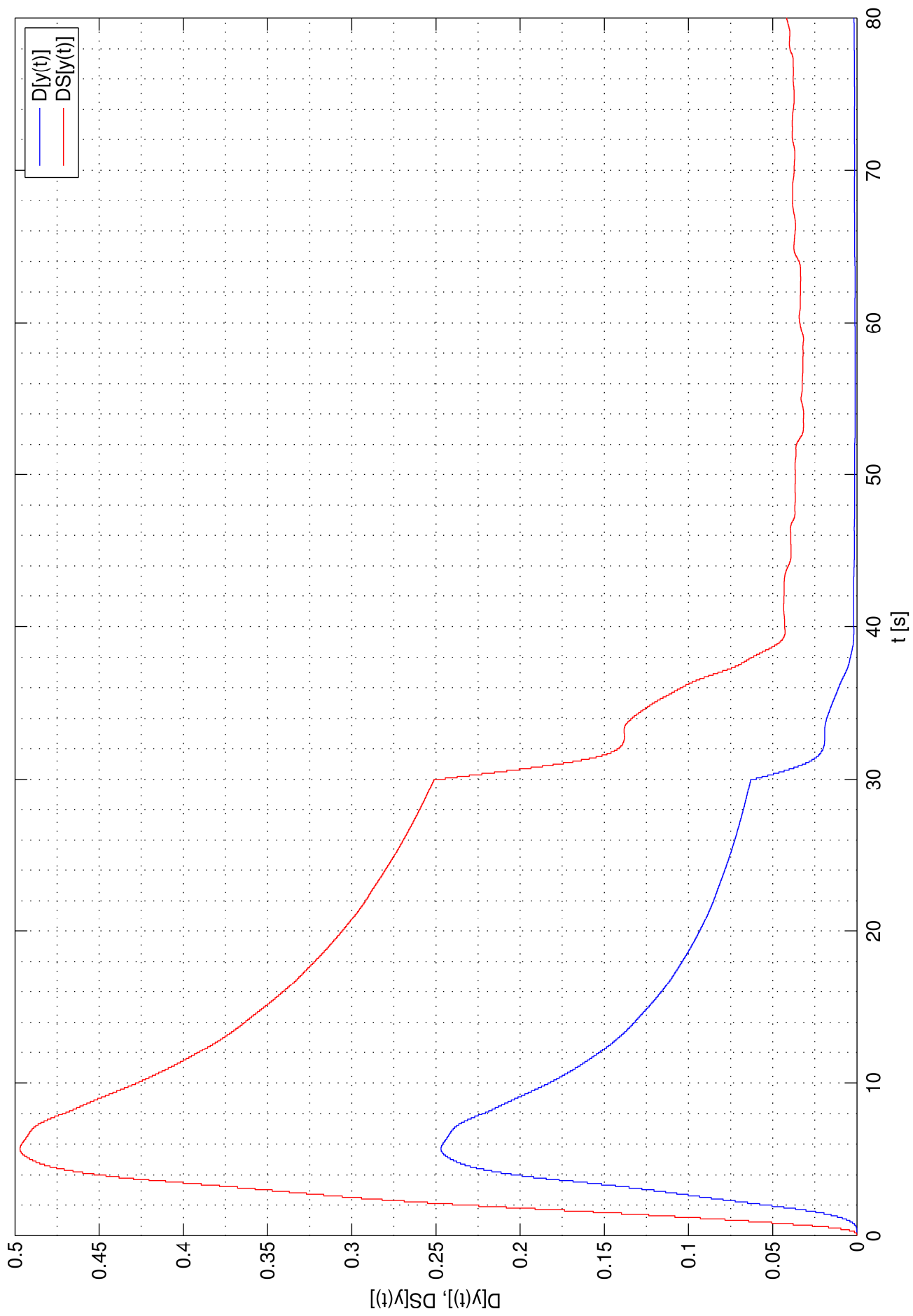
Obr. 5.4: Průběh střední hodnoty $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu ze strany 22	52
Obr. 5.5: Průběh rozptylu a směrodatné odchyly ze strany 22	53
Obr. 5.7: Průběh střední hodnoty $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu ze strany 24	54
Obr. 5.8: Průběh rozptylu a směrodatné odchyly ze strany 24	55
