

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**FAKULTA STROJNÍ**  
**ÚSTAV PŘÍSTROJOVÉ A ŘÍDÍCÍ TECHNIKY**



**IDENTIFIKACE LINEÁRNÍCH KMITAVÝCH SOUSTAV**

IDENTIFICATION OF LINEAR OSCILLATORY SYSTEMS

**BAKALÁŘSKÁ PRÁCE**

**2023**

**JAN ROLLO**

**Studijní program:** B2341\_Strojírenství

**Studijní obor:** 2612R022\_Informační a automatizační technika

**Vedoucí práce:** prof. Ing. Milan Hofreiter, CSc.

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Rollo** Jméno: **Jan** Osobní číslo: **483928**  
Fakulta/ústav: **Fakulta strojní**  
Zadávací katedra/ústav: **Ústav přístrojové a řídicí techniky**  
Studijní program: **Strojírenství**  
Studijní obor: **Informační a automatizační technika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Identifikace lineárních kmitavých soustav**

Název bakalářské práce anglicky:

**Identification of Linear Oscillatory Systems**

Pokyny pro vypracování:

1. Proveďte rešerši metod vhodných pro identifikaci stabilních kmitavých soustav z přechodové charakteristiky.
2. Simulačně v prostředí Matlab ověřte vyhledané metody.
3. Porovnejte vybrané metody z hlediska praktické použitelnosti.

Seznam doporučené literatury:

Hofreiter M.: Základy automatického řízení. Skriptum, Vydavatelství ČVUT, Praha, 2012  
Šulc B., Vítečková M.: Teorie a praxe návrhu regulačních obvodů, Vydavatelství ČVUT, Praha, 2004

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**prof. Ing. Milan Hofreiter, CSc. U12110.3**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **30.04.2022** Termín odevzdání bakalářské práce: **19.08.2022**

Platnost zadání bakalářské práce: \_\_\_\_\_

\_\_\_\_\_  
prof. Ing. Milan Hofreiter, CSc.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Ing. Michael Valášek, DrSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## **PROHLÁŠENÍ**

*Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně s využitím zdrojů, které jsou všechny v práci uvedeny. Uděluji souhlas s tím, že výsledky mohou být dále využity dle uvážení vedoucího této práce.*

V Praze dne .....

Podpis .....

## **PODĚKOVÁNÍ**

Na tomto místě bych rád poděkoval prof. Ing. Milanu Hofreiterovi, CSc. za jeho podněty a odborné připomínky, které mi pomohly při psaní této bakalářské práce. Zároveň bych rád poděkoval své rodině za jejich všestrannou podporu.

## ABSTRAKT

Bakalářská práce se zabývá srovnáním několika dostupných metod používaných k identifikaci lineárních kmitavých soustav. Zvolené metody jsou využívány k získání matematického modelu neznámého systému. Pomocí softwaru MATLAB byla naměřena vstupní a výstupní data z tohoto systému, která byla dále použita k výpočtu modelu. Cílem výsledného modelu je co nejpřesněji napodobit chování původního systému. Konečná přesnost jednotlivých modelů je srovnávána mezi sebou a tím je také určeno pořadí vhodnosti využití jednotlivých metod. V závěrečném pořadí je možné určit, které metody identifikace se nejvíce hodí pro původně měřený systém a které metody k identifikaci tohoto systému nejsou vhodné.

## KLÍČOVÁ SLOVA

identifikace systémů, analýza, matematický model, MATLAB, lineární kmitavé soustavy, metoda nejmenších čtverců, metoda ploch, metoda postupné integrace, metoda aproximace přechodové charakteristiky s kmitavým průběhem

## ABSTRACT

The aim of this to bachelor thesis is compare several methods that are used for experimental identifiaton of linear oscillatory systems. These methods are designed to obtain a mathematical model from an unknown system. Using MATLAB software, input and output data from above mentioned **SYSTEM WERE** measured, were used to compute the final model. The purpose of this model is to mimic the behaviour of the original system as accurate as possible. The resulting accuracy of the individual models is compared with each other to determine the ranking of the suitability of each method. In the model accuracy ranking, it is possible to see which identification methods are most appropriate for the original system and which methods are not suitable for identification of the system.

## KEY WORDS

identification of systems, analysis, mathematical model, MATLAB, linear oscillatory systems, least squares method, surface method, gradual integration method, method for approximation of transient characteristic with oscillating waveform

# OBSAH

<b>PŘEHLED POUŽITÝCH ZKRATEK A SYMBOLŮ .....</b>	<b>7</b>
<b>1 ÚVOD .....</b>	<b>8</b>
<b>2 TEORETICKÁ ČÁST .....</b>	<b>9</b>
2.1 KMITAVÝ POHYB – FYZIKÁLNÍ PODSTATA .....	9
2.2 IDENTIFIKACE SYSTÉMŮ .....	9
2.3 MATEMATICKÝ MODEL .....	12
2.4 METODY VÝPOČTU MODELŮ SYSTÉMŮ .....	14
2.4.1 Metoda nejmenších čtverců (MNČ) .....	14
2.4.2 Metoda aproximace přechodové charakteristiky s kmitavým průběhem (PKS) .....	16
2.4.3. Metoda ploch (MP, MetPloch) – Parametrizace přechodové charakteristiky .....	19
2.4.3. Metoda postupné integrace (MPI) .....	21
<b>3 PRAKTICKÁ ČÁST .....</b>	<b>23</b>
3.1 NAVRHOVANÉ PRINCIPY MĚŘENÍ .....	23
3.2 PŘÍKLADY VÝPOČTU S POUŽITÍM TEORIE DANÝCH MATEMATICKÝCH METOD: .....	25
3.2.1 Příklad 1 - Základní případ .....	25
3.2.2 Příklad 2 - Příklad s vyšším kmitáním .....	26
3.2.3 Příklad 3 – Příklad s jedním nulovým bodem .....	28
3.2.4 Příklad 4 - Příklad se zpožděním signálu .....	30
3.2.5 Příklad 5 - Model 3. řádu .....	32
3.2.6 Příklad 6 - Nekmitavý případ .....	34
<b>4 VYUŽITÍ V PRAXI .....</b>	<b>36</b>

4.1 TEORIE PRAKTICKÉHO MĚŘENÍ .....	36
4.2 VÝSLEDKY EXPERIMENTÁLNÍ IDENTIFIKACE NA REÁLNÉ LABORATORNÍ ÚLOZE .....	39
4.3. SROVNÁNÍ VÝSLEDKŮ .....	42
<b>5 ZÁVĚR .....</b>	<b>45</b>
<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>48</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>50</b>
<b>SEZNAM TABULEK .....</b>	<b>51</b>
<b>PŘÍLOHY .....</b>	<b>52</b>
VÝVOJ PROGRAMU .....	52

## PŘEHLED POUŽITÝCH ZKRATEK A SYMBOLŮ

	$(R_i =  R_i e^{j\theta_i})$
$M_i$	moment
$R_i$	reziduální nebo komplexní koeficient
$T_s$	perioda odběru vzorků
$\alpha_i$	faktor tlumení
$\theta_i$	fázový úhel
$\omega_i$	úhlová frekvence ( $\omega_i = 2\pi f_i$ )
$G$	model systému
$h(t)$	normovaný výstup
ICT	Information and Communication
MetPloch, <u>MP</u>	metoda ploch
MNČ (MNC)	metoda nejmenších čtverců
MPI	metoda postupné integrace
PKS	metoda aproximace přechodové charakteristiky s kmitavým průběhem
T	perioda Technologies
u	výstup
<b>u</b>	výstupní data
y	vstup
<b>yreal</b>	vstupní data
$A$	amplituda
$K$	zesílení
$L$	Laplaceova transformace
$N$	počet vzorků
$S$	plocha
$t$	čas
$\Omega$	vlastní úhlová frekvence
$\xi$	součinitel poměrného tlumení



# 1 ÚVOD

Analýza dat zahrnuje velmi širokou oblast technik zpracování i využívání informací. Poptávka po analýze dat společně s pokročilejší technikou, automatizací a digitalizací stále roste. Analýza je využívána denně, a to nejen v průmyslu ale i v běžném životě.

Z tohoto důvodu je stále větší zájem o efektivnější, spolehlivější a rychlejší metody zpracování těchto dat, o předpovídání budoucího chování původních systémů a následnou regulaci. Ačkoliv je matematické zpracování dat velmi starou vědní disciplínou teprve v současnosti, díky pokročilé technice, získává stále větší uplatnění. V průmyslu se snažíme nahradit nebezpečné a náročné práce automatizovanými roboty. I běžné domácí přístroje využívají senzorů, jenž kontrolují jejich bezpečný chod. Vše uvedené má společné právě to, že by bez dobře analyzovaných informací nemohly efektivně fungovat. Informace, jejich rozpoznávání a jejich zpracování je tedy s modernizací úzce spjato.

Cílem této práce je vytvoření modelů, které co nejpřesněji popisují chování původního zkoumaného systému. Nejprve budou pro snazší pohled do problematiky vysvětleny některé základní pojmy a objasněna problematika procesu identifikace a modelování obecně. Následně budou blíže popsány jednotlivé použité metody identifikace. V praktické části si pak na testovaných příkladech v prostředí MATLAB demonstrujeme možnosti iterativního postupu identifikace systému. Dále budou získané výsledky u jednotlivých modelů srovnány mezi sebou a tím se i určí, které metody identifikace se nejvíce hodí pro původně využitý systém. Poté budou metody aplikovány na reálných datech naměřených v laboratorní úloze nazvané „Vodní levitace“. I tyto výsledky jednotlivých metod budou srovnány a vyhodnotí se nejvhodnější metoda pro řešení dané laboratorní úlohy. Na závěr budou všechny výsledky metod porovnány, tedy teoretické i experimentální, a určí se nejspolehlivější metoda.

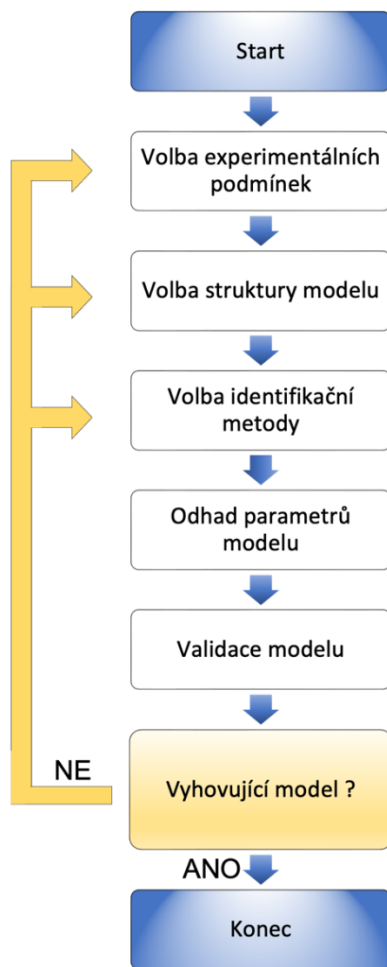
## 2 TEORETICKÁ ČÁST

### 2.1 Kmitavý pohyb – fyzikální podstata

Základ vlnových jevů tvoří netlumené kmity, primárně lineárních soustav [1]. Ve všech fyzikálních oborech se s kmitavým dějem setkáváme (například elektrické kmitavé obvody, seismické vlny, molekulární spektra a další). Kmitavým pohybem rozumíme periodický pohyb, kdy sledované těleso prochází jejich rovnovážnou polohou. Nejkratší doba, za kterou se pohyb tělesa opakuje je perioda  $T$  (doba kmitu). Počet kmitů za 1 sekundu se nazývá frekvence (kmitočet). Jednotkou frekvence je Hertz (Hz). V praxi se častěji používají jednotky KHz, MHz. Kmitavý pohyb je pohyb nerovnoměrný [2].

### 2.2 Identifikace systémů

Identifikace systémů je poznávací proces zabývající se hledáním, s cílem nalezení co nejpřesnějšího matematického modelu (popisu) na základě získaných vstupních dat [3]. Při tomto postupu určujeme parametry a strukturu modelu. Aby model systému vystihl podstatné vazby mezi příčinami (vstupy) a důsledky (výstupy), které jsou významné z hlediska sledovaného cíle, je nutné uvažovaný systém dobře identifikovat. Hledání vhodného modelu pro daný účel nazýváme identifikací [4]. Cílem identifikace systému je tedy vytvořit takový model systému, jehož chování se při stejných podmínkách co nejvíce blíží chování původního reálného systému. Je ale třeba brát vždy na zřetel, že pokud jsou data sledovaného tělesa získána z reálného prostředí, je toto těleso v neustálé interakci s tímto prostředím. Je vhodné, aby se v tomto procesu identifikace pracovalo s reálně měřitelnými vstupními a výstupními veličinami, aby srovnání metod a jejich výsledných modelů nejlépe odpovídalo reálnému využití identifikace [5].



Obr. 1: Postup identifikace a hledání vhodného matematického modelu [6]

Při hledání modelu jsou dva základní přístupy identifikace, identifikace analytická a identifikace experimentální. Analytická i experimentální identifikace má své klady i zápory, proto je v praxi často důležité je vzájemně kombinovat [7].

Identifikace analytická, nazývaná rovněž jako matematicko-fyzikální analýza, zjišťuje matematický popis identifikovaného systému analyticky za použití fyzikálních, chemických, biologických, ekonomických zákonů. Měření na reálných zařízeních není nutné. Výhodou proti zmiňované experimentální identifikaci je, že analytická identifikace může být použita již v okamžiku přípravy projektu, protože nevyžaduje existenci skutečné soustavy. Výsledkem této identifikace je popis vnitřních vazeb systému. V angličtině se užívá název této metody jako „white box“ [4] (metoda bílé skříňky). Analytické metody identifikace tedy vyžadují především znalost fyzikálních a matematických procesů a způsob jejich popisu.

Experimentální identifikace je analýza, která se opírá o údaje zjištěné pozorováním identifikovaného systému při experimentu. Z tohoto vyplývá, že vyžaduje

existenci zkoumaného objektu a možnost s ním experimentovat. Identifikace experimentální je tedy volena v případě, že je zkoumaný systém složitější a nelze pro něj vytvořit jednoduchý ekvivalent analytického modelu. Vytvořený model popisuje parametry ze změřených dat bez fyzikálního významu. Výsledkem této identifikace je vnější popis systému, jelikož vnitřní vazby jsou skryty. V angličtině se užívá název této metody „black box“ (metoda černé skříňky) [4]. Experimentální metody jsou díky přístrojovému vybavení nákladnější. Potřebná znalost procesu je však nižší než u metod analytické identifikace.

Tabulka 1: Vlastnosti analytické a experimentální identifikace [8]

<b>Vlastnost modelu</b>	<b>Analytická identifikace</b>	<b>Experimentální identifikace</b>
Struktura modelu	Struktura vyplývá z přírodních zákonů.	Struktura musí být zvolena.
Popis systému	Systém je popsán pomocí vnitřních stavových proměnných a chování na vstupu a výstupu.	Systém je popsán pouze pomocí relace vstup – výstup.
Parametry modelu	Parametry modelu jsou funkcemi systémových veličin, mají fyzikální význam.	Parametry modelu jsou analytické proměnné, které neumožňují většinou nalézt souvislost s fyzikálními systémovými proměnnými.
Platnost modelu	Model platí pro celou třídu typů procesu a pro různé provozní stavy.	Model platí pouze pro zkoumaný proces a konkrétní provozní stav.
Existence originálního systému	Model může být vytvořen i pro neexistující systém.	Model může být identifikován pouze pro existující systém.
Znalost vnitřní struktury	Důležité vnitřní procesy systému musí být známé a matematicky popsatelné.	Vnitřní procesy nemusí být známé.
Opakované použití metody	Každá tvorba modelu představuje opakovanou aplikaci fyzikálních zákonů. Zvyšování know-how řešitele.	Metody nezávislé na jednotlivých systémech, vytvořené programové vybavení může být opakovaně použito pro identifikaci různých systémů.
Časová náročnost	Tvorba modelu vyžaduje větší časové nároky.	Menší časové nároky.

## 2.3 Matematický model

Matematický model je zjednodušené abstraktní znázornění zkoumaného systému, který pomocí matematického jazyka (rovnice, funkce, vzorce atd.) a matematických pojmů popisuje určité chování systému [9].

Matematický model můžeme rozdělit na parametrický a neparametrický [8]. U parametrických modelů (analytická identifikace) jsou dány struktury a z matematického hlediska jsou popsány rovnicemi, algebraickými vztahy, představující parametry modelu. Neparametrické modely (experimentální identifikace) jsou funkčně závislé na vstupním signálu a odpovídajícím výstupním signálu, který je vyjádřen v grafické podobě (tabulka, graf, apod.).

Z jiného hlediska lze matematický model dělit také na model statický a model dynamický. Statický model popisuje chování systému v ustáleném stavu, zatímco model dynamický popisuje chování systému výstupních veličin při změně vstupních veličin, tedy změny výstupu na současných změnách vstupu a minulých změnách vstupu a výstupu [4].

