



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**SOFTWARE PRO IDENTIFIKACI DYNAMICKÝCH
SYSTÉMŮ**

SOFTWARE FOR IDENTIFICATION OF DYNAMIC SYSTEMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Zimek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. et Ing. Stanislav Lang, Ph.D.

BRNO 2019

Zadání diplomové práce

Ústav: Ústav automatizace a informatiky
Student: **Bc. Tomáš Zimek**
Studijní program: Strojní inženýrství
Studijní obor: Aplikovaná informatika a řízení
Vedoucí práce: **Ing. et Ing. Stanislav Lang, Ph.D.**
Akademický rok: 2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Software pro identifikaci dynamických systémů

Stručná charakteristika problematiky úkolu:

Práce je zaměřena na problematiku identifikace dynamických systémů. Rešeršní část práce popíše používané metody (uvedeny budou metody pro identifikace dynamických systémů v časové i frekvenční oblasti). V rámci implementační části práce student otestuje vybrané metody na zvoleném dynamickém systému (může být provedeno v prostředí Matlab & Simulink), a dále některou z metod implementuje v podobě samostatné aplikace pro MS Windows (programovací jazyk dle volby studenta).

Cíle diplomové práce:

Proveďte rešerši v oblasti identifikace dynamických systémů.

Otestujte vybrané metody v prostředí Matlab & Simulink.

Implementujte vybrané identifikační metody mimo prostředí Matlab & Simulink (volba programovacího jazyka či interpreteru dle libosti, avšak zajistěte, aby provozování vytvořeného softwaru nebylo vázáno na placený produkt).

Připravte grafické uživatelské rozhraní programu.

Vytvořte stručný návod k obsluze programu.

Seznam doporučené literatury:

ŠVARC, Ivan, ŠEDA, Miloš a VÍTEČKOVÁ, Miluše. Automatické řízení. Brno: Akademické nakladatelství CERM, 2007. ISBN 978-80-214-3491-2.

BLAHA, Petr a Petr VAVŘÍN. Řízení a regulace I: Základy regulace lineárních systémů - spojité a diskrétní. Brno: VUT FEKT, 2005.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Práce se zabývá metodami identifikace dynamických systémů v časové i frekvenční oblasti. Rozebírá používané neparametrické a parametrické metody identifikace. Vybrané metody jsou implementovány a otestovány v prostředí Matlab & Simulink. V závěru práce je popsán návod k obsluze vytvořené aplikace v jazyce Python, jenž implementuje zvolenou metodu identifikace.

ABSTRACT

The thesis deals with methods of identification of dynamic systems in time and frequency domain. Nonparametric and parametric methods of identification are analyzed. Selected methods are implemented in Matlab & Simulink software. Finally, the user guide for the created Python application is introduced.

KLÍČOVÁ SLOVA

Dynamické systémy, identifikační metody, metoda nejmenších čtverců, spektrální analýza.

KEYWORDS

Dynamic systems, identification methods, least squares method, spectral analysis.

BIBLIOGRAFICKÁ CITACE

ZIMEK, Tomáš. *Software pro identifikaci dynamických systémů*, Brno, 2019, 61s.
Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí diplomové práce Ing. et Ing. Stanislav Lang, Ph.D.

PODĚKOVÁNÍ

Děkuji Ing. et Ing. Stanislavovi Langovi, Ph.D. za pedagogickou pomoc při tvorbě práce, její odborné vedení, cenné rady, veškerou ochotu a za inspiraci při tvorbě praktické části.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. et Ing. Stanislava Langa, Ph.D. a s použitím literatury uvedené v seznamu literatury.

V Brně dne 24. 5. 2019

.....

Bc. Tomáš Zimek

OBSAH

1	ÚVOD.....	15
2	DYNAMICKÉ SYSTÉMY	17
2.1	Možnosti popisu dynamických systémů	17
2.2	Přístupy k modelování dynamických systémů	18
3	IDENTIFIKAČNÍ METODY	21
3.1	Metody k získání charakteristiky soustavy (neparametrické metody)	22
3.1.1	Identifikace z přechodového děje	22
3.1.2	Identifikace z frekvenčního vstupu	25
3.1.3	Korelační metoda	28
3.1.4	Spektrální analýza	29
3.2	Metody k získání parametrů systému (parametrické metody)	32
3.2.1	Parametrické modely	32
3.2.2	Jednorázová metoda nejmenších čtverců	34
3.2.3	Rekurzivní metoda nejmenších čtverců	35
4	REALIZACE VYBRANÝCH IDENTIFIKAČNÍCH METOD V PROSTŘEDÍ MATLAB	37
4.1	Identifikace nekmitavého systému z přechodové charakteristiky	37
4.1.1	Implementace	37
4.1.2	Výsledky pro testované systémy	38
4.2	Identifikace parametrů soustavy jednorázovou metodou nejmenších čtverců ..	41
4.2.1	Implementace	41
4.2.2	Výsledky pro testované systémy	42
4.3	Identifikace parametrů soustavy rekurzivní metodou nejmenších čtverců rozšířenou o techniku zapomínání	45
4.3.1	Implementace	45
4.3.2	Výsledky pro testované systémy	46
4.4	Identifikace systému ve frekvenční oblasti spektrální analýzou	46
5	SAMOSTATNÁ APLIKACE PRO IDENTIFIKACI SOUSTAVY	49
5.1	Tvorba GUI ve wx knihovně	51
6	ZÁVĚR.....	55
7	SEZNAM POUŽITÉ LITERATURY	57
8	SEZNAM POUŽITÝCH ZKRATEK	59
9	SEZNAM PŘÍLOH	61

1 ÚVOD

K dosažení efektivního řízení procesů a objektů řízení je potřeba navrhnout kvalitní regulaci daného systému. Tento návrh se odvíjí od chování regulované soustavy a znalostí o ní získaných. Ve skutečnosti je úplný popis určitého systému na základě odvození z matematicko-fyzikálních zákonů nerealizovatelný, případně velmi složitý. Řešením je sestavit matematický model, který se skutečnému systému přibližuje ve všech podstatných charakteristických vlastnostech a je možné jej řídit. Modelování, tedy tvorba daného modelu, se neobejde bez identifikace řízeného systému.

Cílem práce je provést rešerši v oblasti problematiky metod identifikace dynamických systémů. Vybrané metody by měly být implementovány v prostředí výpočetního standardu pro technické výpočty Matlab & Simulink tak, aby mohly sloužit např. pro názornou výuku tématu modelování a identifikace systémů. Zvolené metody budou taky naprogramovány v podobě samostatné aplikace nevázané na placený produkt. K této aplikaci bude vytvořeno grafické uživatelské rozhraní a návod na jeho obsluhu.

Vlastní práce je rozdělena na rešeršní a praktickou část. Rešeršní část je obsažena ve druhé a třetí kapitole. Kapitola 2 je úvodem do dynamických systémů, možností jejich popisu a modelování. Navazuje kapitola 3 zaměřená na identifikaci dynamických systémů. V této rešeršní části jsou popsány parametrické metody identifikace včetně typů parametrických modelů, uvedeny jsou neparametrické metody jak v časové, tak i ve frekvenční oblasti. Praktickou část práce tvoří kapitoly 4 a 5. V kapitole 4 jsou implementovány vybrané metody na zvolených systémech. Kapitola 5 obsahuje návod k obsluze vytvořeného programu.

2 DYNAMICKÉ SYSTÉMY

Dynamický systém je abstraktní objekt popisující reálný fyzický systém, jehož chování lze popsat diferenciálními rovnicemi. Je to taky model popisující řízenou soustavu, která se v určitém časovém úseku vyvíjí. K tomu tento systém využívá veličiny umožňující popis tohoto systému v čase t . Veličiny obvykle nezachycují všechny detaily reálného systému. Jejich výběr je takový, aby kompletně reprezentovaly matematický model. Znalost hodnot těchto veličin v určitém čase poskytuje souhrnné poznatky o stavu dynamického systému. Rozměr (dimenze) dynamického systému je rovna počtu těchto proměnných vyjadřující stav systému.

Znalost stavových veličin v určitém čase určuje vývoj systému v budoucích stavech. Spojitý dynamický systém je takový, kde se jeho stav vyvíjí nepřetržitě v čase t , jenž je obvykle reprezentován reálnými čísly. Pokud se systém vyvíjí v časových krocích (tedy v diskrétním čase složeného z časových okamžiků, které jsou obvykle reprezentovány celými čísly), potom časová osa diskrétního děje lze vyjádřit jako $t = k \cdot T$, kde T je vzorkovací perioda a $k = 0, 1, 2, \dots$. O takovém systému mluvíme jako o diskrétním dynamickém systému. Diskrétní systém lze vytvořit tak, že všechny veličiny spojitého systému měříme pouze v diskrétních časových okamžicích, čímž tento systém vlastně vzorkujeme. [1, 2, 5, 7]

Na řízený dynamický systém působí v každém čase t vstupní veličina (označuje se také jako řídicí veličina) $\mathbf{u}(t)$ o m složkách:

$$\mathbf{u}(t) = [u_1(t), u_2(t), \dots, u_m(t)] \quad (1)$$

V tomto čase t z něj také vystupuje výstupní veličina $\mathbf{y}(t)$ o n složkách:

$$\mathbf{y}(t) = [y_1(t), y_2(t), \dots, y_n(t)] \quad (2)$$

Výstup systému závisí na současném i předcházejícím průběhu vstupní veličiny. Stav systému $\mathbf{x}(t)$ v čase t popisuje vnitřní vlastnosti systému včetně záznamu o minulém vývoji systému až do toho času t :

$$\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_p(t)] \quad (3)$$

Potom přechod stavu systému v čase τ do stavu v čase t je dán funkcí přechodu:

$$\mathbf{x}(t) = f(t, \tau, \mathbf{x}(\tau), \mathbf{u}) \quad (4)$$

2.1 Možnosti popisu dynamických systémů

Popsat dynamické systémy lze ze dvou rovin, rozlišuje se vnější a vnitřní popis.

Vnitřní popis

Vnitřní popis systému vyjadřuje jeho dynamické vlastnosti vztahy mezi vstupem, stavem systému a výstupem podle rovnice (5):

$$\mathbf{y} = f(\mathbf{u}, \mathbf{x}) \quad (5)$$

Stav systému v sobě také uchovává počáteční stav, k dosažení této funkce je třeba znát fyzikální pochody v systému a jeho vnitřní uspořádání. Výše zmíněná stavová veličina

$x(t)$ má složky, které jsou funkcemi času. Tyto v systému jakožto náhradě řízeného procesu nemusí ve skutečnosti existovat, mohou to být funkce výstupní veličiny, její derivace apod. [1, 2, 5]

Vnější popis

Tento popis vychází ze závislosti mezi výstupem a vstupem dynamického systému, nebere v úvahu vnitřní uspořádání systému:

$$y = f(u) \quad (6)$$

Systém si lze představit jako „black box“ (ilustrace tohoto modelu je na obr. 1), jenž nám neposkytuje informaci o fyzikální podstatě a struktuře tohoto dynamického systému, zkoumáme časovou závislost vstupní a výstupní veličiny. Realizaci vnějšího popisu se dosáhne pomocí diferenciální rovnice systému, přenosu, impulsní funkce, přechodové funkce, frekvenčního přenosu, polohy pólů a nul přenosu. [1, 5]



Obr. 1: Model černé skříňky (black box)

2.2 Přístupy k modelování dynamických systémů

Na základě poznatků (získaných dat) o soustavě je možno konstruovat její model. Přístupovat k modelování dynamických systémů lze ze dvou směrů, analytickým (teoretickým) modelováním a experimentálním modelováním, oba přístupy jsou znázorněny na obr. 2.

Teoretické modelování

Tvorba analytického modelu je založena na zjednodušení získaných předpokladů o řízeném dynamickém systému. Výsledný model je získán kombinací metod kalkulu s aplikací rovnic (bilanční rovnice hmoty, energie, hybnosti; stavové rovnice popisující vratné děje; rovnice popisující nevratné děje jako je např. přenos tepla; apod.) a fyzikálních zákonů. Model je ve formě parciálních diferenciálních rovnic nebo obyčejných diferenciálních rovnic, které jsou obvykle tak složité, že se nadále zjednodušují podle potřeby konkrétní aplikace. [6, 7]

Experimentální modelování

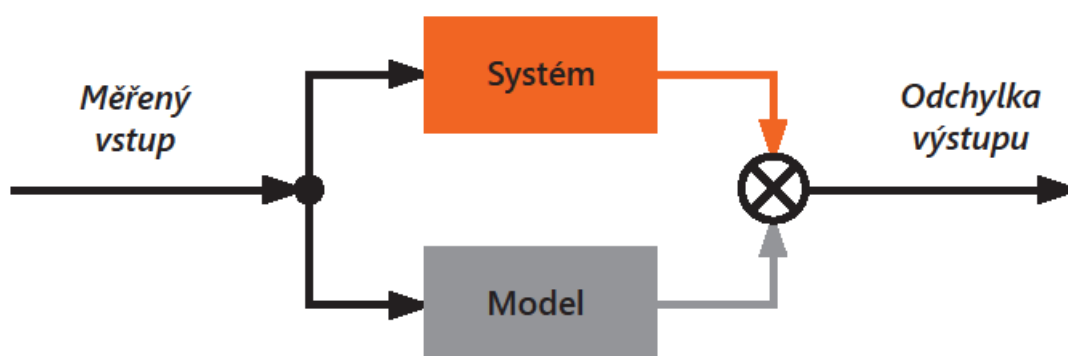
Experimentální modelování je identifikace systému na základě dat získaných ze soustavy. Vychází se z předchozích znalostí daného systému získaných z teoretické analýzy nebo experimentů již vykonaných na soustavě. Měří se vstupní a výstupní signály dynamického systému a na tyto signály se aplikují metody identifikace, které jako výstup poskytují určitý matematický model popisující obecný vztah mezi vstupem a výstupem. Vstupní signály mohou být přirozené signály působící na systém nebo se jedná o uměle vytvořené signály s předem definovaným chováním. Výsledný experimentální model je v závislosti na použití identifikační metody parametrický nebo neparametrický. [6, 7, 8]

Experimentální modelování	Analytické modelování
<ul style="list-style-type: none"> • Struktura modelu je předpokládána • Pouze chování na vstupu a výstupu je možno identifikovat • Parametry modelu jsou čísla • Model lze vytvořit pouze pro skutečný systém • Složitost modelu lze upravit pro potřeby konkrétní aplikace • Model je platný pouze pro zkoumaný systém v rámci provozních limitů 	<ul style="list-style-type: none"> • Struktura modelu vychází z přírodních zákonů • Modelován je vstup, výstup i vnitřní popis • Parametry modelu jsou funkce vlastností systému • Model lze vytvořit i pro neexistující systém • Modely jsou poměrně složité • Model je platný pro celou třídu procesů určitého typu v rámci různých provozních podmínkách • Vnitřní chování systému musí být známé a popsitelné matematickými zákony

Obr. 2: Přístupy k modelování dynamických systémů [7]

3 IDENTIFIKAČNÍ METODY

Identifikační metody mají za cíl rozpoznat charakteristiky nebo parametry modelovaného systému tak, že lze vytvořit model, který se co nejvíce přibližuje k chování procesu. Model použitý pro návrh regulace řízeného systému musí dostatečně umožňovat regulaci tohoto procesu. Na základě zadaného vstupu jsou výstupy ze systému a z modelu porovnány a vzniká odchylka výstupu ε , která by měla být u správně vytvořeného modelu minimální, pro představu ilustrace na obr. 3. [1, 2, 3, 4]



Obr. 3: Odchylka modelu od řízeného systému

Podle způsobu získávání parametrů modelu se identifikace rozděluje na průběžnou identifikaci (parametry modelu se postupně upravují s přibývajícími naměřenými daty) a na jednorázovou identifikaci (parametry modelu se získávají z celého souboru naměřených dat dynamického systému). Jak již bylo zmíněno podle typu získaného experimentálního modelu se identifikační metody rozdělují na neparametrickou identifikaci (modely bez specifické struktury s možným nekonečným počtem parametrů) a na parametrickou identifikaci (model se strukturou o konečném počtu parametrů).

Model obvykle reprezentuje funkce f , která vyjadřuje závislost výstupní veličiny ze systému $y(t)$ na vstupní veličině (obvykle akční veličina) $u(t)$ a dalších měřitelných veličin např. poruchové veličině $v(t)$. Na výstup mají také vliv neměřitelné veličiny okolí procesu, které obvykle nelze jasně popsat, označují se jako náhodná (stochastická) veličina $\eta(t)$. Obecný tvar modelu je v tomto případě podle rovnice 7. [2, 4, 6, 7]

$$y(t) = f[u(t), v(t), t] + \eta(t) \quad (7)$$

Postup identifikace systémů zahrnuje následující kroky [9]:

- Plánování a návrh experimentu
- Měření a uložení dat
- Výběr vhodné struktury modelu
- Výběr metody pro odhad parametrů
- Určení správnosti modelu

3.1 Metody k získání charakteristiky soustavy (neparametrické metody)

3.1.1 Identifikace z přechodového děje

Tento typ identifikace používá pro buzení soustavy jako vstupní signál skokovou změnu. Výstupem (odezvou systému) je přechodová charakteristika. Předpokládají se nulové počáteční podmínky a dále, že na soustavu nepůsobí kromě vstupní veličiny veličina jiná, např. poruchová. Identifikace je velmi citlivá na šum, potom je následný model nepřesný. [1, 4]

