

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Metody identifikace stabilních dynamických  
systémů 1. řádu s dopravním zpožděním

Petr Nevečeřal

Bakalářská práce

2013

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

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr Nevečeřal**  
Osobní číslo: **I09469**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Řízení procesů**  
Název tématu: **Metody identifikace stabilních dynamických systémů 1. řádu s dopravním zpožděním**  
Zadávací katedra: **Katedra řízení procesů**

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

#### Cíl práce:

Cílem práce je prakticky implementovat a porovnat několik metod aplikovatelných pro stabilní dynamické systémy 1. řádu s dopravním zpožděním. Budou testovány zejména postupy vycházející z tvaru přechodové charakteristiky, metody momentů a metody odhadu na základě minimalizace celkové chyby.

#### Teoretická část:

Teoretická část bude obsahovat úvod do problematiky a popis jednotlivých metod.

#### Praktická část:

Praktická část zahrnuje implementaci jednotlivých algoritmů a porovnání výsledků pro data odpovídající několika typům dynamických systémů z dané třídy. Pro simulace a výpočty bude využit software MATLAB.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**ASTRÖM, K. J. HÄGGLUND, T. PID Controllers: Theory, Design, and Tuning. 2nd edition. Instrument Society of America(ISA), 1995.**

**KIONG, T.K. QUING-GUO, W. - CHIECH, H.C. - HÄGGLUND, T.J. Advances in PID Control, Springer Verlag, 1999.**

Vedoucí bakalářské práce:

**doc. Ing. Jan Cvejn, Ph.D.**

Katedra řízení procesů

Datum zadání bakalářské práce: **11. prosince 2012**

Termín odevzdání bakalářské práce: **10. května 2013**

prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.

Ing. Daniel Honc, Ph.D.  
vedoucí katedry

V Pardubicích dne 29. března 2013

## **Prohlášení autora**

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 9. 5. 2013

Petr Nevečeřal

## **Poděkování**

V této části bych chtěl velice poděkovat panu docentu Cvejnovi za spolupráci při řešení této úlohy, ochotu a poskytnutí všech možných materiálů, potřebných pro vytvoření této bakalářské práce. A rovněž také své rodině a přátelům, co mne po celou dobu podporovali.

## **Anotace**

Cílem bylo implementovat a prakticky ověřit vlastnosti několika metod identifikace pro stabilní dynamické systémy 1. řádu s dopravním zpožděním. Pro aplikaci těchto metod identifikace budou použita simulovaná data. Pro implementaci těchto metod bude použit výpočtový software Matlab se simulačním prostředím Simulink.

## **Klíčová slova**

dynamické systémy, dopravní zpoždění, identifikace, výpočtový software, simulační prostředí

## **Title**

Methods of identification of stable dynamic systems of the 1st order with dead time

## **Annotation**

The target was to implement and verify in practice the properties of several methods for the identification of stable dynamic systems 1st order with delay. For application the methods of identification are used simulated data. To implement these methods will be used computational software Matlab with simulation environment Simulink.

## **Keywords**

dynamic systems, dead time, identification, calculation software, simulation environment

## **Obsah**

<b>Seznam obrázků.....</b>	<b>8</b>
<b>Seznam zkratk a symbolů.....</b>	<b>10</b>
<b>1 Úvod.....</b>	<b>11</b>
<b>2 Teoretická část.....</b>	<b>12</b>
<b>2.1 Úvod do problematiky.....</b>	<b>12</b>
<b>2.2 Metoda na základě shody ve dvou bodech.....</b>	<b>14</b>
2.2.1 Popis metody.....	14
2.2.2 Matematické odvození.....	16
<b>2.3 Metoda ploch.....</b>	<b>18</b>
<b>2.4 Metoda na základě minimalizace celkové kvadratické chyby.....</b>	<b>22</b>
<b>3 Praktická část.....</b>	<b>25</b>
<b>3.1 Simulace pomocí Simulinku.....</b>	<b>25</b>
3.1.1 Vytvoření modelu soustavy vyššího řádu.....	25
3.1.2 Vytvoření modelu dynamické systému 1. řádu s dopravním zpožděním.....	30
<b>3.2 Práce s daty.....</b>	<b>31</b>
<b>3.3 Výsledky bakalářské práce.....</b>	<b>34</b>
<b>3.4 Další možnosti práce s daty.....</b>	<b>38</b>
3.4.1 Metoda importu dat pomocí To Workspace.....	39
3.4.2 Import dat do Matlabu.....	40
<b>4 Závěr.....</b>	<b>42</b>
<b>Literatura.....</b>	<b>44</b>
<b>Příloha A–Zdrojový kód metody na základě shody ve dvou bodech.....</b>	<b>A – 1</b>
<b>Příloha B–Zdrojový kód metody ploch.....</b>	<b>B – 1</b>
<b>Příloha C–Zdrojový kód metody na základě minimalizace celk. kvadr. chyby.....</b>	<b>C – 1</b>

## Seznam obrázků

Obrázek 1: Jednotkový skok.....	12
Obrázek 2: Přejchodová charakteristika soustavy vyššího řádu .....	13
Obrázek 3: Přejchodová charakteristika soustavy 1. řádu s dopravním zpožděním.....	14
Obrázek 4: Určení hodnot $h_1$ a $h_2$ .....	15
Obrázek 5: Určení doby $t_i$ .....	19
Obrázek 6: Zobrazení plochy I nad přejchodovou charakteristikou .....	19
Obrázek 7: Lichoběžníková metoda .....	20
Obrázek 8: Zobrazení plochy $A_1$ pod přejchodovou charakteristikou a určení doby $t_{AR}$ .....	21
Obrázek 9: Blok Transfer Fcn .....	25
Obrázek 10: Tabulka nastavení bloku Transfer Fcn.....	26
Obrázek 11: Blok Step.....	26
Obrázek 12: Tabulka nastavení bloku Step .....	27
Obrázek 13: Blok Band-Limited White Noise .....	27
Obrázek 14: Tabulka nastavení bloku Band-Limited White Noise.....	28
Obrázek 15: Přejchodová charakteristika soustavy vyššího řádu .....	28
Obrázek 16: Dynamický systém vyššího řádu.....	29
Obrázek 17: Přejchodová charakteristika soustavy vyššího řádu .....	29
Obrázek 18: Blok Transport Delay .....	30
Obrázek 19: Tabulka nastavení bloku Transport Delay .....	31
Obrázek 20: Soustava 1. řádu s dopravním zpožděním.....	31
Obrázek 21: Hlavní ovládací panel bloku Scope.....	32
Obrázek 22: Tabulka nastavení parametrů bloku Scope .....	32
Obrázek 23: Tabulka nastavení parametrů simulace .....	33
Obrázek 24: Ukázka panelu Workspace po simulaci .....	34
Obrázek 25: Výsledky metody na základě shody ve dvou bodech.....	35
Obrázek 26: Výsledky metody ploch.....	35
Obrázek 27: Výsledky metody ploch (po úpravě).....	36
Obrázek 28: Určení minimální celkové kvadratické chyby.....	36
Obrázek 29: Výsledky metody na základě minimalizace celkové kvadratické chyby .....	37
Obrázek 30: Porovnání metod identifikace mezi sebou .....	37
Obrázek 31: Blok Chirp Signal.....	38
Obrázek 32: Porovnání metod identifikace mezi sebou po změně šumu .....	38



Obrázek 33: Blok To Workspace.....	39
Obrázek 34: Tabulka nastavení To Workspace .....	39
Obrázek 35: Struktura proměnné To Workspace .....	40
Obrázek 36: Struktura podproměnné signals z To Workspace.....	40
Obrázek 37: Tabulka Import Wizard .....	41

## Seznam zkratk a symbolů

$u(t)$	vstupní veličina
$F(s)$	přenos soustavy
$y(t)$	výstupní veličina
$t$	čas
$h(t)$	přechodová charakteristika
$\eta(t)$	jednotkový skok
$T$	časová konstanta
$\tau$	dopravní zpoždění
$K$	statické zesílení
$a$	koeficient diferenční rovnice
$b$	koeficient diferenční rovnice
$\mathcal{E}_c$	celková kvadratická chyba
$\mathcal{E}_k$	šum
$\mathcal{E}$	vektor hodnot šumů
$A_1$	vypočtená plocha
$I$	vypočtená plocha
$t_i$	pomocná konstanta
$t_{AR}$	pomocná konstanta
$d$	časový posun
$\mathbf{x}$	matice s neznámými koeficienty
$A$	matice se simulovanými daty
$b$	matice se simulovanými daty

# 1 Úvod

Cílem této bakalářské práce je porovnat několik metod identifikace pro stabilní dynamické systémy 1. řádu s dopravním zpožděním.

Teoretická část zahrnuje úvod do problematiky a teoretický rozbor jednotlivých metod. Úvod do problematiky popisuje základní teoretické poznatky, které se této bakalářské práce týkají. Především poznatky o dynamických systémech a identifikaci. Následně jsou popsány metody vycházející z tvaru přechodové charakteristiky, metody momentu a metody odhadu na základě minimalizace celkové kvadratické chyby. Popis všech uvedených metod identifikace obsahuje teoretický i matematický popis.

Praktická část bakalářské práce popisuje postup získání simulovaných dat z dynamického systému vyššího řádu, které jsou potřeba pro aplikaci metod identifikace a také popis manipulace s těmito daty v prostředí Matlab. Aplikované metody identifikace budou psány ve skriptech Matlabu (soubory s příponou .m). Zdrojové kódy jednotlivých metod identifikace jsou součástí příloh bakalářské práce. Výsledkem metod jsou získané parametry, potřebné pro vytvoření modelu dynamického stabilního systému 1. řádu s dopravním zpožděním. Model bude vytvořen v simulačním prostředí Simulink, který je součástí výpočtového softwaru Matlab (vytváří soubory s příponou .mdl). Data získaná z tohoto modelu jsou porovnána s daty naměřenými v jednom grafu. Chování jednotlivých metod je porovnáno pro různé úrovně šumového signálu. Působení tohoto šumového signálu vyvolá chyby v simulovaných datech a na základě porovnání všech výsledků bude vyhodnoceno celkové chování jednotlivých metod. Především jejich chování na různé typy šumu.

Bakalářská práce také obsahuje důležité poznatky, které byly rovněž důležité v praktické části. Především je tato kapitola zaměřena na několik způsobů získání dat z modelu a práci s těmito daty v prostředí Matlab.

V závěrečné části bude shrnuto hodnocení výsledků bakalářské práce. Především chování všech tří metod při působení různých typů šumu, slabé stránky a důležité kroky praktické aplikace metod identifikace.

V této úloze bude použit výpočtový software Matlab verze R2009a se simulačním prostředím Simulink.

