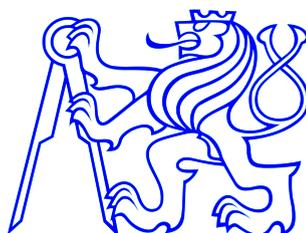


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Identifikační toolbox pro Scilab

Martin Novotný



Diplomová práce

Fakulta elektrotechnická
Katedra řídicí techniky

12. května 2011

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Martin Novotný**

Studijní program:
Obor: Systémy a řízení

Název tématu: **Identifikační toolbox pro Scilab**

Pokyny pro vypracování:

1. Seznamte se s používanými lineárními identifikačními metodami a strukturami modelů.
2. Navrhněte strukturu identifikačního toolboxu a implementujte vybrané struktury modelů a vybrané identifikační metody.
3. Diskutujte možnost inicializačních metod pro identifikaci systémů.
4. Implementujte Prediction Error Method (PEM) pro vybrané struktury modelů inicializovanou pomocí subspace identifikačních metod

Seznam odborné literatury:

- [1] Yuacai Zhu, Multivariable System Identification for process Control, Oxford, 2001
- [2] Lennart Ljung, System Identification, New Jersey, 1999
- [3] Michel Verhaegen, Vincent Verdult, Filtering and system identification: a least squares approach, Cambridge University Press, 2007 New York
- [4] Christian Lyzell, Initialization Methods for System Identification, Dissertation Thesis, Linköping, Sweden, 2009

Vedoucí: Ing. Samuel Prívára

Platnost zadání: do konce zimního semestru 2011/2012



prof. Ing. Michael Šebek, DrSc.
vedoucí katedry



prof. Ing. Boris Šimák, CSc.
děkan

V Praze dne 15. 9. 2010

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne _____

podpis

Poděkování

Na tomto místě bych rád poděkoval Ing. Samuelu Prívarovi za příkladné vedení této práce, jeho podporu, trpělivost, rady a inspirace. Velký dík patří také Ing. Zdeňku Váňovi za cenné rady a připomínky k praktickým implementacím.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Anotace

Fakulta elektrotechnická

Katedra řídicí techniky

Diplomová práce

Tato diplomová práce se zabývá vývojem identifikačního toolboxu pro programové prostředí *Scilab*, podobnému *System Identification ToolBox* (SITB), který je implementován v *Matlabu*.

Práce je zaměřena jednak na implementaci samotného identifikačního toolboxu (Identifikační toolbox pro *Scilab*) s vybranými standardními prvky, které nabízí i komerční SITB, a jednak na novinky a nezvyklé implementace z problematiky oblasti identifikace lineárních dynamických systémů, jakými jsou např. použití metody *NonNegative Garrote* (NNG) pro redukci řádu lineárních modelů, *Errors-In-Variables* (EIV) identifikace, *Subspace State-Space System Identification* (4SID), *Prediction Error Method* (PEM) inicializovaná 4SID metodou, využití *Black-Box* (BB) modelování pro identifikaci fyzikálních parametrů systému a implementace *square-root* Kalmanova filtru.

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Abstract

Faculty of Electrical Engineering
Department of Control Engineering

Master Thesis

This thesis deals with the development of the identification toolbox for `Scilab` environment, which is similar to the *System Identification ToolBox* (SITB) for `Matlab`.

First, the thesis is focused on the development of the identification toolbox (Identification Toolbox for `Scilab`) with selected standard features, which also offers the commercial SITB. Second, we deal with new methods from the system identification research area, such as *NonNegative Garrote* (NNG) method for order selection of linear models, *Errors-In-Variables* (EIV) identification, *Subspace State-Space System IDentification* (4SID), *Prediction Error Method* (PEM) initialized by 4SID method, the use of *Black-Box* (BB) modeling for estimating a physical parameters of a system and the *square-root* implementation of Kalman filter.

Obsah

Anotace	iv
Abstract	v
Použité značení	ix
Seznam obrázků	x
Seznam tabulek	xii
1 Úvod	1
1.1 Dynamické systémy	1
1.2 Modely dynamických systémů	1
1.3 Identifikace dynamických systémů	2
1.4 Scilab	3
1.5 Cíle práce	4
1.6 Struktura práce	4
2 Příprava dat, návrh identifikace	5
2.1 Návrh identifikačního experimentu	5
2.1.1 Volba vzorkovací frekvence	6
2.1.2 Analýza přechodové charakteristiky	7
2.1.3 Doba trvání experimentu	7
2.1.4 Volba vstupního signálu	7
2.2 Předzpracování dat	8
2.2.1 Filtrace	9
2.2.2 Převzorkování dat a odstranění trendů	9
2.3 Výběr struktury modelu	10
2.4 Identifikace	11
2.5 Validace modelu	11
3 Matematické nástroje	13
3.1 Metoda nejmenších čtverců (LS)	13
3.2 Problém úplných nejmenších čtverců (TLS)	15
3.2.1 Implementace metody TLS	16
3.2.2 Řešení TLS regularizací Tikhonovova typu (RTLS)	17
3.3 Metoda instrumentálních proměnných	19
3.4 Nonnegative Garrote	19

3.4.1	Řešení NNG problému	21
3.4.2	Redukce řádu ARX modelu	22
3.4.3	Redukce řádu ARMAX modelu	25
4	Systémy a modely	28
4.1	Struktura obecného LTI modelu	28
4.1.1	Terminologie	29
4.2	Optimální prediktor	31
4.3	Používané LTI modely	31
4.3.1	ARX	31
4.3.2	ARMAX	33
4.3.3	Output Error (OE)	34
4.3.4	Box - Jenkins (BJ)	35
4.3.5	State space	37
4.4	Kalmanův filtr	37
4.4.1	Asymptotický pozorovatel stavu	37
4.4.2	Definice KF	38
4.4.3	Implementace KF	39
4.4.4	Square-root implementace KF	39
4.4.5	Použití Kalmanova filtru k odhadu stavu systému 2. řádu	41
5	Identifikační metody	45
5.1	LMS odhad	45
5.2	ML odhad	45
5.3	Prediction error metoda (PEM)	46
5.4	Output error metoda (OEM)	47
5.5	Subspace	48
5.5.1	Princip Subspace identifikace	48
5.5.2	Použití subspace identifikace	52
5.6	Inicializace PEM pomocí subspace	53
5.6.1	PEM identifikace ARMAX modelu inicializovaná pomocí subspace	57
5.7	Errors-in-variables (EIV) identifikace	57
5.7.1	EIV identifikace ARX modelu	58
5.8	Odhad fyzikálních parametrů systému s využitím Black-Box (BB) identifikace	58
5.8.1	Zahrnutí strukturální informace do black-box identifikace	61
5.8.2	Použití black-box identifikace pro odhad fyzikálních parametrů systému	62
6	Dokumentace k identifikačnímu toolboxu a navrženým algoritmům	64
6.1	Struktura toolboxu	64
6.2	Obslužné skripty	64
6.3	Utility	65
6.4	Matematické nástroje	65
6.5	Předzpracování dat	66
6.5.1	Úprava signálu obsahujícího NaN hodnoty	66
6.5.2	Převzorkování signálů	67
6.5.3	Odstranění středních hodnot a trendů	67
6.5.4	Filtry	67
6.6	Identifikační algoritmy	68

6.7	Algoritmy pro redukci řádu modelu	70
6.8	Volba identifikačního kritéria a verifikace modelu	71
6.8.1	Volba identifikačního kritéria	71
6.8.2	Verifikace modelu	71
6.9	Dokumentace	72
6.10	Grafické uživatelské rozhraní	72
7	Závěr	74
A	Obsah priloženého CD	78

Použité značení

\mathbb{R}	množina všech reálných čísel
0_m	nulová matice rozměru $m \times m$
I_m	jednotková matice rozměru $m \times m$
$E[x]$	střední hodnota x
$\delta(k)$	jednotkový puls v čase k
\hat{x}	odhad x
$\text{tr}(X)$	stopa matice X
θ	vektor parametrů
$\ x\ _2$	$(\sum_{i=1}^n x_i^2)^{1/2}$
$\ x\ _F$	$\text{tr}(X.X^T)$

ATOMS	AuTomatic mOdules Management for Scilab
AIC	Akaikes Information Criterion
BB	Black-Box
EIV	Errors In Variables
IV	Instrumental Variables
LMS	Linear Mean Square
LS	Least Squares
LTI	Linear Time Invariant
MDL	Minimum Description Length
ML	Maximum Likelihood
NNG	NonNegative Garrote
OEM	Output Error Method
PEM	Prediction Error Method
SID	Subspace Identification
SITB	System Identification Toolbox [®]
SRCF	Square-Root Covariance Filter
SVD	Singular value decomposition
TLS	Total Least Squares