Obr. 5.10: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ ze strany 26	56
Obr. 5.11: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ ze strany 26	57
Obr. 5.12: Průběh amplitudy a reálné a imaginární rozklad složky $u(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ ze strany 27	58
Obr. 5.13: Průběh amplitudy a reálné a imaginární rozklad složky $u(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ ze strany 27	59
Obr. 5.15: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy ze strany 29	60
Obr. 5.16: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy ze strany 29	61
Obr. 5.17: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy ze strany 30	62
Obr. 5.18: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy ze strany 30	63
Obr. 5.19: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy a „rozpojené“ zpětné vazbě ze strany 31	64

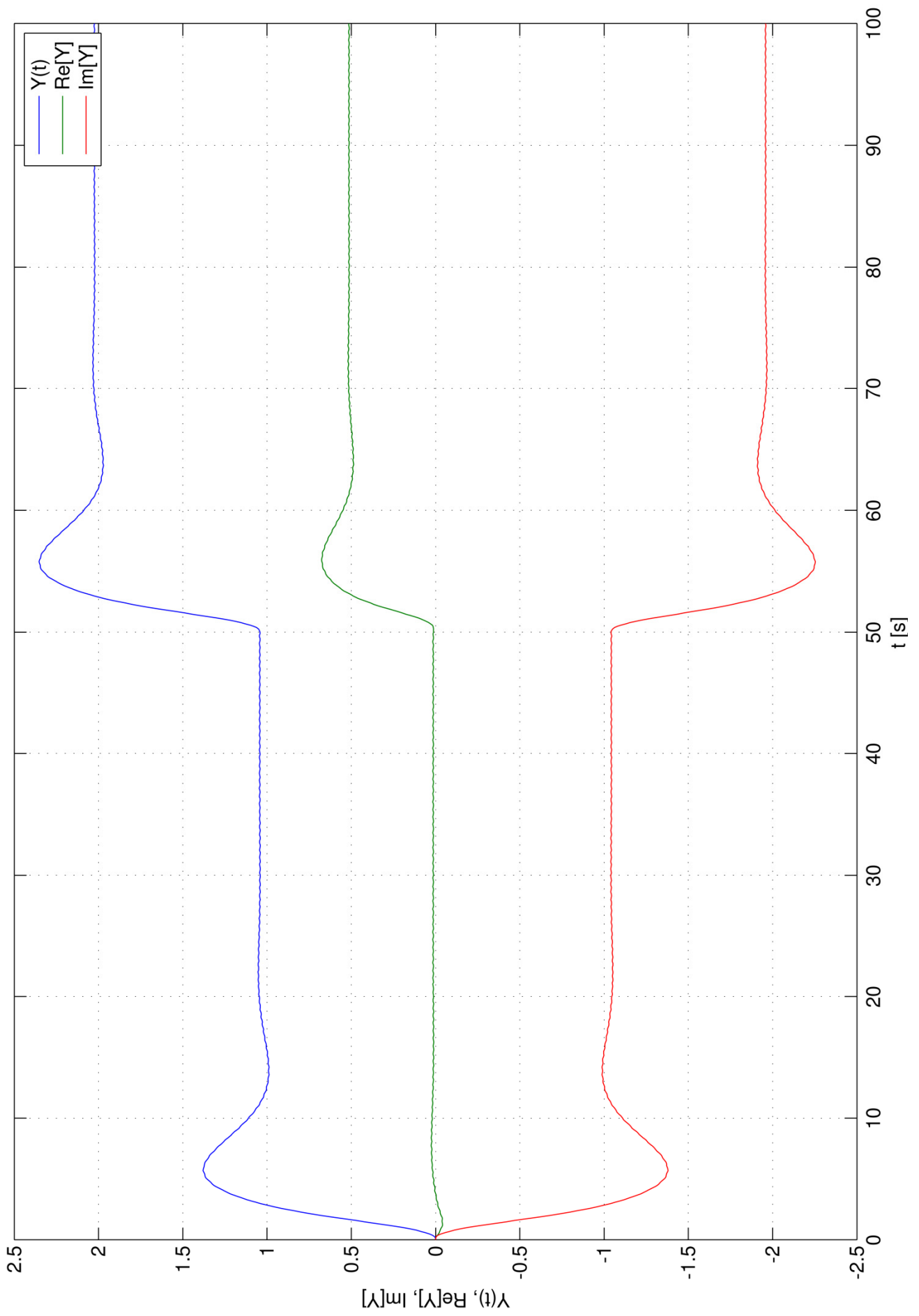


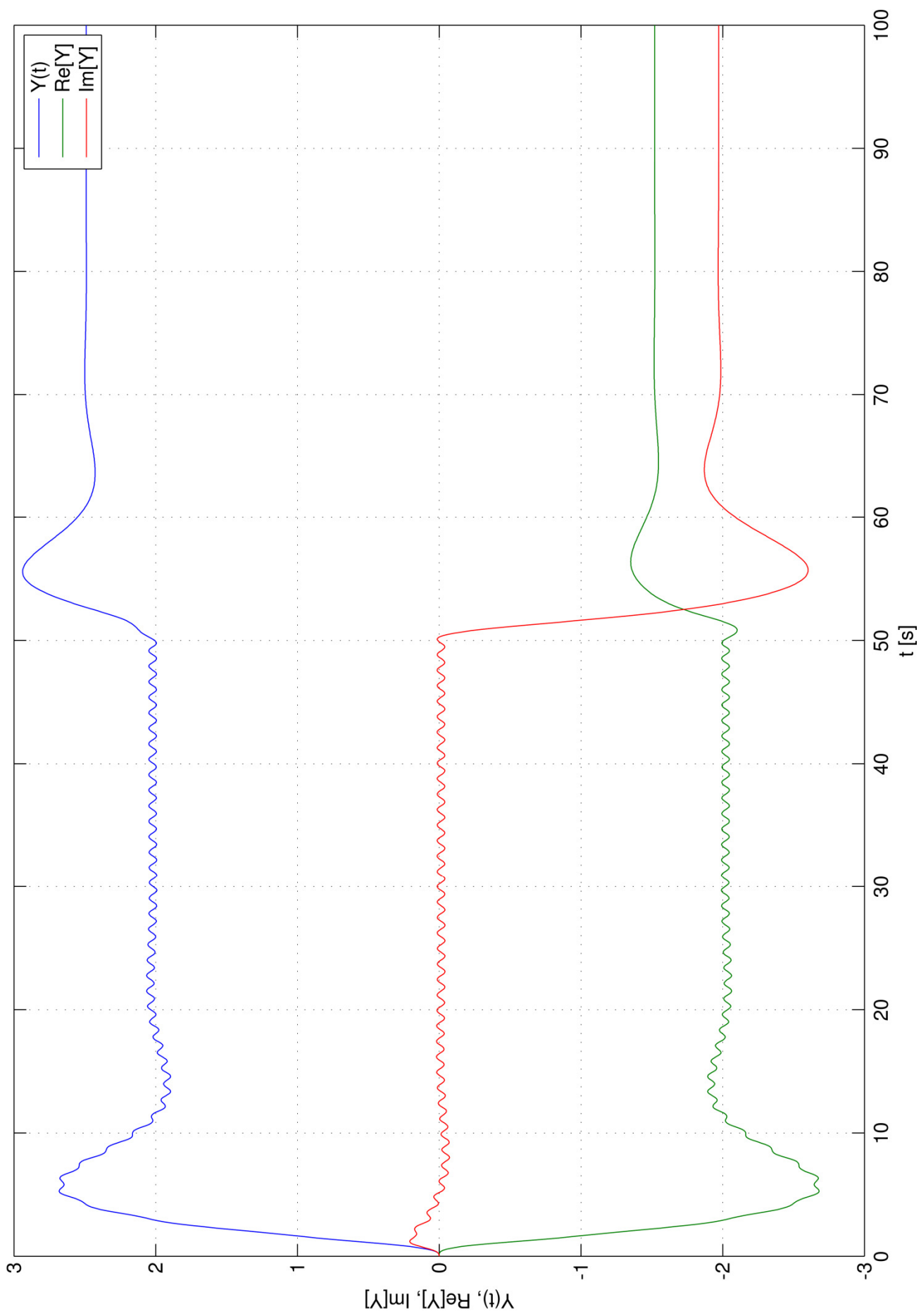


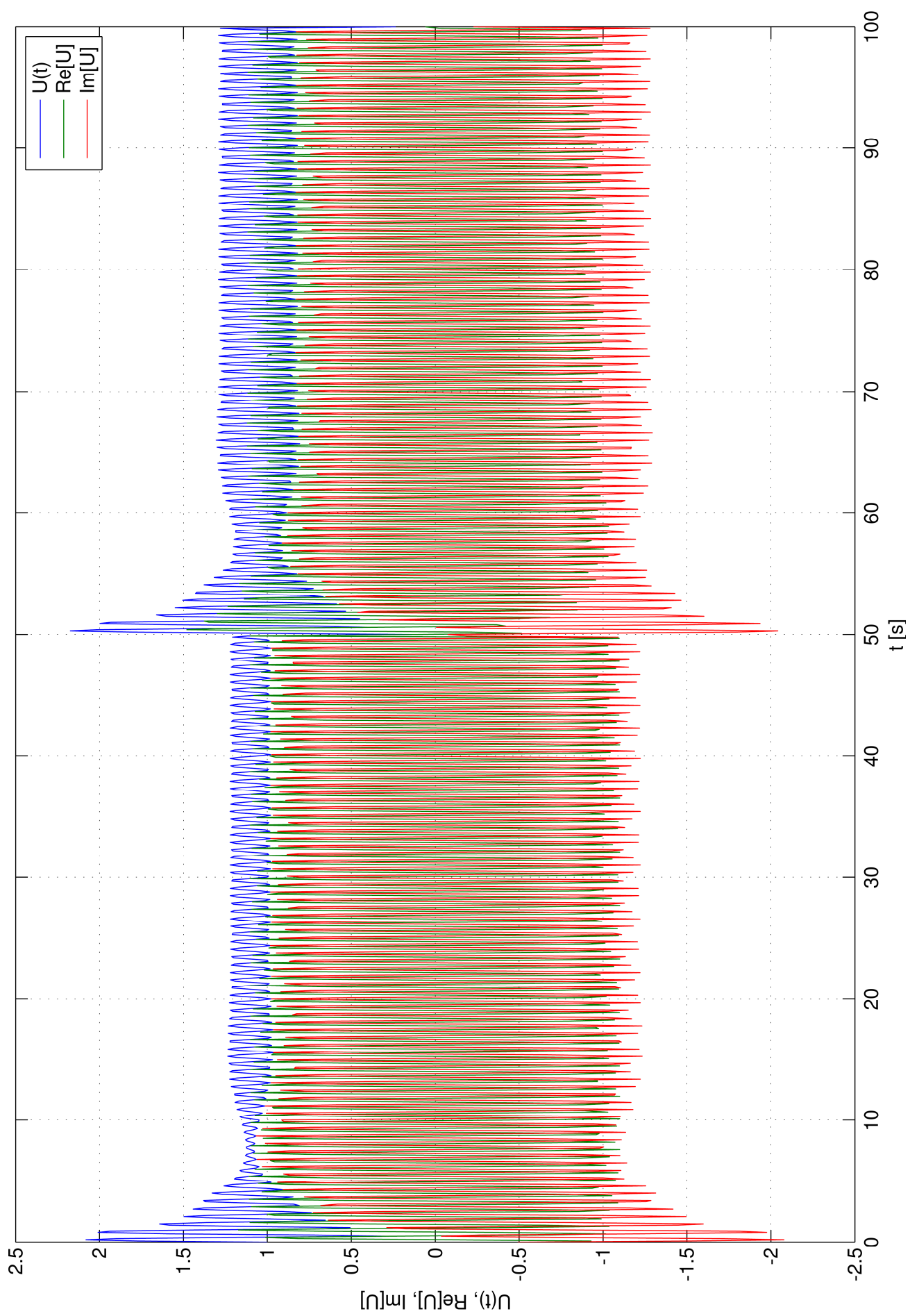


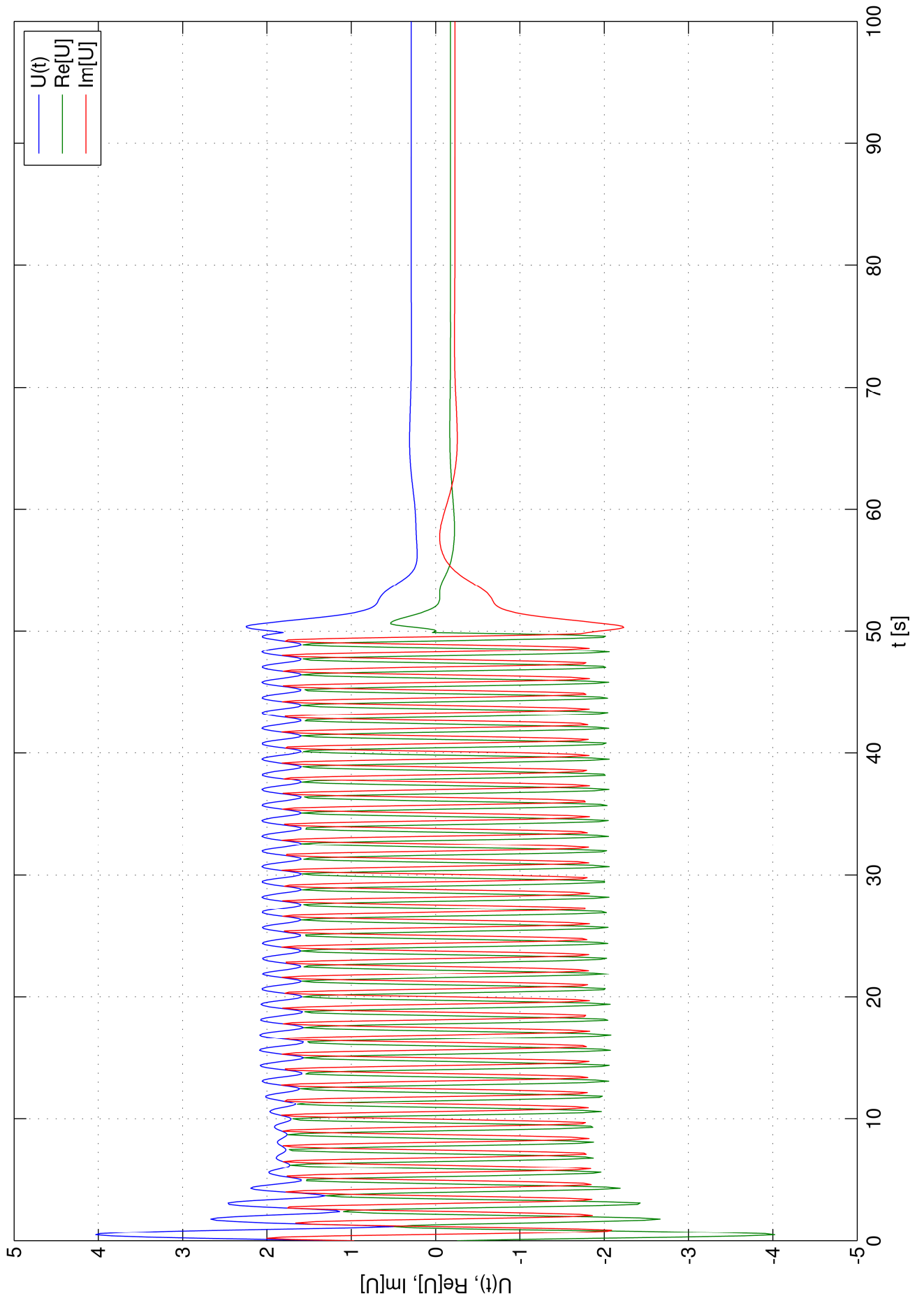


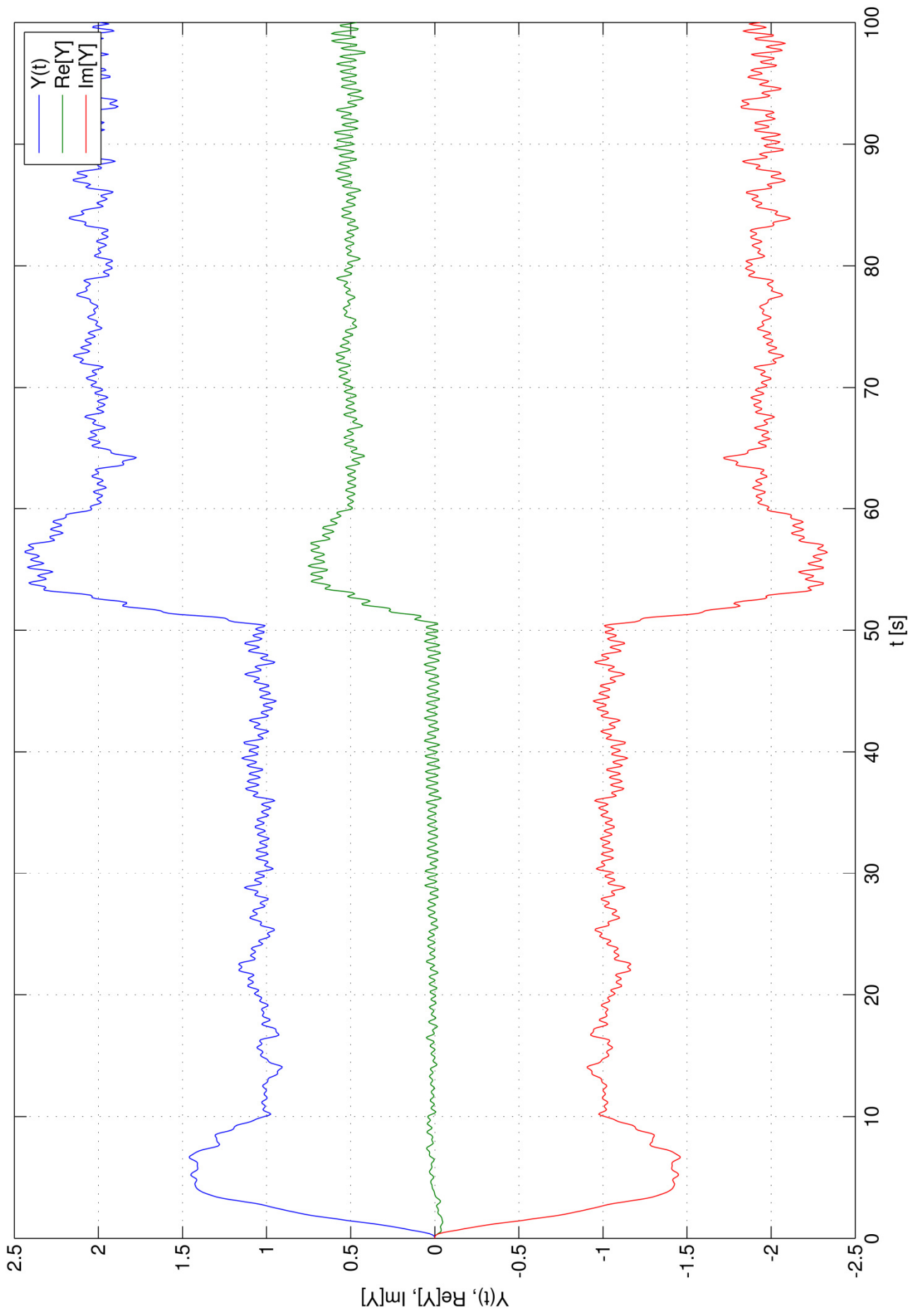


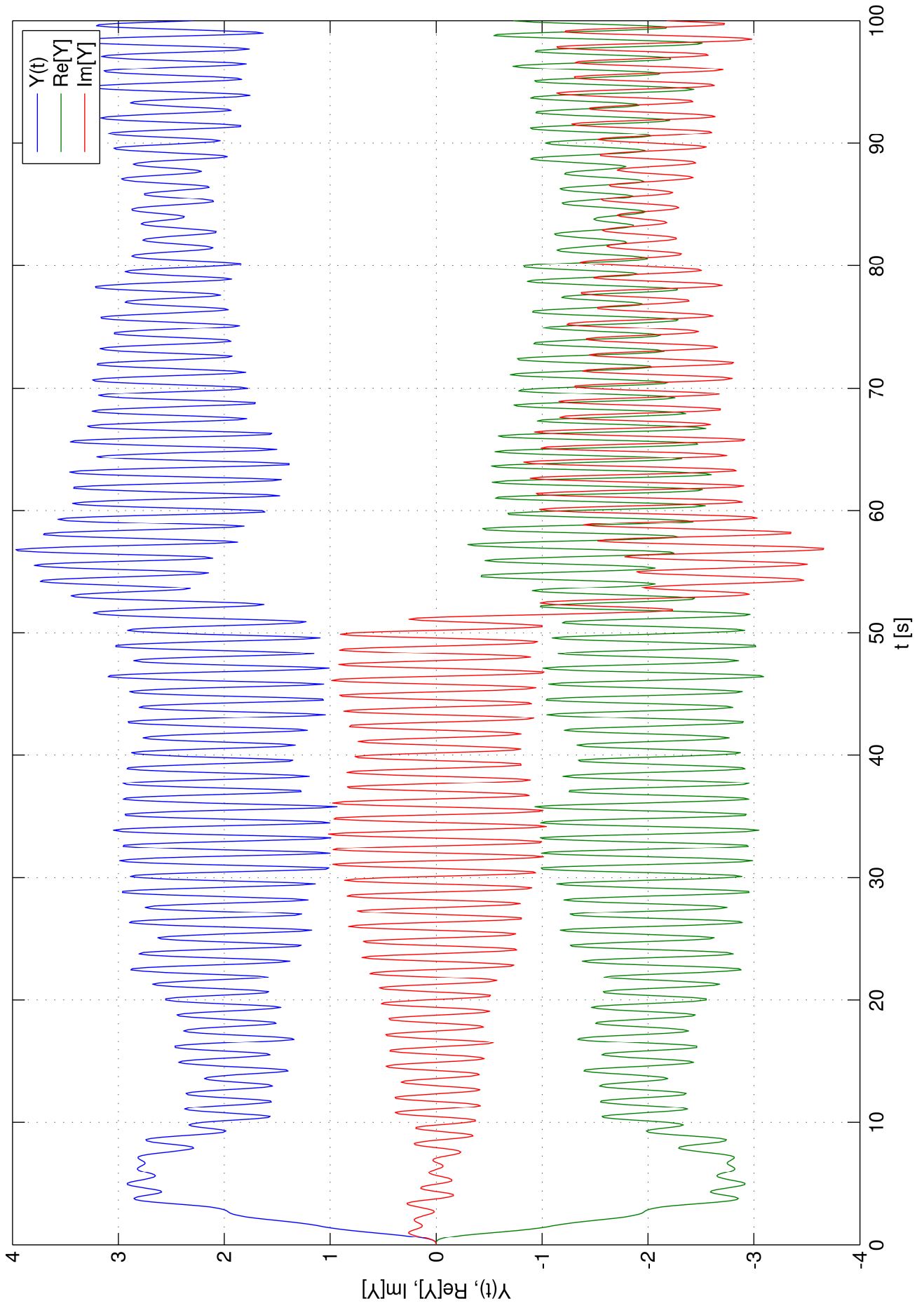