Nekmitavá (přetlumená) soustava prvního řádu s dopravním zpožděním

Model odpovídá přenosové funkci $F(p)$ ve tvaru:

$$F(p) = \frac{k}{Tp+1} e^{-dp} \quad (8)$$

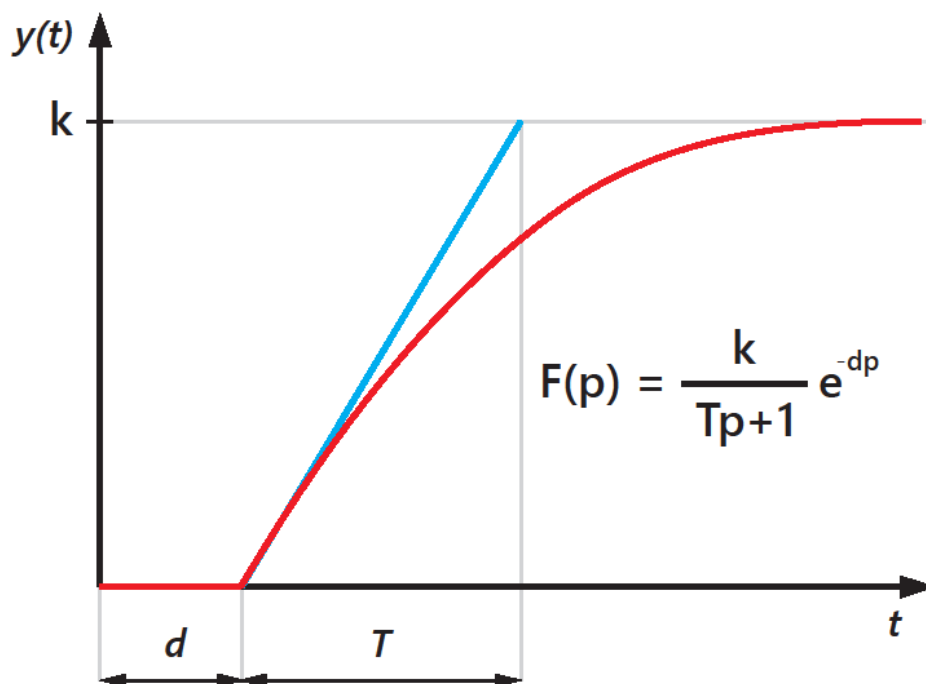
d je dopravní zpoždění, potom je odezva $y(t)$ na skokovou změnu u_0 :

$$y(t) = ku_0(1 - e^{-\frac{t-d}{T}}) \quad (9)$$

kde k je zesílení a je dáno vztahem:

$$k = \frac{y(\infty)}{u_0} \quad (10)$$

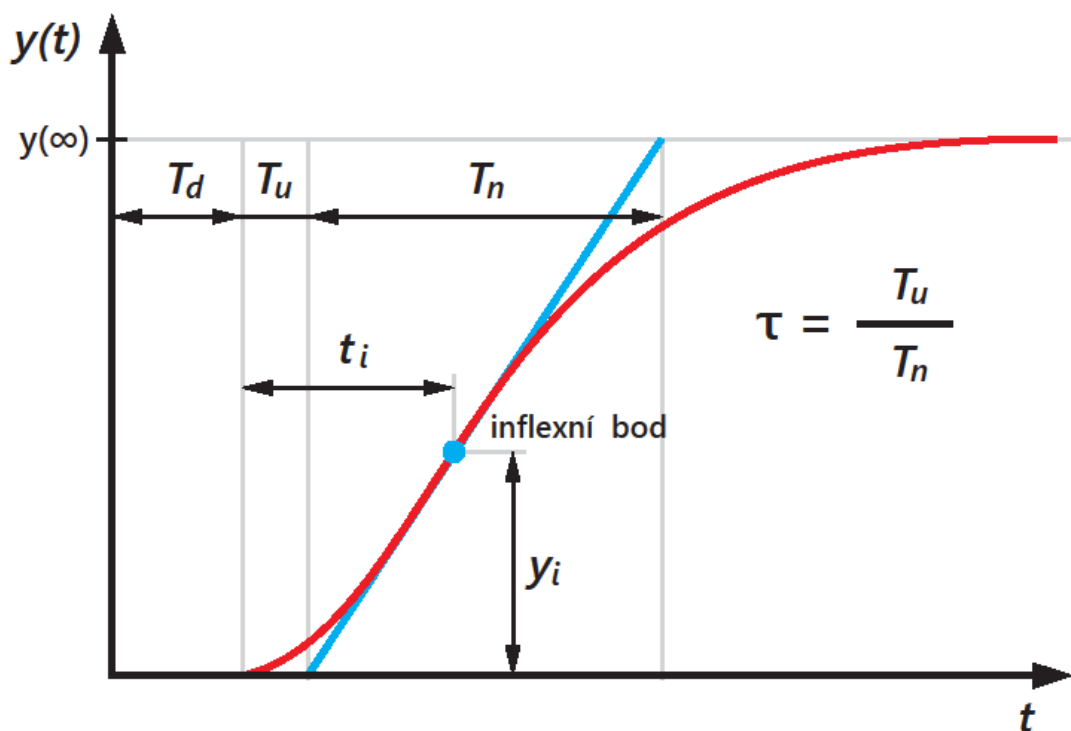
Jak lze vidět z obr. 4 - identifikací ze skokové změny se určí zesílení k a časová konstanta T . Skoková odezva dosahuje své 63%, 95%, 99% konečné ustálené hodnoty v časech $t = T, 3T, 5T$. Časová konstanta se získá konstrukcí tečny ke skokové odezvě v čase t_1 . Průsečík tečny s přímkou reprezentující ustálenou hodnotu odezvy $y(\infty)$ v čase t_2 . Časová konstanta je pak určena rozdílem $T = t_2 - t_1$ [1, 2, 7].



Obr. 4: Přechodová charakteristika nekmitavé soustavy 1.řádu s dopravním zpožděním

Nekmitavá (přetlumená) soustava druhého řádu s dopravním zpožděním

Skoková odezva je charakterizována dobou přechodu, která je rovna součtu doby průtahu T_u a doby náběhu T_n . Tyto úseky protíná tečna na časové ose a na ustálené hodnotě $y(\infty)$ vedena v inflexním bodě skokové odezvy, jak je zřejmé z obr. 5.



Obr. 5: Přechodová charakteristika nekmitavé soustavy 2.řádu s dopravním zpožděním T_d

Podle poměru $\tau = \frac{T_u}{T_n}$ doby průtahu a doby náběhu se vybere vhodný typ modelu. Pokud $\tau < 0,1$, potom má přenosová funkce $F(p)$ modelu soustavy tvar:

$$F_1(p) = \frac{k}{(T_1 p + 1)(T_2 p + 1)} e^{-dp} \quad (11)$$

Pokud je $\tau \geq 0,1$, pak má přenosová funkce $F(p)$ tvar:

$$F_2(p) = \frac{k}{(T p + 1)^n} e^{-dp} \quad (12)$$

V případě rozdílných časových konstant T_1 a T_2 se odečte t_1 pro hodnotu skokové odezvy $y_1 = 0,72y$. Součet s časových konstant je potom roven $s = T_1 + T_2 = \frac{t_1}{1,2564}$. Vypočítá se $t_2 = 0,3574s$, podle něj se odečte y_1 . Z tab. 1 se určí $\tau_2 = \frac{T_2}{T_1}$ podle normalizovaného $y_n = \frac{y(t_2)}{k}$.

Tab. 1: Tabulka pro různé časové konstanty T_1 a T_2 [9]

y_n	0,3	0,28	0,26	0,24	0,22	0,2	0,18	0,1611
τ_2	0,0	0,0435	0,0837	0,128	0,1838	0,2639	0,4031	1,0

Poměr τ_2 a součet s časových konstant umožňuje určit hledané konstanty T_1 a T_2 .

V druhém případě, tj. stejných časových konstant, se z hodnoty τ a souřadnice inflexního bodu y_i tab. 1 Tab. 1: určí podle tab. 2 Tab. 2: řád systému n .

Tab. 2: Tabulka pro stejné časové konstanty T [7]

n	2	3	4	5	6	7	8	9	10
y_i	0,104	0,218	0,319	0,410	0,493	0,570	0,642	0,709	0,773
τ	0,264	0,323	0,353	0,371	0,384	0,394	0,401	0,407	0,413

Pomocí časové souřadnice inflexního bodu t_1 se vypočte hledaná časová konstanta $T = \frac{t_i}{n-1}$. Postup a odvození této metody pomocí konstrukce tečny v inflexním bodě je převzato z [1, 7, 9].

Nekmitavé integrační soustavy (s astatizmem)

Model odpovídá přenosové funkci $F(p)$ ve tvaru:

$$F(p) = \frac{k}{p(Tp+1)} e^{-dp} \quad (12)$$

Pro identifikaci soustav s astatizmem se používají předchozí zmíněné metody, ale místo přechodové charakteristiky se provede její derivace (impulsní charakteristika), systém se převede na problém určení přenosu bez astatizmu.

Přechodová charakteristika s kmitavým průběhem

Tato charakteristika je popsána modelem:

$$F(p) = \frac{k\omega_0^2}{p+2\xi\omega_0p+\omega_0^2} e^{-dp} \quad (13)$$

kde ξ je součinitel poměrného tlumení. Odezva $y(t)$ je určena jako:

$$y(t) = k[1 - \frac{1}{\sqrt{1-\xi^2}} e^{-\xi\omega_0 t} \sin(\omega_0\sqrt{1-\xi^2}t + \arccos\xi)] \quad (14)$$

Určí se postupně parametry, zesílení k :

$$k = \frac{y(\infty)}{u_0} \quad (15)$$

Z polohy prvního maxima a minima se určí t_1 a t_2 , odečtou se hodnoty odezvy pro $y_1 = y(t_1)$ a $y_2 = y(t_2)$. Ze vztahů $y_1 = k(1 + M)$ a $y_2 = k(1 - M^2)$ se určí M .

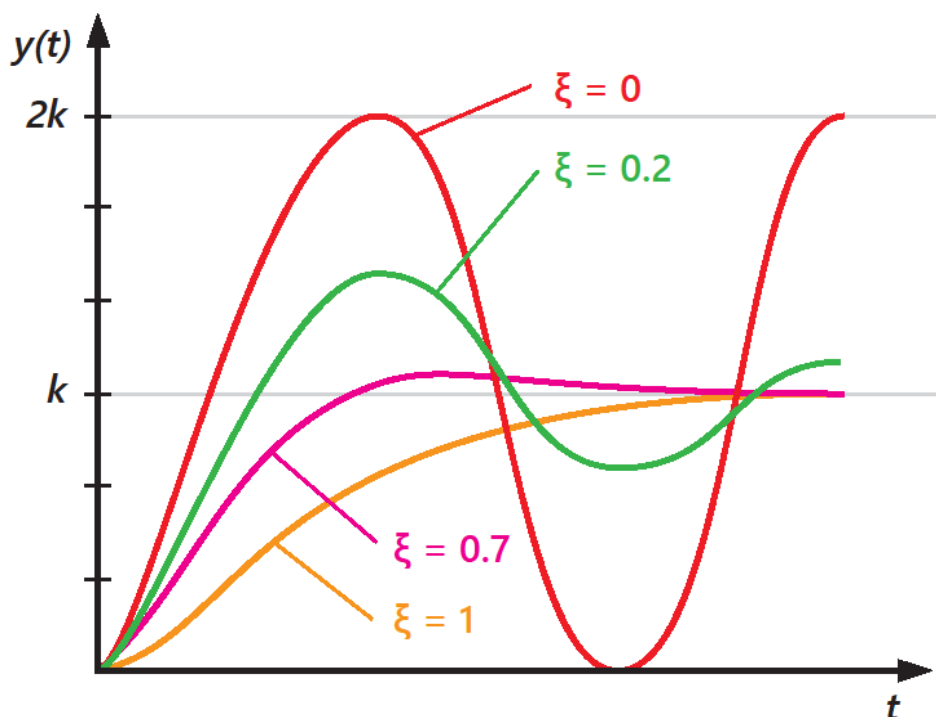
Součinitel poměrného tlumení dle vztahu:

$$\xi = \frac{-\ln M}{\sqrt{\pi^2 + (\ln M)^2}} \quad (16)$$

Z periody kmitů T se vypočte

$$\omega_0 = \frac{2\pi}{T\sqrt{1-\xi^2}} \quad (17)$$

Kmitavou odezvu ilustruje obr. 6. Postup a odvození této metody je převzato z [9].

Obr. 6: Odezvy pro různá tlumení ξ

3.1.2 Identifikace z frekvenčního vstupu

Identifikace používá jako vstupní signál $u(t)$ typu funkce sinus:

$$u(t) = U \sin(\omega t) \quad (18)$$

kde U je amplituda vstupu a ω je jeho úhlová frekvence. Po ustálení přechodového děje (systém je asymptoticky stabilní) je na výstupním signálu $y(t)$ jiná amplituda Y a fázový posun φ :

$$y(t) = Y \sin(\omega t + \varphi) \quad (19)$$

Vstupní a výstupní signál lze vyjádřit v komplexním tvaru:

$$u(t) = U e^{j\omega t} \quad (20)$$

$$y(t) = Y e^{j(\omega t + \varphi)} \quad (21)$$

jež představují v komplexní rovině vektory otáčející se úhlovou rychlostí ω . Jejich poměr je vyjádřen frekvenčním přenosem soustavy:

$$G(j\omega) = \frac{y(t)}{u(t)} = \frac{Y}{U} e^{j\varphi} \quad (22)$$

Grafickým vyjádřením frekvenčního přenosu se získá frekvenční charakteristika (amplitudo-fázová, viz obr. 7). Přenos se upraví na složkový tvar části reálné a komplexní:

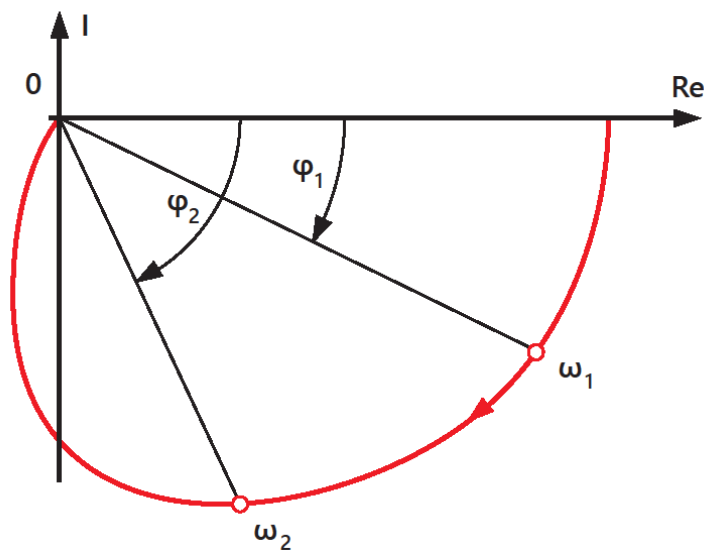
$$G(j\omega) = \text{Re}[G(j\omega)] + j\text{Im}[G(j\omega)] \quad (23)$$

nebo na exponenciální tvar získaný ze složkového tvaru komplexního čísla:

$$G(j\omega) = A e^{j\varphi(\omega)} \quad (24)$$

$$A = |G(j\omega)| \quad (25)$$

$$\varphi = \arctg \left(\frac{\text{Im}[G(j\omega)]}{\text{Re}[G(j\omega)]} \right) \quad (26)$$



Obr. 7: Amplitudo-fázová frekvenční charakteristika

Výše popsanou charakteristiku lze rozdělit na amplitudovou a fázovou. Lineární souřadnice se nahradí logaritmickými z důvodu rozšíření frekvenčního pásma. Obě tyto charakteristiky mají vodorovnou osu tvořenou frekvencí v logaritmickém měřítku. V amplitudové charakteristice je na svislé ose vynášena amplituda přenosu A v jednotkách decibel:

$$A[dB] = 20 \log(A[-]) = 20 \log\left(\left|\frac{Y}{U}\right|\right) \quad (27)$$

Na svislou osu fázové charakteristiky je vynášena fáze φ ve stupních nebo radiánech.

Konstrukce amplitudové a fázové charakteristiky

Přenos jednoduchého systému:

$$G(s) = \frac{k}{1+Ts} \quad (28)$$

se převede na exponenciální tvar frekvenčního přenosu:

$$G(j\omega) = \frac{k}{\sqrt{1+\omega^2 T^2}} e^{-j \arctg(\omega T)} \quad (29)$$

Průběh amplitudové charakteristiky se určí pro frekvence menší, než je lomová frekvence $\omega_l = \frac{1}{T}$ (viz obr. 8). Amplituda se potom podle rovnice (27) dále počítá:

$$A[dB] = 20 \log(|G(j\omega)|) = 20 \log(k) - 20 \log(\sqrt{1 + \omega^2 T^2}) \quad (30)$$

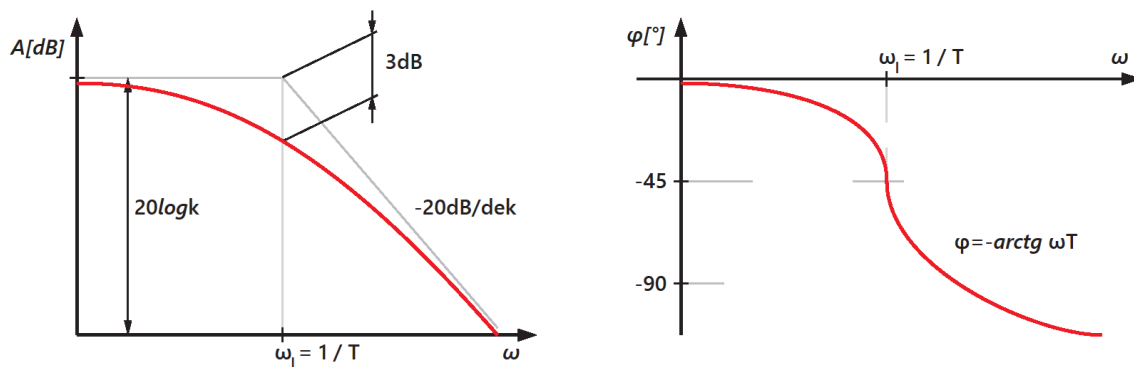
kde T je některá časová konstanta přenosu. Pro tyto frekvence platí, že pro $\omega T < 1$ lze zjednodušit výpočet amplitudy:

$$A[dB] = 20 \log(k) \quad (31)$$

amplitudovou charakteristiku lze vlevo od lomové frekvence ztotožnit s vodorovnou přímkou, která představuje asymptotu skutečného průběhu charakteristiky. Vpravo od lomové frekvence platí, že $\omega T > 1$, analogicky se zanedbá jednička ve výrazu pod odmocninou a amplitudová charakteristika se určí jako:

$$A[dB] = 20 \log(k) - 20 \log(\omega T) \quad (32)$$

V této části má asymptota pro přenos (28) s působením jednoho pólu sklon -20dB/dek , kde *dek* je rozmezí frekvencí mezi ω a 10ω . Na obr. 8 je také naznačena konstrukce fázové charakteristiky, ze které je patrné, že lomová frekvence odpovídá fázi 45° , tedy $\varphi = \arctg\left(\frac{1}{T}\right) = \arctg(1) = \frac{\pi}{4}$.



Obr. 8: Amplitudová a fázová charakteristika

Postup a odvození této metody je převzato z [1, 7].

Frekvenční analýza

Pro skutečný systém určený modelem:

$$Y(s) = G(s)U(s) + E(s) \quad (33)$$

je rovnice vstupu:

$$u(t) = A\sin(\omega t) \quad (34)$$

a reakce na tento vstup je potom rovnice výstupu:

$$y(t) = B\sin(\omega t + \varphi) + e(t) \quad (35)$$

kde A je amplituda vstupu, $e(t)$ je stochastická porucha, $E(s)$ její Laplaceova transformace. V důsledku přítomnosti šumu nelze přesně určit amplitudu B a fázový posun φ . Užitím korelační techniky lze omezit vliv šumu, protože se zpracuje informace z delšího časového intervalu. Pro redukci šumu touto metodou je žádoucí provést dlouhé identifikační experimenty.