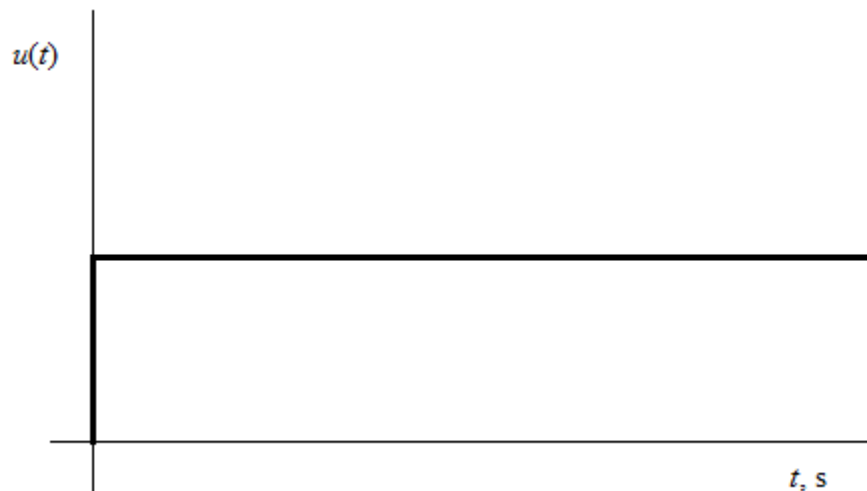
## 2 Teoretická část

### 2.1 Úvod do problematiky

Tato část bakalářské práce je zaměřena na základní teorii, která je důležitá pro pochopení celkového významu této úlohy. [1]

Dynamický systém je obecně systém, jenž má vstupy a výstupy a je charakterizován tak, že hodnoty jeho výstupních veličin nezávisí pouze na aktuální hodnotě vstupů, ale také na přechodí historii vstupů a výstupů. Pro zjištění informací o chování nějakého skutečného fyzického dynamického systému je třeba definovat systém a vyšetřit jeho vlastnosti. Jedním ze způsobů, jak toto provést, je změřit hodnoty veličin na tomto zařízení. Proto se zavádí identifikace, kdy na základě měřených dat se získá matematický model systému. Prakticky se to provádí tak, že z naměřených dat se získají určitými metodami důležité parametry pro vytvoření modelu.

Při identifikaci lze tyto parametry získat na základě odezvy systému na speciální typ vstupního signálu. Odezva je právě reakcí systémů na tento signál. Často se využívá přechodová charakteristika, což je reakce systému na jednotkový skokový signál, nebo též jednotkový skok, který je na obr. 1.



Obr. 1 – Jednotkový skok

Přechodová charakteristika je často označována jako  $h(t)$  a jednotkový skok pak jako  $\eta(t)$ , kdy v čase  $t = 0$  dochází ke skokové změně a  $\eta(t) = 1$ .

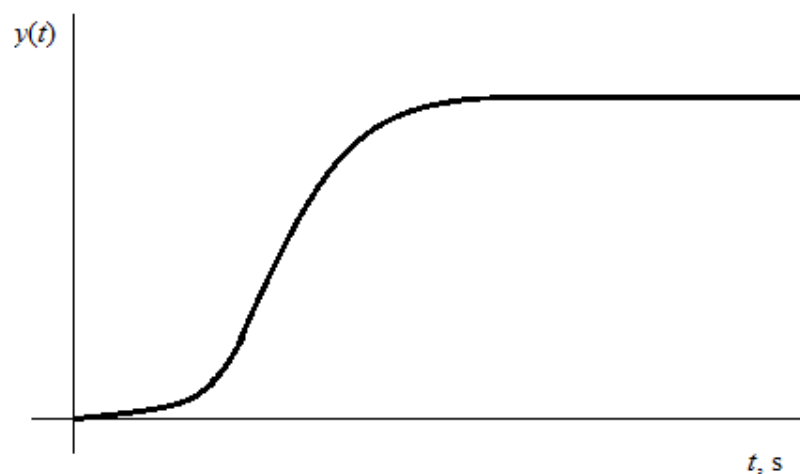
V praxi se tyto získané parametry používají pro nastavení regulátoru určitého typu. Aby byl regulátor nastaven správně, je třeba tyto parametry získat s velkou přesností. Zde testované metody jsou známé a jsou popsány v různých publikacích. [2] [3] V podstatě lze nahradit soustavu vyššího řádu soustavou 1. řádu s dopravním zpožděním. Každou soustavou, lze nějakým způsobem charakterizovat. Pro tyto účely byl zaveden obrazový přenos soustavy  $F(s)$ . Tento způsob popisu systému vyplývá z Laplaceovy transformace. Přenos vyššího řádu je popsán následujícím způsobem.

$$F(s) = \frac{K}{(T_1s + 1) + \dots (T_Ns + 1)} \quad (1)$$

Kde  $K$  je statické zesílení soustavy a  $T_1 \dots T_N$  označuje časové konstanty. Soustava vyššího řádu je charakterizována právě větším počtem časových konstant  $T$ . Kde největší vliv na dynamiku systému má časová konstanta  $T_1$ . Při modelování je často vytvářen sériovým spojením několika dynamických systémů 1. řádu. Tento dynamický systém, pak může být nahrazen dynamickým systémem 1. řádu s dopravním zpožděním, který je popsán takto.

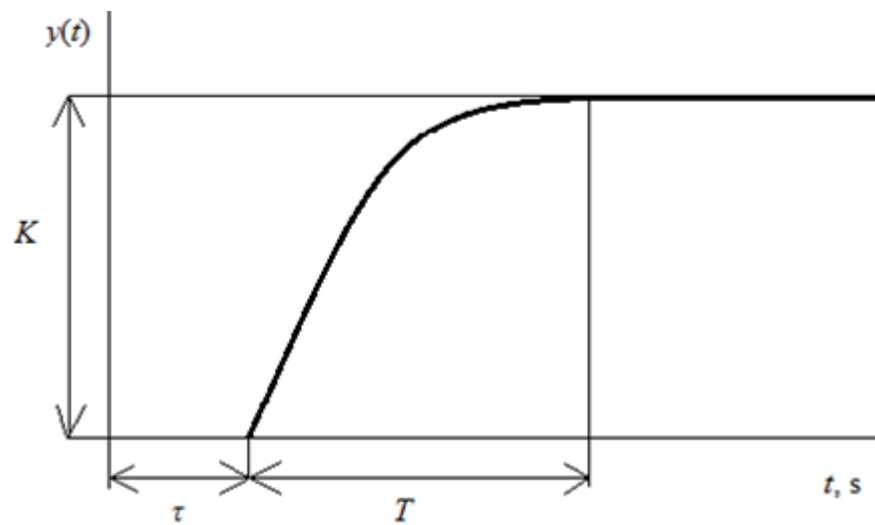
$$F(s) = \frac{K}{(T_1s + 1)} \cdot e^{-\tau s} \quad (2)$$

Kde  $\tau$  je dopravní zpoždění a společně s časovou konstantou  $T$  a zesílením  $K$  tvoří neznámé parametry, které je potřeba získat na základě metod identifikace. Na obr. 2 je příklad přechodové charakteristiky soustavy vyššího řádu.



Obr. 2 – Přechodová charakteristika soustavy vyššího řádu

A na obr. 3 je příklad přechodové charakteristiky soustavy 1. řádu s dopravním zpožděním.



Obr. 3 – Přechodová charakteristika soustavy 1. řádu s dopravním zpožděním

Její tvar samozřejmě závisí na získaných parametrech. Existují i jiné způsoby, než které jsou popsány dále. Nejběžnějším způsobem je získání náhradního modelu s využitím tečny přechodové charakteristiky v inflexním bodě. Použité metody identifikace v této bakalářské práci vracejí ale lepší výsledky. V další kapitole bude věnován popisu vybraných metod identifikace.

## 2.2 Metoda na základě shody ve dvou bodech

### 2.2.1 Popis metody

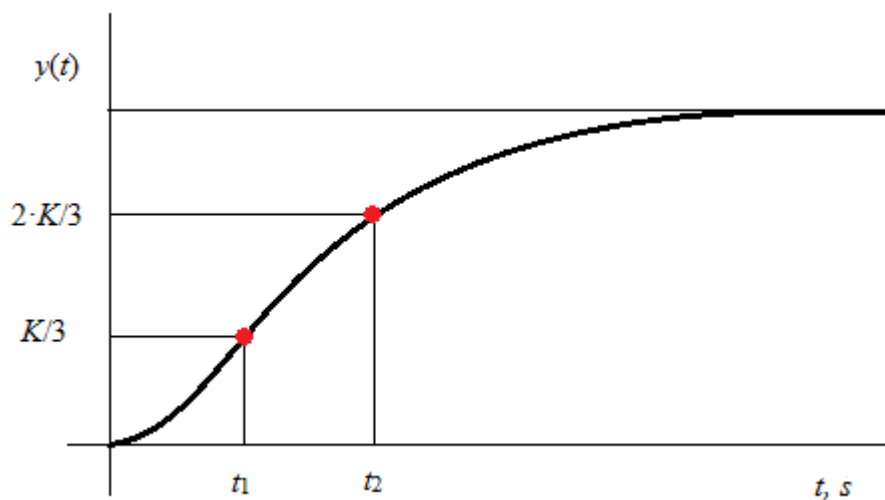
V této metodě se vychází z matematického popisu přechodové charakteristiky dynamického systému 1. řádu s dopravním zpožděním, který je uveden níže.

$$h(t) = K \cdot \left(1 - e^{-\left(\frac{t-\tau}{T}\right)}\right) \quad (3)$$

Kde  $h(t)$  je označení přechodové charakteristiky,  $K$  je statické zesílení,  $t$  je čas,  $T$  je časová konstanta a  $\tau$  je dopravní zpoždění.

Pro nastavení regulátoru musíme tedy získat neznámé parametry: časovou konstantu  $T$ , statické zesílení  $K$  a dopravní zpoždění  $\tau$ .

Tato metoda je založená na odvození těchto parametrů přímo z popisu přechodové charakteristiky. Ze simulovaných dat soustavy vyššího řádu si vybereme dva body, tj. dvě hodnoty z naměřených dat. Hodnoty  $h_1$  v čase  $t_1$  a  $h_2$  v čase  $t_2$  vybereme následovně. Hodnota  $h_1$  by měla odpovídat zhruba  $1/3$  zesílení a  $h_2$  zhruba  $2/3$  zesílení. Ještě předtím však samozřejmě je nutné určit samotné statické zesílení  $K$ . Například graficky nebo nějakou vybranou funkcí. Na obr. 4 je znázorněno určení hodnot  $h_1$  a  $h_2$ .



Obr. 4 – Určení hodnot  $h_1$  a  $h_2$

Pro tyto dvě hodnoty, pak získáme tyto dvě rovnice.

$$h_1(t) = K \cdot \left(1 - e^{-\frac{(t_1 - \tau)}{T}}\right) \quad (4)$$

$$h_2(t) = K \cdot \left(1 - e^{-\frac{(t_2 - \tau)}{T}}\right) \quad (5)$$

Z těchto dvou rovnic pak matematicky odvodíme parametry  $T$  a  $\tau$ .

$$\tau = T \cdot \ln\left(1 - \frac{h_2}{K}\right) + t_2 \quad (6)$$

$$T = \frac{t_2 - t_1}{\ln\left(\frac{1 - \frac{h_1}{K}}{1 - \frac{h_2}{K}}\right)} \quad (7)$$

Nyní stačí tyto vztahy správně aplikovat v prostředí Matlab. Kompletní matematické odvození je v další kapitole.

### 2.2.2 Matematické odvození

V této kapitole bude postupně popsáno celé odvození parametrů  $T$  a  $\tau$  z matematického popisu přechodové charakteristiky. Po určení dvou shodných bodů vzniknou tyto dvě rovnice.

$$h_1(t) = K \cdot \left(1 - e^{-\frac{(t_1-\tau)}{T}}\right) \quad (8)$$

$$h_2(t) = K \cdot \left(1 - e^{-\frac{(t_2-\tau)}{T}}\right) \quad (9)$$

Pozn.: pro větší přehlednost bude označení  $h_1(t)$  a  $h_2(t)$  bude zkráceno na  $h_1$  a  $h_2$ .