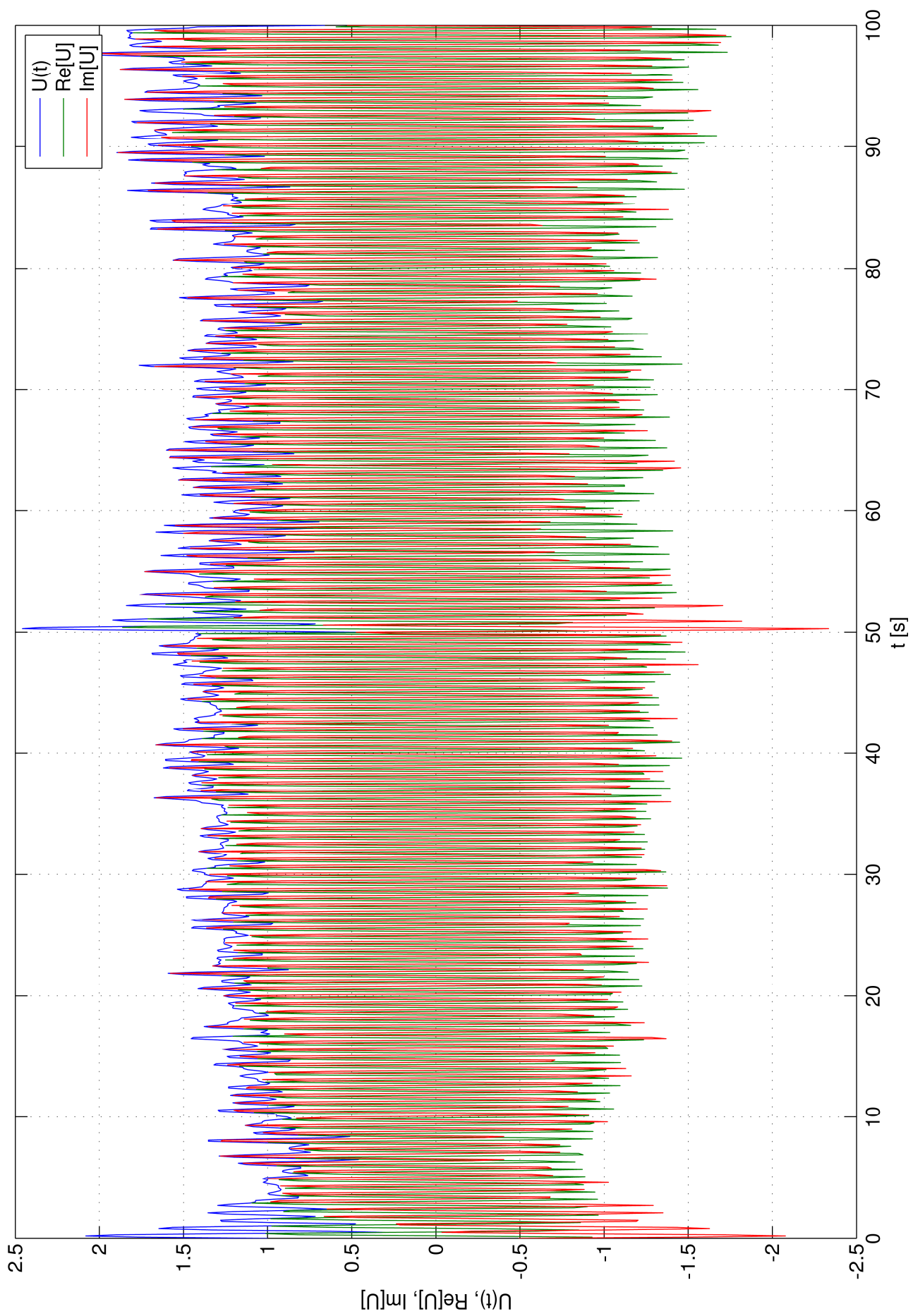


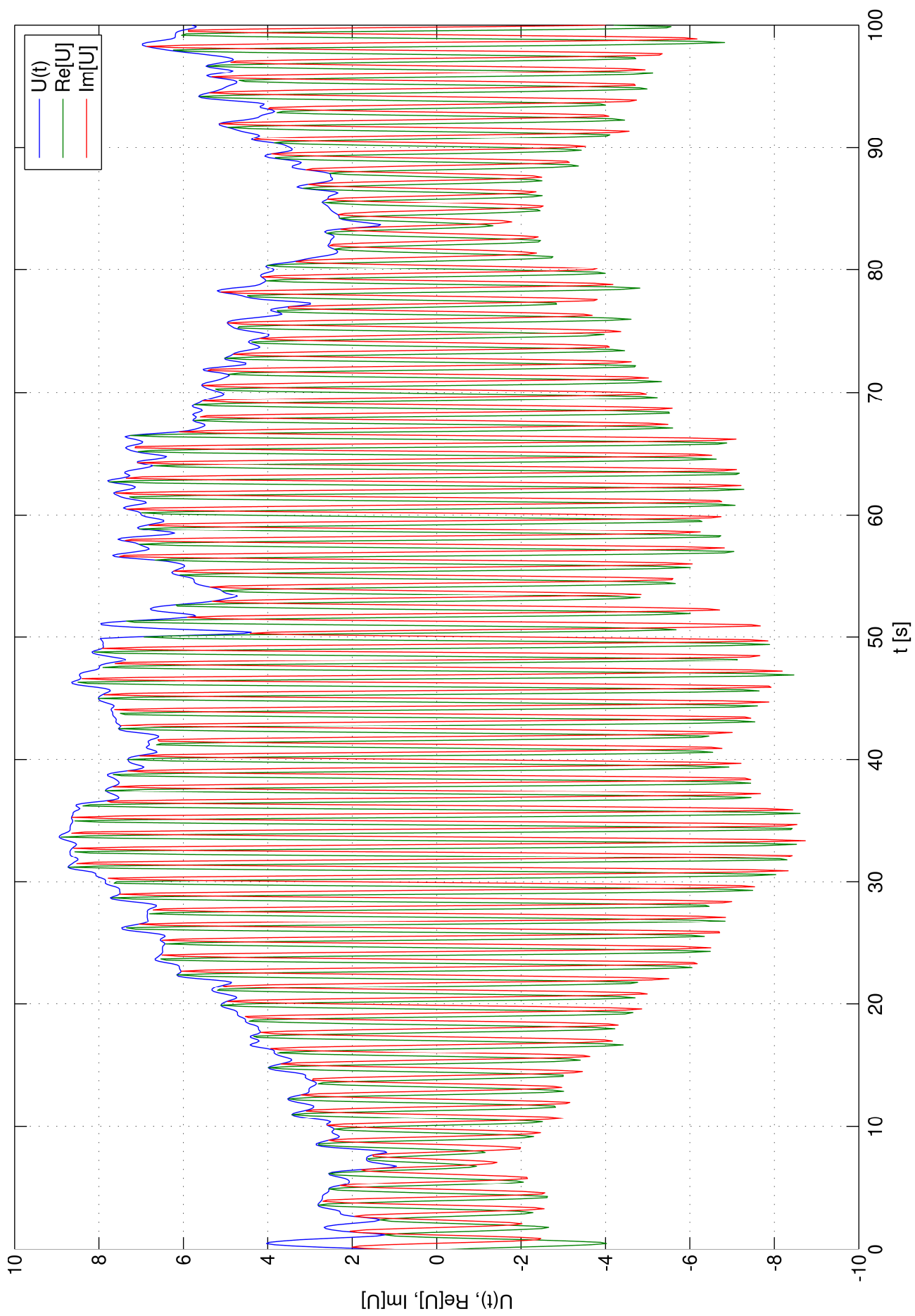


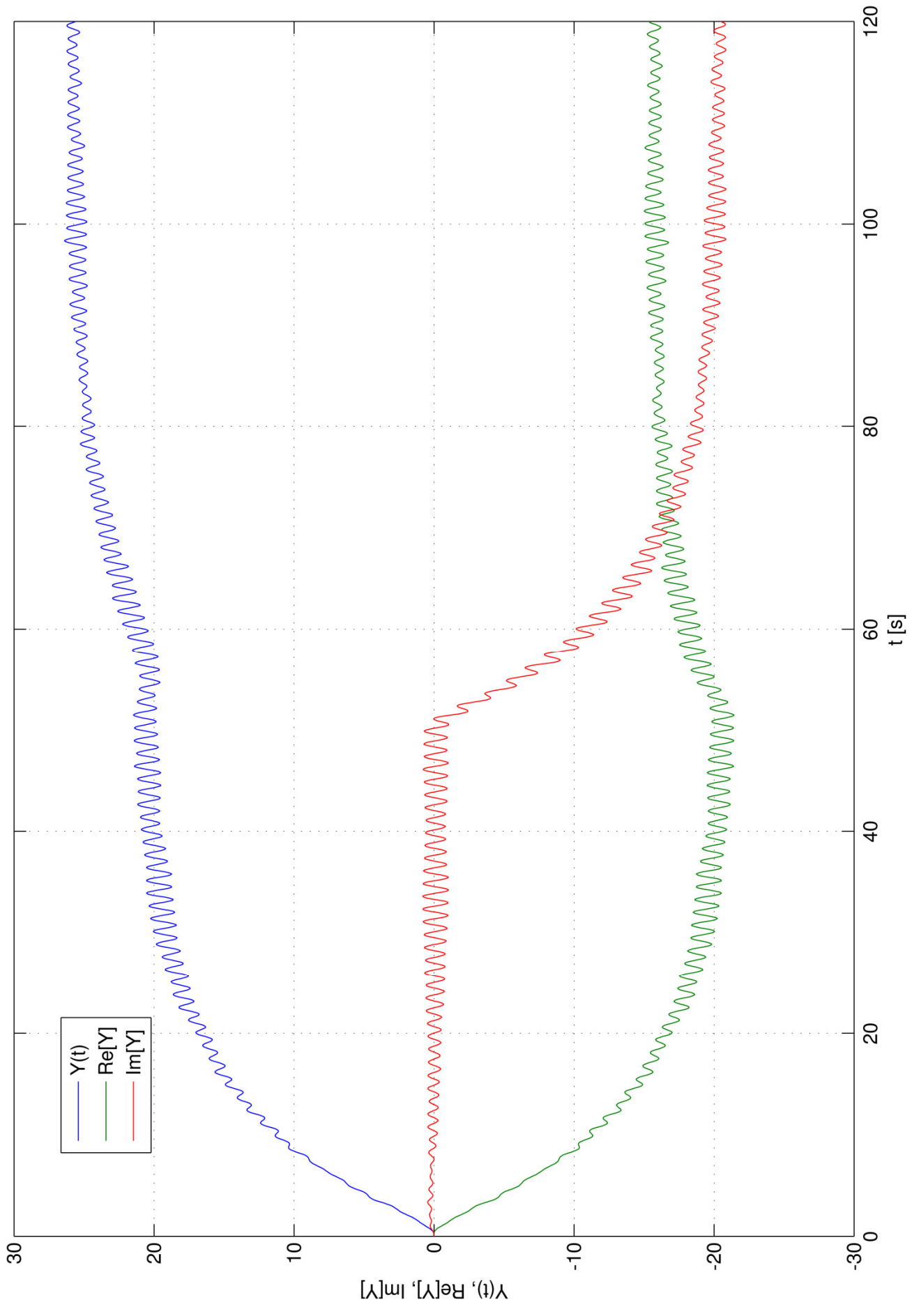












## B-4 Celostránkové verze grafů z kapitoly 5.2

### Obsah

Obr. 5.23: Průběh střední hodnoty $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu ze strany 34	66
Obr. 5.24: Průběh rozptylu a směrodatné odchylky ze strany 34	67
Obr. 5.26: Průběh střední hodnoty $y(t)$ , jejího rozdílu proti žádané hodnotě a absolutní hodnoty tohoto rozdílu ze strany 36	68
Obr. 5.27: Průběh rozptylu a směrodatné odchylky ze strany 36	69
Obr. 5.29: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ ze strany 38	70
Obr. 5.30: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ ze strany 38	71
Obr. 5.31: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ ze strany 39	72
Obr. 5.32: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ ze strany 39	73
Obr. 5.33: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy ze strany 40	74
Obr. 5.34: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy ze strany 40	75
Obr. 5.35: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $10 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy ze strany 41	76
Obr. 5.36: Průběh amplitudy a reálné a imaginární složky $u(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy ze strany 41	77
Obr. 5.37: Průběh amplitudy a reálné a imaginární složky $y(t)$ o frekvenci $5 \text{ rad} \cdot \text{s}^{-1}$ při působení poruchy a „rozpojené“ zpětné vazbě ze strany 42	78