Naměřený výstupní signál se vynásobí funkcí $\sin(\omega t)$ a provede se integrace přes jednu periodu T , získá se signál y_s . Stejně tak se výstup násobený funkcí $\cos(\omega t)$, integruje se přes jednu periodu $y(t)$ (což je T) a získá se signál y_c . Tyto signály lze vyjádřit jako:

$$\begin{aligned} y_s(T) &= \int_0^T y(t)\sin(\omega t)dt = \\ &= \frac{BT}{2}\cos(\varphi) - \frac{B}{2}\int_0^T y(t)\cos(2\omega t + \varphi)dt + \int_0^T e(t)\sin(\omega t)dt \end{aligned} \quad (36)$$

$$\begin{aligned} y_c(T) &= \int_0^T y(t)\cos(\omega t)dt = \\ &= \frac{BT}{2}\sin(\varphi) - \frac{B}{2}\int_0^T y(t)\sin(2\omega t + \varphi)dt + \int_0^T e(t)\cos(\omega t)dt \end{aligned} \quad (37)$$

Pokud měření je bez šumu ($e(t) = 0$) a doba integrace přes periodu T je násobek periody funkce sinus ($T = k2\pi/\omega$), lze vztahy (37) a (38) psát jako:

$$y_s(T) = \frac{BT}{2} \cos(\varphi) \quad (39)$$

$$y_c(T) = \frac{BT}{2} \sin(\varphi) \quad (40)$$

Z těchto vztahů se určí hledaná amplituda B a fázový posun φ výstupního signálu. Dále podle vztahů:

$$y_s(T) = \frac{AT}{2} \operatorname{Re}[G(j\omega)] \quad (41)$$

$$y_c(T) = \frac{AT}{2} \operatorname{Im}[G(j\omega)] \quad (42)$$

lze určit $|G(j\omega)|$ pro popis $G(j\omega)$. Postup a odvození frekvenční analýzy je převzato ze zdrojů [7, 8, 9].

3.1.3 Korelační metoda

Tato analýza používá na vstupu bílý šum. Model je váhová funkce:

$$y(t) = \sum_{k=0}^{\infty} h(k)u(t-k) + v(t) \quad (43)$$

kde $h(k)$ je váhová sekvence, $v(t)$ je porucha (šum). Předpoklad je, že vstup nezávisí na poruše (šumu). Pro autokovarianční funkci platí Wiener-Hopfova rovnice:

$$r_{yu}(\tau) = \sum_{k=0}^{\infty} h(k) r_u(\tau - k) \quad (44)$$

jejíž členy r_{yu} , r_u mohou být odhadnuty jako:

$$\hat{r}_{yu}(\tau) = \frac{1}{N - \max(\tau, 0) + \min(\tau, 0)} \sum_{t=1 - \min(\tau, 0)}^{N - \max(\tau, 0)} y(t + \tau)u(t) \quad (45)$$

pro $\tau = 0, \pm 1, \pm 2, \dots$

$$\hat{r}_u(\tau) = \frac{1}{N - \tau} \sum_{t=1}^{N - \tau} u(t + \tau)u(t) \quad (46)$$

pro $\tau = 0, 1, 2, \dots$

Odhad váhové funkce je potom určen soustavou s nekonečným počtem rovnic. Při použití bílého šumu jako vstupu platí $r_u(\tau) = 0$ pro $\tau \neq 0$. A tedy

$$h(k) = \frac{r_{yu}(k)}{r_u(0)} \quad (47)$$

jenž se spočítá odhadem $\hat{r}_{yu}(k)$. Odhad váhové sekvence $\hat{h}(k)$ je dán nekonečným množstvím rovnic, proto se používá useknutá váhová funkce: $h(k) = 0$ pro $k \geq M$, kde M je velké ve srovnání s dominantní časovou konstantou. Pak pro odhad platí:

$$\hat{r}_{uy}(\tau) = \sum_{k=0}^{M-1} \hat{h}(k) \hat{r}_u(\tau - k) \quad (48)$$

$$\begin{pmatrix} \hat{r}_{uy}(0) \\ \vdots \\ \hat{r}_{uy}(M-1) \end{pmatrix} = \begin{pmatrix} \hat{r}_u(0) & \cdots & \hat{r}_u(M-1) \\ \vdots & \ddots & \vdots \\ \hat{r}_u(M-1) & \cdots & \hat{r}_u(0) \end{pmatrix} \begin{pmatrix} \hat{h}(0) \\ \vdots \\ \hat{h}(M-1) \end{pmatrix} \quad (49)$$

Postup a odvození korelační metody vychází ze zdrojů [8, 9].

3.1.4 Spektrální analýza

Metoda není citlivá na šum, možnost jejího použití je téměř pro libovolný vstupní signál. Podstatou spektrální analýzy je zjistit, jak jsou dané frekvence zastoupeny v analyzovaném signálu. Přenosová funkce je ve formě frekvenční charakteristiky. Předpokládá se, že $u(t)$ je stacionární stochastický proces, autokovarianční funkce je

$$r_u(\tau) = E[u(t + \tau) - m_u][u(t) - m_u] \quad (50)$$

Dle definice stochastického procesu je spektrální hustota

$$\Phi_u(\omega) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} r_u(\tau) e^{-i\omega\tau} \quad (51)$$

Vztah pro výpočet autokovarianční funkce ze spektrální hustoty je:

$$r_u(\tau) = \int_{-\pi}^{\pi} \Phi_u(\omega) e^{i\omega\tau} d\omega \quad (52)$$

Perioda vzorkování se uvažuje $T_{vz} = 1$, interval pro ω $(-\pi, \pi)$. Pokud se neuvažuje určená perioda vzorkování, tak obecně je frekvence Nyquistova $\omega = \pi/T_{vz}$. Interval pro integraci tedy je $(-\omega, \omega)$. Předpokládá se dále průchod vstupu $u(t)$ lineárním systémem:

$$y(t) = \sum_{k=0}^{\infty} h(k)u(t - k) \quad (53)$$

kde posloupnost $h(k)$ představuje váhovou funkci. Tento filtr je stabilní tehdy, když $k \rightarrow \infty$ pak $\|h(k)\| \rightarrow 0$. Potom je signál $y(k)$ stacionární. Charakteristikami tohoto signálu jsou vzájemná spektrální hustota $\Phi_y(\omega)$, vzájemná kovarianční funkce $r_{yu}(\tau)$ a střední hodnota m_y . K jejich výpočtu se zavádí operátor přenosu:

$$H(q^{-1}) = \sum_{k=0}^{\infty} h(k)q^{-k} \quad (54)$$

kde q^{-k} je operátor zpětného k -krokového posunu. Potom lze psát průchod vstupu $u(t)$ lineárním systémem vztahem:

$$y(t) = H(q^{-1})u(t) \quad (55)$$

Vypočítá se střední hodnota:

$$m_y = Ey(t) = \sum_{k=0}^{\infty} h(k)Eu(t - k) = \sum_{k=0}^{\infty} h(k)m_u = H(1)m_u \quad (56)$$

kde $H(1)$ je zesílení filtru v ustáleném stavu. Zároveň platí:

$$\tilde{y}(t) = y(t) - m_y \quad (57)$$

$$\tilde{u}(t) = u(t) - m_u \quad (58)$$

Potom je odhad $\tilde{y}(t)$:

$$\tilde{y}(t) = \sum_{k=0}^{\infty} h(k)u(t - k) - \sum_{k=0}^{\infty} h(k)m_u = H(q^{-1})\tilde{u}(t) \quad (59)$$

Výpočet kovarianční funkce za předpokladu $m_u = 0$:

$$r_{yu}(\tau) = Ey(t + \tau)y(t) = \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} h(j)r_u(\tau - j + k)h(k) \quad (60)$$

Podle výše zmíněného definičního vztahu pro spektrální hustotu vypočteme $\Phi_y(\omega)$, jenž nám následujícím vztahem vyjadřuje závislost spektrální hustoty výstupního signálu na přenosové funkci $H(e^{i\omega})$:

$$\begin{aligned}\Phi_y(\omega) &= \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} r_y(\tau) e^{-i\omega\tau} = \dots = \left[\sum_{j=0}^{\infty} h(j) e^{-ij\omega} \right] \Phi_u(\omega) \left[\sum_{k=0}^{\infty} h(k) e^{-ik\omega} \right] = \\ &= H(e^{-i\omega}) \Phi_u(\omega) H(e^{i\omega})\end{aligned}\quad (61)$$

Vztah pro vzájemnou kovarianční funkci:

$$r_{yu}(t) = Ey(t + \tau)u(t) = \sum_{j=0}^{\infty} h(j)r_u(\tau - j) \quad (62)$$

a pro vzájemnou spektrální hustotu:

$$\begin{aligned}\Phi_{yu}(\omega) &= \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} r_{yu}(\tau) e^{-i\omega\tau} = \sum_{j=0}^{\infty} h(j) e^{-ij\omega} \left[\frac{1}{2\pi} \sum_{\tau'=-\infty}^{\infty} r_u(\tau') e^{-i\tau'\omega} \right] = \\ &= H(e^{-i\omega}) \Phi_u(\omega)\end{aligned}\quad (62)$$

kde

$$\Phi_{yu}(\omega) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} r_{yu}(\tau) e^{-i\tau\omega} \quad (63)$$

$$\Phi_u(\omega) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} r_u(\tau) e^{-i\tau\omega} \quad (64)$$

$$H(e^{-i\omega}) = \sum_{k=0}^{\infty} h(k) e^{-ik\omega} \quad (65)$$

Potom odhad přenosové funkce je podle vztahu:

$$\hat{H}(e^{-i\omega}) = \hat{\Phi}_{yu}(\omega) / \hat{\Phi}_u(\omega) \quad (66)$$

k výpočtu odhadů spektrálních hustot se použijí vztahy:

$$\hat{\Phi}_{yu}(\omega) = \frac{1}{2\pi} \sum_{\tau=-N}^N \hat{r}_{yu}(\tau) e^{-i\tau\omega} \quad (67)$$

$$\hat{\Phi}_u(\omega) = \frac{1}{2\pi} \sum_{\tau=-N}^N \hat{r}_u(\tau) e^{-i\tau\omega} \quad (68)$$

Naznačení výpočtu $\hat{\Phi}_{yu}(\omega)$ po dosazení za $\hat{r}_{yu}(\tau)$:

$$\hat{\Phi}_{yu}(\omega) = \frac{1}{2\pi N} \sum_{\tau=-N}^N \sum_{t=1-\min(\tau,0)}^{N-\max(\tau,0)} y(t + \tau) u(t) e^{-i\tau\omega} \quad (69)$$

substitucí $s = t + \tau$ se dostane:

$$\hat{\Phi}_{yu}(\omega) = \frac{1}{2\pi N} \sum_{s=1}^N \sum_{t=1}^N y(s)u(t)e^{-is\omega} e^{it\omega} = \frac{1}{2\pi N} Y_N(\omega)U_N(-\omega) \quad (70)$$

kde Fourierovy transformace sekvencí $y(t)$ $u(t)$ jsou:

$$Y_N(\omega) = \sum_{s=1}^N y(s)e^{-is\omega} \quad (71)$$

$$U_N(\omega) = \sum_{s=1}^N u(s)e^{-is\omega} \quad (72)$$

Pro $\omega = \frac{0,2\pi}{N}, \frac{4\pi}{N}, \dots, \pi$ se mohou vypočíst rychlou Fourierovou transformací FFT. Obdobně pak výpočet odhadu $\hat{\Phi}_u(\omega)$ (také označovaný jako periodogram):

$$\hat{\Phi}_u(\omega) = \frac{1}{2\pi N} |U_N(\omega)|^2 \quad (73)$$

Odhad přenosu je pak:

$$\hat{H}(e^{-i\omega}) = \frac{\hat{\Phi}_{yu}(\omega)}{\hat{\Phi}_u(\omega)} = \frac{Y_N(\omega)}{U_N(\omega)} \quad (74)$$

Tento postup vede k nepřesným výsledkům z důvodu nepřesného odhadu $\hat{r}_{yu}(\tau)$ pro velké hodnoty τ a jednotlivé členy $\hat{r}_{yu}(\tau)$ při výpočtu $\hat{\Phi}_{yu}(\omega)$ jsou váženy stejnou vahou. Pokud členy s velkými hodnotami τ mají jinou váhu, lze dosáhnout přesnějších výsledků. Namísto rovnice (63) se zavádí výpočet vzájemné spektrální hustoty pomocí „zpožďovacího okénka“ $w(\tau)$:

$$\hat{\Phi}_{yu}(\omega) = \frac{1}{2\pi} \sum_{\tau=-N}^N \hat{r}_{yu}(\tau)w(\tau)e^{-i\tau\omega} \quad (75)$$

Volí se tak, aby při $\tau = 0$ bylo $w(0) = 1$, aby klesalo s rostoucím τ a aby bylo $w(\tau) = 0$ pro velké τ (myšleno $5 \div 10$ % z počtu N dat). Analogicky pro odhad spektrální hustoty $\hat{\Phi}_u(\omega)$.

Pravouhlé okénko

$$w_1(\tau) = 1 \text{ pro } |\tau| \leq M$$

$$w_1(\tau) = 0 \text{ pro } |\tau| > M$$

Barlettovo okénko

$$w_2(\tau) = 1 - |\tau|/M \text{ pro } |\tau| \leq M$$

$$w_2(\tau) = 0 \text{ pro } |\tau| > M$$

Hammingovo okénko

$$w_3(\tau) = \frac{1}{2} \left(1 + \cos \left(\frac{\pi\tau}{M} \right) \right) \text{ pro } |\tau| \leq M$$

$$w_3(\tau) = 0 \text{ pro } |\tau| > M$$

Postup a odvození spektrální analýzy je čerpán ze zdrojů [7, 8, 9].

3.2 Metody k získání parametrů systému (parametrické metody)

Parametrické metody identifikace se používají k získání parametrů určitého modelu. Rozlišujeme několik modelů, které budou popsány v následující podkapitole.

3.2.1 Parametrické modely

Model MA (Moving Average)

MA model lze zapsat jako:

$$y(k) = b_1 u(k-1) + b_2 u(k-2) + \dots + b_n u(k-n) \quad (76)$$

kde b_i jsou parametry modelu. Model v diskrétním tvaru je určen z:

$$F(z) = B(z) \quad (77)$$

Např. model ve tvaru:

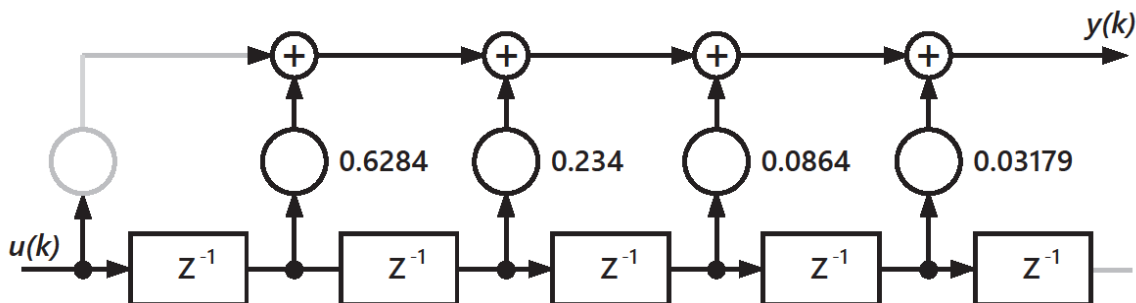
$$F(s) = \frac{e^{-0,1s}}{10s+1} \quad (78)$$

se převede do diskrétního tvaru, zvolí se perioda vzorkování např. $T = 10s$. Filtr „konečná impulsová odezva“ FIR (Finite Impulse Response) je model určen tvarem:

$$F(z) = \frac{0,6284z^{-1} + 0,0037z^{-2}}{1 - 0,3679z^{-1}} \quad (79)$$

Parametry modelu z obr. 9 se určí z rovnice odezvy na impuls o velikosti 1:

$$Y(z) = F(z)U(z) = F(z)1 = 0,6284z^{-1} + 0,234z^{-2} + 0,0864z^{-3} + \dots$$



Obr. 9: Model MA filtru FIR

Pokud se zkrátí perioda vzorkování, počet členů modelu se zvyšuje. Dále filtr typu „klouzavý průměr“ řádu n má všechny parametry modelu o stejné velikosti:

$$b_i = \frac{1}{n+1} \quad (80)$$

Perioda vzorkování se volí:

$$T = \left(\frac{1}{6} \div \frac{1}{15} \right) T_{95} \quad (81)$$

kde T_{95} je čas po jednotkovém vstupním skoku, za kterou dosáhne výstup hodnoty 95%.

Model AR (Auto-Regresive)

V diskretním tvaru je tento model typu „nekonečná impulsová odezva“ IIR (Infinite Impulse Response) určen funkcí:

$$F(z) = \frac{1}{A(z)} = \frac{Y(z)}{U(z)} \quad (82)$$

Model podle přenosové funkce z rovnice (79):

$$A(z) = \frac{1 - 0,3679z^{-1}}{0,6284z^{-1} + 0,0037z^{-2}} = \frac{Y(z)}{U(z)} = 1,591z^{-1} - 0,5945z^{-2} - 0,0035z^{-3} + 0,00002z^{-4} + \dots$$

po úpravě

$$F(z) = \frac{0,6284z}{1 - 0,3736z^{-1} - 0,0022z^{-2}}$$

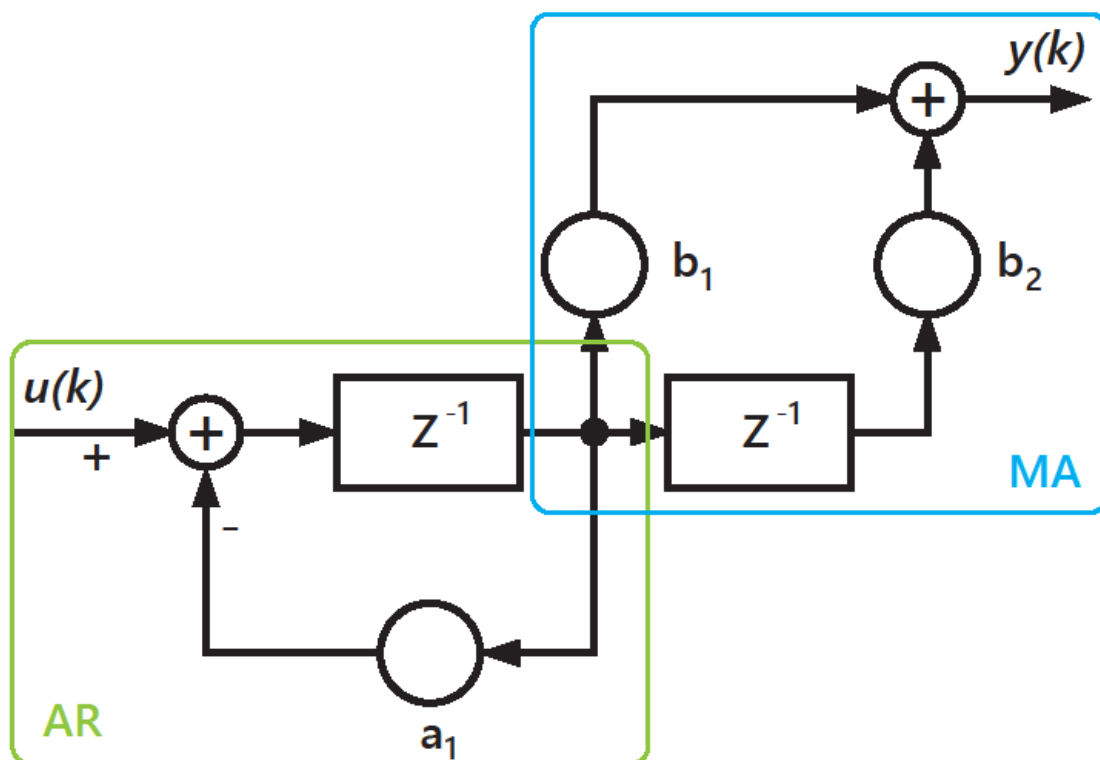
$$Y(z) = 0,6284zU(z) + (0,3736z^{-1} + 0,0022z^{-2})Y(z)$$

$$y(k) = 0,6284u(k+1) + 0,3736y(k-1) + 0,0022y(k-2)$$

Výstup závisí na okamžité hodnotě vstupu a také na předchozích hodnotách výstupů násobených parametry a_i .