Cílem identifikace systému je nalézt takový model, který bude sloužit pro predikci výstupů systému. K dosažení optimální identifikace matematického modelu je zásadní volba správné struktury modelu systému. Do jisté míry platí, že získání co nejdokonalejší shody modelu s reálným systémem závisí na složitosti struktury modelu [6]. Při výběru vyhovující struktury je žádoucí využít znalostí profilu systému a stupně přesnosti, který budeme od matematického modelu požadovat.

Matematické modely jsou využitelné v přírodních vědách, technických oborech i ve společenských vědách. Matematické modely mají mnoho podob (dynamické systémy, statické modely, logické modely, teoretické modely her, diferenciální rovnice atd.), přičemž se mohou vzájemně prolínat. Tedy matematický model pracuje s množinou vstupních a výstupních proměnných, konstant, parametrů, množinou matematických struktur a řešením. Ty pak určují vazby systému, stavy a vztahy mezi parametry a proměnnými. Matematický model je tak stanoven pouze na základě vstupně – výstupních experimentálních hodnot a na výslednou kvalitu matematického modelu má vliv výběr vhodné struktury modelu a identifikační metody. Je tedy třeba si zároveň uvědomit, že identifikační metody mohou být časově velmi náročné, a použití získaného matematického modelu je tak omezené. Parametry modelu, získané identifikací, zcela neodpovídají reálným parametrům systému a nenesou tak skutečný (např. fyzikální)

význam. Validita takového modelu je omezená množstvím experimentálních dat, na základě kterých byl model vytvořen [6]. Shoda matematických modelů postavených na teoriích s výsledky opakovatelných experimentů často záleží na kvalitě konkrétního vědního oboru. Pokud tomu tak není, je snaha tento model upravit. Úpravy modelu následně dospějí k lepším teoriím popisující fakta daného oboru [10].

Vlastnosti, které by měl mít navrhovaný matematický model je především jednoduchost, užitečnost, stabilita, citlivost, objektivnost a univerzálnost. I když se matematické modely jeví jako čistě hypotetický koncept, není tomu tak. V každodenním světě se vyskytuje velké množství aspektů, které se běžně řídí matematickými modely. Tyto matematické modely jsou formulované pro určitou funkci, například při řízení vozidla [11].

Zjednodušený proces vývoje pro vytvoření matematického modelu je následující [12]:

- 1) Určení problému a cíle.
- 2) Získání potřebných dat
- 3) Definování modelu pomocí matematického jazyka a identifikování relevantní proměnné (nezávislé a závislé)
- 4) Stanovení předpokladu a metody testování
- 5) Použití matematických vzorců k řešení a předpovědi chování systému
- 6) Srovnání získaných hodnot s hodnotami skutečnými
- 7) V případě, že výsledné hodnoty se neblíží očekávání, je nutné opravit matematický model

Ačkoliv je matematické modelování velmi starou disciplínou popisu chování určitých systémů, teprve v šedesátých letech minulého století proniklo díky informačním a komunikačním technologiím [Information and Communication Technologies (ICT)] více do vědních oborů, praxe i do výuky. Nyní si již matematické modelování nedovedeme bez ICT představit.

Existuje mnoho komerčních programů používaných k získání funkčních matematických modelů jako například programy Maple od Maplesoft Inc. ([www.maplesoft.com](http://www.maplesoft.com)), MathCAD od MathSoft Inc., ([www.mathsoft.com](http://www.mathsoft.com)), Mathematica od Wolfram Research, Inc. OCTAVE pro numerické výpočty ([www.gnu.org/software/octave/index.html](http://www.gnu.org/software/octave/index.html)), MATLAB od společnosti Mathworks Inc. ([www.mathworks.com](http://www.mathworks.com)). S posledním jmenovaným programem MATLAB provádíme

výpočty potřebné k získání matematických modelů v této bakalářské práci. Výše zmiňovaný aplikační program MATLAB jsme využili při matematickém modelování během celého výpočetního procesu, tedy při identifikaci, analýze, vývoji, implementaci, řešení a ověřování matematického modelu v praxi.

„Ukazuje se, že přesnost matematického modelu je pouze iluze, neboť je principiálně nedosažitelná. Snaha o absolutní přesnost nás vždy dovede ke sporu. Není však třeba litovat, neboť vágnost, kterou přirozený jazyk umí dokonale využít, je jeho hlavní silou, nikoliv nedostatkem.“ [9].

## **2.4 Metody výpočtu modelů systémů**

### **2.4.1 Metoda nejmenších čtverců (MNČ)**

Metoda nejmenších čtverců (MNČ) je jednou z nejznámějších a nejvíce využívaných metod k identifikaci systémů. Důvodem je jednoduchost metody a rychlost výpočtu. Základní myšlenkou metody je vytvořit model systému sloužící k co nejpresnějšímu ověřování experimentálně zjištěných dat pomocí aproximační funkce, která by se co nejvíce přibližovala (co nejmenším součtem obsahů čtverců), reálnému systému. Nejmenšími čtverci rozumíme druhé mocniny chyb odchylek (reziduí) mezi řešením zjištěnými hodnotami a původními naměřenými hodnotami [15]. MNČ je nápomocná při určování takových hodnot koeficientů řešení, aby byl součet těchto odchylek co možná nejmenší. Od tohoto je i odvozen název metody, tj. Metoda nejmenších čtverců [16].

MNČ byla vyvinuta Carlem Friedrichem Gaussem již roce 1795. Tuto metodu vytvořil za účelem vyloučení chyb ze získaných hodnot při geodetickém vyměřování. Postupně si tato metoda nacházela stále více uplatnění i v mnoha dalších oborech. V dnešních ICT je metoda běžnou součástí každodenně používaných programů, například programu Microsoft Excel.

Primárně je důležité si určit jakou formu bude náš hledaný model mít. Toto rozhodování závisí hlavně na tvaru původního systému a také na znalostech a zkušenostech člověka, jenž provádí výpočet.

Nejjednodušší z hledaných forem modelů je přímka. Z tohoto důvodu si právě na přímce ukážeme celý proces výpočtu modelu, pomocí metody MNČ.

Přímka

Rovnici přímky  $f(x) = a \cdot x + b$  je možné proložit polynomem 1. řádu (přímkou), pomocí parametrů, které získáme rovnicemi:

$$a = \frac{n \cdot \sum_i^n x_i \cdot y_i - (\sum_i^n x_i) \cdot (\sum_i^n y_i)}{n \cdot (\sum_i^n x_i^2) - (\sum_i^n x_i)^2} \quad (1)$$

$$b = \frac{(\sum_i^n x_i^2) \cdot (\sum_i^n y_i) - (\sum_i^n x_i) \cdot (\sum_i^n x_i \cdot y_i)}{n \cdot (\sum_i^n x_i^2) - (\sum_i^n x_i)^2} \quad (2)$$

$x_i$ -získaná data osy x

$y_i$ -získaná data osy y

Kvadratické kritérium:

$$J = \sum_{i=1}^n (x_i - y_i)^2 \quad (3)$$

J – Kvadratické kritérium

$x_i$  –výstupní signál

$y_i$  – predikce výstupního signálu

$n$  – počet naměřených dat

Platí, že čím je kvadratické kritérium menší, tím více se vytvořený model blíží zkoumanému reálnému systému. V ideálním případě by se hodnota kvadratického kritéria měla rovnat nule. Toto však v praxi není možné, jelikož zkoumaná data jsou zpravidla získávána reálným měřením. Měření a pak tedy i naměřená data jsou ovlivněna i svým okolím. Tomuto ovlivnění dat se v praxi říká šum. Z důvodu, že všechna naměřená data obsahují určité množství šumu, který je v datech nerovnoměrně rozšířený, tak matematický model, který výpočtem získáme, není zcela totožný s modelem zkoumaného systému.

Metoda nejmenších čtverců bývá používána i jako součást dalších metod využívaných k identifikaci systémů, nebo k filtraci šumu vyskytujícímu se v zadaných datech.



## 2.4.2 Metoda aproximace přechodové charakteristiky s kmitavým průběhem

### (PKS)

K nejužívanějším postupům získání lineárního parametrického modelu patří geometrická analýza přechodové charakteristiky. To se logicky projevuje i ve velkém množství publikovaných metod aproximace přechodové charakteristiky [4, 8].

Výsledné parametrické modely jsou pak odlišné strukturou i parametrizací, protože pro popis reálného systému nejde určit jeden absolutní model. Postupy jsou tedy děleny podle toho, zda je identifikovaný systém proporcionální, derivační nebo integrační. Toto lze z průběhu přechodové charakteristiky snadno odlišit.

Základní dělení proporcionálního systému je na proporcionální nekmitavý systém a proporcionální kmitavý systém. Proporcionální kmitavý systém je dále využit v experimentální části této bakalářské práce.

#### 2.4.6.1. Proporcionální kmitavý systém

Pokud přechodová charakteristika vykazuje kmity, je evidentní, že model má nejméně dva kořeny komplexně sdružené. Z toho vyplývá, že aproximující charakteristika bude určena přenosem, jenž obsahuje proporcionální člen se setrvačností 2. řádu a součinitelem poměrného tlumení  $0 < \xi < 1$ . [4, 8].

Tento model systému obsahuje alespoň dva komplexně sdružené kořeny.

$$G_{(s)} = \frac{S}{T_0^2 \cdot s^2 + 2 \cdot \xi \cdot T_0 \cdot s + 1}, \quad \text{pro } 0 < \xi < 1. \quad (4)$$

Póly přenosu  $G_{(s)}$  jsou

$$s_{1,2} = \frac{-2 \cdot \xi \cdot T_0 \pm \sqrt{4 \cdot \xi^2 \cdot T_0^2 - 4 \cdot T_0^2}}{2 \cdot T_0^2} = \frac{-\xi}{T_0} \pm j \cdot \frac{1}{T_0} \cdot \sqrt{1 - \xi^2} \quad (5)$$
$$= \alpha \pm j\Omega,$$

kde vlastní útlum

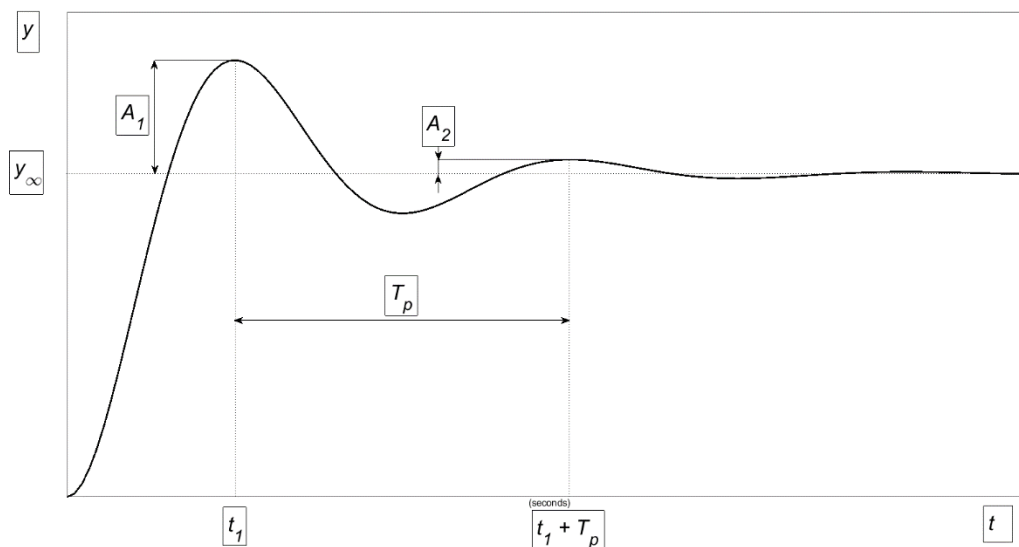
$$\alpha = \frac{-\xi}{T_0}, \quad (6)$$

a vlastní úhlová frekvence

$$\Omega = \frac{1}{T_0} \cdot \sqrt{1 - \xi^2}. \quad (7)$$

Přechodovou odezvu  $y(t)$  na skokovou změnu vstupu  $u$  lze popsat funkcí

$$y(t) = C \cdot e^{\alpha t} \cdot \cos(\Omega \cdot t + \gamma) + y_\infty \quad (8)$$



Obr. 2: Přechodová odezva kmitavého proporcionálního členu [4].

U přechodové odezvy  $y(t)$  pro poměr amplitud  $\frac{A_2}{A_1}$  vzdálených o periodu  $T_p$  (viz obrázek č.2) platí

$$\frac{A_2}{A_1} = \frac{C \cdot e^{\alpha(t_1+T_p)} \cdot \cos(\Omega \cdot (t_1 + T_p) + \gamma)}{C \cdot e^{\alpha t_1} \cdot \cos(\Omega \cdot t_1 + \gamma)} = e^{\alpha T_p} = e^{\alpha \cdot \frac{2 \cdot \pi}{\Omega}}, \quad (9)$$

neboť pro vlastní úhlovou frekvenci platí

$$\Omega = \frac{2 \cdot \pi}{T_p}. \quad (10)$$

Zavedeme-li označení

$$\vartheta = \ln\left(\frac{A_2}{A_1}\right), \quad (11)$$

z čehož plyne

$$\vartheta = \ln\left(\frac{A_2}{A_1}\right) = 2 \cdot \pi \cdot \frac{\alpha}{\Omega} = -\frac{2 \cdot \pi \cdot \xi}{\sqrt{1 - \xi^2}}. \quad (12)$$

A víme, že ze známého vzorce využití limitní věty o konečné hodnotě (u Laplaceovy transformace) ze které pak

$$\xi = \frac{\vartheta}{\sqrt{(2 \cdot \pi)^2 + \vartheta^2}} \quad (13)$$

a jelikož

$$\Omega = \frac{2 \cdot \pi}{T_p} = \frac{1}{T_0} \cdot \sqrt{1 - \xi^2}, \quad (14)$$

pak

$$T_0 = \frac{T_p}{2 \cdot \pi} \cdot \sqrt{1 - \xi^2}. \quad (15)$$

Hledané parametry  $T_0, \xi$  tedy určíme pomocí vztahů  $\vartheta = \ln\left(\frac{A_2}{A_1}\right), \xi = \frac{\vartheta}{\sqrt{(2 \cdot \pi)^2 + \vartheta^2}}, T_0 = \frac{T_p}{2 \cdot \pi} \cdot \sqrt{1 - \xi^2}$ , kde  $T_p, A_2, A_1$  můžeme přímo odečíst z přechodové odezvy. Statické zesílení  $S$  určíme opět z poměru ustálených hodnot výstupu  $y_\infty$  a vstupu  $u_\infty$ , tj. vztahu

$$S = \frac{y_\infty}{u_\infty}. \quad (16)$$

Hledané parametry  $T_0, \xi$  můžeme též určit z hodnoty maximálního relativního přeregulování  $\sigma_{max}$ , pro které platí

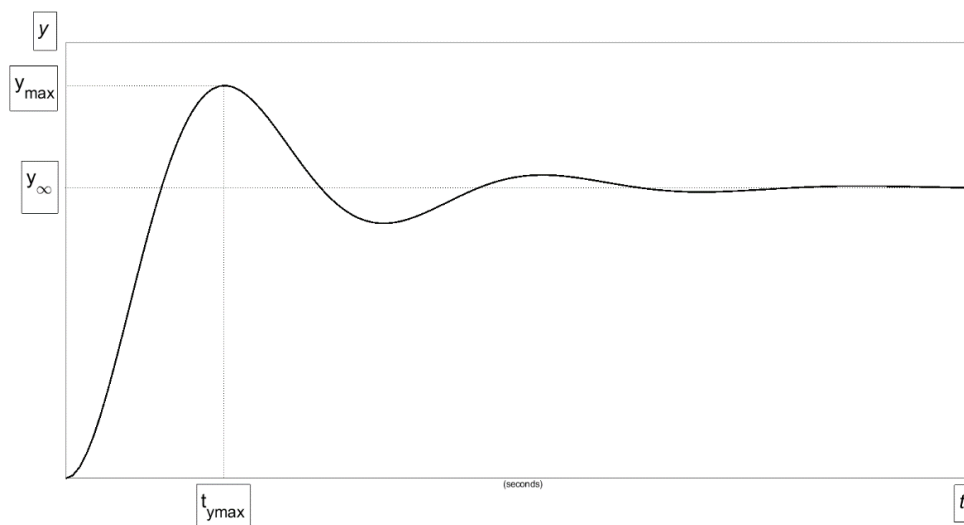
$$\sigma_{max} = \frac{y_{max} - y_\infty}{y_\infty} \quad (17)$$

a času  $t_{y_{max}}$ , ve kterém k němu došlo. Z podmínky extrému přechodové funkce, viz obrázek 3, lze odvodit

$$T_0 = \frac{t_{y_{max}} \cdot \sqrt{1 - \xi^2}}{\pi}, \quad (18)$$

$$\xi = \left| \frac{\frac{\ln(\sigma_{max})}{\pi}}{\sqrt{1 + \left(\frac{\ln(\sigma_{max})}{\pi}\right)^2}} \right| \quad (19)$$

Naměřená odezva vždy neodpovídá odezvě pouze členu 2. řádu, proto může být přijatelné rozšířit tento model o sériově řazený proporcionální člen se setrvačností 1. řádu, především když jsou naměřené kmity tlumenější, než je aproximující odezva [8].