Obě strany budou vynásobeny  $1/K$ .

$$\frac{h_1}{K} = 1 - e^{-\frac{(t_1-\tau)}{T}} \quad (10)$$

$$\frac{h_2}{K} = 1 - e^{-\frac{(t_2-\tau)}{T}} \quad (11)$$

Poté se k oběma stranám rovnic přičte  $-1$ .

$$\frac{h_1}{K} - 1 = -e^{-\frac{(t_1-\tau)}{T}} \quad (12)$$

$$\frac{h_2}{K} - 1 = -e^{-\frac{(t_2-\tau)}{T}} \quad (13)$$

Následně budou rovnice vynásobeny  $-1$ .

$$1 - \frac{h_1}{K} = e^{-\frac{(t_1-\tau)}{T}} \quad (14)$$

$$1 - \frac{h_2}{K} = e^{-\frac{(t_2-\tau)}{T}} \quad (15)$$



Obě rovnice budou nyní zlogaritmovány přirozeným logaritmem.

$$\ln\left(1 - \frac{h_1}{K}\right) = \frac{-(t_1 - \tau)}{T} \quad (16)$$

$$\ln\left(1 - \frac{h_2}{K}\right) = \frac{-(t_2 - \tau)}{T} \quad (17)$$

Další úpravou bude vynásobení obou rovnic  $T$  a odstranění závorek na pravé straně rovnic.

$$-t_1 + \tau = T \cdot \ln\left(1 - \frac{h_1}{K}\right) \quad (18)$$

$$-t_2 + \tau = T \cdot \ln\left(1 - \frac{h_2}{K}\right) \quad (19)$$

Nyní budou oběma stranám rovnic se přičte  $-\tau$ .

$$-t_1 = T \cdot \ln\left(1 - \frac{h_1}{K}\right) - \tau \quad (20)$$

$$-t_2 = T \cdot \ln\left(1 - \frac{h_2}{K}\right) - \tau \quad (21)$$

Toto je typický příklad soustavy dvou rovnic o dvou neznámých. Takže například z druhé rovnice vyjádříme  $\tau$ .

Přičteme nejdříve  $\tau$  k oběma stranám rovnice.

$$-t_2 + \tau = T \cdot \ln\left(1 - \frac{h_2}{K}\right) \quad (22)$$

Teď k oběma stranám rovnice přičteme  $t_2$  a získáme parametr  $\tau$ .

$$\tau = T \cdot \ln\left(1 - \frac{h_2}{K}\right) + t_2 \quad (23)$$

Pro získání parametru  $T$  dosadíme do první rovnice za  $\tau$ .

$$-t_1 = T \cdot \ln\left(1 - \frac{h_1}{K}\right) - T \cdot \ln\left(1 - \frac{h_2}{K}\right) - t_2 \quad (24)$$

Nejdříve se k oběma stranám rovnice přičte  $t_2$ .

$$-t_1 + t_2 = T \cdot \ln\left(1 - \frac{h_1}{K}\right) - T \cdot \ln\left(1 - \frac{h_2}{K}\right) \quad (25)$$

Na pravé straně dojde teď k vytknutí konstanty  $T$ .

$$-t_1 + t_2 = T \cdot \left[ \ln\left(1 - \frac{h_1}{K}\right) - \ln\left(1 - \frac{h_2}{K}\right) \right] \quad (26)$$

Z rozdílu dvou logaritmů vznikne logaritmus podílu.

$$-t_1 + t_2 = T \cdot \left[ \ln\left(\frac{1 - \frac{h_1}{K}}{1 - \frac{h_2}{K}}\right) \right] \quad (27)$$

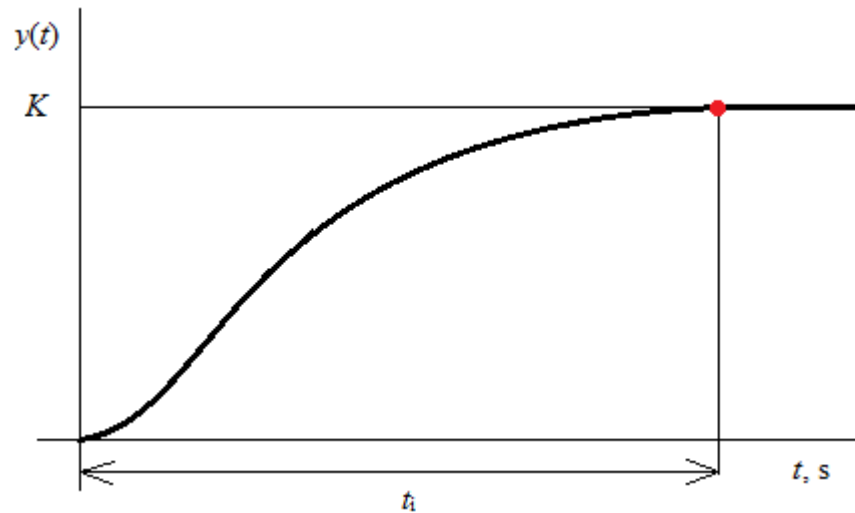
Převedením logaritmu na levou stranu bude získán poslední parametr  $T$ .

$$T = \frac{t_2 - t_1}{\ln\left(\frac{1 - \frac{h_1}{K}}{1 - \frac{h_2}{K}}\right)} \quad (28)$$

### 2.3 Metoda ploch

Místo metody momentů byla nakonec vybrána metoda ploch. Jelikož obě metody dávají stejné výsledky, tak je jedno, která z nich bude vybrána. Metoda ploch je pro aplikaci trochu zajímavější.

Prvním krokem je určení zesílení  $K$ , které může být určeno stejně jako v předchozím případě, například dohledáním maximální hodnoty v datech nebo určením z poslední naměřené hodnoty. Dalším krokem je určení plochy  $I$  nad přechodovou charakteristikou. Plocha bude vypočítána pomocí určitého integrálu. Předtím je však nutné určit horní mez  $t_i$ . Doba  $t_i$  je určena v okamžik, kdy hodnota přechodové charakteristiky se shoduje s hodnotou již určeného statického zesílení  $K$ . Určení doby  $t_i$  je na obr. 5.

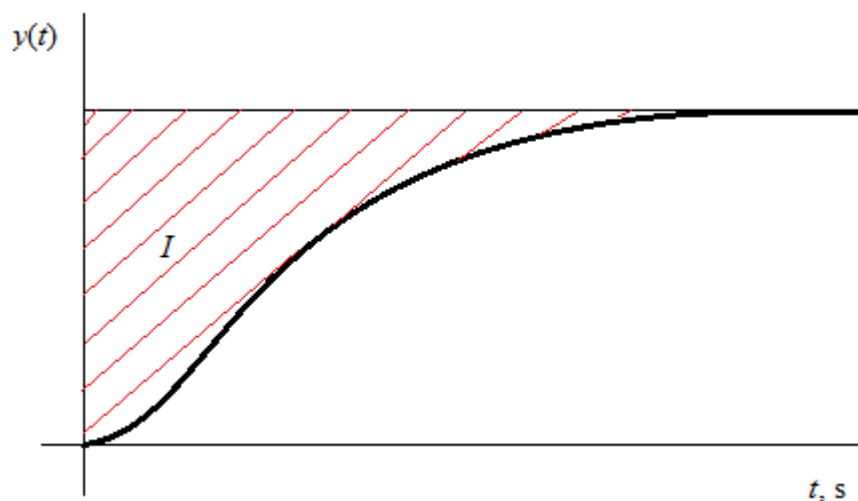


Obr. 5 – Určení doby  $t_i$

Po zjištění doby  $t_i$  lze vypočítat následovně plochu  $I$  nad přechodovou charakteristikou.

$$I = \int_0^{\infty} (h(\infty) - h(t)) dt \approx \int_0^{t_i} (K - h(t)) dt \quad (29)$$

Na obr. 6 je zobrazena vypočtená plocha  $I$ , která je nad přechodovou charakteristikou.

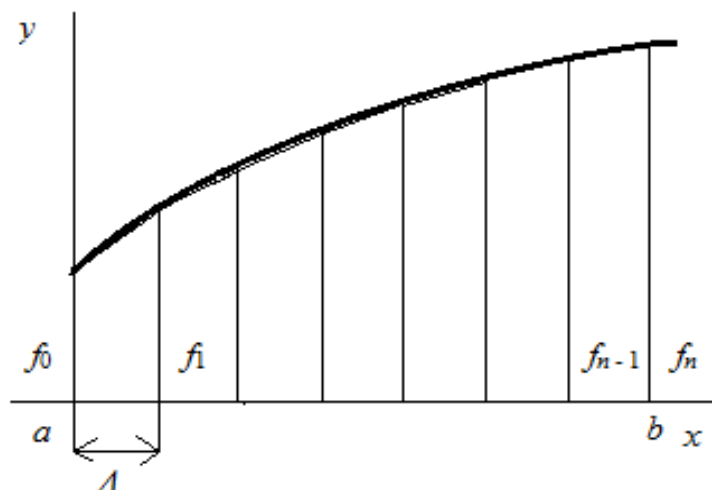


Obr. 6 – Zobrazení plochy  $I$  nad přechodovou charakteristikou

Bohužel v prostředí Matlab takto nebude možné tento integrál vypočítat. Matlab pro výpočet integrálu bude očekávat nějakou funkci. V našem případě je ale přechodová charakteristika

reprezentována naměřenými daty, nebo jinak řečeno čísly. Proto bude nutné tento určitý integrál vypočítat numericky pomocí lichoběžníkové metody. [4]

Na obr. 7 je ukázka aplikace lichoběžníkové metody určené pro výpočet určitého integrálu funkce  $f(x)$ .



Obr. 7 – Lichoběžníková metoda

Lichoběžníková metoda se používá pro numerický výpočet určitého integrálu. Integrál určuje plochu pod křivkou na daných mezích. Tato metoda spočívá v tom, že daná plocha je rozdělena na menší plochy–lichoběžníky. Plochy jsou rozděleny pevným krokem  $\Delta$ , který se nemění. Součtem těchto obdélníků pak vznikne výsledná plocha.

Výsledná hodnota určitého integrálu je pak vypočtena následovně.

$$\int_a^b f(x)dx \approx \left( \frac{f_0 + f_n}{2} + \sum_{i=1}^{n-1} f_i \right) \cdot \Delta \quad (30)$$

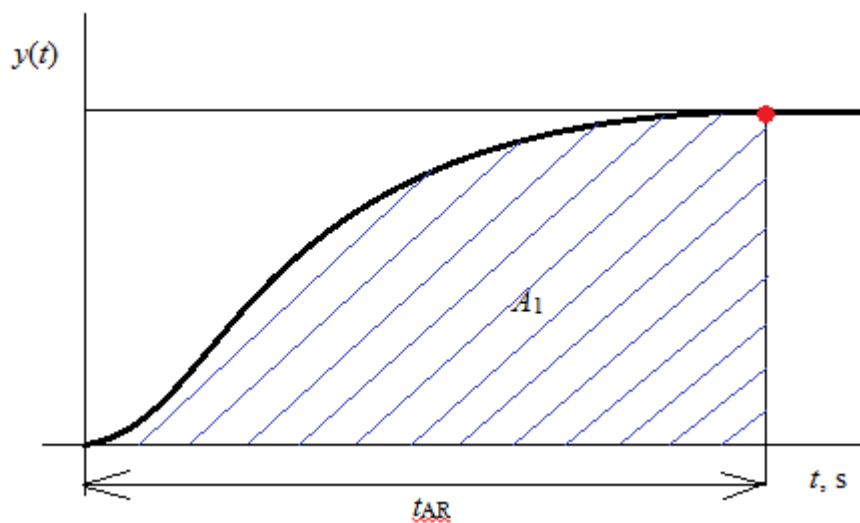
Nejdříve je tedy proveden výpočet  $K - h(t)$ , a pak je pro výpočet plochy I aplikována lichoběžníková metoda.