Seznam obrázků

1.1	Stavový popis systému	2
1.2	Schématické znázornění identifikačního procesu na neznámém systému	3
2.1	Identifikační procedura [1]	6
2.2	Různé druhy vstupních signálů pro identifikaci [2]	8
2.3	Odstranění trendu z dat	10
3.1	Geometrická interpretace metody LS	14
3.2	Geometrická interpretace metody TLS	16
3.3	ARX: Závislost fit faktoru na počtu iterací λ	23
3.4	ARX: Vizualizace obsazenosti koeficientů v θ v závislosti na počtu iterací λ (plné body označují nenulové koeficienty)	24
3.5	ARX: Porovnání NNG a <code>selstruc</code>	25
3.6	ARX: Konvergence parametrů θ k nule s rostoucím počtem iterací λ v detailním pohledu na nejdůležitější koeficienty	25
3.7	ARMAX: Závislost fit faktoru na počtu iterací λ	26
3.8	ARMAX: Vizualizace obsazenosti koeficientů v θ v závislosti na počtu iterací λ (plné body označují nenulové koeficienty)	27
3.9	ARMAX: Konvergence parametrů θ k nule s rostoucím počtem iterací λ v detailním pohledu na nejdůležitější koeficienty	27
3.10	ARMAX: Porovnání NNG a <code>balred</code>	27
4.1	Struktura obecného LTI modelu [3]	30
4.2	Terminologie LTI modelů	30
4.3	Simulace a predikce	32
4.4	ARX model	32
4.5	ARMAX model	34
4.6	OE model	35
4.7	BJ model	36
4.8	Průběhy odhadu stavů systému Kalmanovým filtrem a square-root kovariančním filtrem	43
4.9	Průběhy odhadu výstupů systému Kalmanovým filtrem a square-root kovariančním filtrem	43
4.10	Průběhy chyb odhadů výstupů systému Kalmanovým filtrem a square-root kovariančním filtrem	44
4.11	Autokorelační funkce chyb odhadů výstupu systému Kalmanovým filtrem a square-root kovariančním filtrem	44
5.1	Prediction-error metoda	46
5.2	Output-error metoda	47

5.3	Testovací data <code>dryer.mat</code>	52
5.4	Global optimum	54
5.5	EIV identifikační problém	58
5.6	Zahrnutí strukturální informace do black-box identifikace	62
5.7	Chyba odhadu neznámých parametrů θ_0 v závislosti na počtu iterací	63
6.1	Grafické uživatelské rozhraní	73

Seznam tabulek

3.1	ARX: Porovnání NNG a <code>selstruc</code>	24
3.2	ARMAX: Porovnání NNG a <code>balred</code>	26
4.1	Porovnání výpočetních rychlostí algoritmů odhadu stavu systému klasickým (KF) a square-root kovariančním filtrem (SRKF) v <code>Matlabu</code> a ve <code>Scilabu</code>	43
5.1	Porovnání různých implementací subspace identifikace (pozn. matice A je pro úsporu místa vypsána do sloupce; $\hat{A} \sim \text{vec}(\hat{A})$)	53
5.2	PEM identifikace ARMAX modelu, PEM identifikace ARMAX modelu inicializovaná IV a 4SID metodami	57
5.3	EIV: Porovnání úspěšnosti identifikace ARX modelu s použitím LS, TLS a RTLS při různých úrovních vstupního a výstupního šumu, měřenou fit faktorem	59
6.1	Struktura toolboxu	64

Kapitola 1

Úvod

1.1 Dynamické systémy

Dynamický systém sestává ze stavového prostoru, jehož souřadnice popisují stav systému v daném čase a z dynamických podmínek, které popisují změnu tohoto systému v čase. Stav systému je popsán vektorem, který leží ve stavovém prostoru. Dynamické podmínky jsou určeny modelem systému (Kap. 1.2), které popisují změnu stavového vektoru v čase. Dynamický systém může být deterministický nebo stochastický, spojitý nebo diskrétní. Systémy se dále dělí na lineární a nelineární. Lineární systém je takový systém, v němž lze uplatnit princip superpozice. To lze ilustrovat následujícím příkladem: jestliže $f(x) = 0$ a současně $f(y) = 0$, potom také $f(x + y) = 0$. Chování lineárních systémů lze při znalosti počátečních podmínek a modelu předpovědět i do budoucnosti. Oproti tomu v nelineárním systému platí princip superpozice pouze pro malou množinu izolovaných bodů, tzv. pracovních bodů. Je-li systém nelineární a nelze využít principu superpozice, je nutné pro výpočet změny stavu systému řešit diferenciální rovnice, což je mnohdy značně obtížné. Také zde není zaručeno, že se podaří předpovědět stav systému do budoucnosti. Dále uvažujme pouze lineární, časově invariantní (LTI) diskrétní systémy, neboť právě jejich identifikaci se práce zabývá. Podrobnosti o dynamických systémech nabídne [4].

1.2 Modely dynamických systémů

Analýza a syntéza dynamických systémů se realizuje pomocí matematického modelu. Dynamické vlastnosti reálných systémů se všemi vazbami a interakcemi lze, jen stěží, vyjádřit dostatečně obecným a v praxi použitelným matematickým modelem. Zavádí se proto nejdříve zjednodušující předpoklady, které umožní vytvořit zjednodušený fyzikální model. Matematický model se pak odvozuje z fyzikálních zákonů aplikovaných na tento fyzikální model nebo, pomocí různých identifikačních metod, na základě pozorování vstupů a výstupů zkoumaného dynamického systému.

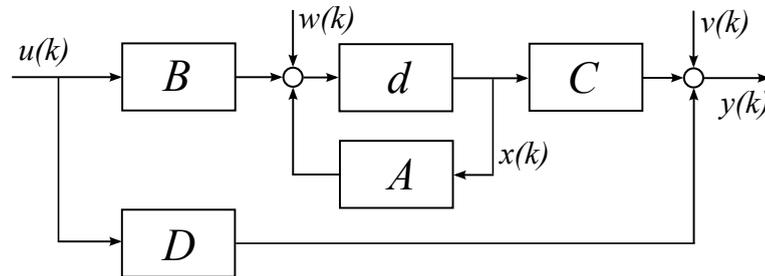
Dynamiku systému lze vyjádřit vnějším nebo vnitřním popisem. Vnější popis systému je vyjádření dynamických vlastností pomocí relací mezi vstupní a výstupní veličinou $u(k)$ a $y(k)$. Lineární, časově invariantní diskrétní dynamický systém popisuje diferenční rovnice s konstantními koeficienty

$$y(n) + a_{n-1}y(n-1) + \dots + a_1y(1) + a_0y = b_mu(m) + \dots + b_1u_1 + b_0u + c_{m_c}v(m_c) + \dots + c_1v(1) + c_0v, \quad (1.1)$$

kde $v(k)$ představuje poruchovou veličinu (šum). Takový popis neposkytuje informaci o vnitřních stavech systému. Pozorováním vstupní a výstupní veličiny můžeme získat pouze vnější popis systému. Vnitřní popis systému chápeme jako relaci mezi vstupní veličinou $u(k)$, stavem systému $x(k)$ a výstupní veličinou $y(k)$. Hovoříme pak o stavovém popisu systému, který v současné době převládá. V případě, že je diskrétní dynamický systém lineární, časově invariantní, pak je jeho stavový popis dán rovnicemi

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= Cx(k) + Du(k) + v(k), \end{aligned} \quad (1.2)$$

kde $w(k)$ představuje šum procesu a $v(k)$ šum měření.



OBRÁZEK 1.1: Stavový popis systému

1.3 Identifikace dynamických systémů

Identifikace se zabývá nalezením modelu systému z pozorovaných dat. Identifikační metody jsou užívány v disciplínách zabývajících se automatickým řízením a zpracováním signálů, ale rovněž v ekonomii, ekologii či biologii a lékařství. Umožňují získat vhodné modely pro předvídání vývoje sledovaných veličin, sledování objektů, navigaci, detekci poruch, filtraci signálů, simulaci a zvyšování znalostí o identifikovaném systému, ale především pro návrh regulátorů.

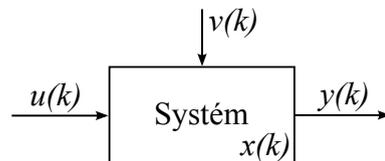
V disciplínách, zabývajících se automatickým řízením je identifikace bezpochyby klíčovým prvkem následného řízení. Aby bylo možné nějaký systém dobře řídit, je třeba vytvořit k němu kvalitní model. Existují dva základní přístupy, jak model dynamického systému vytvořit. Prvním z nich je tzv. **matematicko - fyzikální analýza** [5], kdy na základě různých fyzikálních,

ekonomických, či jiných zákonitostí, hledáme vztahy mezi veličinami, které jsou předmětem našeho zájmu. V rovnicích, popisujících vztahy mezi veličinami, se často vyskytují konstanty, jejichž hodnoty se nesnadno hledají. Model dynamického systému vzniklý po takové identifikaci bývá většinou natolik složitý, že ho není možné použít pro potřeby následného řízení. Druhý způsob vytváření modelu dynamického systému se nazývá **experimentální identifikace** [5] a je založen na určování vzájemných vztahů mezi sadou vstupně - výstupních dat, naměřených přímo na systému. Na systém nahlížíme jako na černou skříňku a pozorujeme pouze vstupní a výstupní data. Tomuto principu se říká **black - box** identifikace. V praxi se často využívá i kombinace matematicko - fyzikální analýzy s black - box přístupem. Pomocí analýzy určíme např. strukturu či řád systému. Při experimentální analýze se potom z černé skříňky stává skříňka šedá a hovoříme o tzv. **gray - box** identifikaci.

Základní myšlenka identifikace je založena na snaze nalézt vhodný matematický model popisující co nejpřesněji neznámý systém, máme-li k dispozici pozorování vstupu $u(k)$ a výstupu $y(k)$ na systému. Množinu pozorování označme jako

$$Z^N = (u(k), y(k))_{k=1}^N. \quad (1.3)$$

Obr. 1.2 schématicky znázorňuje pozorování na neznámém systému pro účely identifikace. Signál $v(k)$ představuje poruchu vstupující do systému. Existují různé přístupy k identifikaci neznámého systému. Ty nejběžnější jsou obvykle založeny na *Prediction-Error Method* (PEM) přístupu [1]. Mimo to existují i další metody, jejichž popisem se zabývá Kap. 5.