Obr. 3: Přechodová odezva proporcionálního kmitavého systému s vyznačením  $y_{max}$ ,  $t_{y_{max}}$ ,  $y_{\infty}$  [4]

Pokud je neznámým systémem proporcionální kmitavý člen 2. řádu, model je tak určen přesně. Pokud je však kmitavý proporcionální systém vyššího řádu s dominantní dvojicí komplexně sdružených pólů, může být tento model přiměřenou aproximací neznámého systému.

### 2.4.3. Metoda ploch (MP, MetPloch) – Parametrizace přechodové charakteristiky

Metodu ploch publikovali v roce 1992 pánové J.C. Simo-J.W. Ju v [Simo-Ju 1957, Isermann 1992]. Tato metoda byla originálně využita pro výpočet izotropních modelů poškození založeného na deformaci [14].

Metodou ploch je možné vytvořit model přechodové charakteristiky systému [8] přenosu ve tvaru:

$$G_{MetPloch} = \frac{K}{a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_1 \cdot s + 1} \quad (20)$$

Vypočteme zesílení systému  $K$  ze vztahu:

$$K = \frac{y_\infty}{u_\infty} \quad (21)$$

Následně vydělíme zadaná data pro výstup  $y$  hodnotou  $K$ .

$$h(t) = \frac{y(t)}{K} \quad (22)$$

Nyní vypočteme koeficienty jmenovatele polynomu obrazu  $a_i$ .

Nejdříve vypočteme tzv. momenty  $M_i$  integrací [8]:

Pro Laplaceův obraz naší funkce  $f(t) = 1 - h(t)$  platí, z definice Laplaceovy transformace, funkce:

$$L\{f(t)\} = \int_0^\infty f(t) \cdot e^{-s \cdot t} dt$$

Kde člen  $e^{-s \cdot t}$  rozepíšu do tvaru:

$$e^{-s \cdot t} = \sum_{i=0}^{\infty} \frac{(-t)^i}{i!} \cdot s^i$$

Po dosazení do rovnice bude platit vztah:

$$L\{1 - h(t)\} = \int_0^\infty [1 - h(t)] \cdot \sum_{i=0}^{\infty} \frac{(-t)^i}{i!} \cdot s^i dt = \sum_{i=0}^{\infty} M_i \cdot s^i$$

Z čehož vyplívá vztah pro moment  $M_i$ :

$$M_i = \int_0^\infty \frac{[1 - h(t)] \cdot (-t)^i}{i!} dt \quad (23)$$

$i$  – index momentů. Odpovídá řádu výsledného modelu.

$M_i$  – moment

$t$  – čas

$L$  – Laplaceova transformace

Následně vypočítáme plochy  $S_i$ , které jsou díky počátečnímu vydělení získaných dat zesílením  $K$  rovny hodnotám koeficientů jmenovatele modelu  $a_i$ .

$$a_i = S_i = M_i + \sum_{j=0}^{i-1} S_{i-j-1} \cdot M_{j+1} \quad (24)$$

V našem případě

$$a_1 = S_1 = M_1 \quad (25)$$

$$a_2 = S_2 = S_1 \cdot M_1 + M_2 \quad (26)$$

$$a_3 = S_3 = S_2 \cdot M_1 + S_1 \cdot M_2 + M_3 \quad (27)$$

$$a_4 = S_4 = S_3 \cdot M_1 + S_2 \cdot M_2 + S_1 \cdot M_3 + M_4 \quad (28)$$

Maximální použitelný řád modelu je takový, dokud vyjde nejvyšší hodnota  $a_i$  a všechny menší hodnoty koeficientů kladné. Jinak řečeno, pokud vyjde hodnota  $a_{i+1}$  záporná a  $a_i, a_{i-1}, \dots, a_1$  jsou kladné, tak je nejvyšší řád systému roven hodnotě  $i$ .

Výsledek je model:

$$G_{MetPloch} = \frac{K}{1 + \sum_1^i a_i \cdot s^i} \quad \text{pro } i = \text{řád modelu} \quad (29)$$

### 2.4.3. Metoda postupné integrace (MPI)

Pomocí metody postupné integrace dokážeme přibližně určit parametry matematického modelu. K jejich výpočtu je zapotřebí znát naměřené odezvy pro známé průběhy vstupních signálů [17]. Důležitou podmínkou je to, že se vždy vychází z jednoho ustáleného stavu a končí v druhém ustáleném stavu.

Pokud tedy uvažujeme ustálený stav jak vstupní veličiny  $u$  tak výstupní veličiny  $y$ , na počátku měření a na jeho konci [17] a tyto stavy se nerovnají, pak musí platit diferenciální rovnice:

$$a_n \cdot y^n(t) + a_{n-1} \cdot y^{n-1}(t) + \dots + a_2 \cdot \ddot{y}(t) + a_1 \cdot \dot{y}(t) - a_0 \cdot [y_\infty - y(t)] = -[u_\infty - u(t)] \quad (30)$$

Z této rovnice popisující charakteristiku systému za pomoci integrace lze získat jednotlivé složky  $a_0, a_1, a_2, \dots, a_n$ .

Nejprve pro složku  $a_0$  platí:

$$a_0 = \frac{u_\infty}{y_\infty} \quad (31)$$

Následně vypočteme  $a_1$  integrujeme-li rovnici od 0 do  $\infty$ , pak bude platit následující:

$$a_1 \cdot y_\infty - a_0 \cdot \int_0^\infty [y_\infty - y(t)] dt = - \int_0^\infty [u_\infty - u(t)] dt \quad (32)$$

$$a_1 = \frac{a_0 \cdot \int_0^\infty [y_\infty - y(t)] dt - \int_0^\infty [u_\infty - u(t)] dt}{y_\infty} \quad (33)$$

Pro parametr  $a_2$  po integraci od  $t$  do  $\infty$  a integraci od 0 do  $\infty$  a stejným způsobem vypočítáme i  $a_3$ :

$$\begin{aligned} -a_2 \cdot y_\infty + a_1 \cdot \int_t^\infty [y_\infty - y(t)] dt - a_0 \cdot \int_0^\infty \int_t^\infty [y_\infty - y(t)] (dt)^2 \\ = \int_0^\infty \int_t^\infty [u_\infty - u(t)] dt \end{aligned} \quad (34)$$

$$\begin{aligned} a_2 = -\frac{1}{y_\infty} \cdot \left\{ a_0 \cdot \int_0^\infty \int_t^\infty [y_\infty - y(t)] (dt)^2 - a_1 \right. \\ \left. \cdot \int_0^\infty [y_\infty - y(t)] dt - \int_0^\infty \int_t^\infty [u_\infty - u(t)] dt \right\} \end{aligned} \quad (35)$$

$$\begin{aligned} a_3 \cdot y_\infty - a_2 \cdot \int_0^\infty [y_\infty - y(t)] dt + a_1 \cdot \int_0^\infty \int_t^\infty [y_\infty - y(t)] (dt)^2 - a_0 \\ \cdot \int_0^\infty \int_t^\infty \int_t^\infty [y_\infty - y(t)] (dt)^3 = - \int_0^\infty \int_t^\infty \int_t^\infty [u_\infty - u(t)] (dt)^3 \end{aligned} \quad (36)$$

$$\begin{aligned} a_3 = \frac{1}{y_\infty} \cdot \left\{ a_0 \cdot \int_0^\infty \int_t^\infty \int_t^\infty [y_\infty - y(t)] (dt)^3 - a_1 \right. \\ \left. \cdot \int_0^\infty \int_t^\infty [y_\infty - y(t)] (dt)^2 - a_2 \cdot \int_0^\infty [y_\infty - y(t)] dt \right. \\ \left. - \int_0^\infty \int_t^\infty \int_t^\infty [u_\infty - u(t)] (dt)^3 \right\} \end{aligned} \quad (37)$$

V bakalářské práci budeme pouze počítat s odezvou výstupu  $y$  na skokový vstup  $u$ . Z tohoto důvodu bude platit:

$$\int_0^{\infty} [u_{\infty} - u(t)] dt = 0 \quad (38)$$

Proto lze upravit vzorce na:

$$a_1 = \frac{a_0 \cdot \int_0^{\infty} [y_{\infty} - y(t)] dt}{y_{\infty}} \quad (39)$$

$$a_2 = \frac{1}{y_{\infty}} \cdot \left\{ a_0 \cdot \int_0^{\infty} \int_t^{\infty} [y_{\infty} - y(t)] (dt)^2 - a_1 \cdot \int_0^{\infty} [y_{\infty} - y(t)] dt \right\} \quad (40)$$

$$a_3 = \frac{1}{y_{\infty}} \cdot \left\{ a_0 \cdot \int_0^{\infty} \int_t^{\infty} \int_t^{\infty} [y_{\infty} - y(t)] (dt)^3 - a_1 \cdot \int_0^{\infty} \int_t^{\infty} [y_{\infty} - y(t)] (dt)^2 - a_2 \cdot \int_0^{\infty} [y_{\infty} - y(t)] dt \right\} \quad (41)$$

Pomocí parametrů  $a_0, a_1, \dots, a_n$  vytvoříme model:

$$G_{MPI} = \frac{1}{\sum_{j=0}^n a_j \cdot s^j} \quad \text{pro } n \text{ rovno řádu modelu} \quad (42)$$

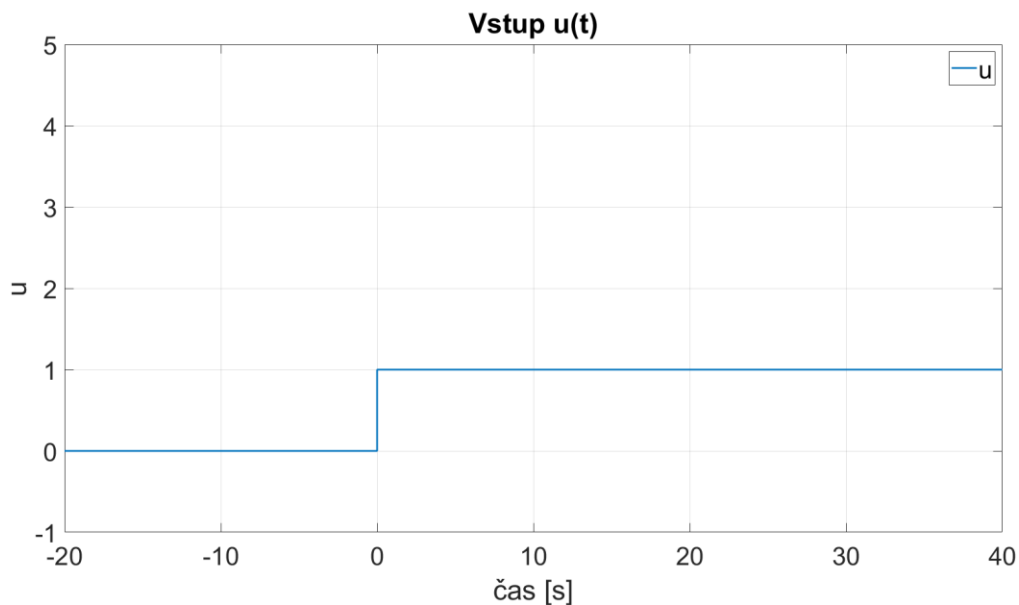
## 3 PRAKTICKÁ ČÁST

### 3.1 Navrhované principy měření

V této části je demonstrováno využití jednotlivých metod k identifikaci uměle vytvořených dat systému. Následně budou srovnány výsledky identifikace jak s původními daty, tak mezi jednotlivými výsledky.

K získání dat byl pokaždé použit identický vstup  $u(t)$ , a vždy se jednalo o jednotkový skok:





Obr. 4: Jednotkový skok vstupu

Pro lepší přehlednost následně uváděných grafů u jednotlivých příkladů není uváděn jednotkový skok vstupu (obr. 4).

Výsledná data byla srovnána pomocí vzorců:

Pro jednotlivé body v daném čase  $t$  vytvoříme procentuální shodu s původními daty:

$$p_i = \left(1 - \frac{|y_{real_i} - y_i|}{y_{real_i}}\right) \cdot 100 [\%]$$

Pro  $i = 1, 2, \dots, n$ , kde  $n = \text{počet získaných dat}$ .

$p_i$  je procentuální shoda dat pro daný čas  $t_i$ .

Průměr:

$$\text{Průměr} = \frac{[\sum_{i=1}^n (p_i)]}{n}$$

Medián:

$$\text{Medián} = \text{median}(p_i)$$

Nejmenší shoda:

$$\text{Nejmenší shoda} = \min(p_i)$$

## 3.2 Příklady výpočtu s použitím teorie daných matematických metod:

### 3.2.1 Příklad 1 - Základní případ

Tento základní případ byl zvolen, aby byla zjištěna správná funkce programů všech čtyř zvolených metod. Zadaný model má kmitavou charakteristiku, bez zpoždění. Samotný model neobsahuje nulové body a je druhého řádu.

Bylo očekáváno, že všechny naprogramované metody budou schopny dobře popsat chování původního systému a jejich výsledné modely se budou velice blížit původnímu modelu. Bylo předpokládáno, že výsledné křivky získané za pomoci vypočtených modelů budou nerozeznatelné od křivky vzniklé ze zvoleného modelu.

$$G_{real} = \frac{0,4}{0,2 \cdot s^2 + 0,2 \cdot s + 1} \quad (43)$$

**Výsledky:**

**PKS:**

$$G_{PKS} = \frac{0,4}{0,2 \cdot s^2 + 0,2 \cdot s + 1} \quad (44)$$

**MNČ:**

$$G_{MNC} = \frac{0,4}{0,2 \cdot s^2 + 0,2 \cdot s + 5} \quad (45)$$

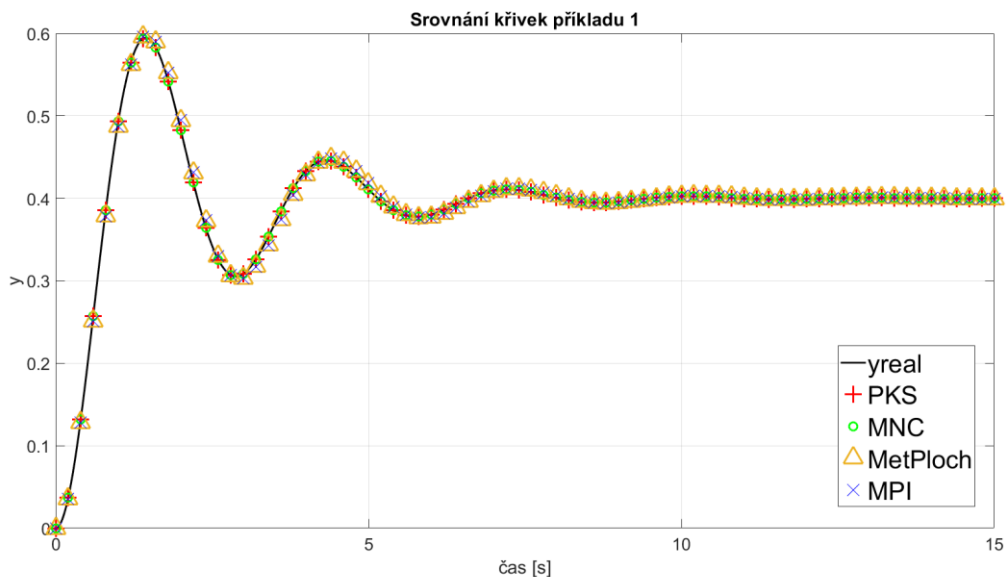
**MetPloch:**

$$G_{MP} = \frac{0,4}{0,2076 \cdot s^2 + 0,1992 \cdot s + 1} \quad (46)$$

**MPI:**

$$G_{MPI} = \frac{0,4}{0,2076 \cdot s^2 + 0,19924 \cdot s + 1} \quad (47)$$

(Jednotlivé výsledné vektory a matice budou uvedeny v přílohách)



Obr. 5: Srovnání původních dat s daty vypočtených modelů příkladu 1

Tabulka 2: Srovnání výsledků u příkladu 1

Metoda	Průměr shody (%)	Medián shody (%)	Nejmenší shoda (%)
PKS	99,9961	99,9962	99,9884
MNČ	99,9999	99,9999	99,9999
PLOCH:	99,3800	99,8524	96,3406
MPI	99,3794	99,8522	95,6452

Výsledná data splnila očekávání. Všechny čtyři křivky vzniklé z výpočtů zvolených identifikačních metod přesně kopírují zadanou křivku.