Model ARMA (Auto-Regresive Moving Average)

Tento smíšený model je typu IIR, vzniká kombinací modelů AR a MA, jak je zřejmé z obr. 10.



Obr. 10: Model ARMA

Dosud uvedené modely je možné použít pro jednorázovou identifikaci, dále budou popsány modely vhodné pro průběžnou identifikaci.

Modely ARMAX, ARX

Pro popis dalších modelů se uvažuje struktura modelu pro n_y rozměrný vstup y , n_u rozměrný výstup u , případně další měřitelné veličiny ovlivňující vstup jako měřené poruchové veličiny v :

$$A(z^{-1})y = B(z^{-1})u + D(z^{-1})v + C(z^{-1})e \quad (83)$$

kde

$$A(z^{-1}) = 1 + a_1 z^{-1} + \dots + a_{na} z^{-na} \quad (84)$$

$$B(z^{-1}) = b_1 z^{-1} + \dots + b_{nb} z^{-nb} \quad (85)$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + \dots + c_{nc} z^{-nc} \quad (86)$$

$$D(z^{-1}) = d_1 z^{-1} + \dots + d_{nd} z^{-nd} \quad (87)$$

Na výstupu ze soustavy se projevují také neměřitelné vlivy okolí, tyto se reprezentují jako porucha, která projde předem známým filtrem a získá se náhodná měření nepřístupná složka e . Vektor hledaných parametrů modelu ARMAX je dán:

$$\theta = [a_1, \dots, a_{na}, b_1, \dots, b_{nb}, d_1, \dots, d_{nc}, c_1, \dots, c_{nc}]^T \quad (88)$$

Při identifikaci parametrů polynomů A , B , C , D z naměřených dat vyvstává problém v rozlišení, co je ve změřených hodnotách šum, protože složka e není měřitelná, nelze určit koeficienty polynomu $C(z^{-1})$.

Model ARX (regresní model typu ARMA) modeluje výstup soustavy podle vztahu:

$$A(z^{-1})y = B(z^{-1})u + D(z^{-1})v + e \quad (89)$$

Model se často zapisuje ve vektorové formě:

$$y(k) = \theta^T(k) \varphi(k-1) + e(k) \quad (90)$$

Vektor hledaných parametrů:

$$\theta(k) = [a_1, \dots, a_{na}, b_1, \dots, b_{nb}, d_1, \dots, d_{nc}]^T \quad (91)$$

a vektor dat tzv. regresor:

$$\begin{aligned} \varphi^T(k-1) &= \\ &= [-y(k-1), \dots, -y(k-na), u(k-1), \dots, u(k-nb), v(k-1), \dots, v(k-nd)] \end{aligned} \quad (92)$$

Postupy a odvození vztahů pro parametrické modely uvedené v podkapitole 3.2.2 jsou převzaty z [3, 5, 6, 7, 8].

3.2.2 Jednorázová metoda nejmenších čtverců

Předpokládá se lineární statický systém popsáný vztahem:

$$\mathbf{y} = \Phi \boldsymbol{\theta} \quad (93)$$

kde $\boldsymbol{\theta}$ je vektor parametrů, $\mathbf{y} = [y_N, y_{N-1}, \dots, y_1]^T$, $\Phi = [\varphi_N, \varphi_{N-1}, \dots, \varphi_1]^T$, výstup y_i je tvořen kombinací vstupních hodnot φ_i . Takto popsaná soustava lineárních rovnic nemusí mít žádné řešení. Modeluje se pak systém jako:

$$\tilde{\mathbf{y}} = \Phi \tilde{\boldsymbol{\theta}} \quad (94)$$

kde $\tilde{\boldsymbol{\theta}}$ je vektor odhadů parametrů systému. Z porovnání odhadovaného výstupu z modelu a skutečného výstupu ze systému se získá odchylka $\boldsymbol{\varepsilon} = \tilde{\mathbf{y}} - \mathbf{y}$.

Z toho vyplývá model:

$$\mathbf{y} = \Phi \tilde{\boldsymbol{\theta}} + \boldsymbol{\varepsilon} \quad (95)$$

Pro co největší přiblížení parametrů $\tilde{\boldsymbol{\theta}}$ skutečným musí být odchylka $\boldsymbol{\varepsilon}$ co nejmenší. Používá se kritérium součtu kvadrátů odchylek:

$$V(\tilde{\boldsymbol{\theta}}) = \frac{1}{2} \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} = \frac{1}{2} (\mathbf{y} - \Phi \tilde{\boldsymbol{\theta}})^T (\mathbf{y} - \Phi \tilde{\boldsymbol{\theta}}) \quad (96)$$

hledání jejího minima je hledání její derivace rovna nule:

$$\frac{\partial}{\partial \tilde{\boldsymbol{\theta}}} V(\tilde{\boldsymbol{\theta}}) = 0$$

výsledkem operace je

$$\tilde{\boldsymbol{\theta}} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y} \quad (97)$$

kde $\mathbf{P} = (\Phi^T \Phi)^{-1}$ je kovarianční matice:

$$\tilde{\boldsymbol{\theta}} = \mathbf{P} \Phi^T \mathbf{y} \quad (98)$$

Pro lineární dynamické systémy s jedním vstupem a výstupem je přenosová funkce:

$$F(z) = \frac{b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}} \quad (99)$$

n je řád systému, výstup $y(k)$ v k -tém kroku je:

$$y(k) = b_1 u(k-1) + b_2 u(k-2) + \dots + b_n u(k-n) - a_1 y(k-1) - \dots - a_n y(k-n) \quad (100)$$

pro N vzorků (musí platit $N \geq 2n$) a model $\mathbf{y} = \Phi \boldsymbol{\theta} + \boldsymbol{\varepsilon}$:

$$\begin{bmatrix} y(k) \\ \vdots \\ y(k+N-1) \end{bmatrix} = \begin{bmatrix} u(k-1) & \dots & u(k-n) & -y(k-1) & \dots & -y(k-n) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u(k+N-2) & \dots & u(k-n+N-1) & -y(k+N-2) & \dots & -y(k-n+N-1) \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_n \\ a_1 \\ \vdots \\ a_n \end{bmatrix} + \begin{bmatrix} \varepsilon_k \\ \vdots \\ \varepsilon_{k+N-1} \end{bmatrix} \quad (101)$$

Na základě tohoto vztahu lze vypočítat odhad parametrů během jednoho výpočetního kroku. V praktickém použití je počet řádků matice značně rozměrný, počítá se inverze matice velkého rozměru. Při příchodu dalšího vzorku dat se výpočet opakuje, nevyužívá se předchozího kroku.

3.2.3 Rekurzivní metoda nejmenších čtverců

Do výpočtu v kroku k je začleněn další vzorek z měření. Pro odhad parametrů v dalším kroku $k+1$ po rozšíření matice o další řádek platí:

$$\tilde{\boldsymbol{\theta}}(k+1) = \mathbf{P}(k+1) [\Phi^T(k) \quad \varphi(k+1)] \begin{bmatrix} y(k) \\ y(k+1) \end{bmatrix} \quad (102)$$

$$\tilde{\boldsymbol{\theta}}(k+1) = \mathbf{P}(k+1) [\Phi^T(k) y(k) \quad \varphi(k+1) y(k+1)] \quad (103)$$

$$\mathbf{P}(k+1) = [\Phi^T(k) \Phi(k) + \varphi(k+1) \varphi^T(k+1)]^{-1} \quad (104)$$

po delších úpravách dostáváme:

$$\mathbf{P}(k+1) = \mathbf{P}(k) - \mathbf{P}(k) \varphi(k+1) [1 + \varphi^T(k+1) \mathbf{P}(k) \varphi(k+1)]^{-1} \varphi^T(k+1) \mathbf{P}(k)$$

$$\begin{aligned} \tilde{\boldsymbol{\theta}}(k+1) = & \tilde{\boldsymbol{\theta}}(k) + \mathbf{P}(k) \varphi(k+1) [1 + \varphi^T(k+1) \mathbf{P}(k) \varphi(k+1)]^{-1} (y(k+1) \\ & - \varphi^T(k+1) \tilde{\boldsymbol{\theta}}(k)) \end{aligned}$$

Výsledné rovnice lze psát jako:

$$\tilde{\boldsymbol{\theta}}(k+1) = \tilde{\boldsymbol{\theta}}(k) + \mathbf{K}(k+1)(y(k+1) - \varphi^T(k+1) \tilde{\boldsymbol{\theta}}(k)) \quad (105)$$

$$\mathbf{K}(k+1) = \mathbf{P}(k) \varphi(k+1) [1 + \varphi^T(k+1) \mathbf{P}(k) \varphi(k+1)]^{-1} \quad (106)$$

$$\mathbf{P}(k+1) = \mathbf{P}(k) - \mathbf{K}(k+1) \varphi^T(k+1) \mathbf{P}(k) \quad (107)$$

kde $\mathbf{K}(k+1)$ je váhový součinitel, který vyjadřuje s jakým vlivem se bere v úvahu rozdíl mezi tím, co skutečně v kroku $k+1$ naměříme a tím co předpovídá model se současným nastavením parametrů. Pro použitelný výsledek je třeba provést minimálně $2n$ kroků. Vlastností této metody je, že ve velkém počtu vzorků N metoda pomalu konverguje v důsledku přiřazení stejné váhy jak novým naměřeným vzorkům, tak i všem přechozím.

Naopak rekurzivní metoda nejmenších čtverců je rozšířená o techniku exponenciálního zapomínání. Modifikací této metody je použití faktoru exponenciálního zapomínání λ . Pro určení minimální odchylky se vychází z modifikovaného kritéria:

$$V_e(\tilde{\boldsymbol{\theta}}) = \sum_{i=k_0}^k \lambda^{2(k-i)} \boldsymbol{\varepsilon}^2(i) \quad (108)$$

za podmínky $0 < \lambda^2 \leq 1$.

Výsledné modifikované rovnice jsou pak ve tvaru:

$$\tilde{\boldsymbol{\theta}}(k+1) = \tilde{\boldsymbol{\theta}}(k) + \mathbf{K}(k+1)(y(k+1) - \varphi^T(k+1) \tilde{\boldsymbol{\theta}}(k)) \quad (109)$$

$$\mathbf{K}(k+1) = \mathbf{P}(k) \varphi(k+1) [\lambda_{i+1} \mathbf{I} + \varphi^T(k+1) \mathbf{P}(k) \varphi(k+1)]^{-1} \quad (110)$$

$$\mathbf{P}(k+1) = (\mathbf{P}(k) - \mathbf{K}(k+1) \varphi^T(k+1) \mathbf{P}(k)) / \lambda_{i+1} \quad (111)$$

Pro počáteční nastavení metody volíme:

$$\mathbf{P}(0) = (10^4 \div 10^6) \mathbf{I} \quad (112)$$

$$\tilde{\boldsymbol{\theta}}(0) = (1 \dots 0)^T \quad (113)$$

$$\lambda_0 = 0,995 \div 0,999 \quad (114)$$

Např. pro volbu modelu soustavy třetího řádu

$$\varphi^T = (u(k-1), u(k-2), u(k-3), -y(k-1), -y(k-2), -y(k-3))$$

dostáváme výsledek ve formě parametrů:

$$\tilde{\boldsymbol{\theta}} = (\tilde{b}_1, \tilde{b}_2, \tilde{b}_3, \tilde{a}_1, \tilde{a}_2, \tilde{a}_3)^T$$

Postupy a odvození algoritmů metod uvedených v podkapitolách 3.2.2 a v 3.2.3 jsou převzaty z [3, 6].

4 REALIZACE VYBRANÝCH IDENTIFIKAČNÍCH METOD V PROSTŘEDÍ MATLAB

V rámci této kapitoly bude pojednáno o implementaci čtyř vybraných metod, jejich testování a porovnání mezi sebou. Tyto implementace jsou realizovány na 64-bitovém operačním systému Windows 10 Home v software Matlab verze R2018b (9.5.0.944444). Pro tuto verzi software není potřeba instalovat přídatné rozšíření nebo knihovny. Implementace zvolených metod je součástí přílohy jako skripty souborů *.m.

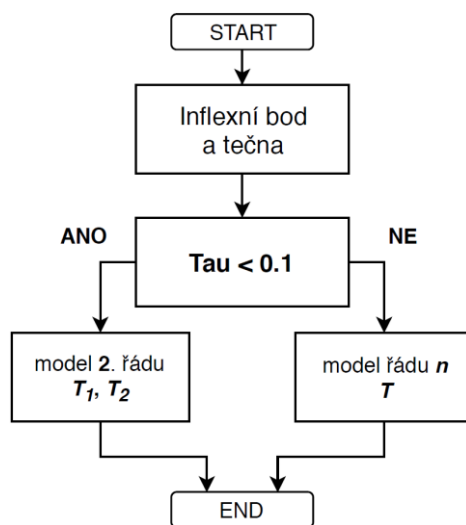
Metody identifikace, které byly zvoleny, jsou vybrány za účelem zastoupení metod jak z oblasti parametrických, tak neparametrických a taktéž pro identifikaci systémů v časové i ve frekvenční oblasti.

4.1 Identifikace nekmitavého systému z přechodové charakteristiky

Vstupem pro tuto metodu je přechodová charakteristika systému, který byl buzen skokovým signálem. Výstupem jsou parametry (zesílení systému k , časové konstanty T_1 a T_2) modelu vycházející z rovnice (11) nebo parametry (zesílení systému k , časová konstanta T a řád n) k vytvoření přenosu podle (12). Implementace vychází z kapitoly 3.1 a je v příloženém souboru stepInput.m.

4.1.1 Implementace

Pro tuto metodu byl zvolen jako vstupní signál jednotkový skok, diskretní čas o periodě 0.01. Po inicializaci vstupního signálu se pomocí přenosové funkce `tf()` vytvoří systém, jenž je předmětem identifikace. Funkce `lsim()` do toho systému pustí vstupní signál a nasimuluje jeho odezvu, kterou vrátí jako výstup. Stejným bodem této metody je výběr daného typu modelu podle velikosti poměru $\tau = \frac{T_u}{T_n}$, jak ukazuje diagram na obr. 11.



Obr. 11: Stejný bod určení typu modelu

K získání hodnot pro výše zmíněný poměr τ je zapotřebí určit inflexní bod a tečnu jím procházející. Obr. 12 ukazuje část implementace, kde je díky diferenci (nahrazující derivaci) přechodové charakteristiky získána směrnice tečny `slopeOfTangent` a souřadnice inflexního bodu y_i, t_i .

```

30 % GETTING THE INFLEX POINT
31 - der1 = diff(y) / tSample; % 1st derivation
32 - der1 = [0 ; der1]; % to have same length as y
33 - [~, indexDer1] = max(abs(der1));
34 - slopeOfTangent = der1(indexDer1);
35 - yi = y(indexDer1); ti = t(indexDer1);

```

Obr. 12: Určení směrnice tečny a souřadnic inflexního bodu

V postupu metody určení modelu 2.řádu s různými konstantami je využíváno tab. 1 ve výpočtu parametrů, tato tabulka je implementována v podobě dat vstupujících do funkce `interpolatedOutput()` (obr. 13).

```

56 - y2 = [0.3 0.28 0.26 0.24 0.22 0.2 0.18 0.1611];
57 - Tau2 = [0.0 0.0435 0.0837 0.128 0.1838 0.2639 0.4031 1.0];
58 - interpolatedOutput = interp1(y2, Tau2, valueToInterpolate, ...
59                               'linear', 'extrap');

```

Obr. 13: Implementace interpolace dat z tab. 1

Podle určeného typu modelu se pomocí získaných parametrů vytvoří model systému stejně tak jako jeho vzor na začátku této implementace a provede se simulace odezvy na vstupní signál. V implementaci je taky možnost si dané modely porovnat, což umožňuje výpis do Command Window Matlabu, anebo vykreslení do grafu.