OBRÁZEK 1.2: Schématické znázornění identifikačního procesu na neznámém systému

1.4 Scilab

Syntaxe jazyka **Scilab** je velice podobná **Matlabu**, navíc **Scilab** je kompatibilní s jeho knihovnamí/skripty. Samotný systém je vyvíjen od roku 1990 vědeckými pracovníky z institucí INRIA a EnPC. O jeho kvalitách svědčí to, že je používán na řadě vědeckých institucí a zejména v průmyslu. Jeho součástí je i balík **Scicos**, který slouží k modelování a simulacím dynamických systémů (ekvivalent **Simulink**). **SciLab** je multiplatformní, tudíž je dostupný nejen ve verzi pro systémy Windows, ale i Linux a unixové systémy včetně Mac OS. Za nejslabší místo konkurentů **Matlabu** je považována absence kvalitních toolboxů. To však neplatí o **Scilabu**, jež nabízí širokou paletu rozšíření. Neobsahuje však žádný toolbox, který by se zabýval identifikací dynamických systémů.

1.5 Cíle práce

Předmětem této práce je vývoj identifikačního toolboxu pro programové prostředí `Scilab`, (<http://www.scilab.org>) podobného *System Identification ToolBoxu* (SITB), který je implementován v `Matlabu` (<http://www.mathworks.com/products/sysid/>). Komerční aplikace jako takové mají obecně problém s tím, že se novinky do výsledného release dostávají velmi pomalu. Tento fakt je ovšem jedním z největších pozitiv open-source aplikací. Motivací je tedy vývoj alternativního nástroje, pružnějšího k novinkám, ale také zaměřeného na aplikaci v rámci projektů na katedře řídicí techniky FEL ČVUT. Pro potenciální komerční využití, zejména menšími firmami, je navíc nasazování `Matlabu` nepřijatelné z důvodu nepřiměřeně vysokých cen. Existuje tedy snaha naprogramovat obdobu v jedné z free alternativ `Matlabu`, v tomto případě `Scilabu`.

Při samotném návrhu toolboxu se jedná o následující dílčí úkoly: návrh struktury toolboxu dle standardu <http://wiki.scilab.org/howto/Create%20a%20toolbox> a jeho začlenění do systému ATOMS (*AuTomatic mOdules Management for Scilab*) <http://wiki.scilab.org/ATOMS>, vytvoření sady skriptů pro úpravu dat, implementaci identifikačních úloh a vytvoření skriptů pro verifikaci. Jednotlivé skripty (funkce) budou dokumentované formou html helpu podobně jako v `Matlabu`. Každá funkce musí mít ošetřené vstupní argumenty. Implementované funkce se dají rozdělit do třech tříd, a to na funkce pro úpravu a předzpracování dat, samotné identifikační algoritmy a funkce pro verifikaci nalezeného modelu.

Práce je zaměřena jednak na implementaci samotného identifikačního toolboxu s vybranými standardními prvky, které nabízí i konkurenční SITB, a jednak na novinky a nezvyklé implementace z problematiky oblasti identifikace lineárních dynamických systémů.

1.6 Struktura práce

- **Kap. 1 - Úvod** - základních pojmy z oblasti identifikace lineárních dynamických systémů.
- **Kap. 2 - Příprava dat, návrh identifikace** - popis jednotlivých kroků identifikační procedury.
- **Kap. 3 - Matematické nástroje** - vybrané matematické nástroje používané v oblasti identifikací.
- **Kap. 4 - Systémy a modely** - vybrané struktury používaných lineárních modelů.
- **Kap. 5 - Identifikační metody** - popis vybraných identifikačních metod.
- **Kap. 6 - Dokumentace k identifikačnímu toolboxu a navrženým algoritmům**
- **Kap. 7 - Závěr** - shrnutí cílů práce a dosažených výsledků.

Kapitola 2

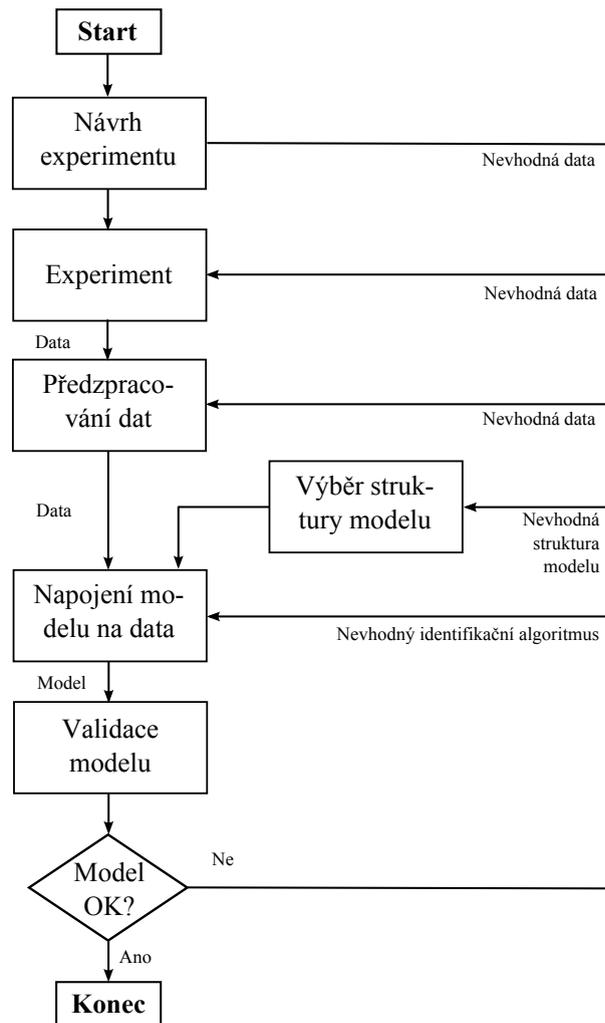
Příprava dat, návrh identifikace

Při současných postupech, kdy se pro sběr dat používají automatizované měřicí systémy, začíná být problémem obrovský objem změřených dat. Abychom mohli vůbec něco usuzovat, je třeba data nějakým způsobem zpracovat a získat pokud možno jednoduché a srozumitelné parametry popisující tato data, případně co možná nejjednodušší vztahy mezi jednotlivými měřenými veličinami.

V této kapitole budou popsány nejdůležitější fáze identifikační procedury znázorněné v Obr. 2.1 [1], od návrhu identifikačního experimentu, přes předzpracování dat, výběr struktury modelu až po validaci navrženého modelu. Cílem identifikační procedury je odhadnout parametry modelu o zvolené struktuře na základě znalosti časových řad vstupních a výstupních signálů. Samotná identifikace parametrů modelu je však pouze jednou z fází celé identifikační procedury. Prvním krokem je získání kvalitních dat. Na začátku celé procedury je třeba správně zvolit mechanismus digitálního zpracovávání dat, vzorkovací frekvenci, případně typy anti-aliasingových filtrů. Tato fáze se nazývá *návrh experimentu* neboli *experiment design* dle [1]. Ve chvíli, kdy vhodně navrheme identifikační experiment, přejdeme k jeho samotnému provedení. Experiment vede k získání takových dat, která jsou vhodná k provedení identifikace neznámého systému buď rovnou, nebo po předchozím předzpracování dat. Následuje volba struktury modelu, kterou [6] formálně definuje jako mapování mezi prostorem parametrů a identifikovaným modelem. Ve fázi, kdy jsou odhadnuty parametry modelu, přichází na řadu krok *validace modelu*, jehož úkolem je posoudit a kvantifikovat kvalitu navrženého modelu. Celou problematikou se detailněji zabývá zejména [1, 6].

2.1 Návrh identifikačního experimentu

Návrh experimentu by se měl odrážet zejména od fyzikálních znalostí o identifikovaném systému. Fyzikální modely jsou obvykle popsány diferenciálními rovnicemi popisujícími fyzikální



OBRÁZEK 2.1: Identifikační procedura [1]

zákonitosti. Tato sekce dále vychází z [6] a zabývá se problematikou vzorkování vstupních a výstupních posloupností dat pro identifikaci. Bude zde diskutována problematika volby doby trvání experimentu a ovlivnění návrhu experimentu vlastnostmi vstupního neboli budícího signálu.

2.1.1 Volba vzorkovací frekvence

Volba vzorkovací frekvence je klíčovým prvkem úspěšné identifikace. Platí Shannonův vzorkovací teorém, který říká, že pokud spojitý, frekvenčně omezený signál obsahuje frekvence z pásma $[-\omega_B, \omega_B]$ (rad/s), tak volba vzorkovací frekvence $\omega_S \geq 2\omega_B$ (rad/s) zajistí přesnou rekonstrukci spojitého signálu z provedených vzorků. Vhodnou volbou vzorkovací frekvence lze předejít známému nežádoucímu efektu *aliasingu* [7].

2.1.2 Analýza přechodové charakteristiky

Dalším mezikrokem při návrhu experimentu může být analýza přechodové charakteristiky systému, ze které lze orientačně určit např. zesílení, časové konstanty apod. Dále můžeme testovat linearitu systému např. generováním různých skokových vstupních signálů a pozorováním výstupu. U MIMO systému lze zjišťovat, který vstup ovlivňuje který výstup. Z přechodové charakteristiky je možné vypočítat i další informace o systému. Např. oscilující odezva na jednotkový skok nám říká, že systém bude mít pravděpodobně imaginární póly atd.

2.1.3 Doba trvání experimentu

Jednou z otázek při návrhu experimentu je volba doby jeho trvání, tzn. vytvoření kompromisu mezi požadovanou přesností odhadu a objemem dat pro identifikaci. Přesnost odhadu parametrů (uvažujeme-li některý z regresních modelů z Kap. 4.3) je nepřímo úměrná počtu vzorků, neboť pokud uvažujeme model

$$y(k) = \varphi(k)\theta_0 + e(k), \quad (2.1)$$

kde $e(k)$ je bílý šum s nulovou střední hodnotou a rozptylem σ_e^2 , tak pro kovarianční matici chyby odhadu parametrů platí

$$E \left[\hat{\theta}_N - \theta_0 \right] \left[\hat{\theta}_N - \theta_0 \right]^T = \frac{\sigma_e^2}{N} \left[J''(\theta_0) \right]^{-1}, \quad (2.2)$$

kde $J(\theta)$ je kritérium, které při hledání odhadu neznámých parametrů θ minimalizujeme. $J''(\theta)$ představuje druhou derivaci daného kritéria. Obecně lze říci, že doba trvání experimentu, pro získání relevantních výsledků, by měla být minimálně desetinásobkem nejdelší časové konstanty identifikovaného systému.