### 3.2.2 Příklad 2 - Příklad s vyšším kmitáním

Zadání druhého příkladu bylo zvoleno tak, že výsledná křivka je opět jednoduše kmitavá křivka, kterou dostaneme z vložení jednotkového vstupu do zvoleného modelu, který je opět modelem druhého řádu, bez nulových bodů a bez zpoždění. Jediný rozdíl od prvního případu je, že výsledná křivka je silněji kmitavá, než tomu bylo u případu 1. Bylo předpokládáno, že toto by mělo ovlivnit některé výpočty zvolených metod.

$$G_{real} = \frac{0,01429}{0,5714 \cdot s^2 + 0,01429 \cdot s + 1} \quad (48)$$

**Výsledky:**

**PKS:**

$$G_{PKS} = \frac{0,01429}{0,5715 \cdot s^2 + 0,01429 \cdot s + 1} \quad (49)$$

**MNČ:**

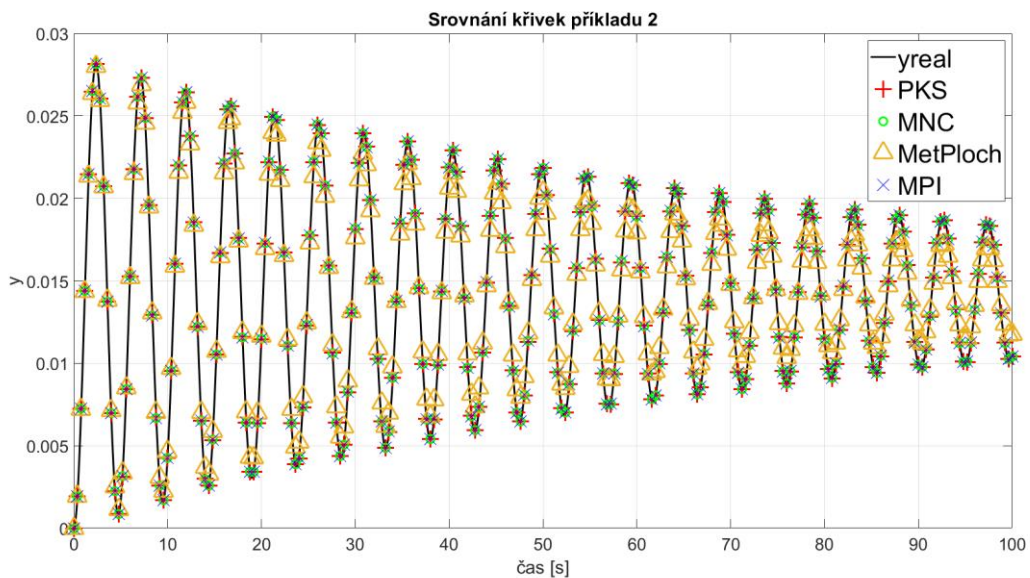
$$G_{MNC} = \frac{0,01429}{0,5714 \cdot s^2 + 0,01429 \cdot s + 1} \quad (50)$$

**MetPloch:**

$$G_{MP} = \frac{0,01429}{0,5716 \cdot s^2 + 0,01929 \cdot s + 1} \quad (51)$$

**MPI:**

$$G_{MPI} = \frac{0,01429}{0,5714 \cdot s^2 + 0,01429 \cdot s + 1} \quad (52)$$



Obr. 6: Srovnání původních dat s daty vypočtených modelů příkladu 2

Tabulka 3: Srovnání výsledků u příkladu 2

Metoda	Průměr shody (%)	Medián shody (%)	Nejmenší shoda (%)
PKS	99,9983	99,9999	99,8679
MNČ	99,9999	99,9999	99,9999
PLOCH:	99,8483	99,9999	66,3886
MPI	99,9991	99,9999	99,9365

Všechny čtyři metody dokázaly úspěšně popsat chování zadaného systému. U MPI a MetPloch bylo u kmitavějšího případu potřeba vytvořit měření na delším časovém úseku než u zbylých metod. Důvodem je, že tyto metody ke správnému výpočtu potřebují, aby se systém co nejvíce ustálil (aby zkonvergoval ke stabilní hodnotě).

Podstatnější rozdíl vypočtených dat MetPloch od zadaných dat vznikl již na počátku datových signálů. Proto jsme přesvědčeni, že metoda ploch je možným způsobem identifikace systémů podobných případu 2, ale ostatní metody jsou vhodnějšími k identifikaci podobných případů.

### 3.2.3 Příklad 3 – Příklad s jedním nulovým bodem

V třetím případě byl zadán model systému, který obsahuje nulový pól pro  $s=(-2)$ . Tento nulový pól by neměla být schopna brát v úvahu metoda PKS, MetPloch a MPI. Je možné nulový pól dodat do vzniklého modelu později, a tím navýšit u dané metody shodu se zadanými daty. Také je u metody MNČ potřeba upravit tvar výsledného modelu. Úprava je jednoduchá, ale pokud není provedena, tak může nastat situace, že se výsledná data nebudou shodovat s daty získanými z původního modelu.

$$G_{real} = \frac{0,2 \cdot s + 0,4}{0,2 \cdot s^2 + 0,2 \cdot s + 1} \quad (53)$$

**Výsledky:**

**PKS:**

$$G_{PKS} = \frac{0,2 \cdot s + 0,4}{0,1996 \cdot s^2 + 0,1998 \cdot s + 1} \quad (54)$$

**MNČ:**

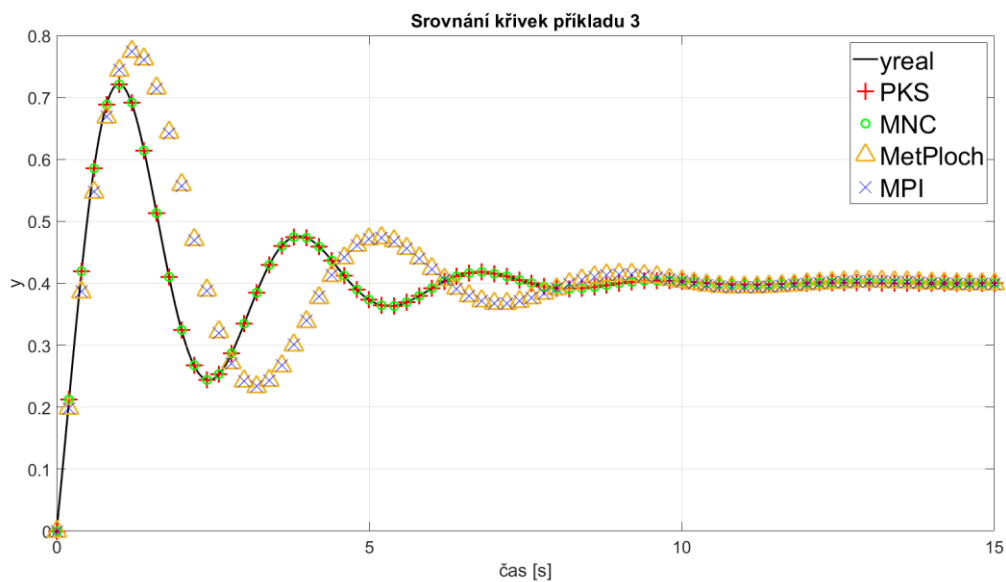
$$G_{MNC} = \frac{0,2 \cdot s + 0,4}{0,2 \cdot s^2 + 0,2 \cdot s + 1} \quad (55)$$

**MetPloch:**

$$G_{MP} = \frac{0,3555 \cdot s + 0,4}{0,36325 \cdot s^2 + 0,30075 \cdot s + 1} \quad (56)$$

**MPI:**

$$G_{MPI} = \frac{0,35616 \cdot s + 0,4}{0,36224 \cdot s^2 + 0,3012 \cdot s + 1} \quad (57)$$



Obr. 7: Srovnání původních dat s daty vypočtených modelů příkladu 3

Tabulka 4: Srovnání výsledků u příkladu 3

Metoda	Průměr shody (%)	Medián shody (%)	Nejmenší shoda (%)
PKS	99,9968	99,9974	99,9856
MNČ	99,9999	99,9999	99,9999
PLOCH:	90,9437	98,9649	23,1439
MPI	91,0064	98,9796	23,7441

V tomto případě měly nejmenší shodu metody MPI a MetPloch. I po dosazení nulových bodů do vzniklého modelu, se shoda modelů MetPloch a MPI významněji nezvýšila. Ze shora uvedeného důvodu bych tyto metody nedoporučoval pro výpočet systémů obsahující nulové póly.

Na rozdíl od MPI a MetPloch byla metoda PKS schopna po dosazení nulového pólu do modelu výrazně navýšit svojí procentuální shodu se zadaným systémem. Je tedy vhodná pro výpočet systémů s malým počtem nulových pólů, které lze následně dopočítat, aby měl model co největší shodu.

MNČ byla schopna pomocí funkce **tfest** dopočítat hodnoty nulových pólů, a to i u složitějších systémů vyšších řádů. Proto je nejvhodnější metodou pro výpočet podobných, ale i náročnějších příkladů.

### 3.2.4 Příklad 4 - Příklad se zpožděním signálu

Hlavním rozdílem případu 4, proti předešlým případům, spočívá v tom, že zadaný dopravní model obsahuje člen  $\exp(-3 \cdot s)$ . Tento člen vytváří časové zpoždění signálu. Jako v předešlém případě očekávám, že metoda PKS nebude schopna opoždění nijak určit a také jako v předešlém případě je potřeba upravit u metody MNČ funkci **tfest**. Tentokrát u funkce **tfest** je potřeba změnit 4 člen vstupu funkce, a to na přibližnou hodnotu opoždění signálu. Tato hodnota, lze zjistit empiricky, výpočtem, nebo pouze přibližně odhadem.

$$G_{real} = \frac{0,4}{0,2 \cdot s^2 + 0,2 \cdot s + 1} \cdot e^{-3 \cdot s} \quad (58)$$

**Výsledky:**

**PKS:**

$$G_{PKS} = \frac{0,4}{0,1996 \cdot s^2 + 0,1998 \cdot s + 1} \cdot e^{-3 \cdot s} \quad (59)$$

**MNČ:**

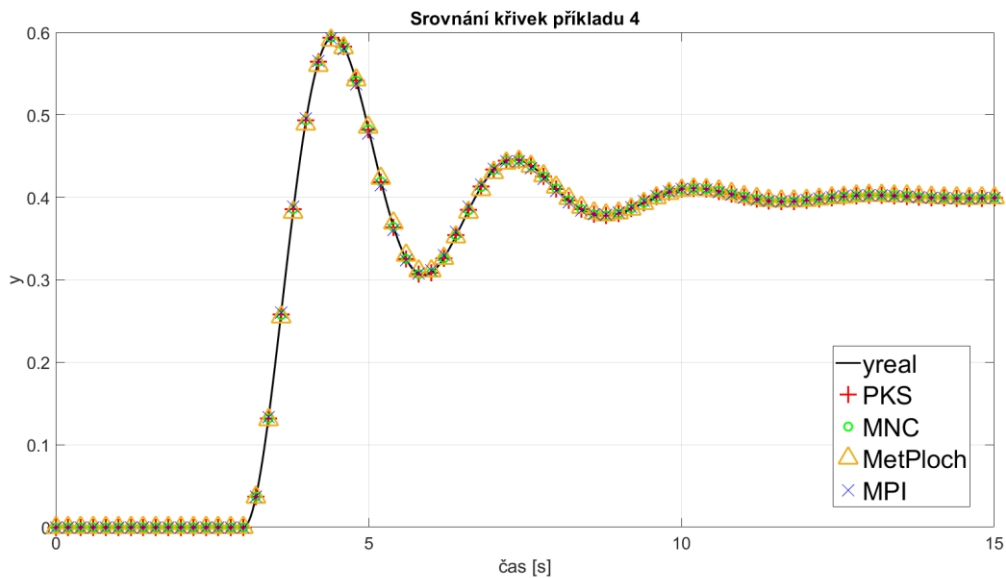
$$G_{MNC} = \frac{0,4}{0,2 \cdot s^2 + 0,2 \cdot s + 1} \cdot e^{-3 \cdot s} \quad (60)$$

**MetPloch:**

$$G_{MP} = \frac{0,4}{0,2018 \cdot s^2 + 0,205 \cdot s + 1} \cdot e^{-3 \cdot s} \quad (61)$$

**MPI:**

$$G_{MPI} = \frac{0,4}{0,1967 \cdot s^2 + 0,2003 \cdot s + 1} \cdot e^{-3 \cdot s} \quad (62)$$



Obr. 8: Srovnání původních dat s daty vypočtených modelů příkladu 4



Tabulka 5: Srovnání výsledků u příkladu 4

Metoda	Průměr shody (%)	Medián shody (%)	Nejmenší shoda (%)
PKS	99,9692	99,9951	99,7980
MNČ	99,9999	99,9999	99,9999
PLOCH:	99,7927	99,9682	98,6371
MPI	99,7561	99,9574	98,3154

V tomto případě byly všechny metody vhodné pro identifikaci systému.

### 3.2.5 Příklad 5 - Model 3. řádu

U případu 5 jsme chtěli zjistit, zda jsou zvolené metody schopné dostatečně napodobit systém vyššího řádu, proto jsme pro změnu zadali model 3. řádu. Opět bylo potřeba před začátkem výpočtu, změnit hledanou formu modelu u metody MNČ. Tentokrát byl změněn 2 prvek funkce třetího řádu na daný hledaný řád (v tomto případě na hodnotu 3).

Víme, že metoda PKS je schopna vytvořit model pouze 2. řádu, takže bylo očekáváno, že bude schopna pouze stěží napodobit systém vyššího řádu.

$$G_{real} = \frac{3,8536}{3,8536 \cdot 10^{-4} \cdot s^3 + 7,3218 \cdot 10^{-3} \cdot s^2 + 8,9788 \cdot 10^{-2} \cdot s + 1} \quad (63)$$

**Výsledky:**

**PKS:**

$$G_{PKS} = \frac{3,854}{0,005752 \cdot s^2 + 0,02278 \cdot s + 1} \quad (64)$$

**MNČ:**

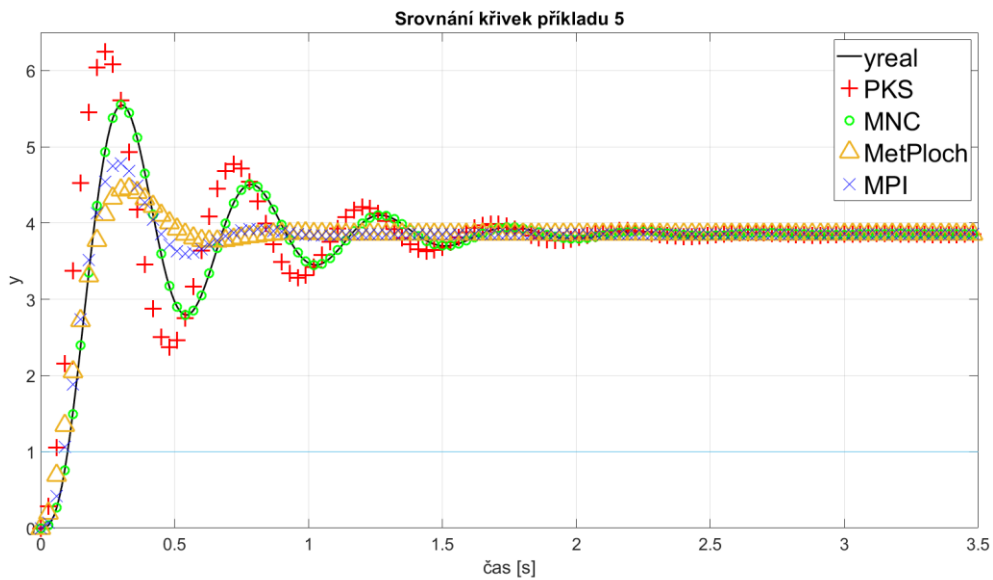
$$G_{MNC} = \frac{3,8536}{3,8536 \cdot 10^{-4} \cdot s^3 + 7,3218 \cdot 10^{-3} \cdot s^2 + 8,9788 \cdot 10^{-2} \cdot s + 1} \quad (65)$$

**MetPloch:**

$$G_{MP} = \frac{3,8536}{0,007794 \cdot s^2 + 0,0901 \cdot s + 1} \quad (66)$$

**MPI:**

$$G_{MPI} = \frac{3,8536}{1,8015 \cdot 10^{-4} \cdot s^3 + 7,4566 \cdot 10^{-3} \cdot s^2 + 0,08971 \cdot s + 1} \quad (67)$$



Obr. 9: Srovnání původních dat s daty vypočtených modelů příkladu 5

Tabulka 6: Srovnání výsledků u příkladu 5

Metoda	Průměr shody (%)	Medián shody (%)	Nejmenší shoda (%)
PKS	86,6161	99,7210	0
MNČ	99,9999	99,9999	99,9999
PLOCH:	91,3638	99,7235	0
MPI	96,0376	99,7251	0

Podle očekávání byla metoda PKS nejhorší metodou pro identifikaci systému vyššího řádu. Přesto byla schopna systém do určité míry napodobit.