4.1.2 Výsledky pro testované systémy

Metoda byla testována na různých systémech a splňuje požadavky na identifikaci z jednotkového skoku díky překrývání se křivky modelu s charakteristikou identifikovaných systémů. Pro výstup, který byl upraven tak, aby v sobě nesl šum (tato implementace je vidět na obr. 14), se tato metoda podle očekávání z jejích vlastností nedokázala vypořádat s určením sklonu tečny a tím výsledný model nebyl dostatečně přesný.

```

27 % SIMULATING OF APPLICATION OF STEP INTO A SYSTEM
28 - [y, ~] = step(sys, t, stepOptions);
29 - a = -0.5; b = 0.5;
30 - randomNumbers = (a + (b-a).*rand(length(y),1))*0.05;
31 - y = y + randomNumbers;

```

Obr. 14: Přidání šumu do výstupního signálu

Testovány byly identifikace následujících soustav:

$$G_1(s) = \frac{5}{(12s+1)(s+1)} \quad (115)$$

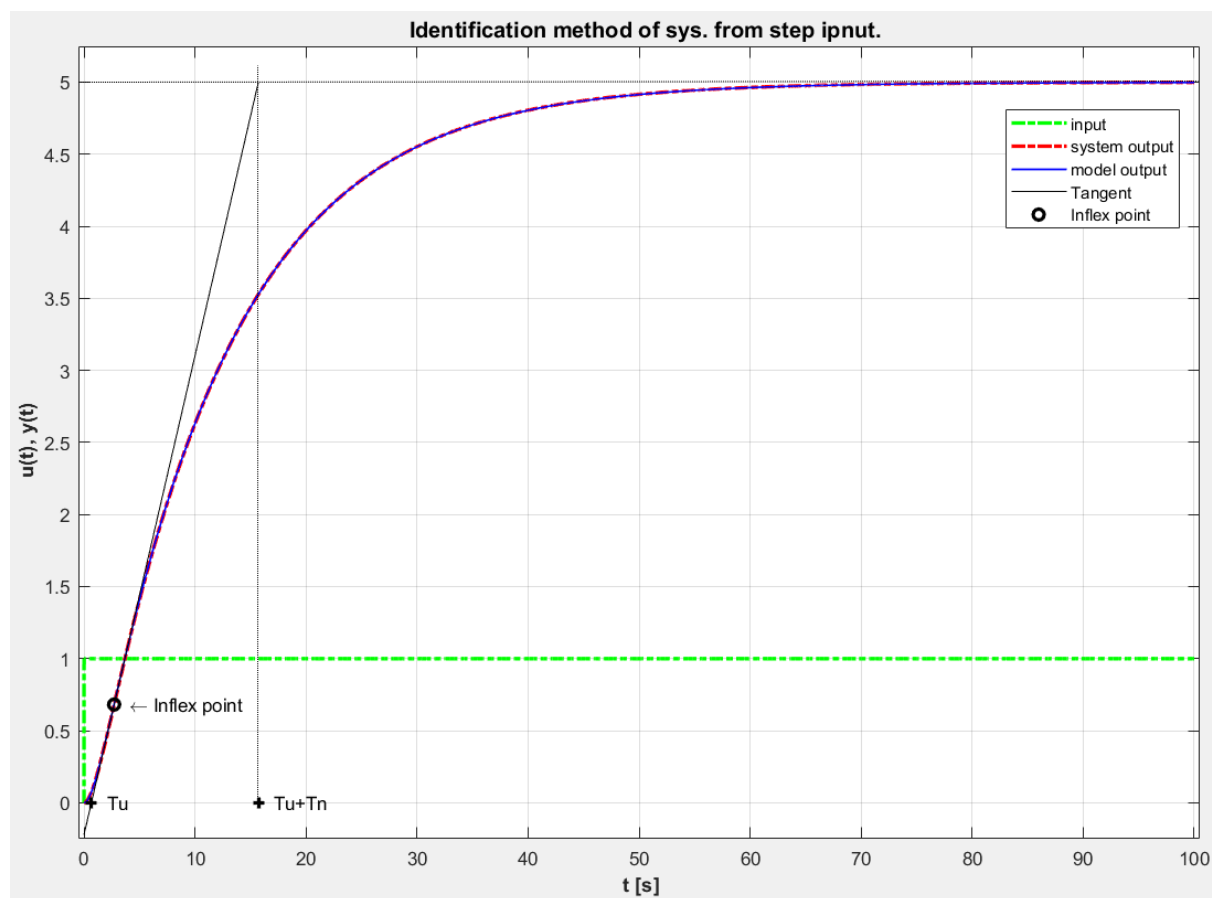
$$G_2(s) = \frac{5}{(3s+1)(2,5s+1)(2,5s+1)} \quad (116)$$

$$G_3(s) = \frac{5}{(0,5s+1)(0,1s+1)(0,5s+1)} \quad (117)$$

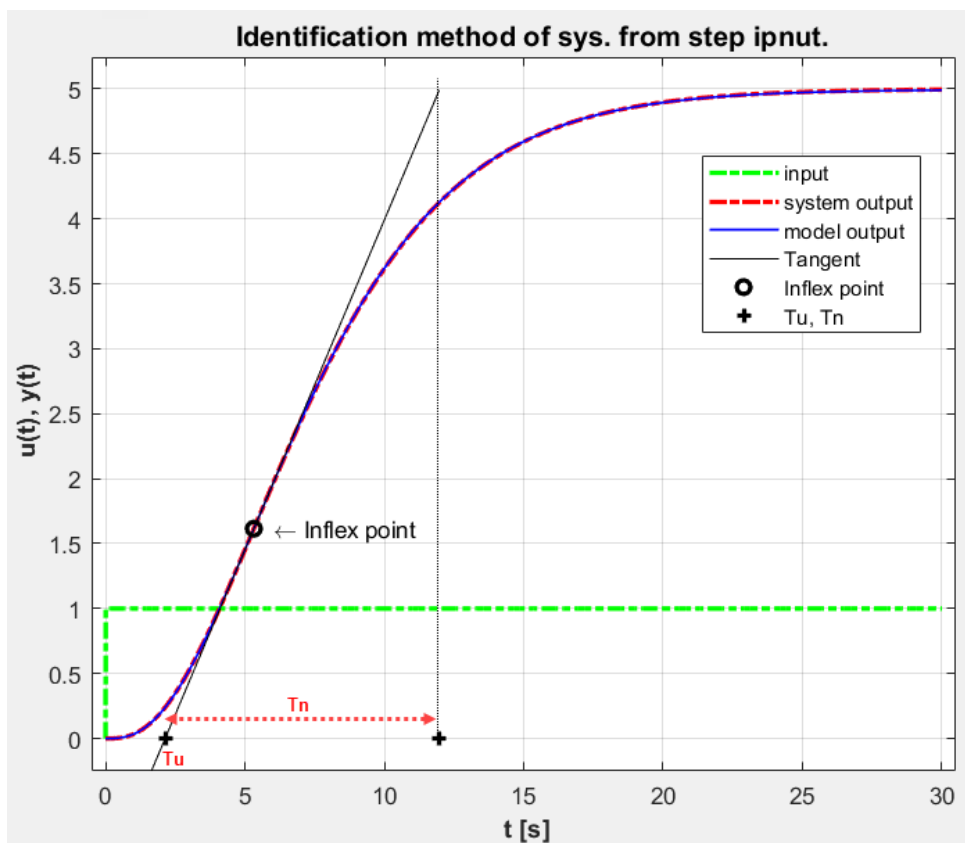
Systémy $G_1(s)$ a $G_2(s)$ byly aproximovány přesně jak je vidět z obr. 15 a obr. 16. Aproximace $G_3(s)$ s menšími časovými konstantami již vykazovala určité zkreslení (obr. 18), nepřesnost aproximace se pak projevila u systému $G_2(s)$ se šumem (obr. 17). Výsledky jednotlivých parametrů jsou shrnuty v tab. 3.

Tab. 3: Výsledky testování systémů

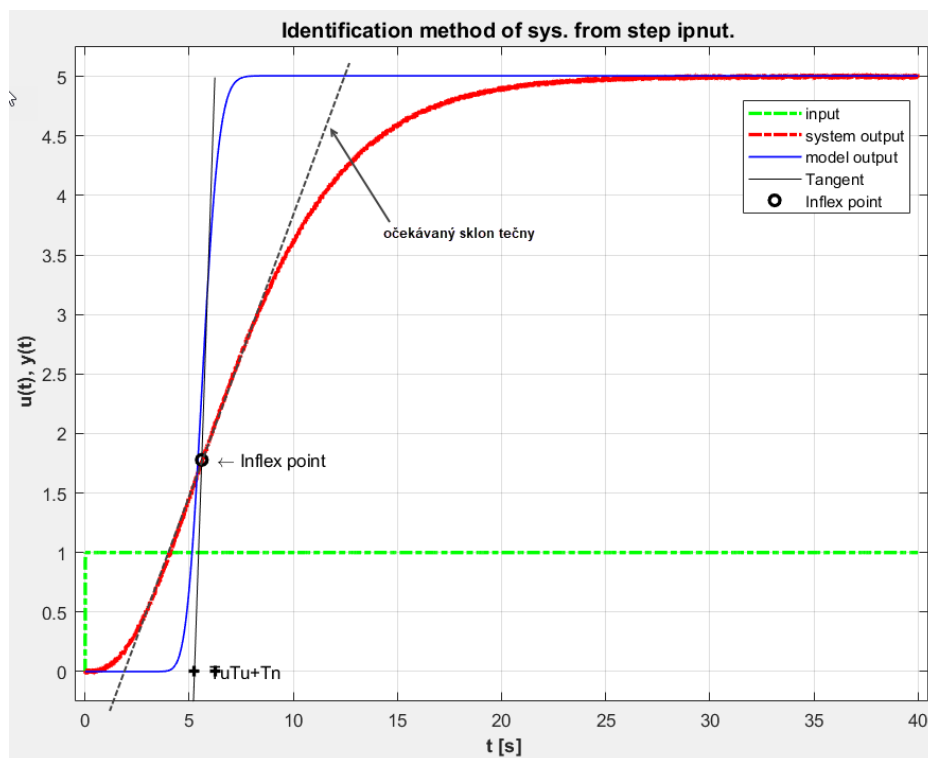
Systém	Model	Parametry
$G_1(s)$	2. řád	$k=5$, $T_1=11.99$, $T_2=1.007$, $n=3$
$G_2(s)$	3. řád	$k=5$, $T=2.67$, $n=3$
$G_2(s) + \text{šum}$	3. řád	$k=5$, $T=0.06 \div 0.8$, $n=60 \div 110$
$G_3(s)$	3. řád	$k=5$, $T=0.32$, $n=3$



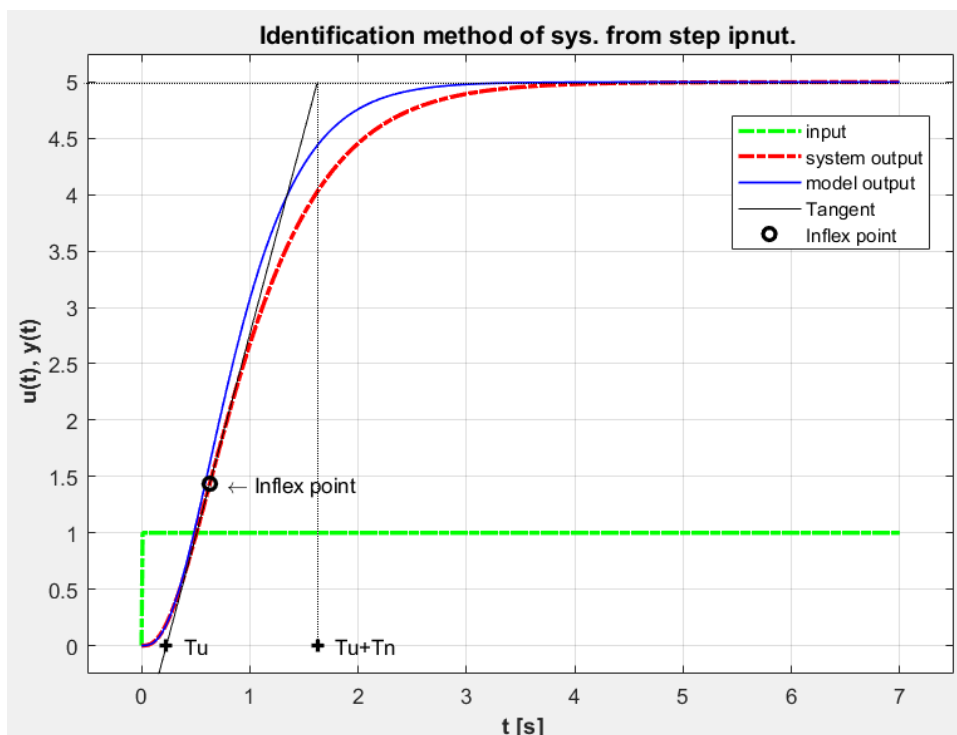
Obr. 15: Přechodová charakteristika systému $G_1(s)$ aproximovaná modelem 2.řádu s rozdílnými časovými konstantami



Obr. 16: Přejchodová charakteristika systému $G_2(s)$ aproximovaná modelem 3.řádu se stejnými časovými konstantami



Obr. 17: Očekávané selhání aproximace modelu $G_2(s)$ se šumem na výstupu ze soustavy



Obr. 18: Nepřesnost aproximace systému $G_3(s)$ modelem 3.řádu se stejnými časovými konstantami

4.2 Identifikace parametrů soustavy jednorázovou metodou nejmenších čtverců

Metoda vychází z teoretického základu uvedeného v kapitole 3.2.2. Požaduje na vstupu data o přechodném ději proběhlém na systému, který je možno budít signálem o kmitavém průběhu na rozdíl od metody uvedené v předchozí podkapitole. Dalším nutným vstupním parametrem je vzorkovací perioda odezvy identifikovaného systému, kde tato perioda musí být konstantní. Výsledný model systému je ve tvaru diskrétního přenosu podle rovnice (99), ale jen pro konkrétní danou vzorkovací periodu. Koeficienty z této rovnice jsou pak složky hledaného $\tilde{\theta}$ podle (101).

4.2.1 Implementace

Implementace této metody je v přiloženém souboru leastSquares.m. Její algoritmus vychází z (97), kde dochází k několika maticovým operacím, které jsou v prostředí Matlab velmi lehce proveditelné. Algoritmus postupuje formálně podobně jako u předchozí metody, vzorkovací perioda byla zvolena 0.01. Po vytvoření systému a simulaci přechodového děje ze vstupu se vytvoří matice Φ , jenž obsahuje 2 vstupní a 2 výstupní složky podle (101), hlavní výpočet je proveden příkazem podle (97).

Výstupem z metody jsou parametry diskrétního přenosu $\tilde{\theta} = (\tilde{b}_1, \tilde{b}_2, \tilde{a}_1, \tilde{a}_2)^T$. Pro možnost srovnání získaného modelu s identifikovaným systémem je třeba převést

získaný model na spojitý přenos (viz obr. 19). Výpis průběhu metody do Command Window Matlabu a vykreslení výsledků do grafu je součástí implementace.

```

38 % IDENTIFICATION MODEL
39 - disp('+++++++');
40 - fprintf('\nDiscrete model\n');
41 - Gz = tf([THETA(3) THETA(4)], [1 THETA(1) THETA(2)], tSample)
42 - disp('+++++++');
43 - fprintf('\nContinuous model\n');
44 - Gs = d2c(Gz) % Discrete model to Continuous
45
46 % SIMULATION OF INPUT ON COMPUTED MODEL
47 - yModel = lsim(Gs, u, t);

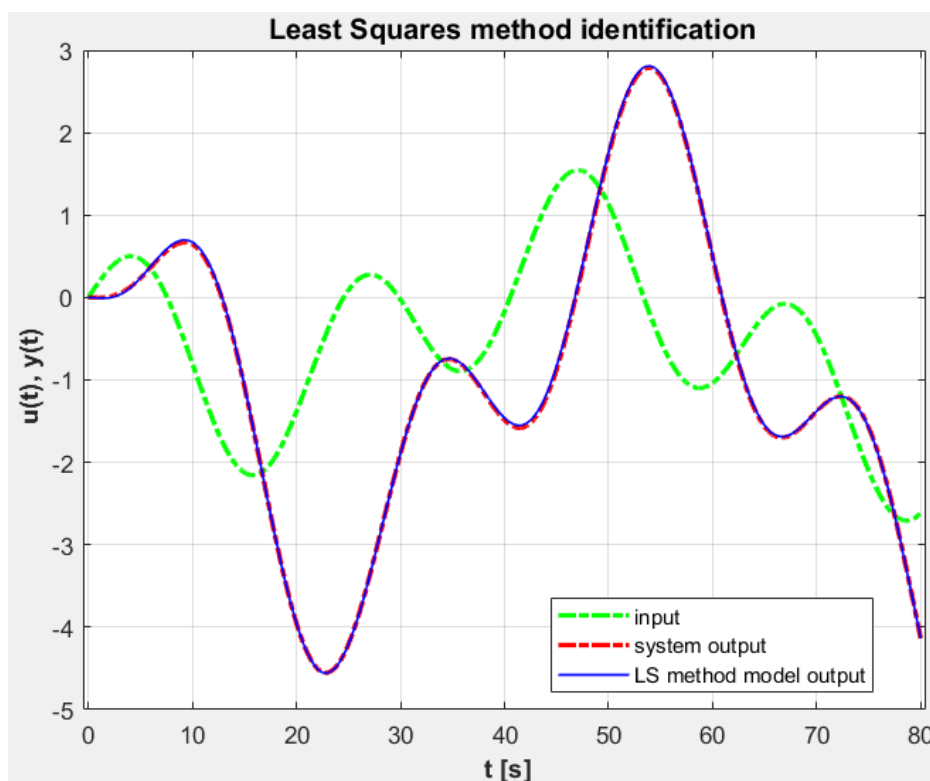
```

Obr. 19: Převod vypočítaného diskrétního modelu na spojitý

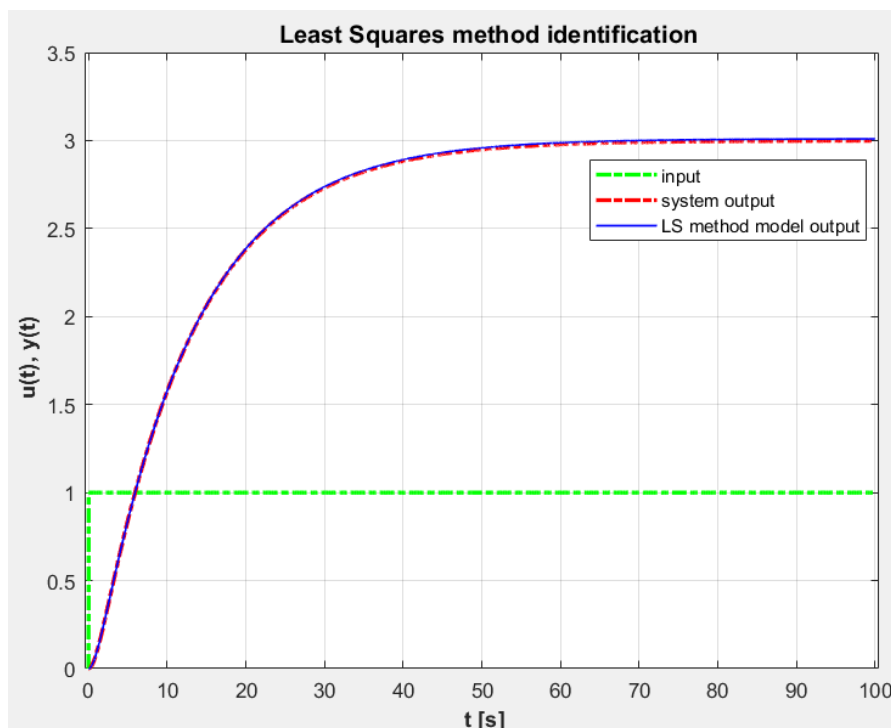
4.2.2 Výsledky pro testované systémy

Metoda byla testována na systémech (115), (116) a (117) se změnou v zesílení systémů z hodnoty 5 na hodnotu 3. Pro vstup byly použity dva signály, jednotkový skok a funkce složená z kombinací sinus $u = \sin(3\pi + 0.1t) + \sin(2\pi + 0.3t) + \sin(3\pi + 0.1t)$.

Výhodou této metody je možná analýza kmitavé odezvy. Výsledky parametrů z testování jsou v tab. 4. Pro oba vstupní signály a všechny systémy se metoda dokázala vypořádat s nelineárním průběhem křivky a tento průběh úspěšně kopírovala (obr. 20 a obr. 21).