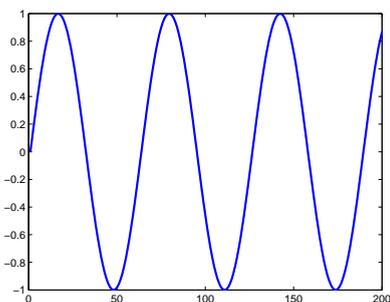
2.1.4 Volba vstupního signálu

Dosud jsme při návrhu experimentu uvažovali pouze vstupní skokové signály pro analýzu základních vlastností systému z přechodových charakteristik. V identifikaci však potřebujeme daleko rozmanitější signály, které nám umožní systém vybudit natolik, abychom v odezvě získali všechny jeho dílčí přenosové funkce. Zároveň je třeba nějakým způsobem stanovit pravidla, která určí, po jakou dobu musí být systém vybuděn apod. Pokud použijeme konstantní nulový signál pro buzení systému, nejsme schopni zjistit žádnou relevantní informaci o přenosu na výstup. Pro kvalitní vybudění je třeba střídát nulový vstup s vhodným nenulovým, abychom byli schopni získat všechny dílčí přenosové funkce systému. Proto se zavádí pojem *vytrvalost buzení*. Vstupní signál $u(k)$, $k = 0, 1, 2, \dots$ je *trvale vybudující* po dobu n právě tehdy, když existuje

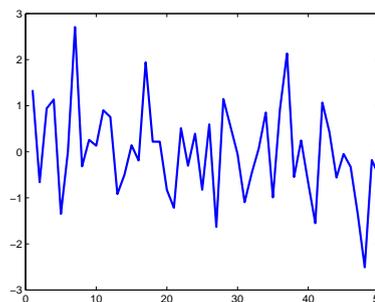
číslo N , zajišťující plnou hodnost n matice (viz. definice a důkaz v [1])

$$U_{0,n,N} = \begin{bmatrix} u(0) & u(1) & \dots & u(N-1) \\ u(1) & u(2) & \dots & u(N) \\ \vdots & & \ddots & \vdots \\ u(n-1) & u(n) & \dots & u(N+n-2) \end{bmatrix}. \quad (2.3)$$

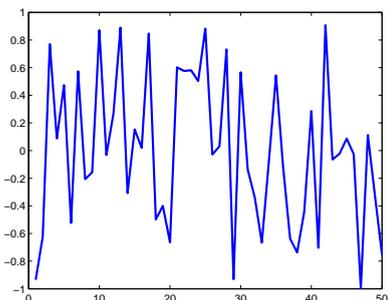
Pro různé účely identifikace používáme různé typy vstupních signálů. Nejběžněji používané signály jsou zobrazeny v Obr. 2.2.



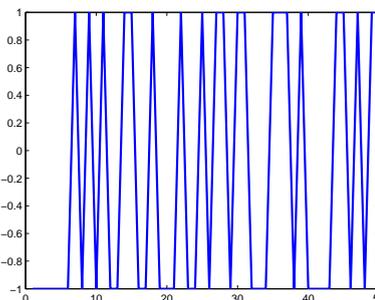
(a) Periodický sinusový signál



(b) Náhodný Gaussovský signál



(c) Součet různých sinusoid



(d) Pseudonáhodný binární signál (PRBS)

OBRÁZEK 2.2: Různé druhy vstupních signálů pro identifikaci [2]

2.2 Předzpracování dat

Základem dobré identifikace jsou kvalitní data. V praxi data často obsahují mnoho nesprávných nebo chybějících hodnot, jsou nekonzistentní a v krajních případech mohou pocházet i z několika různých zdrojů. Krok předzpracování dat představuje zpracování neuspořádaných dat do formy vhodné pro provedení úspěšné identifikace neznámého systému.

2.2.1 Filtrace

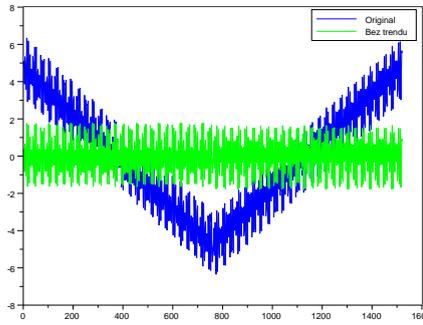
Jedním z hlavních důvodů provádění filtrace je zamezení aliasingu, protože pokud k němu dojde, jeho následky se odstraňují velmi těžce. Při návrhu experimentu se proto při převodu spojitého signálu na diskrétní zařazuje tzv. *antialiasingový filtr* (realizuje se obvykle dolní propustí), který má za úkol odfiltrovat frekvence vyšší než odpovídají Shannonovu teorému (Kap. 2.1.1). Pro filtraci ve frekvenční oblasti se používají i ostatní typy filtrů jako horní propust, pásmová propust, pásmová zadrž nebo amplitudový filtr, podle toho, jaké frekvenční pásmo nás pro daný experiment zajímá.

2.2.2 Převzorkování dat a odstranění trendů

V případě, kdy bylo použito nadbytečně rychlé vzorkování (s ohledem na frekvenční pásmo, které nás zajímá), převzorkujeme data vybráním každého i -tého vzorku. Pokud byla původní frekvence vzorkování ω_S , po následném převzorkování bude ω/i . Je však třeba předcházet aliasingu, a proto se před aplikací takového převzorkování doporučuje použít antialiasingový filtr (realizovaný dolní propustí) s nejvyšší propustnou frekvencí $\omega/2i$. Na fakt, že byla při návrhu experimentu zvolena příliš vysoká frekvence, mohou ukazovat data, obsahující vysokofrekvenční rušení v oblasti frekvenčního pásma, které nás zajímá. Dalším ukazatelem, který může upozornit na příliš vysokou frekvenci vzorkování, je seskupení pólů odhadnutého diskrétního modelu v okolí bodu $z = 1$ komplexní roviny.

Jestliže máme k dispozici příliš řídká data (není dodržen Shannonův teorém), je třeba také použít převzorkování. Po převzorkování se objeví chybějící data (chybějící data se objevují i v případě, že pro dané pozorování není v proměnné uložena žádná hodnota již při samotném měření). Chybějící hodnoty obvykle reprezentuje hodnota NaN (*Not a Number*). Úlohou je odstranění NaN hodnot je data upravit tak, aby se hodnoty NaN zaplnily jinou hodnotou nebo vynechaly, opravila se konzistence, případně se vyřešila nadbytečnost. Pro doplnění chybějících hodnot se obvykle používají dvě základní aproximační metody. První variantou je *zero-order hold* (ZOH), kdy je chybějící hodnota nahrazena nejbližší předchozí hodnotou. Druhou variantou je doplnění hodnot pomocí lineární interpolace.

Vzhledem k tomu, že lineární modely jsou schopny popisovat daný systém pouze v určitém pracovním bodě, nelze jimi modelovat změny, jakými jsou např. různé offsety, lineárních nebo periodické posuny. Tyto vlivy souhrnně označujeme jako trendy. Pokud před provedením identifikace odstraníme z dat trendy, můžeme se zaměřit pouze na oblasti kolísání v okolí trendů a úspěšnost identifikace tak zvýšit. To se provádí např. tím způsobem, že vypočteme na datech nejlepší odhad ve smyslu nejmenších čtverců (viz. Kap. 3.1), a ten od původních dat odečteme. Názorně to popisuje Obr. 2.3.



OBRÁZEK 2.3: Odstranění trendu z dat

2.3 Výběr struktury modelu

Jak se ukáže později v Kap. 4.3 a Kap. 5.3, nejpoužívanějším kritériem, které budeme v souvislosti s běžnými vstupně-výstupními regresními modely minimalizovat, za účelem nalezení optimálního odhadu parametrů, je

$$J_N(\theta, Z_N) = \frac{1}{N} \sum_{k=0}^{N-1} (y(k) - \varphi^T(k)\theta)^2, \quad (2.4)$$

které představuje *prediction error method* (PEM) a je založeno na minimalizaci váženého součtu čtverců chyb predikce. Pokud bychom toto kritérium používali pro výběr optimální struktury modelu, došli bychom k závěru, že nejsložitější model s největším řádem by byl vždy ten nejlepší. Mezi sofistikovanější kritéria pro výběr struktury modelu patří například AIC (*Akaike Information Criterion*) [8]

$$J_N^{AIC}(\theta, Z_N) = J_N(\theta, Z_N) \left(1 + \frac{2 \dim \theta}{N} \right), \quad (2.5)$$

kde $\dim \theta$ je dimenze vektoru parametrů. AIC má ovšem tendenci volit příliš vysoké řády modelu. Dalším z používaných kritérií, je MDL (*Minimum Description Length*) (pro více detailů [9])

$$J_N^{MDL}(\theta, Z_N) = J_N(\theta, Z_N) \left(1 + \frac{\dim \theta \log N}{N} \right). \quad (2.6)$$

Uvedená kritéria jsou používána v úlohách stanovení optimálního řádu modelu s různou úspěšností (detailnějším popisem různých kritérií se zabývá [1]). V této práci se dále (Kap. 3.4)

budeme zabývat novým, dosud nepříliš zkoumaným přístupem k výběru optimálního řádu modelu, založeným na metodě *Nonnegative Garrote* (NNG), která používá kritérium

$$J_N^{NNG}(\theta, Z_N) = \min_w \sum_{k=1}^N \left(y(k) - \sum_{j=1}^n w_j \varphi_j(k) \hat{\theta}_j \right)^2 + \lambda \sum_{j=1}^n w_j \quad (2.7)$$

$$w \succeq 0,$$

kde λ označuje jako parametr složitosti modelu. S rostoucím λ se postupně snižuje složitost modelu. Řešením NNG problému jsou prvky $w_i \hat{\theta}_i$ pro každé jednotlivé λ . Podrobná studie problematiky NNG je dále věnována Kap. 3.4.