MetPloch byla schopna vytvořit model 3. řádu, který dobře popisoval systém, ale tento model nekonvergoval ke stabilní hodnotě. Po snížení řádu byla sice shoda s daty zadaného systému menší pro kmitavou část, ale vzniklý model úspěšně konvergoval ke stabilní hodnotě.

MPI a MNČ se podařilo vytvořit model 3. řádu. Metodě MPI se ale nepodařilo dostatečně napodobit původní systém, proto bych tuto metodu přednostně používal pro výpočet systémů nižšího řádu.

### 3.2.6 Příklad 6 - Nekmitavý případ

V tomto případě bylo zjištěno, jakým způsobem budou metody reagovat na křivku, která nemá kmitavý průběh. Metoda PKS by neměla být schopna se dostat k jakémukoliv použitelnému tvaru modelu, jelikož k výpočtu potřebuje znát hodnoty amplitud kmitání systému.

$$G_{real} = \frac{8,3333 \cdot 10^{-2}}{8,3333 \cdot 10^{-3} \cdot s^2 + 0,1833 \cdot s + 1} \quad (68)$$

#### Výsledky:

##### PKS:

Metoda PKS nemohla být použita, jelikož lze použít pouze pro kmitavé systémy.

**MNČ:**

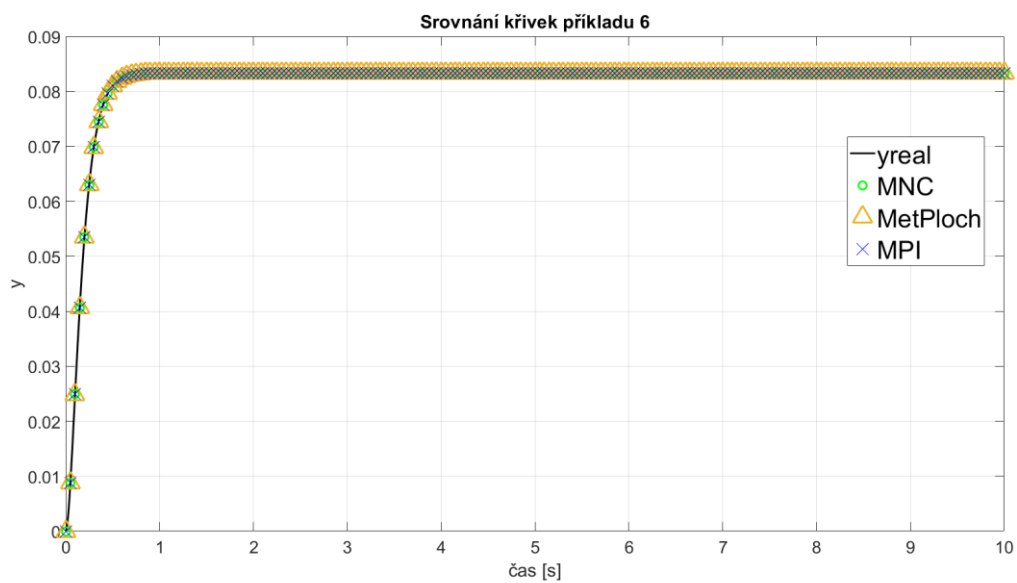
$$G_{MNC} = \frac{8,3333 \cdot 10^{-2}}{8,3333 \cdot 10^{-3} \cdot s^2 + 0,1833 \cdot s + 1} \quad (69)$$

**MetPloch:**

$$G_{MP} = \frac{0,08333}{0,008347 \cdot s^2 + 0,1834 \cdot s + 1} \quad (70)$$

**MPI:**

$$G_{MPI} = \frac{8,3333 \cdot 10^{-2}}{8,3333 \cdot 10^{-3} \cdot s^2 + 0,1833 \cdot s + 1} \quad (71)$$



Obr. 10: Srovnání původních dat s daty vypočtených modelů příkladu 6

Tabulka 7: Srovnání výsledků u příkladu 6

Metoda	Průměr shody (%)	Medián shody (%)	Nejmenší shoda (%)
PKS			
MNČ	99,9999	99,9999	99,9999
PLOCH:	99,9623	99,9999	98,5176
MPI	99,9999	99,9999	99,9982

Podle očekávání nebyla metoda PKS schopna vytvořit jakýkoliv model, který by byl schopen napodobit chování zadaného systému, neboť je stavěná jen na kmitavé systémy.

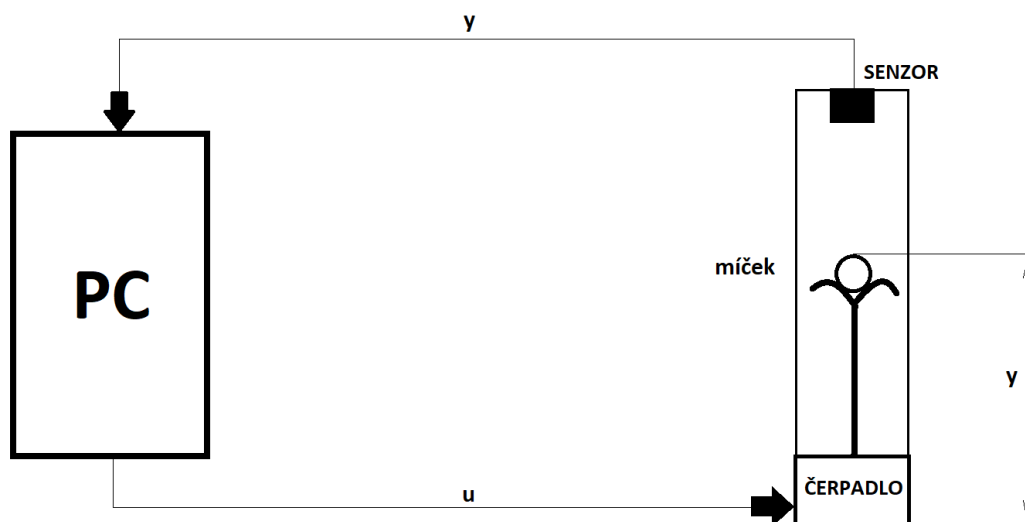
Zbylé tři metody byly schopné velice dobře napodobit chování systému.

## 4 VYUŽITÍ V PRAXI

### 4.1 Teorie praktického měření

Cílem experimentální části bylo ověřit, zda naprogramované metody dokáží zpracovat reálná data a vytvořit tak model daného systému.

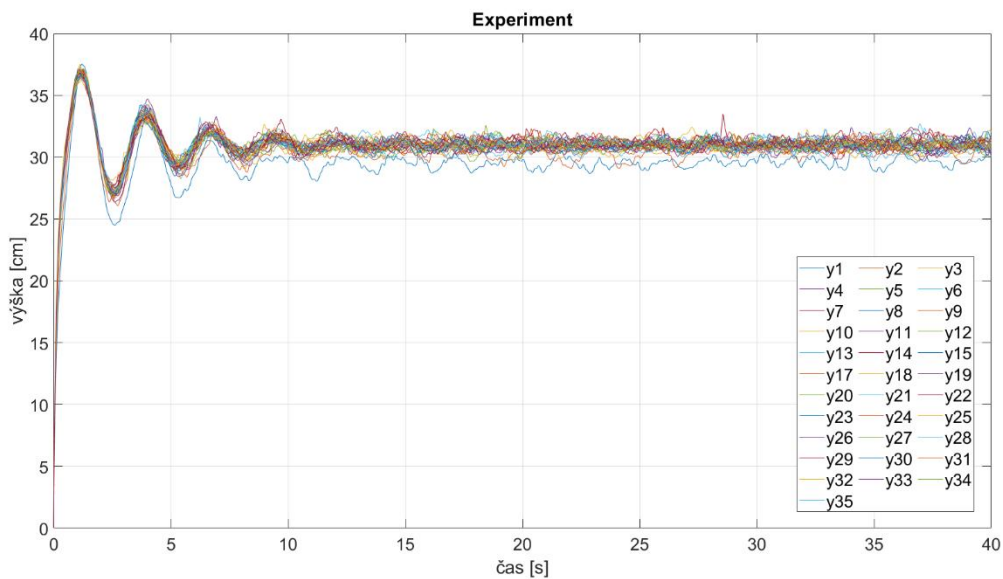
Reálná data jsem získal pomocí laboratorní úlohy Vodní levitace. Bylo třeba upravit model v MATLABu Simulink tak, aby výsledná data vykazovala kmitavý výstup v reakci na jednotkový vstup.



Obr. 11: Diagram experimentálního měření.

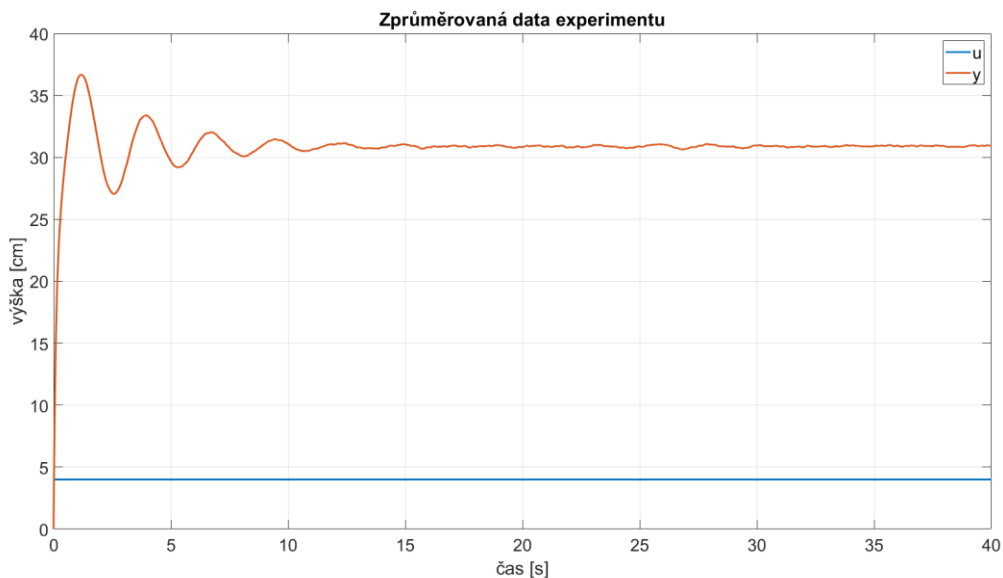
Vodní levitace se zabývá měřením polohy „míčku“, který pomocí vertikálního vodního paprsku z vodního čerpadla mění svou polohu. Míček je nadnášen a balancuje na vertikální ose tohoto paprsku. Na vrcholu měřicí trubice je usazen senzor, který měří vzdálenost míčku od senzoru. Pomocí zmíněného senzoru jsou zapisována výstupní data polohy míčku. Tato data jsou následně využita k určení modelu celého systému laboratorní úlohy.

K odstranění šumu v naměřených datech nebyl využit filtr, ale pouze jsem měření několikrát zopakoval. Následně data z jednotlivých měření byly zprůměrovány a tím byly jednotlivé rozdíly mezi daty odstraněny. Z důvodu dosažení nejlepšího průměru byly rovněž odstraněny nejrozdílnější výsledky měření, aby tak byl výsledný průměr co nejlepší. Díky tomu byl vzniklý šum natolik zmenšen, že ho bylo možné ve výpočtech dále zanedbat.



Obr. 12: Nezprůměrovaná naměřená data experimentu

Výsledná křivka má tlumený kmitavý průběh a již po několika vteřinách konverguje ke konstantní hodnotě.



Obr. 13: Zprůměrovaná data experimentu

Naměřená experimentální data z úlohy Vodní levitace byla následně nahrána ve formě vektorů do programového prostředí MATLAB, ve kterém byla dále zpracovávána.

## 4.2 Výsledky experimentální identifikace na reálné laboratorní úloze

PKS:

$$G_{PKS} = \frac{7,721}{0,1706 \cdot s^2 + 0,273 \cdot s + 1} \quad (72)$$

MNČ:

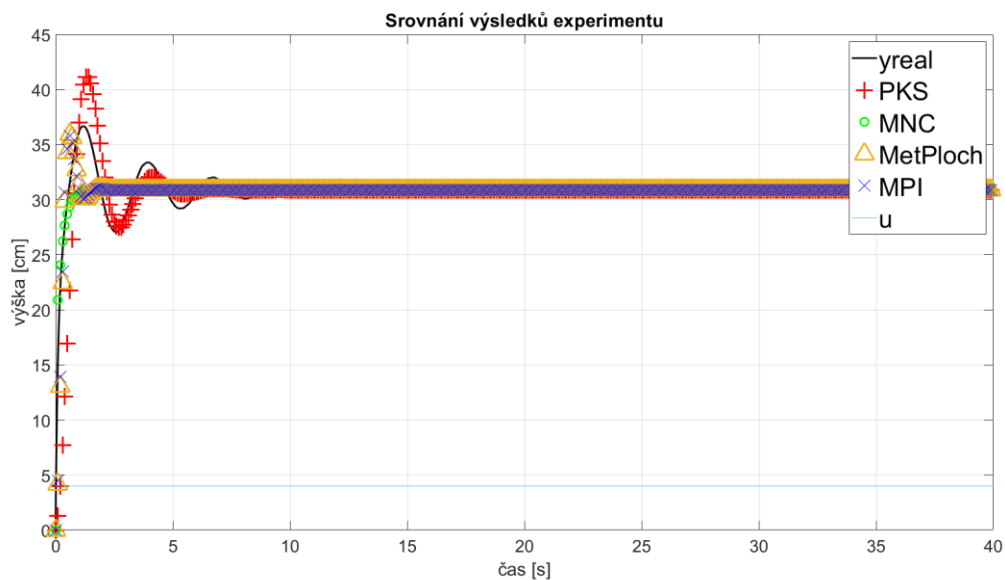
$$G_{MNC} = \frac{4,1051 \cdot s + 7,7398}{3,0543 \cdot 10^{-4} \cdot s^3 + 0,04582 \cdot s^2 + 0,6701 \cdot s + 1} \quad (73)$$

MetPloch:

$$G_{MP} = \frac{7,721}{0,03021 \cdot s^2 + 0,1727 \cdot s + 1} \quad (74)$$

MPI:

$$G_{MPI} = \frac{7,722}{0,02759 \cdot s^2 + 0,01678 \cdot s + 1} \quad (75)$$



Obr. 14: Srovnání původních dat experimentu s daty vytvořenými z vypočítaných modelů.



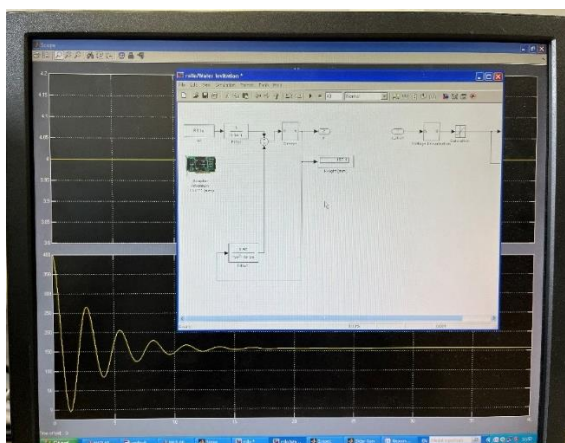
Tabulka 8: Srovnání výsledků praktického experimentu

Metoda	Průměr shody (%)	Medián shody (%)	Nejmenší shoda (%)
PKS	98,0143	99,7492	6,3387
MNČ	98,4745	99,6748	66,5263
PLOCH	98,4255	99,7479	30,2907
MPI	98,4371	99,7481	32,6734

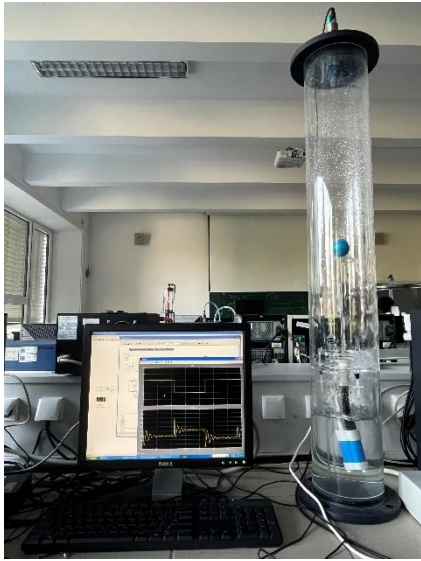
Všem metodám se podařilo vytvořit modely, které do určité míry dokáží popisovat chování systému. Podrobnějším zkoumáním bylo zjištěno, že nejlépe dokáže popsat systém experimentu MNČ, pokud dokáže vytvořit model vysokého řádu:

$$G_{MNC} = \frac{65,21 \cdot s^2 + 18,61 \cdot s + 397,8}{s^3 + 10,31 \cdot s^2 + 11,05 \cdot s + 51,52} \quad (76)$$

Jeho shoda s původními daty se blížila ke 100%, ale vzhledem k vysokému řádu tohoto modelu, byl raději výsledný řád modelu MNČ zvolen s pouze jedním nulovým pólem.