Obr. 20: Aproximace systému $G_2(s)$ se vstupem kombinované funkce sinus

Obr. 21: Aproximace systému $G_1(s)$ na vstup jednotkového skoku

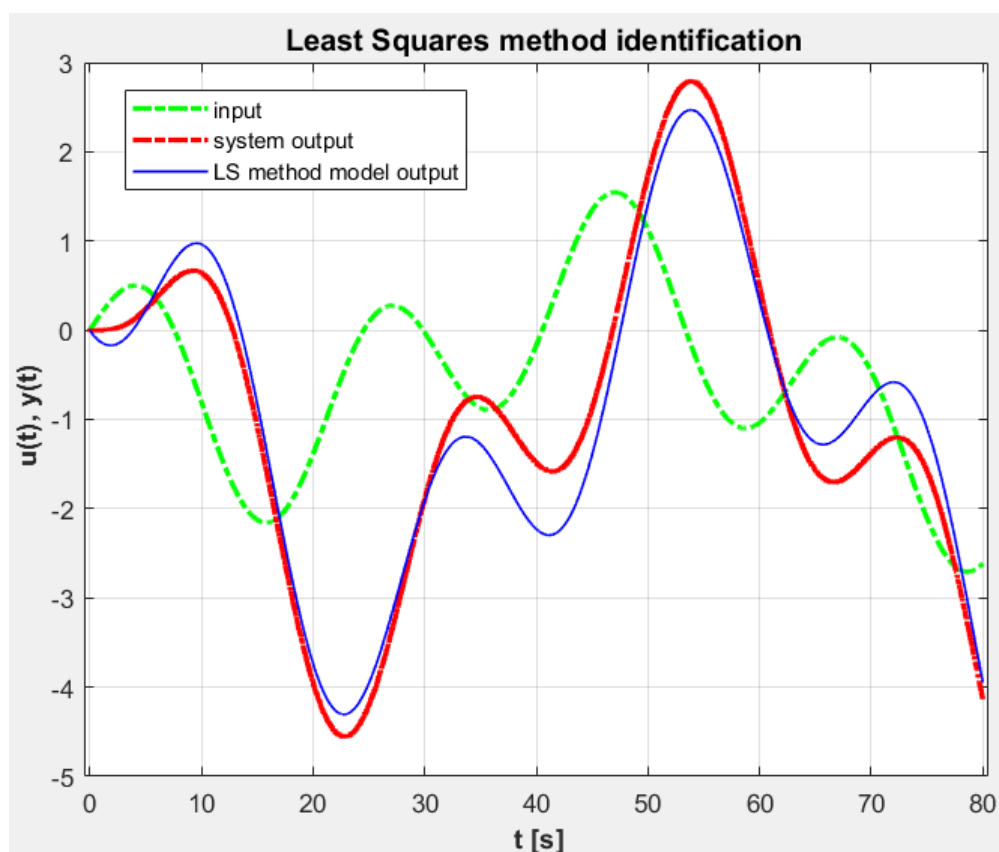
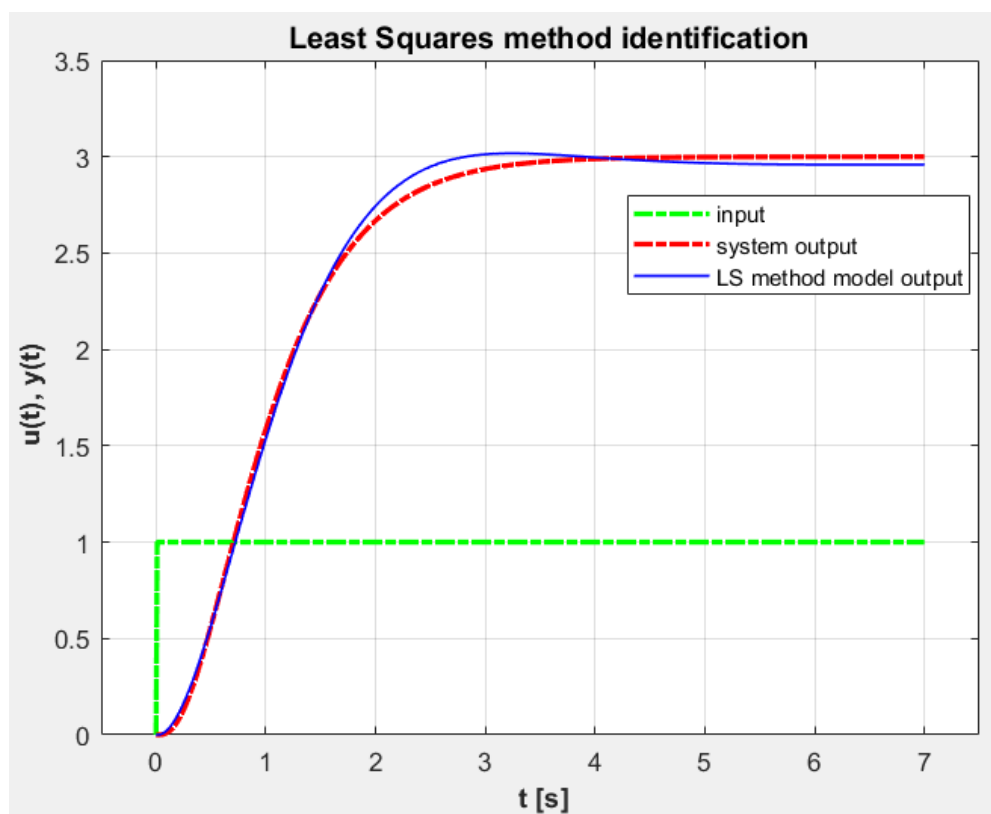
Výstup z modelu $G_2(s)$ byl také úpravou zašuměn, metoda potom kopíruje trend křivky, ale už se s ní tak nepřekrývá, jak lze vidět na obr. 22. Stejný nedostatek se jevil také u aproximace systému $G_3(s)$ s jednotkovým skokem, kde přechodový děj a překmit je značně rychlý a metoda tuto změnu ne zcela stíhá sledovat (obr. 23), což je ale způsobeno nevhodnou volbou vzorkované frekvence.

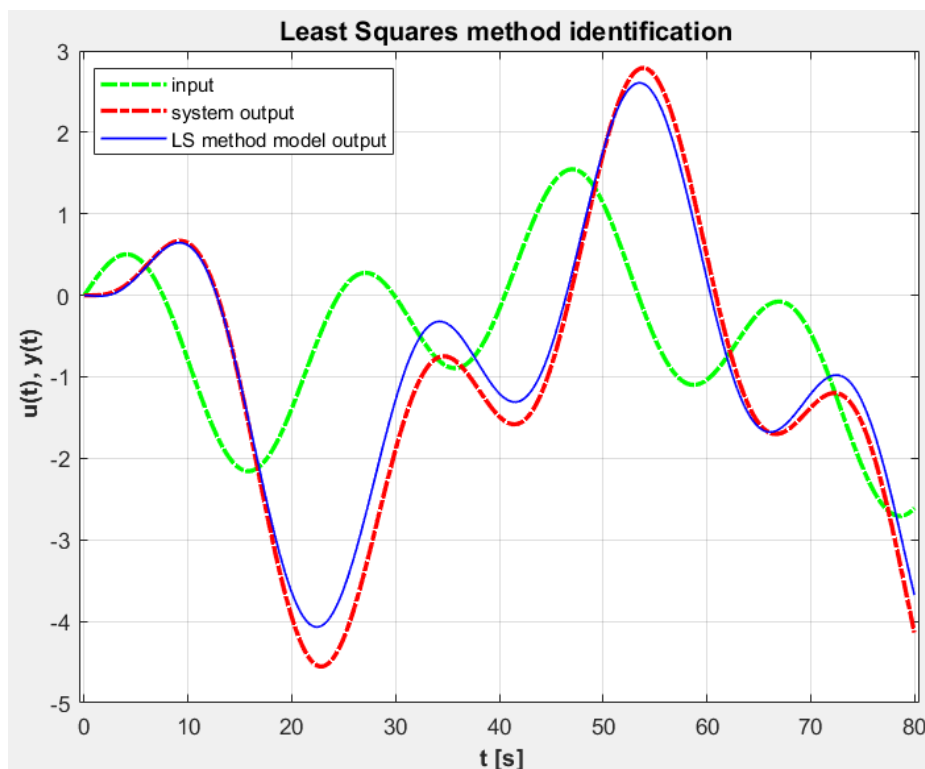
Byla rovněž testována jiná perioda vzorkování na $G_2(s)$, změněna na 0.001. Výsledek na obr. 24 v porovnání s obr.20 je důkaz jiných výsledků pro jinou vzorkovací periodu. Parametry z této modifikace mohou sloužit také pro srovnání s parametry z tab. 4:

$$\begin{aligned} b_1 &= -0.0002, & b_2 &= 0.0002 \\ a_1 &= -1.999, & a_2 &= 0.999 \end{aligned}$$

Tab. 4: Výsledky testování systémů se 2 různými vstupy při vzorkovací periodě 0.01

Systém (k=3)	Parametry	Parametry
	z jednotkového vstupu	z kombinovaného sinu
$G_1(s)$	$b_1=1.246, b_2=1.241$ $a_1=-1.989, a_2=0.989$	$b_1=2.502, b_2=-1.661$ $a_1=-1.989, a_2=0.989$
$G_2(s)$	$b_1=2.6e-8, b_2=1.1-5$ $a_1=-1.997, a_2=0.998$	$b_1=-0.001, b_2=0.001$ $a_1=-1.996, a_2=0.996$
$G_3(s)$	$b_1=1.932, b_2=0.001$ $a_1=-1.996, a_2=0.976$	$b_1=-0.005, b_2=0.006$ $a_1=-1.964, a_2=0.964$

Obr. 22: Aproximace zašuměného systému $G_2(s)$ není zcela přesnáObr. 23: Aproximace systému $G_3(s)$



Obr. 24: Aproximace systému $G_2(s)$ s periodou vzorkování 0.001

4.3 Identifikace parametrů soustavy rekurzivní metodou nejmenších čtverců rozšířenou o techniku zapomínání

Metoda se označuje také jako online, zpracovává v každém kroku nový vzorek dat. Implementace vychází z kapitoly Rekurzivní metoda nejmenších čtverců 3.2.3. Koeficienty hledaného $\tilde{\theta}$ tedy v každém kroku aktualizuje, lze sledovat jejich vývoj v čase. Je rozšířená o techniku exponenciálního zapomínání v podobě koeficientu λ , která přiřazuje novým vzorkům vyšší váhu.

4.3.1 Implementace

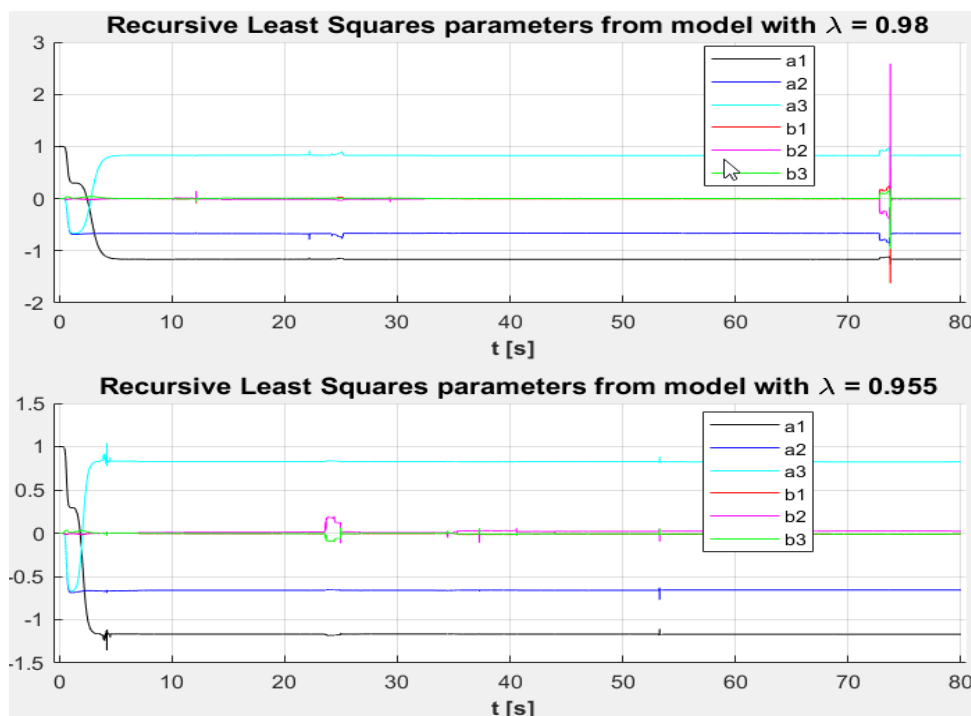
Také tato metoda je součástí přílohy v souborech `recLeastSquaresLambda.m` a `rls.m`. Její implementace vychází ze vzorců (109, 110, 111), kde toto jádro výpočtu metody pro každý nový přicházející vzorek bylo osamostatněno v podobě funkce v souboru `rls.m`, která se v hlavním souboru volá v iteraci s parametry vektorů se vstupními a výstupními daty, které se v každém kroku rozrůstají o další vzorek.

Pro implementaci byla zvolena vzorkovací perioda 0.01. Po vytvoření systému a simulaci přechodového děje ze vstupu se nastaví počáteční podmínky podle (112, 113, 114) a také vektory vstupu a výstupu, do kterých se v následném cyklu `for` iteračně přidávají složky vstupního a výstupního signálu ze soustavy. Při každém volání funkce `rls()` se v této funkci vytvoří vždy nová matice Φ , jenž obsahuje 3 vstupní a 3 výstupní složky podle (101). Výstupem z metody jsou parametry diskrétního přenosu

$\tilde{\theta} = (\tilde{b}_1, \tilde{b}_2, \tilde{b}_3, \tilde{a}_1, \tilde{a}_2, \tilde{a}_3)^T$. Pro možnost srovnání získaného modelu s identifikovaným systémem je opět implementován výpis průběhu metody do Command Window Matlabu a vykreslení výsledků do grafu.

4.3.2 Výsledky pro testované systémy

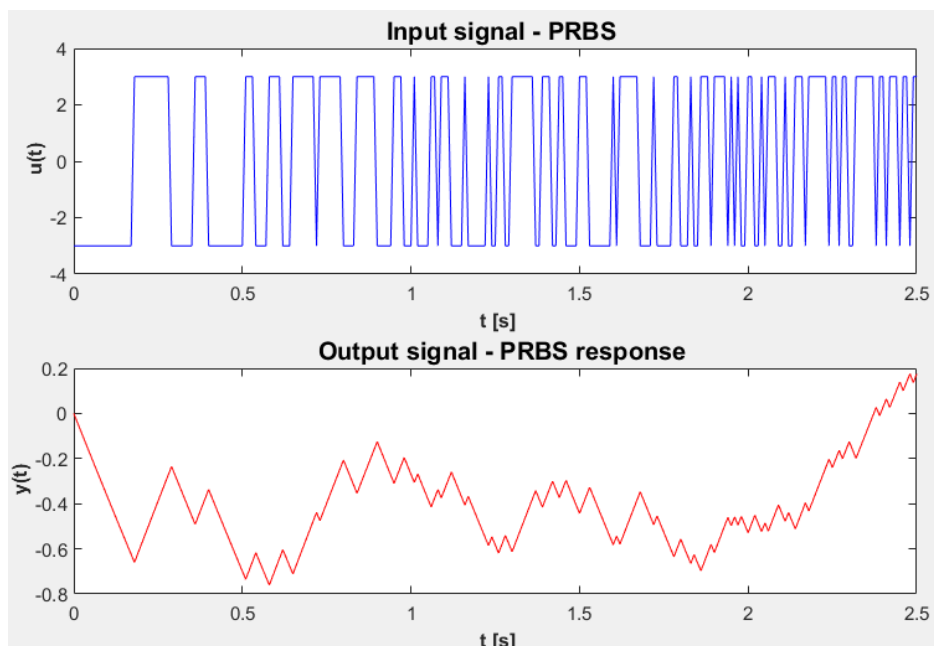
Metoda byla testována za použití stejných systémů a vstupních signálů tak jak je uvedeno v kapitole 4.2.2. Metoda vykazovala dobré výsledky při identifikaci, odezva modelu a originálního systému (na proměnlivý vstupní signál) vykazovala vysokou míru shody. Zároveň bylo možné zaznamenat vývoj hledání parametrů výsledného diskrétního modelu (obr. 25.).



Obr. 25: Vývoj parametrů modelů (různá λ) při aproximaci systému $G_2(s)$

4.4 Identifikace systému ve frekvenční oblasti spektrální analýzou

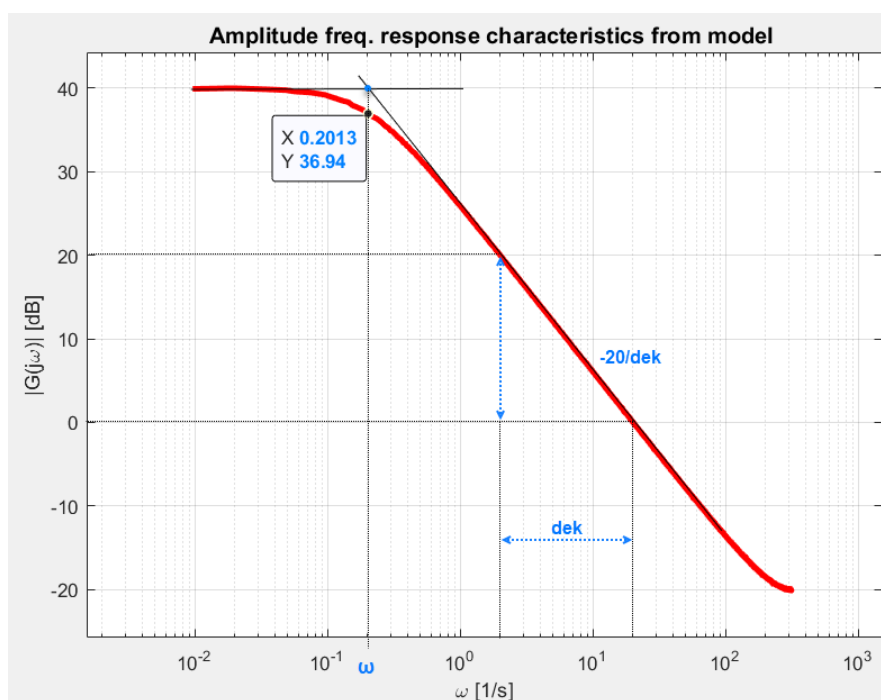
Tato metoda analýzou spektra určí jednotlivé zastoupení daných frekvencí ve výstupním signálu, výstupem z metody je frekvenční charakteristika identifikovaného systému. Implementace vychází z algoritmu popsaného v kapitolách 3.1.2 a 3.1.4 a je obsažena v příloženém souboru spectralAnalysis.m. Perioda vzorkování je nastavena na 0.01 a na zvoleném systému se nasimuluje vstupní signál PRBS (Pseudorandom binary sequence) pomocí funkce `idinput()` a získá se odezva systému (viz obr. 26). Pomocí funkce `cpsd()` se provede výpočet odhadů spektrálních hustot, které jsou pak použity při výpočtu frekvenčního přenosu modelu jako podle (74). Výsledná charakteristika je pak vynesena do grafu v logaritmických souřadnicích pomocí funkce `semilogx()`, z ní se odečte podle asymptot lomová frekvence a podle amplitudy signálu taky zesílení.