2.4 Identifikace

V celé práci uvažujeme tzv. parametrické modely a parametrické metody. Pokud tedy máme zvolenou strukturu modelu, samotná identifikace jeho neznámých parametrů znamená minimalizaci určitého matematického kritéria - podle toho, která identifikační metoda byla zvolena (Kap. 5).

2.5 Validace modelu

Poslední, ale velmi důležitou součástí celé identifikační procedury, je stanovení kvality navrženého modelu. To lze provádět mnoha způsoby (pro více detailů [1]), přičemž uvedeme nejpoužívanější z nich. Přírozenou cestou při posuzování kvality navrženého modelu je zaměřit se na chybu predikce výstupu (reziduum)

$$\varepsilon(k, \hat{\theta}_N) = y(k) - \hat{y}(k|\hat{\theta}_N), \quad (2.8)$$

kde $\hat{\theta}_N$ je odhad parametrů vybraného modelu libovolnou metodou. Nejpoužívanějším kritériem, posuzujícím kvalitu navrženého modelu, je určování míry shody posloupnosti $\hat{y}(k|\hat{\theta}_N)$ vzhledem k posloupnosti $y(k)$, které se nazývá *fit faktor*

$$\text{fit} = 100 \left(1 - \frac{\|\varepsilon(k, \hat{\theta}_N)\|_2}{\|y(k) - \bar{y}\|_2} \right) \%. \quad (2.9)$$

Ve většině případů udává hodnoty z intervalu 0 – 100%. Pro špatný fit však toto číslo může být i záporné.

Dalším jednoduchým kritériem, posuzujícím kvalitu navrženého modelu, může být výpočet kovarianční matice pomocí *vzájemné korelace* mezi rezidui a (vůči nim) minulými vstupy

$$\hat{R}_{\varepsilon u}^N(\tau) = \frac{1}{N} \sum_{k=1}^N \varepsilon(k)u(k - \tau). \quad (2.10)$$

Vysoká hodnota (2.10) pro dané τ může upozorňovat na to, že příslušný vstup $u(k - \tau)$ nese důležitou informaci. Další variantou je výpočet *autokorelace*

$$\hat{R}_{\varepsilon}^N(\tau) = \frac{1}{N} \sum_{k=1}^N \varepsilon(k)\varepsilon(k - \tau), \quad (2.11)$$

jež může být použita např. pro vizuální ověření, zda je chyba predikce bílý šum (osciluje kolem nuly). To se na autokorelační funkci projeví výrazně vysokou amplitudou v nule.

Kapitola 3

Matematické nástroje

V této kapitole budou popsány nejdůležitější matematické nástroje používané v problematice identifikací, se kterými se setkáme v následujících kapitolách. Pozornost je zde věnována zejména metodě lineárních nejmenších čtverců a jejím různým modifikacím. Poslední část kapitoly se zabývá specifickou problematikou redukce řádu lineárních regresních modelů.

Pokud je chyba mezi výstupem identifikovaného systému a výstupem jeho modelu $e(k)$ lineární v parametrech a jako ztrátovou funkci uvažujeme součet kvadrátů těchto chyb, hovoříme o tzv. lineární optimalizaci. Výstup modelu \hat{y} pak závisí lineárně na n parametrech θ_i (u je vstup do systému):

$$\hat{y} = \theta_1\varphi_1 + \theta_2\varphi_2 + \cdots + \theta_n\varphi_n = \sum_{i=1}^n \theta_i\varphi_i, \quad \varphi_i = f(u). \quad (3.1)$$

Obvykle se φ_i označuje jako *regresor* neboli *nezávislá proměnná*, θ_i jsou *regresní koeficienty*, y je *závislá proměnná*. Celý problém nazýváme *lineární regrese*.

Existuje i mnoho případů, kdy jsou lineární optimalizační problémy zaváděny uměle. Pokud je $e(k)$ nelineární funkcí $f(\cdot)$ parametrů, ztrátová funkce se zavede jako součet invertovaných nelinearit $f(\cdot)^{-1}$. Jestliže máme k dispozici nějakou apriorní informaci o systému, i v tomto případě lze mnoho nelineárních optimalizačních problémů převést na lineární. Více podrobností lze nalézt v [1, 3, 10].

3.1 Metoda nejmenších čtverců (LS)

Metoda nejmenších čtverců (Least Squares) [11], někdy též označovaná jako metoda obyčejných nejmenších čtverců, je nejrozšířenější metodou pro řešení problémů lineární optimalizace, kterou

jako první formuloval Carl Friedrich Gauss v roce 1795. Uvažujme soustavu lineárních rovnic

$$Ax = b + e, \quad (3.2)$$

kde matice A o rozměrech $m \times n$ je známá, b je daný vektor pravých stran rozměru m a x představuje n rozměrný neznámý vektor. Pro náhodný vektor chyb platí $\varepsilon\{e\} = 0$, $\text{cov}\{e\} = \varepsilon\{ee^T\} = \sigma^2 I$. Zdůrazněme, že při formulaci LS problému je vektor pravých stran b zatížen chybou, zatímco matici A známe přesně.

Cílem metody je nalézt odhad neznámého vektoru x , který co nejlépe aproximuje výstup b ve smyslu minimalizace kritéria (3.3). Z geometrického pohledu lze říci, že hledáme vektor e s minimální normou (euklidovskou délkou). Z Pythagorovy věty vyplývá, že nejkratší vzdálenost dostáváme tehdy, pokud bude reziduum e ortogonální na obor hodnot $\mathcal{R}(A)$, tj. $e \perp \mathcal{R}(A)$. Musí tedy platit $A^T e = 0$, viz. Obr. 3.1.

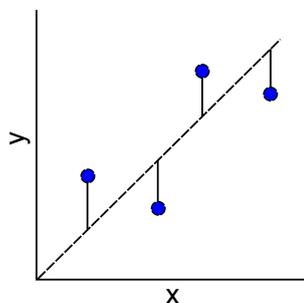
$$\hat{x} = \min_x \{e^T e\} = \min_x \{(Ax - b)^T (Ax - b)\} = \min_x \|Ax - b\|_2^2 \text{ s.t. } Ax = b + e. \quad (3.3)$$

Zjednodušeně řečeno, metoda nejmenších čtverců hledá takové hodnoty koeficientů funkce, udávající vztah mezi A a b , aby součet čtverců odchylek jejich funkčních hodnot od naměřených dat byl nejmenší možný. Minimalizaci kritéria (3.3) provedeme derivováním podle x , kterou položíme rovnu nule:

$$(e^T e)' = [(Ax - b)^T (Ax - b)]' = 2A^T Ax - 2A^T b = 0 \quad (3.4)$$

a vyjádřením získáváme vztah pro nejlepší lineární nevychýlený odhad metodou LS

$$\hat{x}_{LS} = (A^T A)^{-1} A^T b. \quad (3.5)$$



OBRÁZEK 3.1: Geometrická interpretace metody LS

Metoda LS obecně slouží k eliminaci chyb, kterou provádí optimálně vzhledem k pevně danému kritériu. Optimálně eliminovat chyby v datech lze i vzhledem k jiným kritériím, takový postup může vést na metody převoditelné na metodu LS (použitím různých typů vážení, např. když je známo, že chyba některých měření se výrazně liší od zbytku), nebo na metody obecně nepřevoditelné (nebo obtížně převoditelné) na metodu LS (např. problém úplných nejmenších čtverců

v Kap. 3.2). Základní numerické implementace metody LS jsou poměrně známé. Obvykle jsou založeny na QR faktorizaci, nebo singulárním (SVD) rozkladu [12].

Při použití LS v identifikaci předpokládáme, že na systému bylo naměřeno $i = 1, \dots, N$ vzorků vstupních a jim odpovídajících výstupních dat $\{u(i), y(i)\}$. Výstup systému y je zkreslen aditivním bílým šumem $v(k)$ s nulovou střední hodnotou. Počet parametrů θ , které je třeba odhadnout je n . Pro naměřená vstupní a výstupní data lze zkonstruovat n regresorů $\varphi_1, \dots, \varphi_n$. Vzniká problém lineární regrese

$$y(k) = \varphi^T(k)\theta + v(k). \quad (3.6)$$

Jak konkrétně sestavit regresor $\varphi(k)$ pro danou strukturu modelu názorně popisuje např. Kap. 4.3.1.

$$\Phi = \begin{bmatrix} \varphi_1(1) & \varphi_2(1) & \cdots & \varphi_n(1) \\ \varphi_1(2) & \varphi_2(2) & \cdots & \varphi_n(2) \\ \vdots & \vdots & & \vdots \\ \varphi_1(N) & \varphi_2(N) & \cdots & \varphi_n(N) \end{bmatrix} \quad y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}, \quad (3.7)$$

Φ je *matice regresorů*. Odhad neznámých parametrů θ se provede dle vztahu (3.5)

$$\hat{\theta}_{LS} = [\Phi^T \Phi]^{-1} \Phi^T y. \quad (3.8)$$

3.2 Problém úplných nejmenších čtverců (TLS)

Metoda úplných nejmenších čtverců je též známa pod anglickými označeními *total least squares*, nebo *rigorous least squares*, v české literatuře lze obvykle nalézt pod označením ortogonální regrese. Zatímco v předchozím případě metody klasických LS uvažujeme zatížení vektoru pravých stran v rovnici (3.2) bílým šumem e s nulovou střední hodnotou, v případě použití metody TLS uvažujeme, že bílým šumem (E) s nulovou střední hodnotou jsou zatížena i pozorování v matici A :