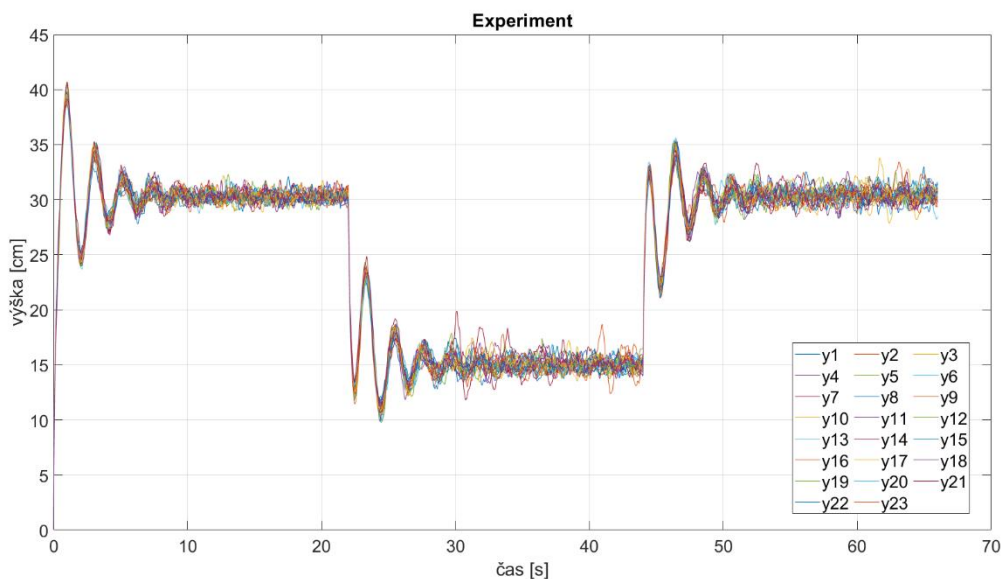
Obr. 15: Snímek PC obrazovky během praktického měření.



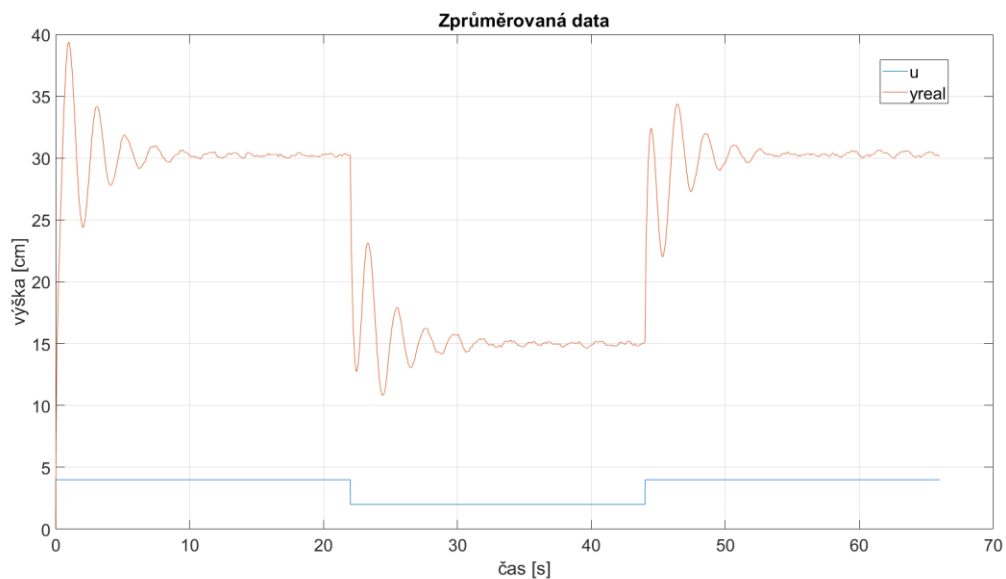
Obr. 16: Foto trubice obsahující míček s  
vodní tryskou

### 4.3. Srovnání výsledků

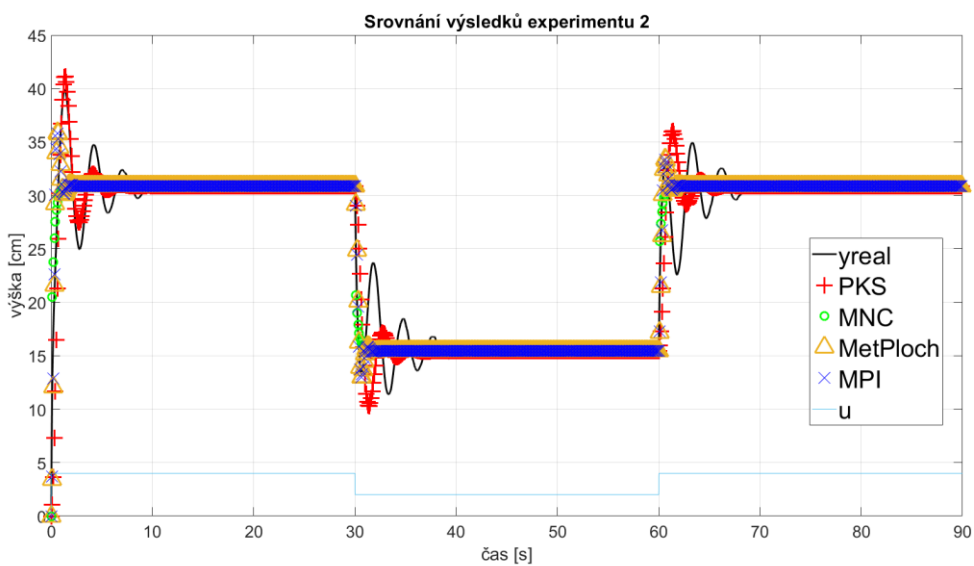
V této části byly využity vypočtené modely z předešlých identifikací experimentu Vodní levitace a snahou bylo použít je na reálném systému s proměnlivým vstupem. Tímto bylo ověřeno, která z metod je nejspolehlivější k popsání chování systému experimentu Vodní levitace.



Obr. 17: Naměřená data



Obr. 18: Zprůměrovaná data



Obr 19: Srovnání původních dat experimentu s daty vytvořenými z vypočtených modelů.

Tabulka 9: Srovnání výsledků praktického experimentu 2

<b>Metoda</b>	<b>Průměr shody (%)</b>	<b>Medián shody (%)</b>	<b>Nejmenší shoda (%)</b>
<b>PKS</b>	96,0010	99,1373	12,2759
<b>MNČ</b>	96,3133	99,0688	12,3088
<b>PLOCH</b>	96,4102	99,1396	47,5312
<b>MPI</b>	96,4046	99,1396	45,5766

Nejvhodnější metoda k popsání chování daného systému je MetPloch. Druhá je metoda MPI, třetí je MNČ a poslední je metoda PKS. Jelikož byl původní systém vyššího řádu a obsahoval vyšší množství nulových pólů, tak metody nebyly schopny chování systému dobře předvídat.

## 5 ZÁVĚR

V této bakalářské práci se podařilo ověřit využití matematických metod identifikace, a to metody nejmenších čtverců, metody aproximace přechodové charakteristiky s kmitavým průběhem, metody ploch a metody postupné integrace, jak na teoretických úlohách, tak i na datech získaných v praxi při experimentu.

### PKS

Metoda aproximace přechodové charakteristiky s kmitavým průběhem dokáže vytvářet modely pouze 2. řádu. Na rozdíl od metody ploch a metody postupné integrace lze dosazením nulových pólů do modelu, získaného metodou PKS, podstatně navýšit shodu s původním systémem, pokud původní systém obsahoval nulové póly. Metoda PKS se hodí pouze pro identifikaci konvergujících kmitavých systémů. Pro ostatní systémy nedokáže vytvořit model, a tak nemůže dojít ke srovnání dat. Při srovnání u teoretických příkladů byla metoda PKS 2. až 3. v pořadí. Tato metoda je vhodná pro kvalitní identifikaci pro systémy 2. řádu a přibližnou identifikaci systémů vyššího řádu.

### MNČ

Metoda nejmenších čtverců je metoda, kterou je možné využít, jak pro predikci chování kmitavých systémů, tak i pro predikci nekmitavých systémů. MNČ byla schopna ze všech identifikovatelných systémů získat model, který byl schopen původní systém dobře napodobit. MNČ byla podle výsledků srovnání jednotlivých teoretických případů nejlepší metodou k jejich identifikaci. Podle srovnání křivek v experimentální části v laboratorní úloze „Vodní levitace“ byla MNČ opět vyhodnocena jako nejvhodnější metoda, i když v experimentu 2 byla metoda nejmenších čtverců až 3 v pořadí, rozdíl byl pouze o přibližně 0,1 % za nejlepší metodou.

Velkou nevýhoda MNČ je, že je u této metody třeba znát finální tvar modelu daného systému. Pokud tento tvar znám není, je nutné měření několikrát zopakovat a tímto opakováním a měněním předpokládaného tvaru modelu se přiblížit k nejpřesnějšímu

tvaru modelu hledaného systému. Také není vhodné dělat model moc složitý, aby se zmírnil vliv šumu v získaných datech.

### **MetPloch**

Metoda ploch je podle výsledků jednou z méně vhodných metod pro identifikaci kmitavých soustav. V experimentální části dokázala napodobit chování systému, ale pouze po podstatnějším snížení šumu získaných dat. U teoretických příkladů byla většinou metodou s nejmenší shodou s původními naměřenými daty. Tato metoda je využitelná v praxi pro jednodušší systémy menšího řádu a ideálně bez nulových pólů.

MetPloch dokázala získat pro nekmitavou soustavu model, který dobře popisoval chování systému. Z tohoto důvodu si myslím že tato metoda je vhodnější pro získávání modelů u nekmitavých soustav.

Po srovnání naměřených dat s vypočítanými daty získanými metodou ploch bylo zjištěno že tato metoda je vhodná pro popis výstupu systémů.

### **MPI**

Metoda postupné integrace je metodou schopnou získat model vyššího řádu. Podobně jako metoda ploch má problém vytvořit model, který by správně popisoval chování systému, pokud původní systém obsahuje nulové póly.

MPI u teoretických příkladů nabývala 2 až 4 pozice a u experimentu byla metoda postupné integrace metodou s nejvyšší shodou. Ale podobně, jako tomu bylo u metody ploch, je potřeba co nejvíce odstranit šum ze získaných dat šum, aby MPI vytvořila co nejpřesnější model.

Podobně jako u metody ploch byla MPI schopna dobře predikovat chování systému u experimentu 2.

### **Celkové shrnutí**

Nejlepší metodou se ukázala být metoda nejmenších čtverců. Druhou nejlepší metodou byla metoda postupné integrace, třetí metoda v pořadí byla metoda aproximace přechodové charakteristiky s kmitavým průběhem a poslední byla metoda ploch.





## SEZNAM POUŽITÉ LITERATURY

- [1] M. Vlachová, "mechanické kmitání a vlnění," ed: Brno, 2006.
- [2] O. Lepil, *Mechanické kmitání a vlnění*. Praha: Prometheus, 1994.
- [3] M. Vrožina, Z. Jančíková, and J. David, *Identifikace systémů*, Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2012.
- [4] M. Hofreiter, Č. v. u. t. v. P. S. fakulta, Ed. *Identifikace systémů I*, Vyd. 1. ed. (no. Book, Whole). Praha: ČVUT, 2009.
- [5] G. P. Liu, *Nonlinear identification and control: a neural network approach*. Springer Science & Business Media, 2001.
- [6] I. Punčochář, "Učební text k předmětu Základy identifikace systémů, detekce chyb." Západočeská univerzita v Plzni: Katedra kybernetiky, 2016.
- [7] M. Witczak, *Modelling and estimation strategies for fault diagnosis of non-linear systems: from analytical to soft computing approaches*. Springer Science & Business Media, 2007.
- [8] P. Noskievič, *Modelování a identifikace systémů*, 1. vyd. ed. (no. Book, Whole). Ostrava: Montanex (in Czech), 1999.
- [9] V. Novák, *Základy fuzzy modelování*, 1. vyd ed. (no. Book, Whole). Praha: BEN (in Czech), 2000.
- [10] N. Oreskes, K. Shrader-Frechette, and K. Belitz, "Verification, validation, and confirmation of numerical models in the earth sciences," *Science*, vol. 263, no. 5147, pp. 641-646, 1994.
- [11] J. D. Sterman, "All models are wrong: reflections on becoming a systems scientist," *System Dynamics Review: The Journal of the System Dynamics Society*, vol. 18, no. 4, pp. 501-531, 2002.
- [12] M. A. Hamilton, "Model validation: an annotated bibliography," *Communications in Statistics-Theory and Methods*, vol. 20, no. 7, pp. 2207-2266, 1991.
- [13] V. Čížek, *Diskrétní Fourierova transformace a její použití*, 1. vyd. ed. (no. Book, Whole). Praha: Nakladatelství technické literatury, 1981.

- [14] J.W.Ju and J.C.Simo, „Strain- and stress-based continuum damage models—I. Formulation,“ *International Journal of Solids and Structures*, pp. 821-840, 18 únor 1987.
- [15] J. Matějka and V. Máša, "Metoda nejmenších čtverců v identifikaci dynamických systémů," *Bakalářská práce*, Fakulta strojního inženýrství, Ústav procesního a ekologického inženýrství, Vysoké učení technické v Brně, Brno, 2009.
- [16] H. L. Nguyen, P. Sibille, and H. Garnier, "A new bias-compensating least-squares method for identification of stochastic linear systems in presence of coloured noise," in *Proceedings of 32nd IEEE Conference on Decision and Control*, 15-17 Dec. 1993 1993, pp. 2038-2043 vol.3, doi: 10.1109/CDC.1993.325556.
- [17] Doc. Ing. Jan Soukup, *Identifikace soustav, SNTL - Nakladatelství technické literatury*, 1990.

# SEZNAM OBRÁZKŮ

OBR. 1: POSTUP IDENTIFIKACE A HLEDÁNÍ VHDNÉHO MATEMATICKÉHO MODELU [6] .....	10
OBR. 2: PŘECHODOVÁ ODEZVA KMITAVÉHO PROPORCIONÁLNÍHO ČLENU [4].....	17
OBR. 3: PŘECHODOVÁ ODEZVA PROPORCIONÁLNÍHO KMITAVÉHO SYSTÉMU S VYZNAČENÍM $y_{max}$ , $t_{max}$ , $y_{\infty}$ [4] ..	19
OBR. 4: JEDNOTKOVÝ SKOK VSTUPU .....	24
OBR. 1: SROVNÁNÍ PŮVODNÍCH DAT S DATY VYPOČTENÝCH .....	26
OBR. 2: SROVNÁNÍ PŮVODNÍCH DAT S DATY VYPOČTENÝCH MODELŮ .....	27
OBR. 3: SROVNÁNÍ PŮVODNÍCH DAT S DATY VYPOČTENÝCH MODELŮ .....	29
OBR. 4: SROVNÁNÍ PŮVODNÍCH DAT S DATY VYPOČTENÝCH MODELŮ .....	31
OBR. 5: SROVNÁNÍ PŮVODNÍCH DAT S DATY VYPOČTENÝCH MODELŮ .....	33
OBR. 6: SROVNÁNÍ PŮVODNÍCH DAT S DATY VYPOČTENÝCH MODELŮ .....	35
OBR. 8: DIAGRAM EXPERIMENTÁLNÍHO MĚŘENÍ. ....	37
OBR. 9: NEZPRŮMĚROVANÁ NAMĚŘENÁ DATA EXPERIMENTU .....	38
OBR. 10: ZPRŮMĚROVANÁ DATA EXPERIMENTU .....	38
OBR. 11: SROVNÁNÍ PŮVODNÍCH DAT EXPERIMENTU S DATY VYTVOŘENÝMI Z VYPOČÍTANÝCH MODELŮ. ....	39
OBR. 12: SNÍMEK PC OBRAZOVKY BĚHEM PRAKTICKÉHO MĚŘENÍ. ....	40
OBR. 14: FOTO TRUBICE OBSAHUJÍCÍ MÍČEK S VODNÍ TRYSKOU .....	41
OBR. 17: NAMĚŘENÁ DATA .....	42
OBR. 18: ZPRŮMĚROVANÁ DATA .....	43
OBR. 16: SROVNÁNÍ PŮVODNÍCH DAT EXPERIMENTU S DATY VYTVOŘENÝMI Z VYPOČTENÝCH MODELŮ. ....	43

## SEZNAM TABULEK

TABULKA 1: VLASTNOSTI ANALYTICKÉ A EXPERIMENTÁLNÍ IDENTIFIKACE [8] .....	11
TABULKA 2: SROVNÁNÍ VÝSLEDKŮ U PŘÍKLADU 1.....	26
TABULKA 3: SROVNÁNÍ VÝSLEDKŮ U PŘÍKLADU 2 .....	28
TABULKA 4: SROVNÁNÍ VÝSLEDKŮ U PŘÍKLADU 3 .....	30
TABULKA 5: SROVNÁNÍ VÝSLEDKŮ U PŘÍKLADU 4 .....	32
TABULKA 6: SROVNÁNÍ VÝSLEDKŮ U PŘÍKLADU 5 .....	34
TABULKA 7: SROVNÁNÍ VÝSLEDKŮ U PŘÍKLADU 6 .....	36
TABULKA 9: SROVNÁNÍ VÝSLEDKŮ PRAKTICKÉHO EXPERIMENTU .....	40
TABULKA 10: SROVNÁNÍ VÝSLEDKŮ PRAKTICKÉHO EXPERIMENTU 2.....	44

# PŘÍLOHY

Všechny přílohy bakalářské práce se nachází na přiloženém CD.