Dalším výpočtem je určení doby  $t_{AR}$  a plochy  $A_1$ . Doba  $t_{AR}$  nemá stejně jako doba  $t_i$  v literatuře žádný odborný název, jde pouze o jakési časové konstanty. Konstanta  $t_{AR}$  je určena takto.

$$t_{AR} = I/K \quad (31)$$

Její výpočet je dán vlastně poměrem plochy  $I$  a zesílení  $K$ . Plocha  $A_1$  je umístěna pod přechodovou charakteristikou a je právě omezena dobou  $t_{AR}$ . Níže je uveden její výpočet a na obr. 8 je vyobrazena plocha  $A_1$  i s dobou  $t_{AR}$ .

$$A_1 = \int_0^{t_{AR}} h(t) dt \quad (32)$$



Obr. 8 – Zobrazení plochy  $A_1$  pod přechodovou charakteristikou a určení doby  $t_{AR}$

Pro výpočet plochy  $A_1$  je znovu aplikována lichoběžníková metoda. Pro plochu  $A_1$  také platí další skutečnost. Pokud do integrálu místo  $h(t)$  dosadíme matematický popis přechodové charakteristiky dynamického systému 1. řádu s dopravním zpožděním a integrál vypočítáme, získáme následující vztah.

$$A_1 = K \cdot T \cdot e^{-1} \quad (33)$$

Z toho lze úpravou získat časovou konstantu  $T$ . Vzorec pro výpočet časové konstanty vypadá takto.

$$T = (A_1 \cdot e)/K \quad (34)$$

e znamená Eulerovo číslo, což je 2,718...

Poslední důležitým parametrem, který je důležité získat je dopravní zpoždění  $\tau$ . To lze získat z konstanty  $t_{AR}$ , pro kterou platí následující.

$$t_{AR} = T + \tau \quad (35)$$

Z toho lze odvodit dopravní zpoždění takto.

$$\tau = t_{AR} - T \quad (36)$$

Nyní na základě této metody získány všechny tři parametry důležité pro vytvoření modelu dynamického systému 1. řádu s dopravním zpožděním.

## 2.4 Metoda na základě minimalizace celkové kvadratické chyby

Při aplikaci této metody je potřeba mít naměřená data nejen výstupní veličiny  $y(t)$ , ale také vstupní veličiny  $u(t)$ . Každý systém lze popsat spojitým přenosem, kdy měřená veličina je spojitá, ale také diskretním přenosem, kdy si měřenou veličinu lze představit jako soubor hodnot, měřených se stejným vzorkovacím krokem. Tato metoda vychází na rozdíl od předchozích dvou metod ne ze spojitého, ale právě z diskretního přenosu. Dynamický systém 1. řádu s dopravním zpožděním má spojitý přenos, který je využíván při vytváření modelu této soustavy. Tímto známým spojitým přenosem lze zmíněný systém popsat.

$$F(s) = \frac{K}{T_s + 1} \cdot e^{-\tau s} \quad (37)$$

A takto ho lze popsat pomocí diskretního přenos.

$$F(z) = K \cdot \frac{1 - e^{-\Delta/T}}{z - e^{-\Delta/T}} \cdot z^{-d} \quad (38)$$

Delta  $\Delta$  je již zmiňovaný vzorkovací krok a  $d$  je časový posun. Tento časový posun je roven podílu dopravního zpoždění a vzorkovacího kroku. Tento vzorec bude následně využit pro získání dopravního zpoždění.

$$d = \tau/\Delta \quad (39)$$

Úpravou přenosu vznikne tato diferenční rovnice.

$$y_{k+1} - e^{-\Delta/T} \cdot y_k = K \cdot (1 - e^{-\Delta/T}) \cdot u_{k-d} \quad (40)$$

Pro řešení diferenčních rovnic se využívá  $Z$ -transformace. Úpravou této rovnice vznikne následující vztah, který popisuje šum  $\varepsilon_k$ , vzniklý při diskretním měření na daném systému zapsaný ve skalárním tvaru.

$$y_{k+d+1} - a \cdot y_{k+d} - b \cdot u_k = \varepsilon_k \quad (41)$$

Z tohoto popisu vyplývá, že tento šum  $\varepsilon_k$  je závislý právě na časovém posunu  $d$ . Řešením této diferenční rovnice jsou neznámé koeficienty  $a$  a  $b$ . Aby bylo možné tyto parametry získat, bude nutné rovnici upravit a zapsat v maticovém tvaru.

$$[y_{k+d} \quad u_k] \cdot \begin{bmatrix} a \\ b \end{bmatrix} = y_{k+d+1} + \varepsilon_k \quad (42)$$

Osamostatněním šumu na jedné straně dosazením  $N - 1 - d$  za index  $k$  vzniknou výsledné matice v následujícím tvaru. Tento tvar popisuje celkovou chybu jako vektor hodnot jednotlivých šumů.

$$\begin{bmatrix} y_{1+d} & u_1 \\ \vdots & \vdots \\ y_{N-1} & u_{N-1-d} \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} - \begin{bmatrix} y_{2+d} \\ \vdots \\ y_N \end{bmatrix} = \varepsilon \quad (43)$$

Proměnná  $N$  označuje počet naměřených dat. Tento vztah lze zapsat obecně ve tvaru.

$$\mathbf{A} \cdot \mathbf{x} - \mathbf{b} = \varepsilon \quad (44)$$

Matice  $\mathbf{A}$  a  $\mathbf{b}$  jsou vytvořené ze simulovaných dat veličin  $u(t)$  a  $y(t)$ . Časový posun bude zadán v cyklu a v určitém intervalu (například od 0 do 30). Obě tyto matice budou plněny daty. Rozměry matice budou pro každý zadaný časový posun jiné. Matice  $\mathbf{x}$  je složena z neznámých koeficientů  $a$  a  $b$ . Osamostatněním matice  $\mathbf{x}$ , podle následujícího vzorce, budou získány neznámé koeficienty.

$$\mathbf{x} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{b} \quad (45)$$

Po získání neznámých koeficientů, lze určit vektor hodnot jednotlivých chyb. Výsledná kvadratická chyba vzniká z jednotlivých šumů a její výpočet je určen následujícím vztahem.

$$\varepsilon_c = \sqrt{(\varepsilon^T \cdot \varepsilon) / N_\varepsilon} \quad (46)$$

$N_\varepsilon$  označuje počet hodnot všech šumů.

Princip této metody spočívá v tom, najít minimální hodnotu kvadratické chyby  $\varepsilon_c$ , která je na časovém posunu  $d$  závislá. Tento časový posun je zadán jako několik hodnot v řadě, aby mohla být nalezena minimální hodnota celkové kvadratické chyby  $\varepsilon_c$ . Pro nejmenší hodnotu celkové kvadratické chyby, pak bude určen výsledný časový posun  $d$ .

Pro tuto hodnotu časového posunu, pak budou z naměřených dat znovu vytvořeny nové matice a z nich určeny stejným způsobem výsledné koeficienty  $a$  a  $b$ . Nyní z těchto koeficientů lze určit výsledné parametry pro vytvoření dynamického systému 1. řádu s dopravním zpožděním. Pro koeficient  $a$  platí následující vztah.

$$a = e^{-\Delta/T} \quad (47)$$

Vzorkovací krok  $\Delta$  lze určit přímo z dat. Takže po úpravě lze z tohoto vztahu určit časovou konstantu  $T$ , jak je tomu zde.

$$T = -\frac{\Delta}{\ln a} \quad (48)$$

Pro koeficient  $b$  platí tento vztah.

$$b = K \cdot (1 - a) \quad (49)$$

Matematickým odvozením z něj lze získat zesílení  $K$  v tomto tvaru.

$$K = b / (1 - a) \quad (50)$$

Dopravní zpoždění  $\tau$  lze získat součinem vzorkovacího kroku a časového posunu.

$$\tau = d \cdot \Delta \quad (51)$$



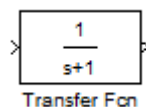
## 3 Praktická část

### 3.1 Simulace pomocí Simulinku

Simulovaná data pro aplikaci metod identifikace nakonec nebyla naměřena na skutečném dynamickém systému vyššího řádu, ale pouze na jeho simulaci. To znamená, že systém vyššího řádu byl vytvořen v simulačním prostředí Simulink a byly simulovány jeho vlastnosti. Na jeho vstupu bude jednotkový skok a na jeho výstupu budou získávána měřená data. K dynamickému systému bude připojen i blok, který bude simulovat určitý způsob šumu (chyby). V praxi tento šum může být určitou chybou vzniklou při měření nebo nějakou nežádanou veličinou, která ovlivňuje chování tohoto systému. Vyzkoušet lze několik typů šumu, protože metody by měly v podstatě vracet podobné výsledky, bez ohledu na typ šumu. To znamená, že by působící šum neměl způsobit nějakou razantní odchylku při získávání parametrů, důležitých pro vytvoření modelu dynamického systému 1. řádu s dopravním zpožděním.

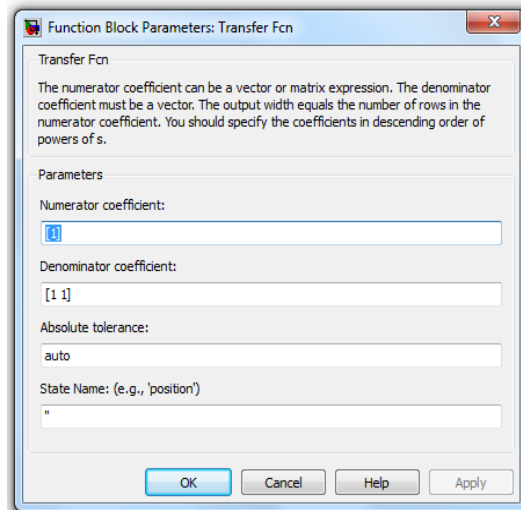
#### 3.1.1 Vytvoření modelu soustavy vyššího řádu

Nyní bude popsáno vytvoření systému vyššího řádu v Simulinku. Ze známého přenosu (který byl vysvětlen v úvodu do problematiky) dynamického systému vyššího bude vytvořen model. Jak již bylo naznačeno, bude vytvořen sériovým spojením několika systémů 1. řádu. Proto bude použit blok Transfer Fcn, který je součástí knihovny Continuous. Tento blok je na obr. 9.



Obr. 9 – Blok Transfer Fcn

Rozkliknutím každého bloku se otevře tabulka nastavení. Tabulka nastavení tohoto bloku je na obr. 10.