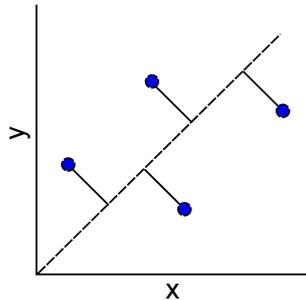
$$(A + E)x = b + e. \quad (3.9)$$

Problém TLS je formulován jako

$$\hat{x} = \min_x \|(E, e)\|_F \quad \text{s.t.} \quad (A + E)x = b + e, \quad (3.10)$$

kde $\|\cdot\|_F$ je Frobeniova norma ($\|X\|_F = \sqrt{\text{tr}(X \cdot X^T)}$). Metodu TLS lze, za výše uvedených předpokladů, geometricky interpretovat tak, že rezidua představují nejkratší možné kolmé vzdálenosti mezi pozorovanými daty a skutečnou křivkou (zatímco u klasických LS představují rezidua nejkratší možné vertikální vzdálenosti mezi pozorovanými daty a skutečnou křivkou), což v praxi

znamená, že vektor reziduí je kolmý na tangentu skutečné křivky (Obr. 3.2). Metoda TLS je někdy též označována jako *dvourozměrná euklidovská regrese* [13]. Problém úplných nejmenších čtverců bude později v Kap. 5.7 použit jako nástroj pro realizaci tzv. *errors-in-variables* (EIV) identifikace.



OBRÁZEK 3.2: Geometrická interpretace metody TLS

3.2.1 Implementace metody TLS

Jedna ze základních implementací metody TLS, známá z [14], je založena na jediném singulárním rozkladu. Rovnici (3.10) lze ekvivalentně přepsat do maticového tvaru

$$[(A + E)(b + e)] \begin{bmatrix} x \\ -I_m \end{bmatrix}. \quad (3.11)$$

Úkolem je nalézt takové $[E \ e]$, které sníží hodnotu $[A \ b]$ o m . Výpočtem SVD dostáváme

$$\begin{bmatrix} A & b \end{bmatrix} = \begin{bmatrix} U_A & U_b \end{bmatrix} \begin{bmatrix} \Sigma_A & 0 \\ 0 & \Sigma_b \end{bmatrix} \begin{bmatrix} V_{AA} & V_{Ab} \\ V_{bA} & V_{bb} \end{bmatrix}^* \quad (3.12)$$

a pokud uvažujeme šумы E a e , platí rovnost

$$[(A + E)(b + e)] = \begin{bmatrix} U_A & U_b \end{bmatrix} \begin{bmatrix} \Sigma_A & 0 \\ 0 & 0_m \end{bmatrix} \begin{bmatrix} V_{AA} & V_{Ab} \\ V_{bA} & V_{bb} \end{bmatrix}^*, \quad (3.13)$$

ekvivalentně lze díky linearitě psát

$$\begin{bmatrix} E & e \end{bmatrix} = - \begin{bmatrix} U_A & U_b \end{bmatrix} \begin{bmatrix} 0_n & 0 \\ 0 & \Sigma_b \end{bmatrix} \begin{bmatrix} V_{AA} & V_{Ab} \\ V_{bA} & V_{bb} \end{bmatrix}^*. \quad (3.14)$$

Díky vzniklým nulovým členům lze zápis zjednodušit na

$$\begin{bmatrix} E & e \end{bmatrix} = -U_b \Sigma_b \begin{bmatrix} V_{Ab} \\ V_{bb} \end{bmatrix}^* = - \begin{bmatrix} A & b \end{bmatrix} \begin{bmatrix} V_{Ab} \\ V_{bb} \end{bmatrix} \begin{bmatrix} V_{Ab} \\ V_{bb} \end{bmatrix}^*. \quad (3.15)$$

Tím zajistíme, že

$$[(A + E)(b + e)] \begin{bmatrix} V_{Ab} \\ V_{bb} \end{bmatrix} = 0. \quad (3.16)$$

Pokud je V_{bb} nesingulární, můžeme obě strany rovnice vynásobit členem V_{bb}^{-1} . V případě, že je V_{bb} singulární, TLS problém nemá řešení.

$$[(A + E)(b + e)] \begin{bmatrix} -V_{Ab} & V_{bb}^{-1} \\ -V_{bb} & V_{bb}^{-1} \end{bmatrix} = [(A + E)(b + e)] \begin{bmatrix} x \\ -I_m \end{bmatrix} = 0. \quad (3.17)$$

Odtud plyne výsledné řešení

$$\hat{x}_{TLS} = -V_{Ab}V_{bb}^{-1}. \quad (3.18)$$

3.2.2 Řešení TLS regularizací Tikhonovova typu (RTLS)

Jako další variantu, vedle klasické implementace metody úplných nejmenších čtverců (Kap. 3.2.1), nyní uvedeme jeden z novějších přístupů, využívající regularizace tzv. Tikhonovova typu [15], známou též pod označením *Ridge* neboli *hřebenová* regrese. Implementace algoritmu řešení této regularizace, využívající metody Lagrangeových multiplikátorů je poprvé prezentována v [16], ale vyžaduje složitý a systematický přístup pro počáteční výběr ladících parametrů. Vylepšení a oproštění od složitého matematického aparátu pro stanovování počátečních parametrů později řeší algoritmus, prezentovaný pod názvem *A Regularized Total Least Squares Algorithm* v [17], kterým se dále budeme zabývat.

Metoda Tikhonovovy regularizace je založena na myšlence, že se lze zbavit nejednoznačnosti a zlepšit stabilitu řešení, pokud kromě empirických dat vezmeme také v úvahu konceptuální data popisující nějakou globální vlastnost hledané funkce. To znamená že omezíme prostor, kde hledáme řešení vyhovující naměřeným datům, pouze na řešení, která mají fyzikální smysl, tj. splňují nějakou předem danou podmínku (jako např. omezený výskyt oscilací vyšších frekvencí) nebo penalizujeme nežádoucí řešení.

Problém Tikhonovovy regularizace TLS (dále jen RTLS) je formulován jako

$$\|(E, f)\|_F \text{ s.t. } (A + E)\theta = b + f, \quad \|L\theta\| \leq \delta, \quad (3.19)$$

kde δ je regularizační parametr, $L \in \mathcal{R}$ je norma řešení (nebo též Tikhonovova matice), která se volí jako jednotková matice, což zajistí prioritní penalizaci řešení s největší normou $\|\cdot\|$ (v některých případech se volí jako aproximace operátoru první nebo druhé derivace - podle toho jakou vlastnost má norma modelovat; ta může modelovat nějakou vlastnost řešení, jako např. hladkost funkce, její lokalizaci apod). Pomocí regularizačního parametru δ lze měnit míru této penalizace.

V [17] je problém regularizace řešen Lagrangeovou metodou. Princip algoritmu spočívá v nalezení optimálního odhadu θ ve smyslu (3.19), tedy na prostoru omezeném regularizačním parametrem δ , přičemž penalizujeme řešení s velkou normou $\|L\theta\|$. Formulace (3.19) pomocí Lagrangeových multiplikátorů má podobu

$$\mathcal{L}(E, f, \theta, \xi, \mu) = \|(E, f)\|_F^2 + \xi^T(A\theta + E\theta - b - f) + \mu(\|L\theta\|^2 - \delta^2), \quad (3.20)$$

což lze napsat pomocí tzv. *Karush-Kuhn-Tucker* (KKT) podmínek

$$\begin{aligned} \mu L^T L \theta A^T f - \|f\|^2 \theta &= 0 \\ (1 + \|\theta\|^2) f &= A\theta - b \\ \mu &\geq 0, \quad \mu(\theta^T L^T L \theta - \delta^2) = 0. \end{aligned} \quad (3.21)$$

Zavedeme θ^* a μ^* jako optimální řešení (3.19). Pokud bude $\mu^* = 0$, znamená to, že $\|L\theta\| \leq \delta$ je neaktivní, což lze způsobit volbou příliš velkého δ a v tomto případě bude zároveň platit $\theta^* = \theta_{TLS}$ ¹. Pokud bude podmínka $\|L\theta\| \leq \delta$ splněna, tak platí $\theta^T L^T L \theta - \delta^2 = 0$ a řešení θ^* zároveň vyhovuje

$$\begin{aligned} (A^T A + \lambda_I I + \lambda_L L^T L) \theta^* &= A^T b \\ \mu > 0, \quad (\theta^*)^T L^T L \theta^* - \delta^2 &= 0, \end{aligned} \quad (3.22)$$

kde

$$\begin{aligned} \lambda_I &= -\frac{\|A\theta^* - b\|^2}{1 + \|\theta^*\|^2} \\ \lambda_L &= \mu(1 + \|\theta^*\|^2) \\ \mu &= -\frac{1}{\delta^2} \left(\frac{b^T(A\theta^* - b)}{1 + \|\theta^*\|^2} + \frac{\|A\theta^* - b\|^2}{(1 + \|\theta^*\|^2)^2} \right). \end{aligned} \quad (3.23)$$

Pokud je omezení $\|L\theta\| \leq \delta$ aktivní, řešení θ^* problému (3.19) vyhovuje

$$B(\theta^*) \begin{bmatrix} \theta^* \\ -1 \end{bmatrix} = -\lambda_I \begin{bmatrix} \theta^* \\ -1 \end{bmatrix}, \quad (3.24)$$

kde λ_I je vlastní vektor vlastních čísel matice

$$B(\theta^*) = \begin{bmatrix} A^T A + \lambda_L(\theta^*) L^T L & A^T b \\ b^T A & -\lambda_L(\theta^*) \delta^2 + b^T b \end{bmatrix}. \quad (3.25)$$

¹Pokud zvolíme $\delta < \|x_{LS}\|$, pak $x_{TLS} = x_{LS}$.