## Vývoj programu

### Implementace metody aproximace přechodové charakteristiky

#### s kmitavým průběhem (PKS) v MATLABu

Vzorkovaná data  $y$  jsou uvedena ve vektoru –  $A$ ,

$A\_1$  – odpovídá hodnotě amplitudy jedna  $A_1$ ,

$A\_2$  – odpovídá hodnotě amplitudy dva  $A_2$ ,

$A\_0$  – značí ustálenou hodnotu  $y_\infty$ ,

$A\_lokal\_max$  – množina všech lokálních maxim v Outputu,

$x1$  – označuje hodnotu času ve kterém se vyskytuje maximální amplituda,

$x2$  – označuje hodnotu času ve kterém je 2. maximální amplituda,

$\Omega$  – značí vlastní úhlovou frekvenci  $\Omega$ ,

$\theta$  – označuje parametr  $\theta$ ,

$\alpha$  – značí vlastní útlum  $\alpha$ ,

$\xi$  – označuje parametr  $\xi$ ,

$T_0$  – je parametr  $T_0$ ,

$T_p$  – je perioda  $T_p$  o kterou jsou vzdáleny amplitudy  $A_1$  a  $A_2$ ,

$S$  – označuje zesílení  $S$ ,

Nejprve je zapotřebí nalézt amplitudy  $A_1$  a  $A_2$ . Toho je docíleno tak, že jsou nalezena všechna lokální maxima

```
X=islocalmax(A);
```

z Outputu  $y$ , tyto lokální maxima jsou pomocí funkce:

```
A_lokal_max= sort((X.*A),'descend');
```

jsou seřazena podle velikosti od největšího k nejmenšímu a určil  $A_1$  a  $A_2$

```
A=yreal;
```

```
X=islocalmax(A);
```

```
procent_konce=10;
```

```
A_0=median(A(end-round(length(A)/procent_konce):end));
```

```
A_lokal_max= sort((X.*A),'descend');
```

```
    if max(A)==yreal(1)
        A_1_max=A_lokal_max(2);
        A_2_max=A_lokal_max(3);
        A_1=A_1_max-A_0
        A_2=A_2_max-A_0
    else
        A_1_max=max(A);
        A_2_max=A_lokal_max(2);
        A_1=A_1_max-A_0
        A_2=A_2_max-A_0
    End
```

Určení času  $t_1$  a získání periody  $T_p$ .

Pomocí for-cyklu najdeme hodnotu času  $x_1$  a  $x_2$  amplitudy  $A_1$  a  $A_2$ . Odečtením hodnoty  $x_1$  od hodnoty  $x_2$  dostaneme periodu  $T_p$ :

```
Y=X.*A;
i=1;
for i=1:length(Y)
    if Y(i)==A_1_max
        x1=i*dt;
    elseif Y(i)==A_2_max
        x2=i*dt;
    end
end
T_p=x2-x1;
```

Nakonec podle již zmíněných vzorců (v teorii) vypočteme vlastní úhlovou frekvenci Omega, tlumení alfa, parametru ksi, parametru theta, parametru  $T_0$  a statického zesílení S:

```
Omega=2*pi/T_p
theta=log(A_2/A_1)
alfa=delta/(2*pi)*Omega
```

$$\begin{aligned} \text{ksi} &= \text{delta} / \sqrt{\text{delta}^2 + (2 * \text{pi})^2} \\ \text{T}_0 &= \text{T}_p / (2 * \text{pi}) * \sqrt{1 - \text{ksi}^2} \\ \text{S} &= \text{A}_0 / (\text{median}(\text{u}(\text{end} - \text{round}(\text{length}(\text{u}) / \text{procent\_konce}) : \text{end}))) \end{aligned}$$

#### 4.1.2 Implementace metody nejmenších čtverců (MNČ) v MATLABu

1) Určení tvaru předpokládaného výsledku modelu pomocí příkazu:

**Gest = tfest(data, 2, 0, 0);**

V tomto příkazu druhý člen odpovídá počtu pólů. Třetí člen odpovídá počtu nulových bodů. Čtvrtý člen odpovídá zpoždění signálu systému.

**tfest(data,np,nz)** – odhaduje přenosovou funkci ve spojitém čase obsahující **nz** nulových bodů a **np** pólů, za pomoci metody least squares.

(Inicializované parametry jsou aktualizovány pomocí nelineární metody nejmenších čtverců)

2) Porovnání předpokládaného modelu se získanými daty a získání finálního modelu

**opt=compareOptions;** – možnosti srovnání dat

**opt** - Sada možností obsahující zadané možnosti pro funkci **compare**,

**compareOptions** – vytvoří výchozí možnosti nastavené pro funkci **compare**,

**compare** – simuluje odezvu dynamického modelu systému a překrývá tuto odezvu přes naměřená data. Graf také zobrazuje míru normalizované střední kvadratické hodnoty dobré shody mezi simulovanou odezvou a naměřenými daty:

**opt.InitialCondition='z';**

**compare(data, Gest, opt);**

#### Implementace metody ploch v MATLABu

Výpočet zesílení  $K$  z dat.

Pro získání přibližné hodnoty  $y_\infty$  a hodnoty  $u_\infty$  ve kmitavém případě bylo vzato několik (V našem případě 200) posledních naměřených dat pro signály **yreal** a **u** a vypočteme jejich medián. Tyto nové hodnoty nazveme **y\_infty** a **u\_infty**. Jejich poměrem získáme hodnotu zesílení **K**:

**y\_infty=median(yreal(end-200:end));**

**u\_infty=median(u(end-200:end))**

**K=y\_infty/u\_infty;**

Následně datové vzorky **yreal** a **u** vydělíme hodnotou **K**.

Výpočet koeficientů polynomu:

Nejdříve vypočteme hodnoty momentů pomocí integrálu funkce:

$$M_i = \int_0^{\infty} \frac{[1 - yreal(t)] \cdot (-t)^i}{i!} d(t)$$

V případě programu je hodnota **t** nahrazena hodnotami **x1(i)** a **x2(i)**. A integrál je řešen pomocí obdélníkové metody. Hodnota **i** je nahrazena hodnotou **k**. String **Vysleny\_integral** označuje vypočtené hodnoty momentů **M<sub>1</sub>** až **M<sub>i</sub>** pro **i** rovno 1 až **L** (**L** rovno počet koeficientů polynomu).

$$\text{Fun1(i)}=(1-yreal(i))*(-x1(i))^k/factorial(k);$$

$$\text{Fun2(i)}=(1-yreal(i))*(-x2(i))^k/factorial(k);$$

$$\text{Vysledny\_integral(k+1)}=(S1+S2)/2$$

Následně pomocí momentů **M<sub>i</sub>** dopočteme hodnoty koeficientů polynomu **a<sub>1</sub>** až **a<sub>i</sub>** pro **i** rovno 1 až **L** (**L** rovno počet koeficientů polynomu). V našem případě bude počet prvků polynomu **L** roven 4.

Koeficienty **a<sub>i</sub>** budou nahrazeny parametry **a<sub>i</sub>**.

$$a_1=\text{Vysledny\_integral(1)}$$

$$a_2=S_1*\text{Vysledny\_integral(1)}+\text{Vysledny\_integral(2)}$$

$$a_3=S_2*\text{Vysledny\_integral(1)}+S_1*\text{Vysledny\_integral(2)}+$$

$$\text{Vysledny\_integral(3)}$$

$$a_4=S_3*\text{Vysledny\_integral(1)}+S_2*\text{Vysledny\_integral(2)}+S_1*\text{Vysledny\_integral(3)}+\text{Vysledny\_integral(4)}$$

## Implementace MPI v MATLABu



Určíme počet  $N$  a krok  $\mathbf{delta\_T}$  datových prvků  $\mathbf{yreal}$  a  $\mathbf{u}$ . Vytvoříme nulové matice  $\mathbf{Y}$  a  $\mathbf{U}$  o rozměrech  $(4,N)$  a do první řady pomocí **for**-cyklu nahrajeme data  $\mathbf{yreal}$  a  $\mathbf{u}$ .

```

N=length(yreal);
delta_T=dt;
Y=zeros(4,N);
U=zeros(4,N);

for i=(1:N)
Y(1,i)=yreal(i);
U(1,i)=u(i);
end

```

Vytvoříme **for** cyklus pro hodnoty  $\mathbf{j}=[2,3,4]$ . Pomocí tohoto cyklu budeme později vytvářet a ukládat nové hodnoty integrací.

```

for j=[2,3,4]

```

Vypočteme rozdíly  $\mathbf{X(i)}$  pro ustálenou hodnotu  $\mathbf{Y(j-1,N)}$  a pro jednotlivé hodnoty  $\mathbf{Y(j-1,i)}$ .

```

for i=(1:N)
X(i)=Y(j-1,N)-Y(j-1,i);
end

```

Provedeme lichoběžníkovou integraci hodnot  $\mathbf{X(i)}$  a výsledné hodnoty nazveme  $\mathbf{V(i)}$ .

```

for i=(2:N)
V(i)=delta_T*((X(1)+X(i))/(2)+S);
S=S+X(i);
end

```

Nahrajeme vypočtené hodnoty  $\mathbf{V(i)}$  do matice  $\mathbf{Y(j,i)}$ .

```

for i=(1:N)
Y(j,i)=V(i);
end

```

Opět vypočteme rozdíly  $\mathbf{X(i)}$ , ale tentokrát pro ustálenou hodnotu  $\mathbf{U(j-1,N)}$  a pro jednotlivé hodnoty  $\mathbf{U(j-1,i)}$  a opět provedeme lichoběžníkovou integraci hodnot  $\mathbf{X(i)}$  a výsledné hodnoty nazveme  $\mathbf{V(i)}$ . Ukončíme **for**-cyklus pro hodnoty  $\mathbf{j}=[2,3,4]$ .

```

    for i=(1:N)
    X(i)= U(j-1,N)-U(j-1,N);
    end
    for i=(1:N)
    U(j,i)=V(i);
    end
    end

```

Nyní vypočteme koeficienty  $a_j$  ( $a_j$ ) pro hodnoty  $j=[0,1,2,3]$ .

```

    a_0=U(1,N)/Y(1,N)
    a_1=(a_0*(Y(2,N)))/((Y(1,N)))
    a_2=-((a_0*(Y(3,N))-a_1*(Y(2,N)))/((Y(1,N))))
    a_3=((a_0*(Y(4,N))-a_1*(Y(3,N))-a_2*(Y(2,N)))/((Y(1,N))))

```

```

clc;
close all;

for KKK=(1:4)
    format long
clearvars -except DD KKK G_MND G_propor G_MetPloch G_MPI;

```

Vložené vstupní data

```

dt =1/100;
q=40%do kolika vteřin [" časových jednotek "] počítáme data

```

```

Haha=1%input("Vložte typ dat který se řeší, 1= vzorec, 2= nahraná data")

```

```

if Haha==1
    metoda=3%input("A) pólové a nulové body = '1', B) generování čitatele a
jmenovatele = '2',C) napsaný polynom = '3'")

```

a)

```

    if metoda == 1
z = [ ]; % Zeros
p = [ -2+13i -2-13i -15]; % Poles
k = 10; % Gain
Greal = zpk(z,p,k)

```

b)

```

elseif metoda == 2
s = tf('s')

num = [ 1 0 ]; % Numerator: s
den = [ 8 -69 420 ]; % Denominator: s^2 + 2 s + 10
Greal = tf(num,den)

```

## Model

```

elseif metoda == 3
s = tf('s')
Ts=3;

Greal=(65.21*s^2+70.48*s+447.5*1)/(s^3+10.31*s^2+11.05*s+51.52)/34.6688*30.8
175%((40*s^2+1*s+70))%*exp(-Ts*s)

end

```

Zvolení kroku systému

```

t=0:dt:q;
u=4*ones(length(t),1);
%u(1:1/dt)=0;

```

```

yreal=lsim(Greal,u,t);

```

```

elseif Haha ==2
    Q=1%input("data z experimentu = '1' a data ze simulinku = '2' ")
    if Q == 1
        a=importdata("Baka/Vodni_levitace/Real/Data_vodni_levitace_2.mat");
        A(1,:)=a.U
        A(2,:)=a.Y

        dt=0.01%40/(2*length(A)-2)
        t=(0:dt:(40-dt));
        yreal=transpose(A(2,1:end-1));
        yreal=transpose([transpose(yreal)])
        %yreal(1201:end,1)=345.64*ones(2800,1)
        plot(t,yreal)
        u=transpose(A(1,1:end-1));
        u=transpose([transpose(u)]% max(u)*ones(1,100*length(u)))
    elseif Q==2
        A=out.Simulink;
        u=A(1:end,1);
        yreal=A(1:end,2);
        T=size(u)
        i=0;
        t=zeros(1:T(1))
        while i < T(1)
            i=i+1;
            t(i)=i;
        end
        dt=0.05
    end
end
end

```

```

plot(t, [u,yreal], 'LineWidth',2)
%axis ([0 q -0.1 0.1 ])
grid on
legend (['u';'y'])
DD(1:length(yreal),1)=yreal;

```

```
if KKK==1
    Y_last_zero=0;
```

PKS

Zpoždění

```
N=length(yreal)
```

```
Yreal=0;
TTTT=0;
i=2;
FIN=0;
while i<N-5 && FIN<1
    if yreal(i-1)==0 && yreal(i)==0 && yreal(i+1)~=0 && yreal(i+2)~=0 &&
yreal(i+1)~=0
        TTTT=i;
        TTT=N;
        FIN=1
        Yreal=zeros(N-TTTT,1);
        Y_original=yreal;
        for J=(TTTT:N-1)
            Yreal(J+1-TTTT)=yreal(J);
        end
        yreal=Yreal;
        N=N-TTTT
    end
    i=i+1;
end

s=0;
i=1;

while i<length (yreal)-2 && s<1
    if u(i)~=0 && yreal(i)==0 && yreal(i+1)==0 && yreal(i+2)==0
        Y_original=yreal;
        Y_delay_zero=i;

        for Q=(i:round(length(yreal)/2))

            if yreal(Q)==0 && yreal(Q+1)~=0 && yreal(Q+2)~=0

                Y_last_zero=Q;

                for E=(Q:length(yreal))
                    Y_real(E+1-Q)=yreal(E);
                end
                Y_real=[transpose(yreal(1:Y_delay_zero)) Y_real];
                Y_zeros=Y_last_zero-Y_delay_zero;
                yreal=transpose(Y_real);
```

```

        end
    end

    s=s+1;

    end
    i=i+1;
end

A=yreal;

X=islocalmax(A);
procent_konce=10;
A_0=median(A(end-round(length(A)/procent_konce):end))%(sum (A(end-
round(length(A)/10):end)))/round(length(A)/10)
A_0 =
    30.884182821202771

```

```

A_lokal_max= sort((X.*A), 'descend');

if max(A)==yreal(1)
    A_1_max=A_lokal_max(2);
    A_2_max=A_lokal_max(3);

    A_1=A_1_max-A_0
    A_2=A_2_max-A_0
else
    A_1_max=max(A);
    A_2_max=A_lokal_max(2);

    A_1=A_1_max-A_0
    A_2=A_2_max-A_0
end

```

```

Y=X.*A;
i=1;

for i=1:length(Y)
    if Y(i)==A_1_max
        x1=i*dt
    elseif Y(i)==A_2_max
        x2=i*dt
    end
end

```

```

    end
end

T_p=x2-x1

Omega=2*pi/T_p %vlastní úhlová frekvence

delta=log(A_2/A_1)

alfa=delta/(2*pi)*Omega %tlumení

ksi=delta/sqrt(delta^2+(2*pi)^2)

T_0=T_p/(2*pi)*sqrt(1-ksi^2)

S=A_0/(median(u(end-round(length(u)/procent_konce):end)))