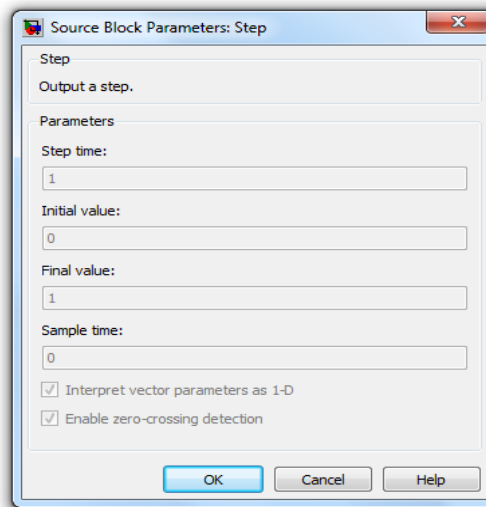
Obr. 10 – Tabulka nastavení bloku Transfer Fcn

Položka Numerator coefficient je nastavení statického zesílení  $K$  a Absolute Tolerance je jmenovatelem přenosu systému, kde první číslice vyjadřuje časovou konstantu  $T$ . Dalším blokem je vstupní signál, tedy jednotkový skok. Ten je reprezentován blokem Step, který se nachází v knihovně Sources. Na obr. 11 je ukázka bloku Step.



Obr. 11 – Blok Step

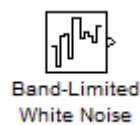
Nastavení tohoto bloku je na obr. 12.



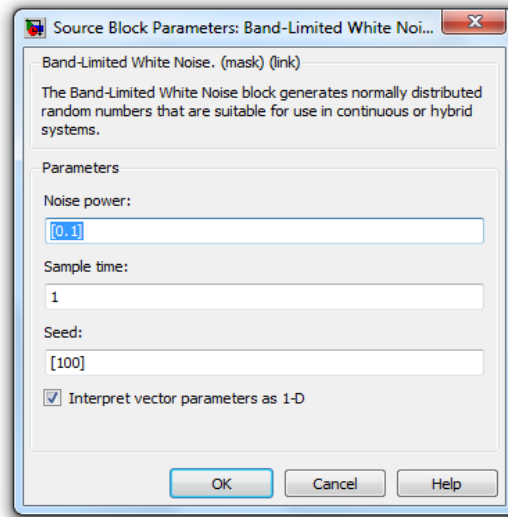
Obr. 12 – Tabulka nastavení bloku Step

Položka Step time udává dobu, kdy k tomuto skoku má dojít. Initial value a Final value, pak udává počáteční a konečnou hodnotu tohoto skoku.

Dalším důležitým blokem je blok, který simuluje šum. Těchto bloků je několik v Simulinku a další šumy lze i vytvořit. V tomto případě byl vybrán blok Band-Limited White Noise, jenž je součástí knihovny Sources. Na obr. 13 je tento blok zobrazen a na obr. 14 je jeho tabulka nastavení.

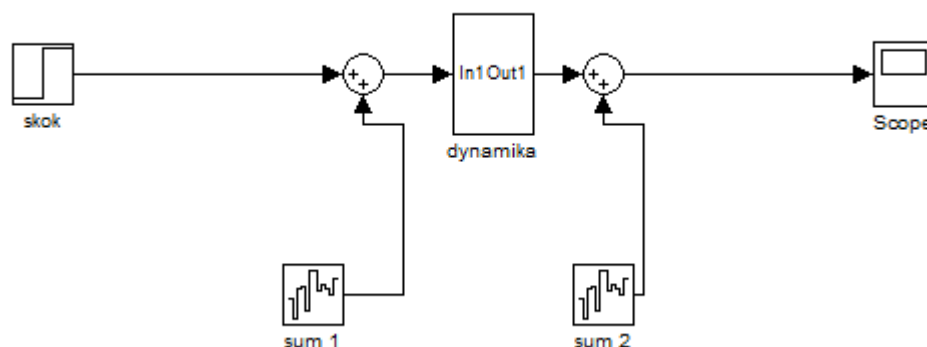


Obr.13 – Blok Band-Limited White Noise



Obr. 14 – Tabulka nastavení bloku Band-Limited White Noise

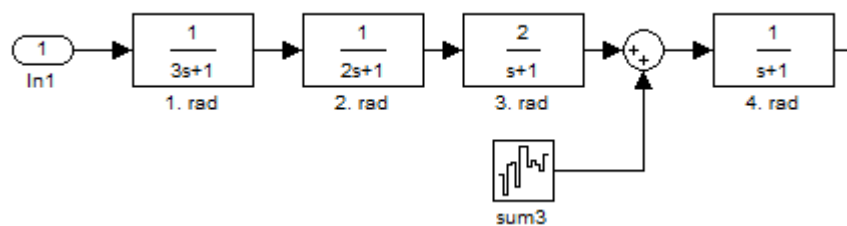
Tomuto šumu se občas říká bílý šum. Důležitou položkou v nastavení je Seed. Tento šum vzniká na základě určitého algoritmu. Tento algoritmus, lze ovlivnit právě nastavením položky Seed. Ta je charakterizována zadaným číslem. Na základě tohoto nastavení je šum generován. Další potřebné bloky pro vytvoření dynamického systému vyššího řádu jsou velice známé a tvoří základ simulačního prostředí Simulink. Proto nemá asi smysl, je nějak složitě popisovat. Pro zobrazení signálu je použit blok Scope a pro připojení šumu do obvodu je použit bloku Sum. Celý výsledný model je na obr. 15.



Obr. 15 – Model dynamického systému vyššího řádu pro simulaci

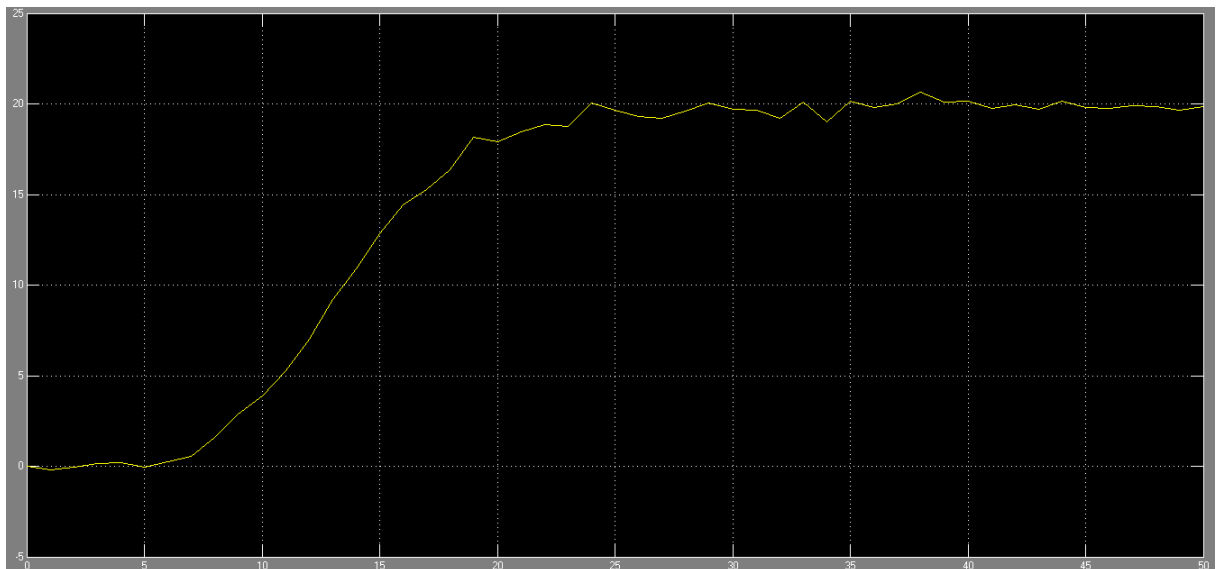
V případě třetí metody identifikace, kdy je nutné ještě získat i naměřené hodnoty vstupní veličiny  $u(t)$ , je tento model ještě rozšířen o známý blok Mux, aby bylo možné do bloku Scope přivést oba signály najednou.

Blok dynamika je subsystem, který je samotným dynamickým systémem vyššího řádu. Z obrázku vyplývá, že šum je připojen, jak na jeho vstup, tak i na jeho výstup. Otevřením bloku dynamika se tento vytvořený systém zobrazí v novém okně. Vytvořený systém je 10. řádu, proto je na obr. 16 níže zachycena pouze jeho část (4. řády systému), aby bylo možné tento model dobře zobrazit v této práci.



Obr. 16 – Dynamický systém vyššího řádu

Z tohoto obrázku je patrné, že i uvnitř systému působí šum, který simuluje nějakou působící chybovou veličinu. Na obr. 17 je možné vidět výslednou přechodovou charakteristiku tohoto systému, která je zobrazena pomocí bloku Scope.



Obr. 17 – Přechodová charakteristika soustavy vyššího řádu v Simulinku

Soustava vyššího řádu je vytvořena a nastavena tak, aby bylo možné dobře aplikovat metody identifikace a systém, co nejlépe simuloval vlastnosti skutečného reálného systému tohoto typu.

Nyní je potřeba získat naměřená data z tohoto systému pro identifikaci.

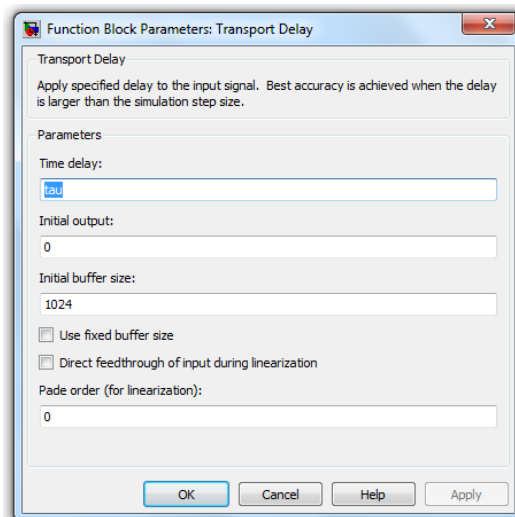
### 3.1.2 Vytvoření modelu dynamické systému 1. řádu s dopravním zpožděním

Po aplikaci jednotlivých metod identifikace, byly získány všechny tři parametry pro vytvoření modelu soustavy 1. řádu s dopravním zpožděním. (V kapitole úvod do problematiky byl vysvětlen přenos této soustavy důležitý pro její vytvoření). Pro vytvoření modelu bude potřeba se seznámit ještě s jedním blokem Simulinku, který se se jmenuje Transport Delay a je součástí knihovny Continuous. Už z toho názvu vyplývá, že tento blok simuluje samotné dopravní zpoždění. Tento blok je na obr. 18.



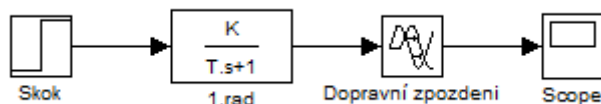
Obr. 18 – Blok Transport Delay

Na obr. 19 níže je zobrazena tabulka nastavení tohoto bloku.



Obr. 19 – Tabulka nastavení bloku Transport Delay

Důležitou položkou tohoto nastavení je položka Time Delay, kde bude zadána jako hodnota velikost získané konstanty dopravního zpoždění  $\tau$ . Celý vytvořený model soustavy 1. řádu s dopravním zpožděním je na obr. 20.