Na základě odvození z [17] odhadujeme vlastní vektor matice $B(\theta^*)$, který pro k -tou iteraci značíme $z^{(k)}$

$$z^{(k)} = \left((\theta^{(k)})^T, -1 \right) / \sqrt{1 + \|\theta^{(k)}\|^2}. \quad (3.26)$$

Dále definujeme reziduum $\rho^k = \|B(\theta^{(k)})z^{(k)} + \lambda_I(\theta^{(k)})z^{(k)}\|$ a toleranci τ , která ukončí celý algoritmus regularizace v případě, že $\frac{\rho^{(k)}}{|\lambda_I^{(k)}|} < \tau$. Uvedený postup regularizace shrnuje Algoritmus 1. Odvození celého algoritmu je uvedeno v [17] a jeho použití následně demonstruje Kap. 5.7.

Algoritmus 1: Tikhonovova regularizace TLS řešení [17]

Vstup: $\theta^{(0)}, \lambda_I^{(0)}, \lambda_L^{(0)}, z^{(0)}$

while $\frac{\rho^{(k+1)}}{|\lambda_I^{(k+1)}|} \geq \tau$ & $\mu(\theta^{(k+1)}) \leq 0$ **do**

Vyřešení $(B(\theta^{(k)}) + \lambda_I^{(k)I})x = z^{(k)}$;
$x^{(k+1)} = -\frac{x^{(1:n)}}{x^{(n+1)}}$;
$z^{(k+1)} = \frac{x}{\ x\ }$;
Aktualizace $\lambda_I^{(k+1)}, \lambda_L^{(k+1)}, \rho^{(k+1)}$;

3.3 Metoda instrumentálních proměnných

V této kapitole připojíme k metodě nejmenších čtverců navíc metodu instrumentálních proměnných neboli *instrumental variables* (IV). Uvažujme (3.6). V případech, kdy jsou regresory $\varphi(k)$ korelované s šumem $v(k)$, LS odhad bude vychýlený. Pro odstranění této nežádoucí vlastnosti lze použít metodu IV [18]. Její základní myšlenkou je nalezení tzv. instrumentů $\zeta(k)$, které jsou nekorelované s šumem $v(k)$, čímž se zajistí nevychýlený odhad²

$$\hat{\theta}_{IV} = [\zeta^T \Phi]^{-1} \zeta^T y. \quad (3.27)$$

Problém spočívá ve stanovení instrumentů $\zeta(k)$. Existují různé metody, které se sestavením $\zeta(k)$ zabývají, za všechny např. IV4 z [19].

3.4 Nonnegative Garrote

Redukce řádu lineárních regresních modelů (viz. Kap. 4.3) je hojně diskutovanou problematikou, zejména ve statistických oborech ([20–22]). Existuje řada úspěšných metod, jako je např. *Lasso* [23] a *Ridge*, které jsou schopny řešit úlohu stanovení optimálního řádu modelu. Obvyklý postup při výběru optimálního modelu dynamického systému je takový, že se nejprve vypočítá model

² $\hat{\theta}_{IV} = \hat{\theta}_{LS}$ pro $\zeta(k) = \varphi(k)$.

vyššího řádu, než je předpokládaný výsledný model, a následně se aplikuje postup, který modifikuje koeficienty navrženého modelu tím způsobem, že některé vynuluje a ostatní optimalizuje do potřebné podoby. Zároveň je třeba udržet kompromis mezi komplexitou a přesností modelu. Např. metoda *Lasso* umožňuje nulovat určité koeficienty modelu, ale není zaručeno, že to budou právě koeficienty u nejvyšších řádů, což je ovšem při redukcí řádu modelu dynamického systému žádoucí. Zřejmou nevýhodou uvedených metod je fakt, že nebyly vyvíjeny ruku v ruce s dynamickými systémy. Poměrně novou metodou, která spatřila světlo světa v roce 1995, je metoda *Nonnegative Garrote* (NNG) [24]. Detailní informace o NNG z hlediska výpočetní složitosti, flexibility a konzistence lze nalézt v [25].

V další části bude pro názornost uvažován model ARX (Kap. 4.3.1), na kterém bude demonstrováno použití NNG metody pro redukcí jeho řádu. V práci bude dále NNG metoda rozšířena na model ARMAX (Kap. 4.3.2). ARX model popisuje rovnice (4.14), kterou lze přepsat do tvaru lineární regrese (4.17). Na základě znalosti sekvence vstupního a výstupního signálu je možné odhadnout parametry $\hat{\theta}$ ARX modelu LS metodou (Kap. 3.1) a výsledným řešením je (3.8), které je výchozím řešením pro NNG metodu. Metoda NNG penalizuje LS řešení vážením jednotlivých koeficientů v $\hat{\theta}_N$. Samotný NNG problém je formulován jako

$$\min_w \sum_{k=1}^N \left(y(k) - \sum_{j=1}^n w_j \varphi_j(k) \hat{\theta}_j \right)^2 + \lambda \sum_{j=1}^n w_j \quad (3.28)$$

$$w \succeq 0.$$

Symbol λ je v [26] a [27] označován jako parametr komplexnosti (složitosti) modelu. S rostoucím λ se postupně snižuje složitost modelu omezením, případně nulováním požadovaných koeficientů. Řešením NNG problému jsou prvky $w_i \hat{\theta}_i$ pro každé jednotlivé λ , kde w je optimálním řešením (3.28). Aby bylo možné docílit prioritní penalizace koeficientů u nejvyšších řádů, je třeba NNG problém (3.28) rozšířit o další omezení na optimální řešení w . Pro uvažovaný ARX model lze omezení vyjádřit jako

$$1 \geq w_1 \geq w_2 \geq \dots \geq w_{n_a} \geq 0 \quad (3.29)$$

$$1 \geq w_{n_a+1} \geq w_{n_a+2} \geq \dots \geq w_{n_a+n_b} \geq 0.$$

Pokud bychom uvažovali model ARMAX (Kap. 4.3.2), přibývá navíc omezení pro polynom C :

$$1 \geq w_{n_a+n_b+1} \geq w_{n_a+n_b+2} \geq \dots \geq w_{n_a+n_b+n_c} \geq 0. \quad (3.30)$$

NNG problém pro výběr řádu ARX modelu reprezentuje kvadratický problém s lineárním omezením

$$\begin{aligned} \min_w \quad & \frac{1}{2} w^T Q w + f^T w + \lambda \mathbf{1}^T w, \\ & A w \preceq b \end{aligned} \quad (3.31)$$

kde $Q = 2\hat{\Theta}\Phi^T\Phi\hat{\Theta}$, $f = -2\hat{\Theta}\Phi^TY$, $\hat{\Theta} \triangleq \text{diag}(\hat{\theta})$. Z (3.29) byly v rámci této práce odvozeny univerzální vztahy pro výpočet matic A a b pro obecný ARX model:

$$\begin{aligned} A' &= \left(\left(\begin{array}{c} I_{n_a \times n_a} \\ 0_{1 \times n_a} \end{array} \right) + \left(\begin{array}{c} 0_{1 \times n_a} \\ -I_{n_a \times n_a} \end{array} \right) \right) \\ A'' &= \left(\left(\begin{array}{c} I_{n_b \times n_b} \\ 0_{1 \times n_b} \end{array} \right) + \left(\begin{array}{c} 0_{1 \times n_b} \\ -I_{n_b \times n_b} \end{array} \right) \right) \\ A &= \left(\begin{array}{cc} (I_{\dim(n_a)} \otimes A' & 0_{\dim(A') \times n_b} \\ 0_{(n_b+1) \times \text{sum}(n_a)} & A'' \end{array} \right) \\ b &= \left(\begin{array}{c} \left(I_{\dim(n_a) \times 1} \otimes \left(\begin{array}{c} 1 \\ 0_{n_a \times 1} \end{array} \right) \right) \\ \left(\begin{array}{c} 1 \\ 0_{n_b \times 1} \end{array} \right) \end{array} \right). \end{aligned} \quad (3.32)$$

Řešením rovnice (3.31) je w_λ pro jednotlivé λ . Výsledný odhad parametrů NNG metodou je dán $\hat{\theta}_\lambda = \hat{\Theta}w_\lambda$, penalizujeme tedy počáteční řešení odhadu parametrů váhou w_λ .