Obr. 26: Vstupní PRBS signál a výstup ze soustavy $G_4(s)$

Testována byla identifikace následující soustavy:

$$G_4(s) = \frac{100}{(5s+1)} \quad (117)$$

Výsledná amplitudová frekvenční charakteristika je na obr. 27. Výpočte se lomová frekvence $\omega_l = 0.2013 \text{ Hz}$, časová konstanta $T = \frac{1}{\omega_l} = 4,97 \text{ s}$ a zesílení $K = 40$ ze vztahu (31) pro vodorovnou asymptotu. Výsledek se shoduje s parametry přenosu $G_4(s)$.



Obr. 27: Identifikace systému z amplitudové frekvenční charakteristiky

5 SAMOSTATNÁ APLIKACE PRO IDENTIFIKACI SOUSTAVY

Pro vytvoření samostatného programu pro identifikaci dynamických systémů byl na 64-bitovém operačním systému Windows 10 Home použit open-source programovací jazyk Python verze 3.6 od společnosti ActiveState Software Inc. Tato kanadská softwarová společnost vyvíjí a podporuje vývojové nástroje pro dynamické jazyky jako jsou Perl, PHP, Python, Ruby. Výběr jazyka Python byl zvolen z důvodu, že jakožto open-source nástroj poskytuje možnost využívat velké množství knihoven, které jsou tvořeny komunitou vývojářů prakticky z celého světa.

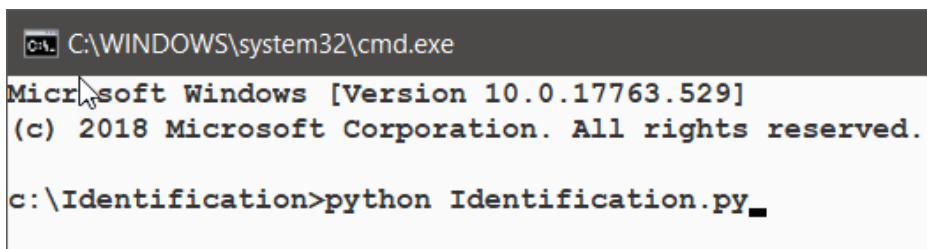
Před instalací je nutné si vytvořit účet na webu platform.activestate.com a přihlásit se. Po vybrání požadované verze jazyka Python si lze stáhnout instalační soubor z webové stránky activestate.com/products/activepython/downloads/. V záložce „Builds“ se volí používaný operační systém, v tomto případě Windows, pro něhož má instalační soubor koncovku *.msi. Spolu s instalací je k dispozici obsáhlá dokumentace jazyka včetně zabudovaných (built-in) knihoven v místě instalace /Python36/Doc/index.html. Python od verze 3 obsahuje už package-management systém **pip**. Ten slouží k pohodlné instalaci open-source knihoven z webu pypi.org pomocí příkazů vyvolaných v Command Prompt systému Windows. **Pip** se aktualizuje v Command Prompt po vyvolání příkazu:

```
python -m pip install --upgrade pip
```

Vytvořená aplikace je napsána za použití open-source knihoven, které je nutno doinstalovat, např. pomocí **pip** následujícími příkazy:

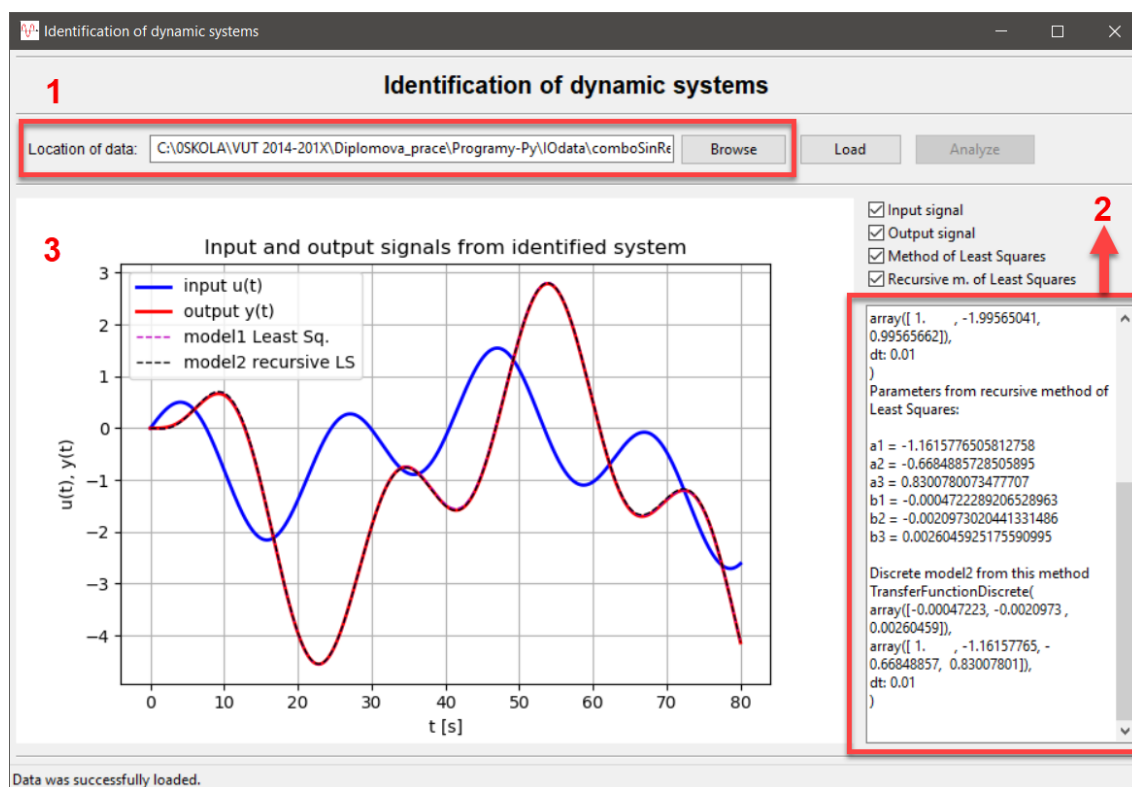
```
python -m pip install scipy
python -m pip install matplotlib
python -m pip install numpy
python -m pip install wx
```

Rozšíření v podobě balíčku numpy bylo zvoleno pro možnost využití funkcionality spojené s maticovým počtem, který je pro výpočet zvolených metod vhodný. Tvorba GUI (Graphical User Interface) bylo realizováno grafickou nádstavbou pro Python, knihovnou wx. Vykreslování křivek přímo v GUI lze naprogramovat při kombinaci knihovny matplotlib s wx. Aplikace má název „Identification“ a obsahuje v el. příloze ve stejnojmenném adresáři soubory icon.png a Identification.py.



Obr. 28: Spuštění aplikace

Program se spouští příkazem podle obr. 28 jímž se spustí hlavní okno aplikace (obr.29). Tato aplikace pro identifikaci systémů implementuje metodu nejmenších čtverců v základní formě a také rekurzivní metodu nejmenších čtverců. Pro identifikaci systému je v aplikaci nutné načíst *.csv soubor tvořen 2 sloupci hodnot o stejném počtu. První sloupec představuje vzorky hodnot vstupního signálu, kterým byl systém buzen, druhý sloupec obsahuje reakci systému v podobě hodnot výstupního signálu.

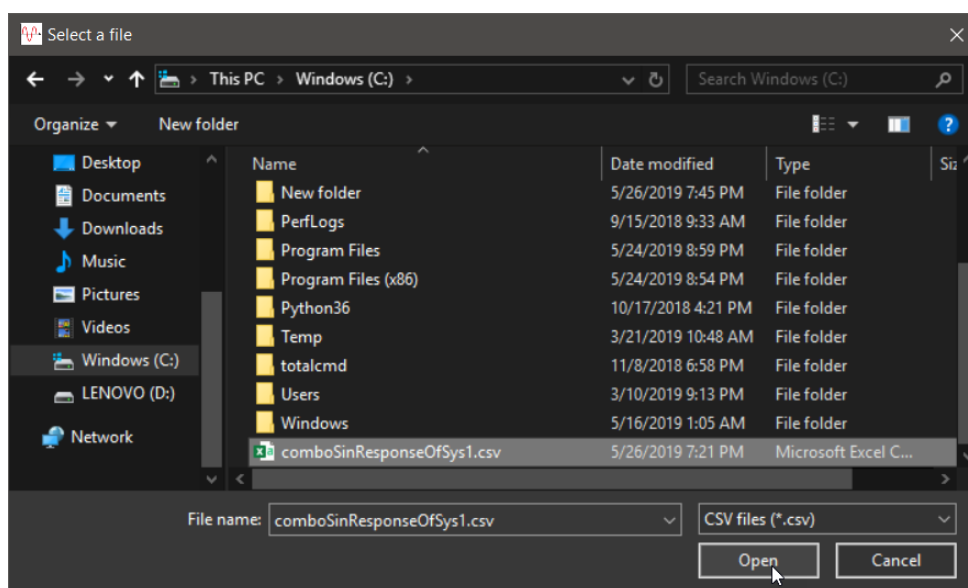


Obr. 29: Hlavní okno aplikace Identification

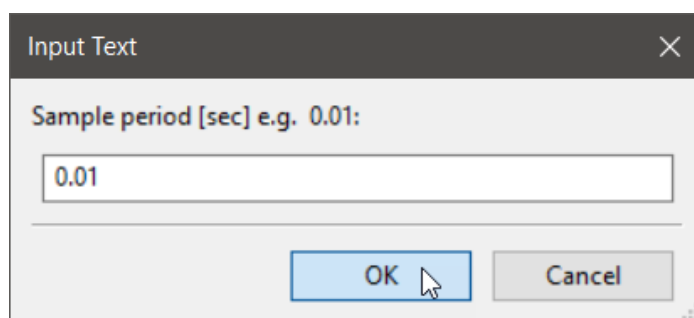
Tento přechodný děj je možné simulovat v software Matlab příkazem:

```
dlmwrite('OUTPUT.csv', [u', y], 'delimiter', ',', 'precision', 15)
```

Pro zadání cesty k tomuto souboru slouží textové políčko vedle popisku „Location of data:“ v oblasti „1“ jak lze vidět na obr. 29. Lze využít i blízké tlačítko „Browse“ umožňující dialogovým oknem daný soubor v systému vyhledat (obr. 30). Stisknutím tlačítka „Load“ se zobrazí dialogové okno pro získání informace o vzorkovací periodě naměřených dat (obr. 31). Po vyplnění údaje se tato data načtou do paměti programu a křivky vstupu a výstupu se vykreslí v zobrazovací části GUI (označenou „3“ na obr. 29). Tlačítkem „Analyze“ se provedou obě implementované metody a výsledek ve formě parametrů diskrétního přenosu se vypíše do textového pole aplikace (část „2“ v obr. 29). a taky se vykreslí do zobrazovací části v podobě nových křivek. Pomocí checkboxů lze tyto křivky skrýt/odkrýt.



Obr. 30: Vyhledání souboru k identifikaci

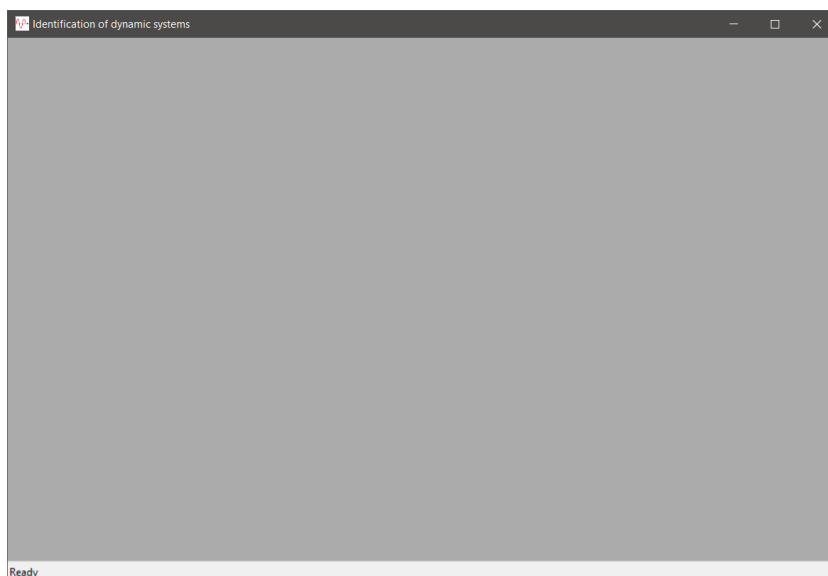


Obr. 31: Dialogové okno k zadání vzorkovací periody

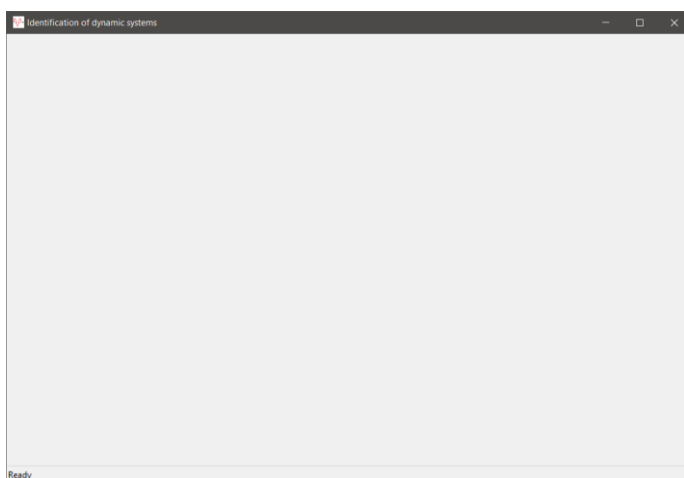
Aplikace byla testována na stejných systémech a při použití stejných budících vstupů, jak bylo použito k testování metody v kapitole 4.2.2. Výsledky identifikovaných systémů v podobě parametrů diskrétních přenosů jsou ve velké míře shody s výsledky parametrů získaných při testování metod v Matlabu.

5.1 Tvorba GUI ve wx knihovně

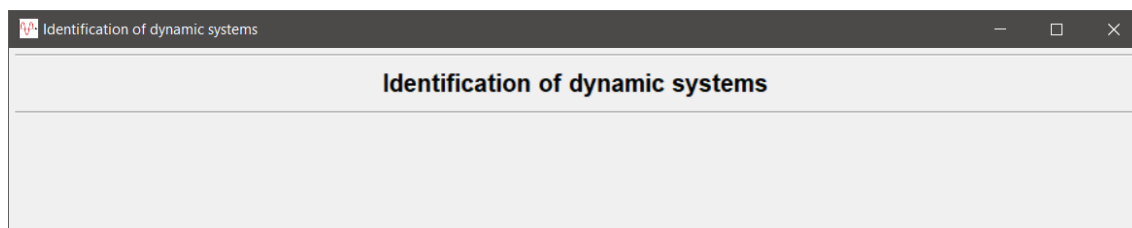
GUI v podobě hlavního okna aplikace je vytvořeno knihovnou wx. Podkladem pro zobrazované prvky je v téměř celém okně vertikální `BoxSizer`, ve spodní části statusbar (obr. 32). `BoxSizer` je základní abstraktní třída pro vkládání prvků, které uživatel uvidí. Tato třída určuje rozměry prvků, které jsou v ní přidány.

Obr. 32: Podklad hlavního okna v podobě `backgroundVBox`

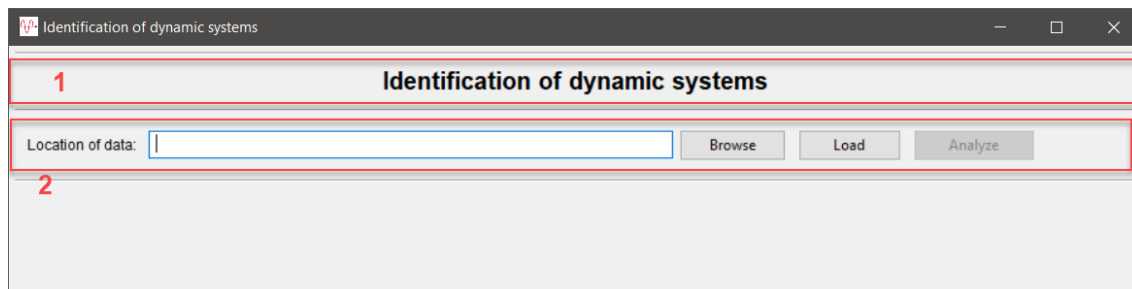
Do objektu této třídy v implementaci označené jako `backgroundVBox` je přidána další objekt `panel` třídy `Panel`, která ji zcela vyplňuje. Tvoří už viditelný podklad, který je možno barevně modifikovat (obr. 33).

Obr. 33: Přidán objekt `panel`

Pro dosažení rozmístění prvků jak to lze vidět v hlavním okně aplikace je do podkladového objektu `panel` přidán další objekt `vBox` třídy `BoxSizer`. Do něho jsou vloženy tři horizontální objekty `hBox1`, `hBox2` a `hBox3` stejné třídy. Objekt `hBox1` už obsahuje objekt `StaticText` s nápisem (obr. 34). `hBox2` je od `hBox1` oddělen objektem třídy `StaticLine` vloženým v objektu `vBox` mezi ně (obr. 35). Součástí `hBox2` je objekt třídy `FilePickerCtrl` sloužící ke zvolení cesty k načítaným datům zadáním to textového pole, nebo stlačením tlačítka „Browse“. Vpravo od tohoto objektu jsou dva objekty třídy `Button`, čili dvě tlačítka „Load“ a „Analyze“ sloužící k obsluze programu k načtení a analýze dat.



Obr. 34: Přidán objekt hBox1



Obr. 35: Přidán objekt hBox2 (2) pod hBox1 (1)

Posledním objektem vloženým do vBox je hBox3. Ten v sobě obsahuje objekt vytvořené třídy `MatlabPlot` (oblast „2“ na obr. 29) a další objekt `vBox3`. Teprve v něm postupně se shora dolů nachází čtyři objekty třídy `CheckBox` a textové pole pro výpis parametrů identifikovaných modelů třídy `TextCtrl` (oblast 3 na obr. 29). Třída `MatlabPlot` umožňující vykreslení grafů má podklad v objektech `matplotlib.figure.Figure` a `backends.backend_wxagg.FigureCanvasWxAgg` importované knihovny `matplotlib`. Při změně rozměrů okna aplikace je třeba zajistit změnu rozměrů vykreslovaného grafu, byla vytvořena funkce `SetToResize`, která se volá v okamžiku vzniku události `wx.EVT_IDL`.

```

38     def SetToResize(self):
39         pixels = tuple(self.GetSize())
40         self.SetSize(pixels)
41         self.canvas.SetSize(pixels)
42         self.figure.set_size_inches([float(x) / self.figure.get_dpi() for x in pixels])

```

Obr. 36: Funkce `SetToResize` implementující přepočítávání rozměrů podkladových objektů pro vykreslování grafu

6 ZÁVĚR

Autor se v rámci rešeršní části práce seznámil s metodami identifikace dynamických systémů, možnostmi jejich aplikací a problematikou programování jejich algoritmů. Prohloubil si také znalosti programování těchto algoritmů v prostředí výpočetního standardu Matlab. V tomto programu byly vytvořeny implementace čtyř zvolených metod. Na vybraných systémech byly simulovány přechodové děje po buzení různými signály. Odezvy systémů byly použity pro testování jednotlivých metod identifikace.