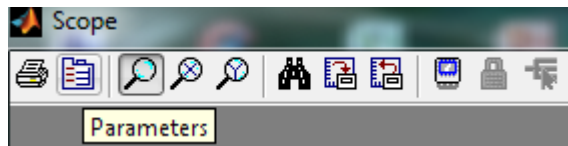
Obr. 20 – Soustava 1. řádu s dopravním zpožděním

V bloku 1. řádu je pak dosazena získaná velikost časové konstanty  $T$  a zesílení  $K$ . Přechodová charakteristika bude zobrazena až v další části této práce, kde budou zobrazeny a porovnány charakteristiky pro všechny tři metody.

### 3.2 Práce s daty

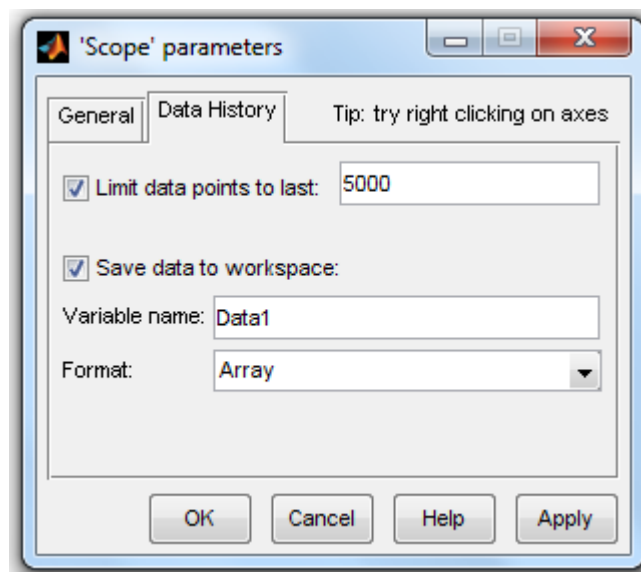
Jedním způsobem jak získat naměřená data z tohoto modelu je využít určitých funkcí Simulinku, pomocí kterých budou data změřena a importována do prostředí Matlab. Jednou z možností jak provést tuto akci je, importovat naměřená data do Matlabu pomocí bloku Scope. Postup je následující. Po rozkliknutí bloku Scope, který je umístěn na výstupu dynamického systému, se zobrazí plocha, na které je vyobrazena přechodová charakteristika

modelu a také ovládací panel, jenž je umístěn vlevo nahoře. Tento ovládací panel je zobrazen na obr. 21.



Obr. 21 – Hlavní ovládací panel bloku Scope

Důležitá je především položka Parameters, který je umístěna jako druhá zleva na ovládacím panelu bloku Scope. Po rozkliknutí se otevře tabulka 'Scope' Parameters, obsahující záložky General a hlavně Data History, která je vidět na obr. 22.



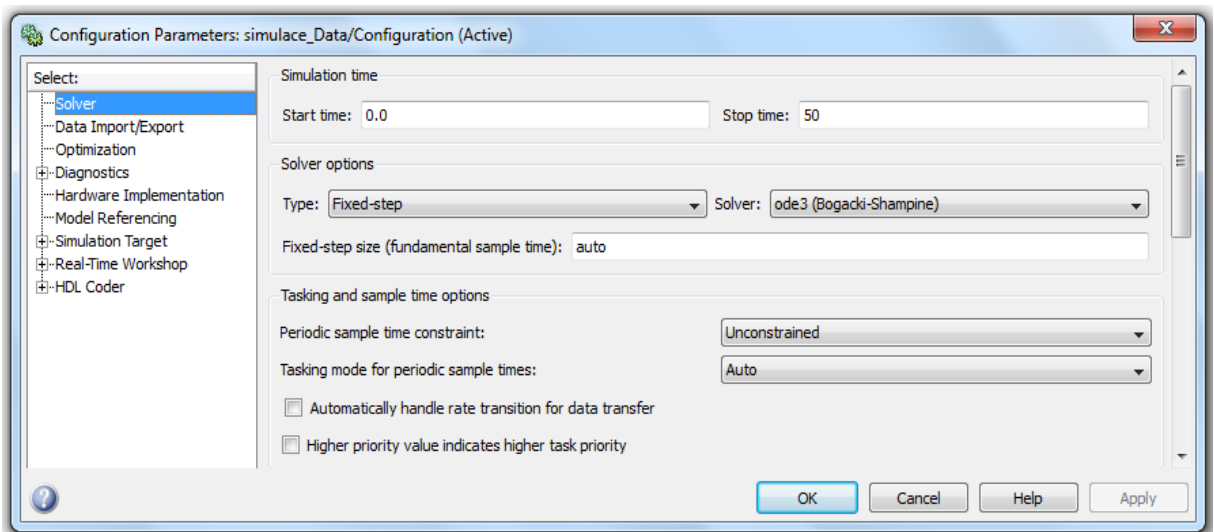
Obr. 22 – Tabulka nastavení parametrů bloku Scope

Na této kartě je nutné upravit nastavení. Zaškrtnutá položka Limit data points to last obsahuje maximální počet dat, které je Simulink schopen zaznamenat a převést do Matlabu (většinou je zvoleno maximum 5000 vzorků, to není nutné nijak měnit). Důležité ale je, aby byla zaškrtnuta možnost Save data to workspace, aby vůbec došlo k uložení naměřených dat. Položka Variable name musí obsahovat název proměnné, do které se tato data budou postupně ukládat (v tomto případě Data1). Položka Format pak říká, jakým způsobem budou data uložena a jak bude možné s nimi pracovat. Je potřeba zaškrtnout možnost Array. Poté budou



data uložena do pole, nebo lépe řečeno do matice. Pomocí tlačítka OK bude tato změna nastavení uložena.

Po zavření bloku Scope je ještě vhodné nastavit pevný vzorkovací krok pro měřená data. Tím dojde k tomu, že data budou měřena ve stejných časových úsecích. Toto nastavení může být často důležité právě při identifikaci a různých výpočtech. V Simulinku je toto nastavení umístěno na hlavním panelu na kartě Simulation ve volbě Configuration Parameters... Otevře se nabídka, která je zobrazena na obr. 23.



Obr. 23 – Tabulka nastavení parametrů simulace

Na záložce Solver v nastavení Solver Options pak musí být nastavena možnost Fixed-step. Stiskem OK se opět tato změna konfigurace uloží.

Nyní po spuštění simulace už dojde k uložení dat do proměnné a jejich importu do Matlabu. Proměnná má specifický název, který jsme zadali v položce Variable name v tabulce Parameters a ihned po spuštění simulace je viditelná v tabulce Workspace Matlabu, stejně jako ostatní vytvořené proměnné. Na obr. 24 níže je panel Workspace s novou proměnnou Data1.

Name	Value	Min	Max
Data1	<51x2 double >	-0.1815	50
tout	<51x1 double >	0	50

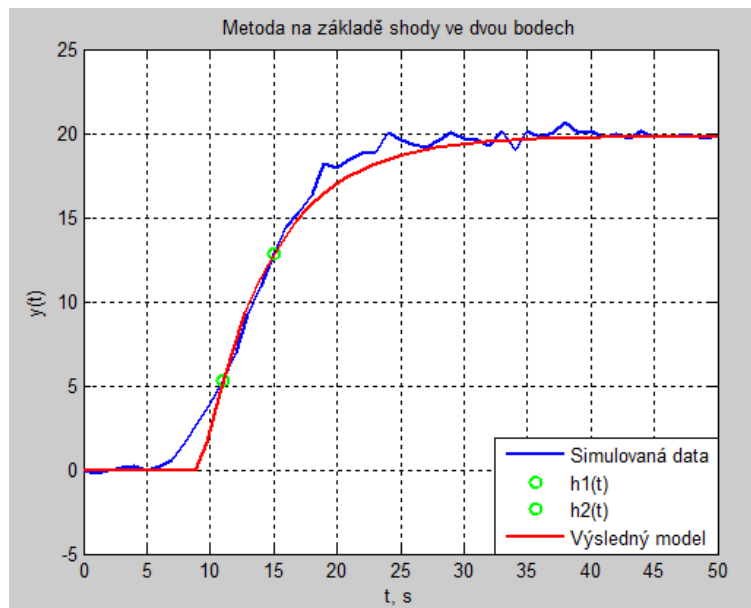
Obr. 24 – Ukázka panelu Workspace po simulaci

Vedle názvu proměnné je uveden i počet naměřených dat a minimální a maximální naměřená hodnota. Data jsou uložena ve dvou sloupcích. První sloupec obsahuje hodnoty času a druhý sloupec hodnoty měřené veličiny. Proto lze k této proměnné přistupovat stejně jako k matici.

### 3.3 Výsledky bakalářské práce

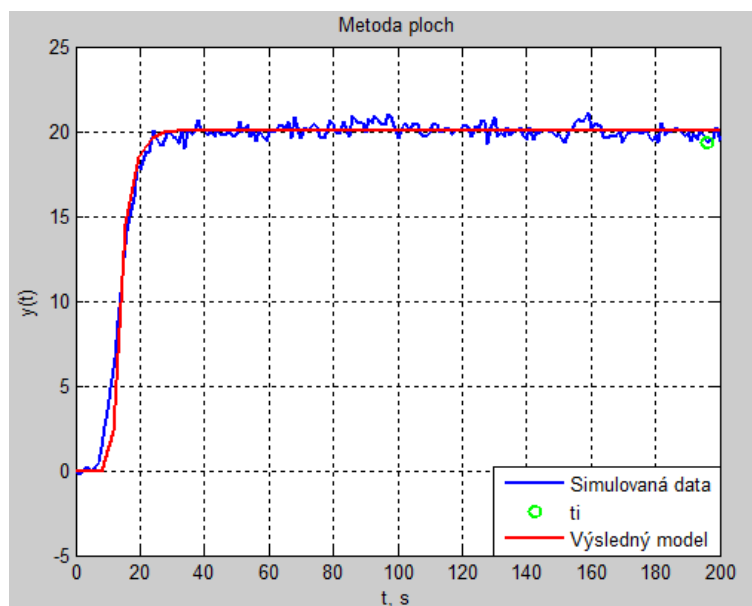
Tato část bakalářské práce obsahuje grafické výsledky jednotlivých metod identifikace, jejich porovnání mezi sebou. Důležité je především porovnat získaná data z dynamického systému vyššího řádu s daty získanými z modelu dynamického systému 1. řádu s dopravním zpožděním, vytvořeného na základě získaných parametrů pomocí metod identifikace. Pak také bude modifikován šum ovlivňující vlastnosti dynamického systému vyššího řádu a bude graficky sledována jejich případná odchylka od předchozích výsledků. Vše bude až zhodnoceno v závěru bakalářské práce.

První metodou byla metoda na základě shody ve dvou bodech. Na obr. 25 je vidět porovnání původních naměřených dat se získanými daty z vytvořeného modelu soustavy 1. řádu s dopravním zpožděním. Jsou zde i zobrazeny shodné body, ze kterých metoda vychází.