```

```

s = tf('s')

G_ss=(S)/((T_0^2*s^2-2*ksi*T_0*s+1)) % pro 0<ksi<1
G_propor=G_ss;
if TTTT~=0
    yreal=Y_original;
    G_propor=G_propor*exp(-t(TTTT)*s)
end
data = iddata(yreal, u, dt);

```

```

yyreal=lsim(G_propor,u,t);
if Y_last_zero~=0
    yyreal=transpose([zeros(1,Y_last_zero)
transpose(yyreal(1:length(yyreal)-Y_last_zero))]);
end
plot(t, yyreal, 'LineWidth',2)
hold on
plot(t, yreal)
opt = compareOptions;
opt.InitialCondition = 'z';
compare (data,G_ss,opt);
set (findall (gca, 'Type','Line'),'LineWidth',2)
grid on
DD(1:length(yyreal),2)=yyreal
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if KKK == 2

```

Choose model structure

```
%alespoň/minimálně druhého stupně  
%přenos zpoždění
```

Identifikovat model aby vyhovaval datům

```
data = iddata(yreal, u, dt);
```

## MNČ

```
Gest = tfest(data, 3, 1, 0) %Druhý člen naznačuje počet pólů. Třetí člen  
naznačuje počet nulových bodů. Čtvrtý naznačuje zpoždění signálu systému
```

```
opt=compareOptions; %možnosti srovnání dat  
opt.InitialCondition='z';  
compare(data, Gest, opt);  
  
set (findall (gca, 'Type', 'Line'), 'Linewidth', 2)  
grid on  
yy=lsim(Gest,u,t);  
DD(1:length(yy),3)=yy  
G_MND=Gest;  
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## Metoda ploch

```
if KKK == 3
```

Zpoždění

Zpoždění

```
N=length(yreal)  
Yreal=0;  
TT=0;  
i=2;  
FIN=0;  
while i<N-5 && FIN<1  
    if yreal(i-1)==0 && yreal(i)==0 && yreal(i+1)~=0 && yreal(i+2)~=0 &&  
yreal(i+1)~=0  
        TT=i;  
        TTT=N;  
        FIN=1  
        Yreal=zeros(N-TT,1);
```



```

        Y_original=yreal;
        for J=(TT:N-1)
            Yreal(J+1-TT)=yreal(J);
        end
        yreal=Yreal;
        N=N-TT
    end
    i=i+1;
end

```

N =  
4001

Začátek

```

KK=1;
    %XXX=max(yreal)
    %yreal=yreal./XXX
y_infty=median(yreal(end-200:end));
u_infty=median(u(end-200:end))

```

```

K=y_infty/u_infty;
if round(K,1)~=1
    Y=yreal;
    KK=K

    yreal=yreal/KK
    plot(t(1:length(yreal)),yreal)

    y_infty=median(yreal(end-200:end))
    u_infty=median(u(end-200:end))
    K=y_infty/u_infty
end
i=0;
S=0;
for k=[0 (1:3)]
    S1=0;
    S2=0;
    pocatek=0;%%od
    konec=q;%%do
    if TT~=0
        konec=q-t(TT);%%do
    end
    Pocet_kroku=length(yreal);%%počet kroků co největší

    krok=(konec-pocatek)/(Pocet_kroku);%%určení kroku

    i=1;
    l=pocatek;
    while l<konec;

```

```

        x1(i)=pocatek+krok*(i-1);
        l=x1(i);
        i=i+1;
    end
    [nepotrebna_hodnota,Pocet_kroku1]=size(x1);

    i=1;
    for i=1:Pocet_kroku1-2
        Fun1(i)=(K-yreal(i))*(-x1(i))^k/factorial(k); %% [S x1(i)!!!]sem
        zadan_funkci 'f(x)' nebo hodnotu 'y(x)'
        i=i+1;
    end

    S1=krok*sum(Fun1(1,1:(Pocet_kroku1-2)));

    i=1;
    l=pocatek+krok/2;
    while l<konec
        x2(i)=l;
        i=i+1;
        l=pocatek+krok*(1+2*(i-1))/2;
    end
    [nepotrebna_hodnota,Pocet_kroku2]=size(x2);

    i=1;
    for i=1:Pocet_kroku2
        Fun2(i)=(K-yreal(i))*(-x2(i))^k/factorial(k); %% [S x2(i)!!!]sem
        zadan_funkci 'f(x)' nebo hodnotu 'y(x)'
        i=i+1;
    end
    plot(t(1:length(yreal)-1 ),Fun1)
    hold on
    S2=krok*sum(Fun2(1,1:Pocet_kroku2));
    Vysledny_integral(k+1)=(S1+S2)/2
    if k==0
        S_1=Vysledny_integral(1)
        a_1=S_1/K
    elseif k==1
        S_2=S_1*Vysledny_integral(1)+Vysledny_integral(2)
        a_2=S_2/K^2
        S_2=S_2
    elseif k==2
        S_3=(S_2*Vysledny_integral(1)+S_1*Vysledny_integral(2)+Vysledny_integral(3))
        a_3=S_3/K^4
    elseif k==3
        S_4=S_3*Vysledny_integral(1)+S_2*Vysledny_integral(2)+S_1*Vysledny_integral(
        3)+Vysledny_integral(4)
        a_4=S_4/K^8
    end

```

```

end

end
hold off

if S_1>=0

rad_modelu=0;
if S_1>0 && S_2<0
    rad_modelu=1
elseif S_1>0 && S_2>0 && S_3<0
    rad_modelu=2
elseif S_1>0 && S_2>0 && S_3>0&& S_4<0
    rad_modelu=3
elseif S_1>0 && S_2>0 && S_3>0&& S_4>0
    rad_modelu=4
else
    disp("EROOR!")
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if rad_modelu==1
    s = tf('s')

    num = [ K ];          % Numerator: s
    den = [ a_1 1 ];      % Denominator: s^2 + 2 s + 10
    G_MetPloch = tf(num,den)
elseif rad_modelu==2
% s = tf('s')
%
%   num = [ K ];          % Numerator: s
%   den = [ a_2 a_1 1 ];  % Denominator: s^2 + 2 s + 10
%   G_MetPloch = tf(num,den)
%
%   s = tf('s')
%
%   num = [ K ];          % Numerator: s
%   den = [ a_2 a_1 1 ];  % Denominator: s^2 + 2 s + 10
%   G_MetPloch2 = tf(num,den)
%
    s = tf('s')

    num = [ K ];          % Numerator: s
    den = [ a_2 a_1 1 ];  % Denominator: s^2 + 2 s + 10
    G_MetPloch = tf(num,den)

elseif rad_modelu==3
    s = tf('s')
    a_2=a_2

```

```

    num = [ K ];          % Numerator: s
    den = [ a_3 a_2 a_1 1]; % Denominator: s^2 + 2 s + 10
    G_MetPloch = tf(num,den)
elseif rad_modelu==4
    s = tf('s')

    num = [ K ];          % Numerator: s
    den = [a_4 a_3 a_2 a_1 1]; % Denominator: s^2 + 2 s + 10
    G_MetPloch = tf(num,den)
end

else
    disp("V základním vzorci jsou záporné nulové póly")
    S_1=-S_1;
    a_1=-a_1;
    rad_modelu=0;
    if S_1>0 && S_2<0
        rad_modelu=1
    elseif S_1>0 && S_2>0 && S_3<0
        rad_modelu=2
    elseif S_1>0 && S_2>0 && S_3>0&& S_4<0
        rad_modelu=3
    elseif S_1>0 && S_2>0 && S_3>0&& S_4>0
        rad_modelu=4
    else
        disp("EROOR!")
    end

    if rad_modelu==1
        s = tf('s')

        num = [ K ];          % Numerator: s
        den = [ a_1 1 ];      % Denominator: s^2 + 2 s + 10
        G_MetPloch = tf(num,den)
        G_MetPloch = G_MetPloch+1*s*G_MetPloch
    elseif rad_modelu==2

        s = tf('s')

        num = [ K ];          % Numerator: s
        den = [ a_2 a_1 1 ]; % Denominator: s^2 + 2 s + 10
        G_MetPloch = tf(num,den)
        G_MetPloch = G_MetPloch*(0.5*s+1)

    elseif rad_modelu==3
        s = tf('s')
        a_2=a_2
        num = [ K ];          % Numerator: s
        den = [ a_3 a_2 a_1 1]; % Denominator: s^2 + 2 s + 10
        G_MetPloch = tf(num,den)

```

```

    G_MetPloch = G_MetPloch*(s+1)
elseif rad_modelu==4
    s = tf('s')

    num = [ K ];          % Numerator: s
    den = [a_4 a_3 a_2 a_1 1]; % Denominator: s^2 + 2 s + 10
    G_MetPloch = tf(num,den)
    G_MetPloch = G_MetPloch*(s+1)
end
end

```

Dodání zpoždění

```

if round(KK,1)~=1
    yreal=Y;
end

if TT~=0
    yreal=Y_original;
    G_MetPloch =G_MetPloch *exp(-t(TT)*s)
end
y_MetPloch =lsim(G_MetPloch ,u,t);
if round(KK,1)~=1
    y_MetPloch =y_MetPloch *KK
    G_MetPloch=G_MetPloch *KK
end
axis([0 250 -1000 1000])
hold off
plot(t,y_MetPloch )
hold on
plot(t(1:length(yreal)),yreal,'LineWidth',2)
legend("a_{2}", "yreal")
axis([0 200 0 1])
hold off
X =1-abs(yreal-y_MetPloch)./yreal;
mean (X (2+TT:end))

```

```

plot(t(1+TT:end),X (1+TT:end))
axis([0 200 0 1])

plot(t(1+TT:end),X(1+TT:end))
DD(1:length(y_MetPloch),4)=y_MetPloch
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

## MPI

```

if KKK == 4
N=length(yreal);

```

```

delta_T=dt;
t=(0:length(yreal)-1)*dt;

```

Zpoždění

```

Yreal=0;
TT=0;
i=2;
FIN=0;
while i<N-5 && FIN<1
    if yreal(i-1)==0 && yreal(i)==0 && yreal(i+1)~=0 && yreal(i+2)~=0 &&
yreal(i+1)~=0
        TT=i;
        TTT=N;
        FIN=1
        Yreal=zeros(N-TT,1);
        Y_original=yreal;
        for J=(TT:N-1)
            Yreal(J+1-TT)=yreal(J);
        end
        yreal=Yreal;
        N=N-TT
    end
    i=i+1;
end
Y=zeros(4,N);
U=zeros(4,N);
for i=(1:N)
    Y(1,i)=yreal(i);
    U(1,i)=u(i);
end
for j=[2,3,4]
    X=0;
    for i=(1:N)
        X(i)=Y(j-1,N)-Y(j-1,i);
    end
end

```

Integrace dodělat

```

V(1)=0;
S=0;
for i=(2:N)
    V(i)=delta_T*((X(1)+X(i))/(2)+S);
    S=S+X(i);
end

for i=(1:N)
    Y(j,i)=V(i);
end
for i=(1:N)
    X(i)= U(j-1,N)-U(j-1,i);
end

```

```
end
end
```

integrace dodělat

```
V(1)=0;
S=0;
for i=(2:N)
    V(i)=delta_T*((X(1)+X(i))/(2)+S);
    S=S+X(i);
end
for i=(1:N)
    U(j,i)=V(i);
end
plot(t(1+TT:end),Y)
a_0=U(1,N)/Y(1,N)
a_1=(a_0*(Y(2,N)))/((Y(1,N)))
a_2=-(a_0*(Y(3,N))-a_1*(Y(2,N)))/((Y(1,N)))
a_3=(a_0*(Y(4,N))-a_1*(Y(3,N))+a_2*(Y(2,N)))/((Y(1,N)))
Y(1,N)
Y(2,N)
Y(3,N)

U(1,N)

rad_modelu=0;
if a_1>0 && a_2<0
    rad_modelu=1
elseif a_1>0 && a_2>0 && a_3<0
    rad_modelu=2
elseif a_1>0 && a_2>0 && a_3>0
    rad_modelu=3
else
    disp("ERROR!")
end
rad_modelu=2%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if rad_modelu==1
    s = tf('s')
    num = [ 1];           % Numerator: s
    den = [a_1 a_0];      % Denominator: s^2 + 2 s + 10
    G_MPI = tf(num,den)
elseif rad_modelu==2
    s = tf('s')
    num = [ 1];           % Numerator: s
    den = [a_2 a_1 a_0];  % Denominator: s^2 + 2 s + 10
    G_MPI = tf(num,den)
elseif rad_modelu==3
    s = tf('s')
    num = [ 1];           % Numerator: s
    den = [a_3 a_2 a_1 a_0]; % Denominator: s^2 + 2 s + 10
    G_MPI = tf(num,den)
```

```

end
else
    disp("V základním vzorci jsou záporné nulové póly")
    a_1=-a_1;
    rad_modelu=0;
    if a_1>0 && a_2<0
        rad_modelu=1
    elseif a_1>0 && a_2>0 && a_3<0
        rad_modelu=2
    elseif a_1>0 && a_2>0 && a_3>0
        rad_modelu=3
    else
        disp("ERROR!")
    end
    rad_modelu=2
    if rad_modelu==1
        s = tf('s')
        num = [ 1];           % Numerator: s
        den = [a_1 a_0];     % Denominator: s^2 + 2 s + 10
        G_MPI = tf(num,den)
        G_MPI = G_MPI*(s+1)
    elseif rad_modelu==2
        s = tf('s')
        num = [ 1];           % Numerator: s
        den = [a_2 a_1 a_0]; % Denominator: s^2 + 2 s + 10
        G_MPI = tf(num,den)
        G_MPI = G_MPI*(0.5*s+1)
    elseif rad_modelu==3
        s = tf('s')
        num = [ 1];           % Numerator: s
        den = [a_3 a_2 a_1 a_0]; % Denominator: s^2 + 2 s + 10
        G_MPI = tf(num,den)
        G_MPI = G_MPI*(s+1)
    end
end
end

```

Dodání zpoždění

```

    if TT~=0
        yreal=Y_original;
        G_MPI=G_MPI*exp(-t(TT)*s)
    end
    u=u(1:end);
    y_MPI=lsim(G_MPI,u(1:end),t);
    plot(t, y_MPI, 'LineWidth',2)
    %axis ([0 q -0.1 0.1 ])
    grid on
    legend (['y_{MPI}'])
    hold off
    t=t(1:end);
    y_MPI=y_MPI((1:end-1));

```



```

XX=Y(1:end,1:end);
plot(t(1:N),XX)
legend ("1", "2", "3", "4")
%y_MPI=[0 y_MPI]
yreal=yreal(1:end-1);
t=t(1:end-1);
plot(t,[yreal,y_MPI])
legend ("y_{real}", "y_{MPI}")
hold off
X=(yreal./y_MPI);
plot(t(1+TT:end),X(1+TT:end))
mean(yreal(2+TT:end)./y_MPI(2+TT:end));
mean (X(2+TT:end));

DD(1:length(y_MPI),5)=y_MPI;
end
end
hold off
Prıklad=plot(t,DD(1:length(t),1), 'k.', t,DD(1:length(t),2), 'y.-'
',t,DD(1:length(t),3), 'r--', t,DD(1:length(t),4), 'g-
.', t,DD(1:length(t),5), 'b-')
hold on
plot(t,u)
title ("Srovnání výsledných modelů experimentu", 'FontSize', 30)
legend ("yreal", "PKS", "MNC", "MetPloch", "MPI", "u", 'FontSize', 20)
xlabel ("čas [s]", 'FontSize', 20)
ylabel("výška [cm]", 'FontSize', 20)
grid on
set(gca, "FontSize", 20)
%axis ([0 40 0 40])
hold off
SS=0;
AA=1;

yreal=DD(1:N,1);
PKS=DD(1:N,2);
MNC=DD(1:N,3);
MetPloch=DD(1:N,4);
MPI=DD(1:N,5);

pks=0;
mnc=0;
metploch=0;
mpi=0;

for i=12:N
    if yreal(i)~=0
        pks(i)=(sum(abs(abs( yreal(i)) - abs(PKS(i)))) /yreal(i))*100;
%mean

```

```

        mnc(i)=(sum(abs(abs( yreal(i)) - abs(MNC(i))) /yreal(i)))*100;
%mean
        metploch(i)=(sum(abs(abs( yreal(i)) - abs(MetPloch(i)))
/yreal(i)))*100; %mean
        mpi(i)=(sum(abs(abs( yreal(i)) - abs(MPI(i))) /yreal(i)))*100;
%mean
    end
end
disp("PKS")
100-mean(pks)
100-median(pks)
100-max(pks)
disp("MNC")
100-mean(mnc)
100-median(mnc)
100-max(mnc)
disp("MetPloch")
100-mean(metploch)
100-median(metploch)
100-max(metploch)

```

```

disp("MPI")
100-mean(mpi)
100-median(mpi)
100-max(mpi(1:end-1))

format long

G_propor
G_MND
G_MetPloch
G_MPI

save ('Experiment_srovnani','Priklad')
DD=zeros(8000,5);

```