Byla implementována metoda pro identifikaci nekaitavých systémů ze skokového vstupu. Při identifikaci zvolených systémů vykazovala metoda dobré výsledky za podmínky správného použití této metody, tedy předpokladu absence působení poruchové veličiny. Další implementovanou metodou byla jednorázová metoda nejmenších čtverců. Vykazovala vysokou míru shody, přičemž tato metoda umožňuje aproximovat mimo jiné i kaitavou odezvu systémů. Ještě jedna parametrická metoda byla naprogramována v prostředí Matlab a sice rekursivní metoda nejmenších čtverců včetně techniky exponenciálního zapomínání, která se používá pro on-line identifikaci systémů. Vyznačuje se značnou rychlostí a bývá používána na mikrokontrolerech pro svou paměťovou nenáročnost. Další naprogramovanou metodou byla spektrální analýza umožňující identifikaci systémů ve frekvenční oblasti. Není citlivá na šum a má široké uplatnění pro analýzu spektra téměř libovolného signálu. V rámci této práce byla vytvořena aplikace pro identifikaci systémů pomocí jednorázové metody nejmenších čtverců a pomocí rekursivní obdoby této metody. Aplikace byla naprogramována v open-source nástrojích, bylo pro ni vytvořeno grafické uživatelské rozhraní.

Práce je úvodem do problematiky metod identifikace dynamických systémů. Implementace výše popsaných metod jsou součástí přílohy. Tyto skripty můžou najít uplatnění při výuce z oblasti modelování a identifikace systémů pro názornou ukázkou. Vytvořená aplikace není vázána na placený produkt, mimo výuku ji lze použít např. v praxi při návrhu regulátoru pro řízení určitého procesu. Součástí poslední kapitoly je i stručný návod na tvorbu jednoduchých GUI pomocí knihovny wxpython, která umožňuje pracovat s abstraktními třídami (Sizers), které určují rozmístění zobrazitelných prvků v nich obsažených.

7 SEZNAM POUŽITÉ LITERATURY

- [1] ŠVARC, Ivan, ŠEDA, Miloš, a VÍTEČKOVÁ, Miluše. *Automatické řízení*. Brno: Akademické nakladatelství CERM, 2007. ISBN 978-80-214-3491-2.
- [2] BLAHA, Petr a Petr VAVŘÍN. *Řízení a regulace I: Základy regulace lineárních systémů - spojité a diskrétní*. Brno: VUT FEKT, 2005 [cit. 2019-05-24].
- [3] ŠOLC, František, VÁCLAVEK, Pavel, VAVŘÍN, Petr. *Řízení a regulace II: Analýza a řízení nelineárních systémů*. Brno: VUT FEKT, 2011 [cit. 2019-05-24].
- [4] PIVOŇKA, Petr. *Optimalizace regulátorů*. Brno: VUT FEKT, 2005 [cit. 2019-05-24].
- [5] ŠTĚCHA, Jan, HAVLENA, Vladimír. *Teorie dynamických systémů*. Praha: ČVUT, 2005 [cit. 2019-05-24].
- [6] PIVOŇKA, Petr. *Číslicová řídicí technika*. Brno: VUT FEKT, 2012 [cit. 2019-05-24].
- [7] ISERMANN, Rolf a Marco MÜNCHHOF. *Identification of dynamic systems: an introduction with applications*. Berlin: Springer, 2011. ISBN 978-3-540-78878-2. [cit. 2019-05-24].
- [8] DUNÍK, Jindřich. *Identifikace systémů a filtrace*. [online]. Plzeň: Západočeská univerzita v Plzni, 2018 [cit. 2019-05-24]. ISBN 978-80-261-3775-0. Dostupné z: <https://doi.org/10.24132/ZCU.2018.07750>
- [9] BLAHA, Petr. *Neparametrické metody identifikace*. [online]. 2014. Brno: VUT FEKT, s. 68 [cit. 2019-05-24]. Dostupné z: <https://sites.google.com/site/modelovaniaidentifikace/>

8 SEZNAM POUŽITÝCH ZKRATEK

apod. a podobně

např. na příklad

GUI (Graphical User Interface) – grafické uživatelské rozhraní

9 SEZNAM PŘÍLOH

Příloha 1: Soubor stepInput.m implementující metodu z kapitoly 4.1

Příloha 2: Soubor leastSquares.m implementující metodu z kapitoly 4.24.1

Příloha 3: Soubory rls.m a recLeastSquaresLambda.m implementující metodu z kapitoly 4.3

Příloha 4: Soubor spectralAnalysis.m implementující metodu z kapitoly 4.4

Příloha 5: Obsah elektronické přílohy 162946_ZimekT.zip

Příloha 1: Soubor stepInput.m implementující metodu z kapitoly 4.1

stepInput.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identification from step input %
% date: 24/5/2019 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all, clear all, clc;

disp('++++++++++++++++++++++++++++++++++++++++');
disp('| Identification from step input |');
disp('++++++++++++++++++++++++++++++++++++++++');

% DATA PREPARATION
stepValue = 1;
tRange = 40;
tSample = 0.01;
t = [0:tSample:tRange];
u = ones(1, length(t)) * stepValue; u(1) = 0;

% DEFINITION OF SYSTEM
fprintf('\nSystem for identification:\n\n');
K = 5;
s = tf('s');
sys = K / ((3*s+1) * (2.5*s+1) * (2.5*s+1))

% SIMULATING OF APPLICATION OF STEP INTO A SYSTEM
[y, ~] = lsim(sys, u, t);
a = -0.5; b = 0.5;

% GETTING THE INFLEX POINT
der1 = diff(y) / tSample; % 1st derivation
der1 = [0 ; der1]; % to have same length as y
 [~, indexDer1] = max(abs(der1));
slopeOfTangent = der1(indexDer1);
yi = y(indexDer1); ti = t(indexDer1);

% GETTING Ks, Tu, Tn, Tau
q = yi - slopeOfTangent*ti; % Tanget: y(t) = kt + q
Ks = y(end) / stepValue;
Tu = -q / slopeOfTangent;
Tn = ((y(end) - q) / slopeOfTangent) - Tu;
Tau = Tu/Tn;
tTanget = [0:tSample:(Tn+Tu)];
```

```

% IDENTIFICATION
fprintf('\nIdentified system:\n');
if (Tau < 0.1) % model of 2nd order different T constants
    y072 = 0.72*y(end);
    t1Index = find(abs(y) > abs(y072), 1);
    sum = t(t1Index) / 1.2564;
    t2 = (0.3574*sum);
    y2Index = find(t > t2, 1);
    Ynorm = y / Ks; %normalization to get y2 for table
    valueToInterpolate = abs(Ynorm(y2Index));

    y2 = [0.3 0.28 0.26 0.24 0.22 0.2 0.18 0.1611];
    Tau2 = [0.0 0.0435 0.0837 0.128 0.1838 0.2639 0.4031 1.0];
    interpolatedOutput = interp1(y2, Tau2, ...
        valueToInterpolate, 'linear', 'extrap');

    % TIME CONSTANTS sum = T1+T2, interpolatedOutput = T2/T1
    % T1 = sum/(1+interpolatedOutput), T2 = sum - T1
    T1 = sum / (1 + interpolatedOutput);
    T2 = sum - T1;
    G = tf(abs(Ks), [T1*T2, T1+T2, 1])

    fprintf('\n');
    disp('Identified model is of 2nd order with parameters:');
    fprintf('\n');
    disp(['T1 = ' num2str(T1)]); disp(['T2 = ' num2str(T2)]);
    disp(['Ks = ' num2str(abs(Ks))]);
else % model of n-th order, same Time constants
    Tau3 = [0.104 0.218 0.319 0.410 0.493 0.570 ...
        0.642 0.709 0.773];
    order = [2 3 4 5 6 7 8 9 10];
    n = round(interp1(Tau3, order, Tau, 'linear', 'extrap'));
    Tm = ti/(n - 1);
    s = tf('s');
    G = abs(Ks)/((Tm*s+1)^n)

    fprintf('\n');
    disp(['Identified model is of order ' num2str(n) ':']);
    fprintf('\n');
    disp(['T = ' num2str(Tm)]); disp(['n = ' num2str(n)]);
    disp(['Ks = ' num2str(abs(Ks))]);
end

% SIMULATION OF INPUT TO COMPUTED MODEL
[yModel, ~] = lsim(G, u, t);

% PLOTTING
plot(t, u, 'g-.', 'LineWidth', 2);
hold on;
plot(t, y, 'r-.', 'LineWidth', 2);
plot(t, yModel, 'b', 'LineWidth', 1);
plot(tTanget, slopeOfTangent*tTanget + q, 'k');
plot(ti, yi, 'ko', 'LineWidth', 2)

```

```

plot(Tu, 0, 'k+', Tn+Tu, 0, 'k+', 'LineWidth', 2);
text(ti, yi, ' \leftarrow Inflex point');
text(Tu, 0, ' Tu'); text(Tn+Tu, 0, ' Tu+Tn');
hold off; grid on;
xlim([-0.5 tRange+0.5]);
if (y(end) > 0)
    ylim([-y(end)*0.05 y(end)*1.05]);
else
    ylim([y(end)*1.05 -y(end)*0.05]);
end
legend('input', 'system output', 'model output', ...
       'Tangent', 'Inflex point');
title('\bf \fontsize{12} Identification method of sys. from
step input.');
```

t [s] **u(t), y(t)**

```

fprintf('\n');
disp('+++++++');
disp('|                END                |');
disp('+++++++');
```


Příloha 2: Soubor leastSquares.m implementující metodu z kapitoly 4.2

leastSquares.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identification with %
% Least Squares method %
% %
% date: 24/5/2019 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all, clear all, clc;

disp('+++++++');
disp('| Method of Least Squares |');
disp('+++++++');

% DATA PREPARATION
tSample = 0.001;
tRange = 80;
t = [0:tSample:tRange];
u = sin(3*pi+0.1*t) + sin(2*pi+0.3*t) + sin(3*pi+0.01*t);

% DEFINITION OF SYSTEM
disp('+++++++');
fprintf('\nSystem for identification\n');
K = 3;
s = tf('s');
sys = K/((3*s+1)*(2.5*s+1)*(2.5*s+1))

% SIMULATING OF APPLICATION OF INPUT SIGNAL INTO A SYSTEM
y = lsim(sys, u, t);

% PREPARING PHI VALUES
Y = y(3:end);
PHI = [-y(2:end-1) -y(1:end-2) u(2:end-1)' u(1:end-2)'];

% COMPUTING PARAMETERS
THETA = (PHI'*PHI)^-1 * PHI'*Y;

% IDENTIFICATION MODEL
disp('+++++++');
fprintf('\nDiscrete model\n');
Gz = tf([THETA(3) THETA(4)], [1 THETA(1) THETA(2)], tSample)
disp('+++++++');
fprintf('\nContinuous model\n');
Gs = d2c(Gz) % Discrete model to Continuous

% SIMULATION OF INPUT ON COMPUTED MODEL
yModel = lsim(Gs, u, t);
```

```
% PLOTTING
plot(t, u, 'g-.', 'LineWidth', 2);
hold on;
plot(t, y, 'r-.', 'LineWidth', 2);
plot(t, yModel, 'b', 'LineWidth', 1);
hold off; grid on;
xlim([-0.5 tRange+0.5]);
legend('input', 'system output', 'LS method model output');
title('\bf \fontsize{12} Least Squares method
identification');
xlabel('\bf t [s]'); ylabel('\bf u(t), y(t)');

fprintf('\n');
disp('+++++++');
disp('|               END               |');
disp('+++++++');
```

Příloha 3: Soubory rls.m a recLeastSquaresLambda.m implementující metodu z kapitoly 4.3

recLeastSquaresLambda.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identification %
% with Recursive Least %
% Squares method %
% %
% date: 24/5/2019 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all, clear all, clc;

disp('+++++++');
disp('| Recursive Method of Least Squares |');
disp('+++++++');

% DATA PREPARATION
tSample = 0.01;
tRange = 80 ;
t = [0:tSample:tRange];
u = sin(3*pi+0.1*t) + sin(2*pi+0.3*t) + sin(3*pi+0.01*t);

% DEFINITION OF SYSTEM
disp('+++++++');
fprintf('\nSystem for identification\n');
K = 3; s = tf('s');
sys = K/((3*s+1)*(2.5*s+1)*(2.5*s+1))

% SIMULATING OF APPLICATION OF INPUT SIGNAL INTO A SYSTEM
y = lsim(sys, u, t);

% PREPARING THETA, COVARIANT MATRIX, LAMBDA, I/O
PCovKcoef = 1e6; % 1e4 - 1e6
PCovariant = PCovKcoef*eye(6);
THETA = [1; 0; 0; 0; 0; 0];
THETA_BUFFER = [];
lambda = 0.98;
U = [0 0 0 0];
Y = [0 0 0 0];

% COMPUTING PARAMETERS
for i = 1:length(u)
    U=[U u(i)];
    Y=[Y y(i)];

    [THETA PCovariant] = rls(U, Y, PCovariant, THETA, lambda);
    THETA_BUFFER = [THETA_BUFFER THETA];
end

End
```

```

% IDENTIFICATION MODEL
warning('off','all');
disp('+++++++');
fprintf('\nDiscrete model 1\n');
Gz = tf([THETA(4) THETA(5) THETA(6)], ...
        [1 THETA(1) THETA(2) THETA(3)], tSample)
disp('+++++++');
fprintf('\nContinuous model 1\n');
Gs = d2c(Gz)      % Discrete model to Continuous

% SIMULATION OF INPUT ON COMPUTED MODEL
tn = [0:(tSample/100):tRange]; % to avoid undersampling
un = sin(3*pi+0.1*tn) + sin(2*pi+0.3*tn) + sin(3*pi+0.01*tn);
yModel = lsim(Gs, un, tn);

% PLOTTING System and model output
figure(1);
plot(tn, un, 'g-.', 'LineWidth', 2);
hold on;
plot(t, y, 'r-.', 'LineWidth', 2);
plot(tn, yModel, 'b', 'LineWidth', 1);
hold off; grid on;
xlim([-0.5 tRange+0.5]);
legend('input', 'system output', 'model ouput');
title('\bf \fontsize{12} Recursive Least Squares method
identification');
xlabel('\bf t [s]'); ylabel('\bf u(t), y(t)');

% PLOTTING time changes in parameters
figure(2); hold on;
plot(t, THETA_BUFFER(1, :), 'k');
plot(t, THETA_BUFFER(2, :), 'b');
plot(t, THETA_BUFFER(3, :), 'c');
plot(t, THETA_BUFFER(4, :), 'r');
plot(t, THETA_BUFFER(5, :), 'm');
plot(t, THETA_BUFFER(6, :), 'g');
hold off; grid on;
xlim([-0.5 tRange+0.5]);
legend('a1', 'a2', 'a3', 'b1', 'b2', 'b3');
title('\bf \fontsize{12} Recursive Least Squares parameters
from model. ');
xlabel('\bf t [s]');

fprintf('\n');
disp('+++++++');
disp('|               END               |');
disp('+++++++');

```

rls.m

```
function [THETA PCovariant] = RLS(U, Y, PCovariant0, THETA0,
lambda)

    PHI = [-Y(end-1) -Y(end-2) -Y(end-3) U(end-1) U(end-2)
U(end-3)]';
    EPS = Y(end) - PHI'*THETA0; %
epsilon
    K = (PCovariant0*PHI) / (lambda + PHI'*PCovariant0*PHI); %
correction
    PCovariant = (1 / lambda) * (PCovariant0 -
K*PHI'*PCovariant0); % update
    THETA = THETA0 + K*EPS;

end
```


Příloha 4: Soubor spectralAnalysis.m implementující metodu z kapitoly 4.4

spectralAnalysis.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Identification                               %
% with Spectral Analysis                       %
%                                              %
% date: 24/5/2019                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all; clear all; clc;

disp('+++++++');
disp('|           Spectral Anaysis           |');
disp('+++++++');

% DATA PREPARATION
tSample = 0.01;
tRange = 2000;
t = [0:tSample:tRange];

%Generating of PRBS signal, expected value (mean) = 0
try
    warning('off','all');
    u = idinput(length(t), 'PRBS', [0 1], [-3 3]);
    warning('on','all');
catch
    PRBS = @(N) randi([-1 1], 1, N);
    u = PRBS(length(t));
end

% DEFINITION OF THE SYSTEM
disp('+++++++');
fprintf('\nSystem for identification\n');
s = tf('s');
K = 100; T = 5;
sys = K/(T*s+1)

% SIMULATING OF APPLICATION OF INPUT SIGNAL INTO A SYSTEM
y = lsim(sys, u, t);

% FOURIER TRANSFORM (cross power spectral density)
% FROM AUTOCORRELATION AND CORRELATION
[FIuu, u_freq] = cpsd(u, u, [], [], [], 1/tSample);
[FIuy, ~] = cpsd(u, y, [], [], [], 1/tSample);

% IDENTIFICATION MODELS - frequency response characteristic
% (complex number)
G = FIuy./FIuu;
```

```
% PLOTTING
figure(1); % plot input and output signals
subplot(2, 1, 1); grid on;
plot(t, u, 'b');
xlim([0 tRange*0.01 + 0.5]);
title('\bf \fontsize{12} Input signal - PRBS');
xlabel('\bf t [s]'); ylabel('\bf u(t)');

subplot(2, 1, 2); grid on;
plot(t, y, 'r');
xlim([0 tRange*0.01 + 0.5]);
title('\bf \fontsize{12} Output signal - PRBS response');
xlabel('\bf t [s]'); ylabel('\bf y(t)');

figure(2); % plot Amplitude freq. response characteristic

% multiply freq. by 2pi to get angular frequency:
semilogx(u_freq*2*pi, 20*log10(abs(G)), 'r', 'linewidth', 3);
hold on; grid on;
title(['\bf \fontsize{12} Amplitude freq. response' ...
      ' characteristic from model']);
xlabel('\omega [1/s]'); ylabel('|G(j\omega)| [dB]');

fprintf('\n');
disp('+++++++');
disp('|           END           |');
disp('+++++++');
```

Příloha 5: Obsah elektronické přílohy 162946_ZimekT.zip

```
C:. .....kořenový adresář
| 2019_DP_Tomas_Zimek_162946.pdf.....elektronická verze této práce
+---Identification
| | icon.png
| | Identification.py.....aplikace k identifikaci systémů
| | ReadMe.txt.....instrukce ke generování systémů z adresáře \IOdata
| |
| \---IOdata
|     comboSinResponseOfSys1.csv
|     comboSinResponseOfSys2.csv
|     comboSinResponseOfSys3.csv
|     stepResponseOfSys1.csv
|     stepResponseOfSys2.csv
|     stepResponseOfSys3.csv
|
| \---Matlab
|     leastSquares.m
|     ReadMe.txt.....instrukce ke generování testovaných systémů
|     recLeastSquaresLambda.m
|     rls.m
|     spectralAnalysis.m
|     stepInput.m
```