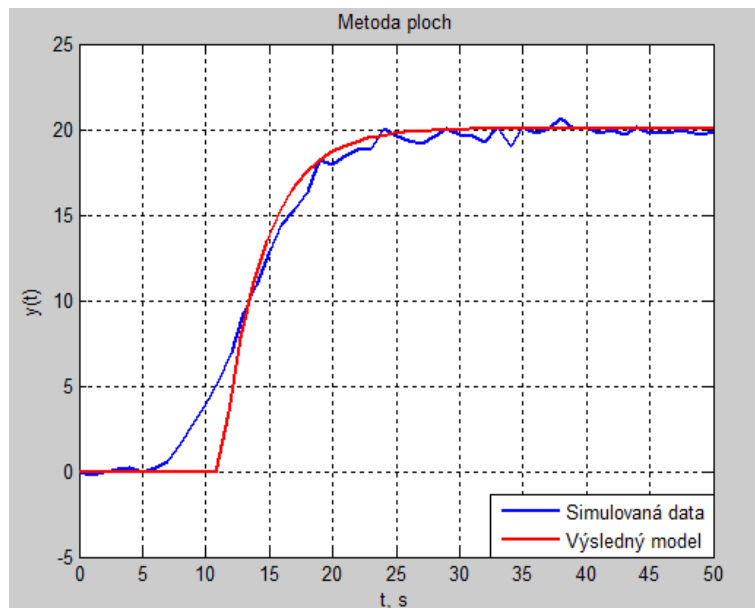
Obr. 25 – Výsledky metody na základě shody ve dvou bodech

Na obr. 26 je zobrazeny výsledné přechodové charakteristiky pro metodu ploch. Pro určení doby  $t_i$ , která je zobrazena ve grafu, byla zvětšena doba simulace. Bohužel z tohoto obrázku nelze vyhodnocovat výsledky této metody.



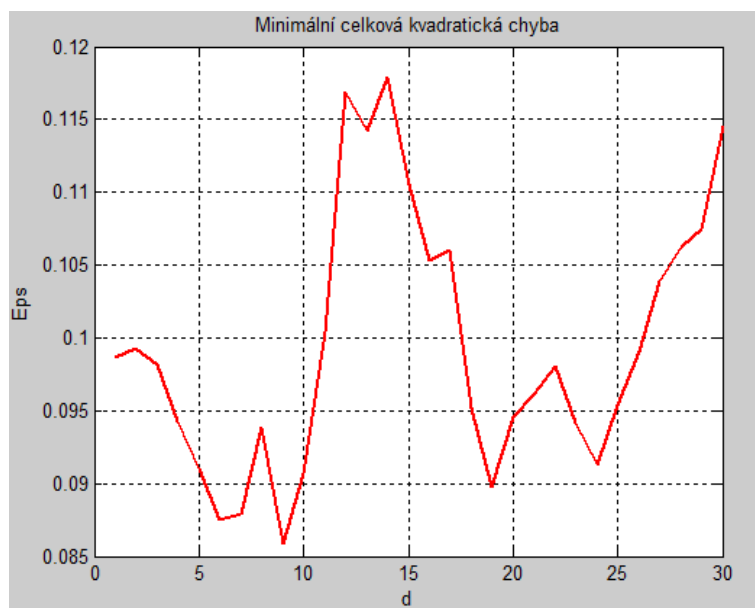
Obr. 26 – Výsledky metody ploch

Proto je na obr. 27 zobrazena ještě jednou s kratší dobou simulace.



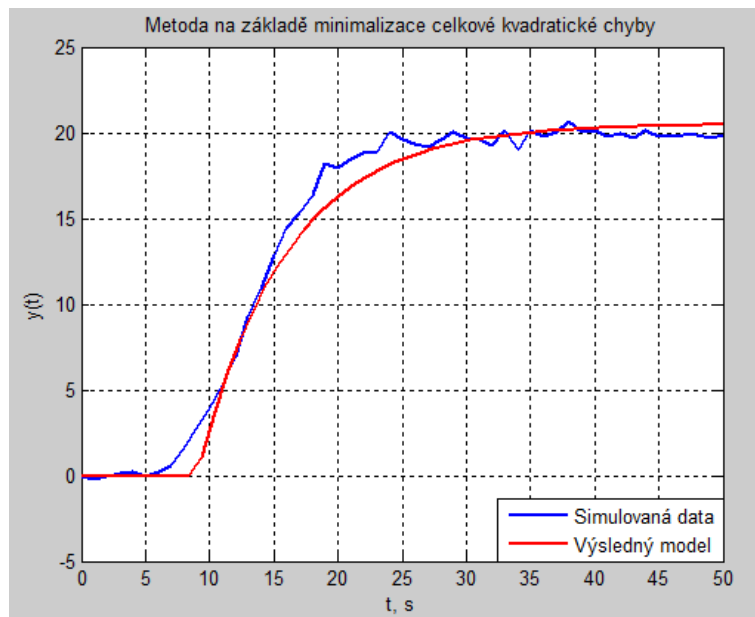
Obr. 27 – Výsledky metody ploch (po úpravě)

Poslední metodou je metoda na základě minimalizace celkové chyby. Na obr. 28 jsou graficky vyobrazeny hodnoty celkových kvadratických chyb pro různé časové posuny. Z Obrázku je vidět, že celková chyba je nejmenší pro časový posun s hodnotou 9.



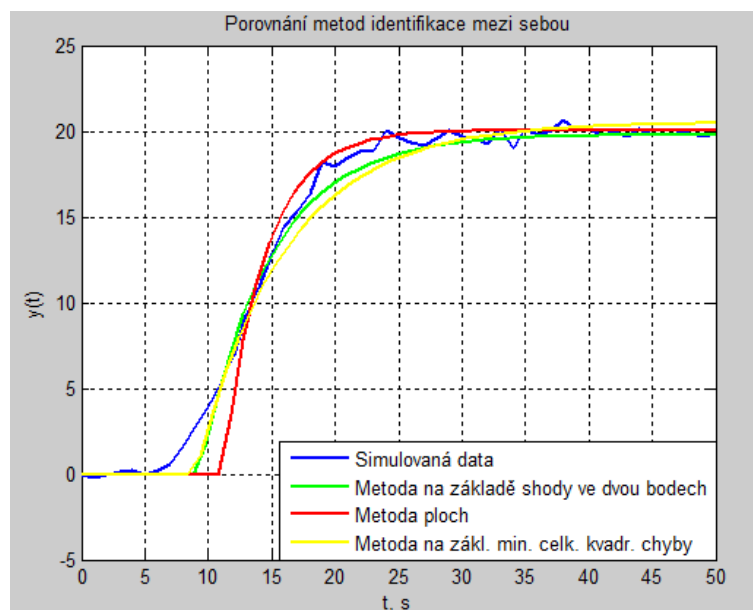
Obr. 28 – Určení minimální celkové kvadratické chyby

Na obr. 29 jsou pak vidět dosažené výsledky pomocí této metody.



Obr. 29 – Výsledky metody na základě minimalizace celkové kvadratické chyby

Na obr. 30 jsou pak všechny tři metody porovnány mezi sebou. Z tohoto obrázku, lze metody porovnat jednoznačně porovnat mezi sebou.



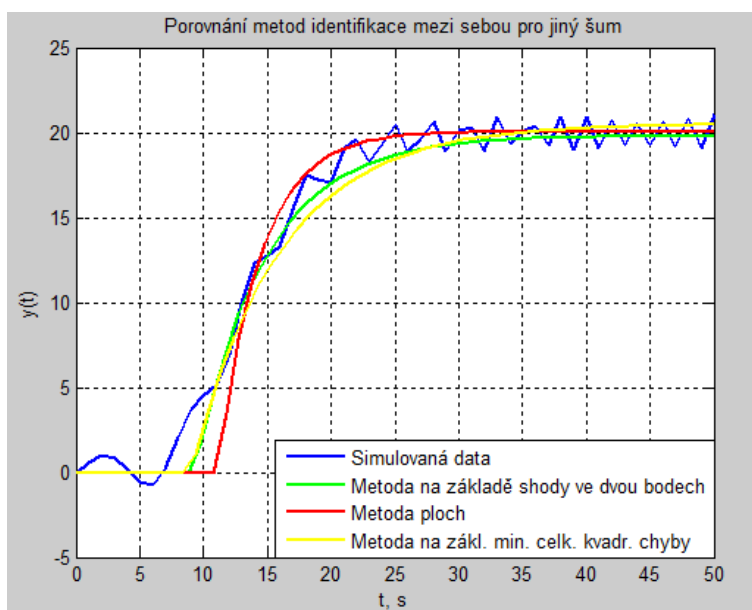
Obr. 30 – Porovnání metod identifikace mezi sebou

Ale aby bylo možné zhodnotit i přesnost výsledků za jiných podmínek. Bude dynamický systém vyššího řádu modifikován, nebo ani ne tak systém, jako působící šum. Na vstupu i na výstupu systému bude místo dosavadního šumu, umístěn blok, který samovolně vytváří šum ve formě různých signálů. Pro tyto účely byl vybrán blok Chirp Signal z knihovny Sources. Jeho nastavení je zbytečné probírat. Ukázka tohoto bloku je na obr. 31.



Obr. 31 – Blok Chirp Signal

Na obr. 32 níže, je možné vidět reakci všech metod na tuto změnu.



Obr. 32 – Porovnání metod identifikace mezi sebou pro jiný šum

### 3.4 Další možnosti práce s daty

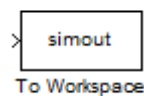
Tato kapitola není zcela součástí zadání bakalářské práce, ale k tomuto tématu se určitě vztahuje. Proto je uvedena až na konci bakalářské práce. Je zaměřena na jiné způsoby zpracování dat v Matlabu a jejich stručný popis. Samozřejmě nemá smysl zde uvádět

všechny, a proto byly vybrány pouze dvě.

### 3.4.1 Metoda importu dat pomocí To Workspace

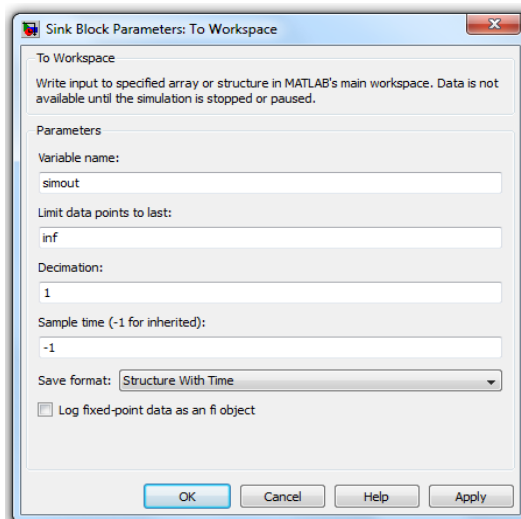
Tato metoda je velmi podobná již popsané použité metodě importu dat z bloku Scope, ale je více používaná. Na výstup dynamického systému bude připojen tentokrát blok To Workspace, který je v knihovně Sinks.

Ukázka tohoto bloku je na obr. 33.



Obr. 33 – Blok To Workspace

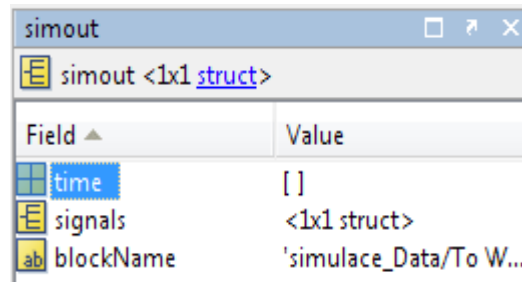
Po otevření bloku To Workspace se otevře tabulka nastavení, která obsahuje několik parametrů pro nastavení. Důležitý je znovu parametr Variable name, kde se opět nastavuje název proměnné, do které se budou data ukládat a Save format, kde musí být nastavena možnost Structure with the time. Celá tabulka je na obr. 34.



Obr. 34 – Tabulka nastavení bloku To Workspace

Po uložení nastavení a spuštění simulace se opět ve Workspace objeví proměnná s daty.

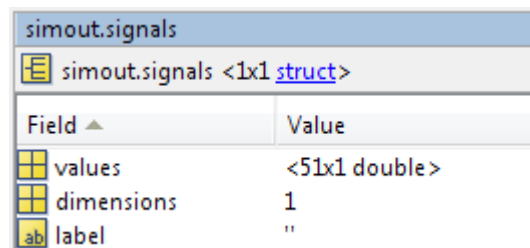
Tentokrát ale jde o datovou strukturu. Při přístupu k těmto datům se využívá tzv. tečková notace (pozn.: ta se využívá například v programovacím jazyce C, právě při přístupu k položkám statických datových struktur). Otevřením této proměnné ve Workspace se ve Variablen editoru otevře struktura této proměnné, která je na obr. 35.



Field ▲	Value
time	[]
signals	<1x1 struct>
blockName	'simulace_Data/To W...

Obr. 35 – Struktura proměnné To Workspace

Důležitá je položka time, kde se ukládá čas a signals, která obsahuje další proměnné. Na obr. 36 je zobrazena struktura podproměnné signals.



Field ▲	Value
values	<51x1 double>
dimensions	1
label	''

Obr. 36 – Struktura podproměnné signals z To Workspace

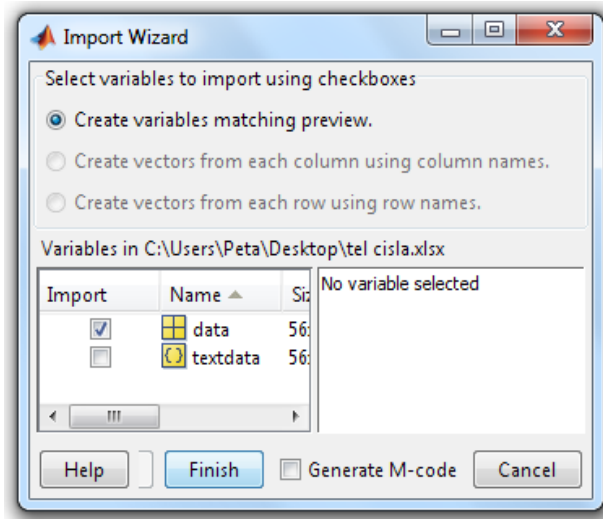
V této proměnné je i položka values, která obsahuje hodnoty měřené veličiny, uloženou ve svislém vektoru. V případě měření více veličin se další hodnoty ukládají v dalších vektorech.

### 3.4.2 Import dat do Matlabu

V případě měření na skutečném zařízení lze data importovat do Matlabu pomocí Import Wizard. Jsou i další způsoby, ale tento je asi nejpoužívanější. Tohoto průvodce importu lze najít na hlavním panelu Matlabu na kartě File ve volbě Import data... Zobrazí se tabulka Import Data pro vyhledání souboru, ve kterém jsou data uložena. Po nalezení se už otevře



Import Wizard, ve kterém je většinou více položek (záleží na obsahu souboru s daty). Položka data obsahuje naměřené hodnoty a v textdata se často ukládají textová data, pokud tam tedy nějaká jsou. Ta nejsou víceméně potřeba, tak nemusí být zaškrtnuta v tabulce Import Wizard pro import. Poté stačí kliknout na tlačítko Finish a data budou načtena do Matlabu, kde se zobrazí ve Workspace. Název proměnné se shoduje s názvem importované položky. Tabulka Import Wizard je zobrazena na obr. 37.



Obr. 37 – Tabulka Import Wizard

## 4 Závěr

Na závěr je důležité říct, že se povedlo splnit cíl této bakalářské práce. Metody identifikace byly úspěšně implementovány ve výpočtovém softwaru Matlab. Původně měly tyto metody být aplikovány na měřená data, získaná ze skutečného reálného systému, ale nakonec byl dynamický systém vyššího řádu simulován v simulačním prostředí Simulink, stejně jako dynamický systém 1. řádu s dopravním zpožděním. Výhodou tohoto řešení je, že lze dynamický systém vyššího řádu jednoduše ovlivňovat šumem a v datech tak vyvolat určitou chybu. Tento šum tak lze jednoduše změnit a skutečně ověřit chování jednotlivých metod identifikace. U skutečného systému by to šlo udělat také, působením nějaké veličiny na tento systém nebo by šlo získat měřená data z více dynamických systémů vyššího řádu a na ně pak aplikovat tyto metody a výsledky pro ně porovnat. To by samozřejmě šlo i u této metody, stačilo by například změnit řád systému. V této úloze byl ale nakonec řád systému zachován a systém byl určitými typy šumů. Šumů je samozřejmě více, nebo si lze nějakou poruchu i vytvořit, tady byly zkoušeny pouze dva základní typy šumu, které byly součástí knihoven Simulinku. Nyní budou vyzkoušené metody hodnoceny. Asi je vhodné říct v tuto chvíli, že názvy metod nejsou asi zcela oficiální. V literatuře se mohou objevit pod jinými názvy nebo jejich název může přeložen trochu jinak.

První metodou byla metoda na základě shody hodnot ve dvou bodech. Vzhledem k působení šumu, který způsoboval chybu v naměřených datech, bylo statické zesílení určeno průměrnou hodnotou z několika posledních naměřených hodnot. Ostatní parametry byly získány již popsáním způsobem. Důležitým krokem této metody bylo správně vybrat právě dva shodné body. Měly zhruba odpovídat  $1/3$  a  $2/3$  statického zesílení, což nebylo možné zcela přesně určit, vzhledem k počtu naměřených datům. U přechodových charakteristik obou systémů bylo dosaženo poměrně dobré shody.

Druhou metodou byla metoda ploch. Tato metoda dostala nakonec přednost, před metodou momentů. Metody jsou sice postaveny na jiných principech, ale výsledky dávají skoro úplně stejné. Princip metody je založen především na výpočtu ploch pod přechodovou a nad přechodovou charakteristikou. Důležité bylo především přesně určit konstanty, které jsou horními mezemi integrálů a taky jejich samotný výpočet, kvůli kterému bylo nutné aplikovat lichoběžníkovou metodu. Zesílení bylo určeno stejně jako u první metody. I u této metody bylo dosaženo poměrně dobré shody.

Poslední metodou byla metoda na základě celkové minimalizace kvadratické chyby. Tato metoda jako jediná je založena principu diskrétního přenosu, kdy se po úpravě vychází z řešení diferenčních rovnic. Prvním krokem bylo jednoznačně určit minimální hodnotu kvadratické celkové chyby, závislou na časovém posunu, což se povedlo poměrně přesně. Poté bylo dohledáno řešení diferenčních rovnic, kterým byly koeficienty těchto diferenčních rovnic. Následně z koeficientů byly určeny parametry pro vytvoření modelu dynamického systému 1. řádu. Rovněž i u poslední metody bylo dosaženo dobré shody.

Podářilo se úspěšně vyzkoušet všechny uvedené metody, které dávají poměrně dobré výsledky. Ale nejlepším způsobem hodnocení by jistě bylo, podle získaných hodnot parametrů, nastavit skutečný regulátor a sledovat jeho regulační pochod.

Tato bakalářská práce byla velmi zajímavá a byla pro mne velkým přínosem a určitě by bylo zajímavé, jí dál rozvíjet.

## Literatura

- [1] CVEJN, J. *Řízení procesů: Elektronický studijní materiál pro předmět Automatizace 1*. Pardubice: Univerzita Pardubice, 2006, 114 s.
- [2] ÅSTRÖM, K. J. and T. HÄGGLUND. *PID Controllers: Theory, Design, and Tuning*. 2nd edition. Instrument Society of America (ISA), 1995.
- [3] KIONG, T. K., QUING-GUO, W., CHIECH, H. C. and T. J. HÄGGLUND. *Advances in PID Control*. Springer Verlag, 1999.
- [4] TAUFER, I., KOTYK, J., aj. *Algoritmy a algoritmizace – vývojové diagramy*. Pardubice: Univerzita Pardubice, 2009, 92 s. ISBN 978-80-7395-182-5.

## Příloha A–Zdrojový kód metody na základě shody ve dvou bodech

```
t=Data1(:,1);
y=Data1(:,2);
plot(t,y,'LineWidth',2)
hold on
grid on
xlabel('t, s')
ylabel('y(t)')
title('Metoda na základě shody ve dvou bodech')
t1=t(12);
h1=y(12);
t2=t(16);
h2=y(16);
plot(t1,h1,'go',t2,h2,'go','LineWidth',2)
K=mean(y(25:end))
T=(t2-t1)/log((1-(h1/K))/(1-(h2/K)))
tau=T*log(1-(h2/K))+t2
plot(Data2(:,1),Data2(:,2),'r','LineWidth',2)
legend('Simulovaná data','h1(t)','h2(t)','Výsledný model');
hold off
```

## Příloha B–Zdrojový kód metody ploch

```
t=Data3(:,1);
y=Data3(:,2);
plot(t,y,'LineWidth',2)
K=mean(y(50:end))
ti=177;
hold on
grid on
plot(t(197),y(197),'go','LineWidth',2)
xlabel('t, s')
ylabel('y(t)')
title('Metoda ploch')
Datapom=K-y
I=((Datapom(1)+Datapom(200))/2+sum(Datapom(2:199)))*1
tar=I/K
tarz=round(tar);
A1=((y(1)+y(tarz))/2+sum(y(2:tarz-1)))*1
T=(exp(1)*A1)/K
tau=tar-T
plot(Data4(:,1),Data4(:,2),'r','LineWidth',2)
legend('Simulovaná data','ti','Výsledný model')
hold off
```

## Příloha C–Zdrojový kód metody na základě minimalizace celkové kvadratické chyby

```
t=Data5(:,1);
u=Data5(:,2);
y=Data5(:,3);
N=length(Data5);
delta=1;
for d=1:30
A=[y(1+d:N-1) u(1:N-1-d)];
B=[y(2+d:N)];
x=inv(A'*A)*A'*B;
Eps=A*x-B;
Eps2(d)=sqrt(Eps'*Eps)/length(Eps);
end
plot(1:30,Eps2,'r','LineWidth',2),grid
xlabel('d')
ylabel('Eps')
title('Minimální celková kvadratická chyba')
Epsmin=min(Eps2)
for d=1:30
if(Epsmin==Eps2(d))
d;
A=[y(1+d:N-1) u(1:N-1-d)];
B=[y(2+d:N)];
x=inv(A'*A)*A'*B;
a=x(1)
b=x(2)
T=-delta/log(a)
K=b/(1-a)
tau=d*delta
end
end
figure
```

```
plot(t,y,'LineWidth',2),grid
hold on
plot(Data6(:,1),Data6(:,2),'r','LineWidth',2)
xlabel('t, s')
ylabel('y(t)')
title('Metoda na základě minimalizace celkové kvadratické chyby')
legend('Simulovaná data','Výsledný model')
hold off
```