3.4.1 Řešení NNG problému

Známostou metodou řešení problému (3.28), popsanou v [28], je parametrická optimalizace. Rovnice (3.28) odpovídá tvaru

$$\min_x L(x) + \lambda J(x), \quad (3.33)$$

kde $L(x)$ je po částech kvadratická a $J(x)$ je lineární funkce. Optimální řešení w rovnice (3.31) je po částech lineární funkcí $\lambda \in \mathbb{R}_+$ ³ [29]. Nejprve je třeba určit počáteční řešení $w(0)$, poté se vypočte celé řešení v podobě průběhu $w(\lambda)$ pro všechny $\lambda \geq 0$. Cílem je hledat optimální průběh $w(\lambda)$ ve smyslu optimálních směrů řešení, které nejvíce snižují kritérium (3.28), dokud nenarazíme na omezení. V případě, že na omezení narazíme, řešení aktualizujeme a pokračujeme v jiném směru λ dokud nedosáhneme $\lambda = \infty$. Pro vytvoření efektivního algoritmu, který nalezne

³ \mathbb{R}_+ je množina všech kladných reálných čísel.

optimální řešení $w(\lambda)$ zavádíme Lagrangian k (3.28) je⁴

$$\mathcal{L}(w, \mu) = \frac{1}{2}w^T Qw + f^T w + \lambda \mathbf{1}^T w + \mu^T (Aw - b), \quad (3.34)$$

odkud plynou KKT podmínky

$$\begin{aligned} Qw + f + \lambda \mathbf{1} + A^T \mu &= 0 \\ Aw - b &\preceq 0 \\ \mu_j (A_j w - b_j) &= 0 \\ \mu &\succeq 0, \lambda \geq 0. \end{aligned} \quad (3.35)$$

Uvažujeme množinu aktivních omezení $\mathcal{J}^a = \{j_1, j_2, \dots, j_{n\mathcal{J}}\}$ pro μ_j . Řešení (3.35) je ekvivalentní řešení

$$\begin{pmatrix} Q & A_{\mathcal{J}^a}^T \\ A_{\mathcal{J}^a} & 0 \end{pmatrix} \begin{pmatrix} w \\ \mu_{\mathcal{J}^a} \end{pmatrix} = \begin{pmatrix} -f \\ b_{\mathcal{J}^a} \end{pmatrix} + \lambda \begin{pmatrix} -\mathbf{1} \\ 0 \end{pmatrix}. \quad (3.36)$$

Derivováním podle λ dostáváme

$$\begin{pmatrix} Q & A_{\mathcal{J}^a}^T \\ A_{\mathcal{J}^a} & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial w}{\partial \lambda} \\ \frac{\partial \mu_{\mathcal{J}^a}}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} -\mathbf{1} \\ 0 \end{pmatrix}. \quad (3.37)$$

Směr optimálního řešení se potom určí jednoduše z (3.37). Pro nalezení počátečního řešení nastavíme $\lambda = 0$, čemuž odpovídá řešení $w = \mathbf{1}$. Pro nalezení $\mu_{\mathcal{J}^a}$ vyřešíme rovnici (3.35). Dá se ukázat, že $\mu_{\mathcal{J}^a} = \mathbf{0}$, neboť regresní matice Φ je ortogonální na rezidua $\Phi \hat{\theta} - Y$ (více viz. [26, 27]).

3.4.2 Redukce řádu ARX modelu

Jako modelový příklad pro otestování NNG algoritmu 2 implementovaného ve Scilabu podle [26] uvažujme libovolně zvolený ARX model řádu [4 4 0] (dále jen (4, 4)), jehož koeficienty jsou v prvním sloupci Tab. 3.1. Na modelu byla provedena simulace s bílým šumem $\mathcal{N}(0, 1)$ délky 6000 jako vstupním signálem. Výsledný výstup byl poté zkruslen bílým šumem $\mathcal{N}(0, 0.1)$. Simulační data byla rozdělena na poloviny. První část byla použita jako identifikační data a druhá část jako validační data. Jako měřítko kvality odhadu byl použit fit faktor [1].

Pomocí funkce `arx` byl vypočten odhad koeficientů ARX modelu řádu (10, 6) LS metodou, který byl použit jako výchozí odhad pro NNG metodu. Pro λ_{15} odpovídá řád modelu skutečnému řádu (4, 4). Numerických hodnot koeficientů odhadu jsou ve druhém sloupci Tab. 3.1. Závislost fit faktoru na λ je vynesena v Obr. 3.3. Ověření očekávané funkčnosti metody NNG vizualizuje Obr. 3.4, kde jsou zobrazeny plnou barvou nenulové koeficienty a prázdná místa značí koeficienty

⁴ μ označuje vektor Lagrangeových multiplikátorů.

Algoritmus 2: Redukce řádu ARX modelu NNG metodou s omezením (3.29) [26]

Inicializace : $\lambda = 0$, $w_0 = \mathbf{1}$, $\mu_{\mathcal{J}^a} = \mathbf{0}$ a $\mathcal{J}^a = \{1, 2, \dots, p-1\}$
Počáteční řešení: $S = \{(\lambda, w)\} = \{(0, \mathbf{1})\}$
while $\lambda \neq \infty$ **do**
 $\delta\lambda = \emptyset$;

 Vyřešení (3.37) pro $\frac{\partial w}{\partial \lambda}$ a $\frac{\partial \mu_{\mathcal{J}^a}}{\partial \lambda}$;

 Nalezení minimálního $\delta\lambda \geq 0$;

if $A_j(w_\lambda + \frac{\partial w}{\partial \lambda} \delta\lambda) = b_j$ & $A_j \frac{\partial w}{\partial \lambda} > 0$ & $j \notin \mathcal{J}^a$ **then**
 $\mathcal{J}^a \leftarrow \mathcal{J}^a \cup j$;

if $\mu_j + \frac{\partial \mu_j}{\partial \lambda} \delta\lambda = 0$ & $\frac{\partial \mu_j}{\partial \lambda} > 0$ & $j \in \mathcal{J}^a$ **then**
 $\mathcal{J}^a \leftarrow \mathcal{J}^a \setminus j$;

if $\delta\lambda == \emptyset$ **then**
 $\lambda = \infty$;

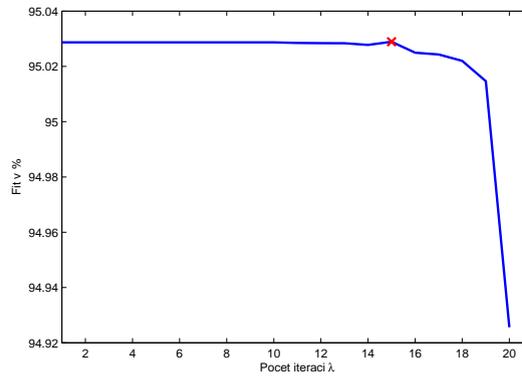
 Aktualizace: $\lambda := \lambda + \delta\lambda$;

 $w_\lambda := w_\lambda + \frac{\partial w}{\partial \lambda} \delta\lambda$;

 $\mu_{\mathcal{J}^a} := \mu_{\mathcal{J}^a} + \frac{\partial \mu_{\mathcal{J}^a}}{\partial \lambda} \delta\lambda$;

 $S := \{S, \{(\lambda, w)\}\}$;

nulové. Vektor parametrů θ je rozdělen přerušovanou linií, přičemž nad ní se nachází polynom A a pod ní polynom B . Je zřejmé, že NNG metoda skutečně zajišťuje postupné nulování koeficientů od nejvyšších řádů. Obr. 3.6 zachycuje konvergenci parametrů θ se zvyšujícím se počtem iterací λ k nule.

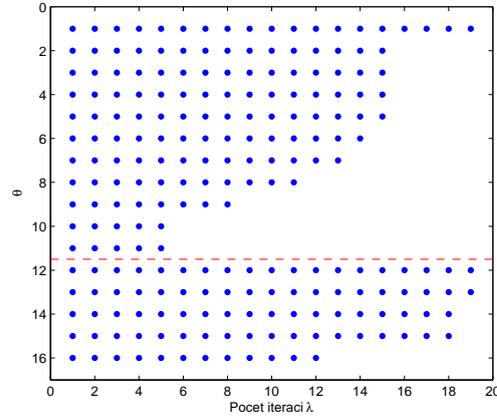

 OBRÁZEK 3.3: ARX: Závislost fit faktoru na počtu iterací λ

Pro srovnání výsledků se SITB⁵ pro Matlab byla použita funkce `selstruc`, která vybírá optimální řád ARX modelu na základě minimalizace kritéria AIC (2.5):

$$V = \text{arxstruc}(z_i, z_v, \text{struc}([1 : n_a], [1 : n_b], 0)); \quad (3.38)$$

$$m = \text{selstruc}(V, 0);$$

⁵SITB = System Identification Toolbox



OBRÁZEK 3.4: ARX: Vizualizace obsazenosti koeficientů v θ v závislosti na počtu iterací λ (plné body označují nenulové koeficienty)

	Skutečný	NNG (λ_{15})	selstruc	arx
$A(q)$	0.1	0.0103	-0.01134	0.07161
	-0.2	-0.0236	-0.03814	- 0.1065
	0.04	0.0077	0.03332	0.05723
	-0.03	-0.0575	-0.07441	- 0.04923
	0	0	0.02134	0
	0	0	- 0.007771	0
$B(q)$	1	1	1	1.001
	2	1.9071	1.889	1.971
	- 0.04	-0.0420	- 0.08791	- 0.004364
	- 0.009	-0.0424	0.2966	0.1847
	0	0	- 0.09869	0
fit		95.03%	95.35%	95.37%

TABULKA 3.1: ARX: Porovnání NNG a `selstruc`

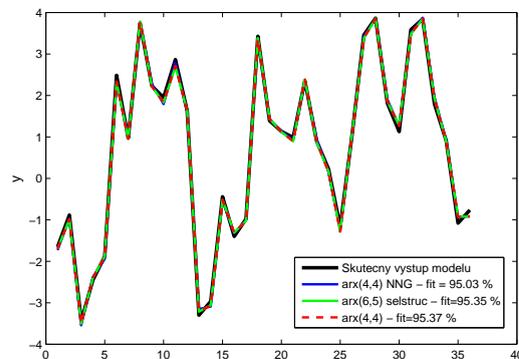
kde z_i jsou identifikační data, z_v validační data, n_a a n_b jsou maximální řády polynomů A a B . Jako nejlepší pro $n_a = 10$ a $n_b = 6$ určila funkce `selstruc` model ARX řádu (6, 5), viz. Tab. 3.1.

Pokud budeme NNG metodu chápat jako metodu pro stanovení optimálního řádu modelu, lze využít stanovené hodnoty řádu modelu (4, 4) pro funkci `arx` a parametry znovu odhadnout s $n_a = n_b = 4$. Aplikováním takového postupu dostáváme ARX model, který je téměř totožný s původním modelem, viz. poslední sloupec Tab. 3.1.

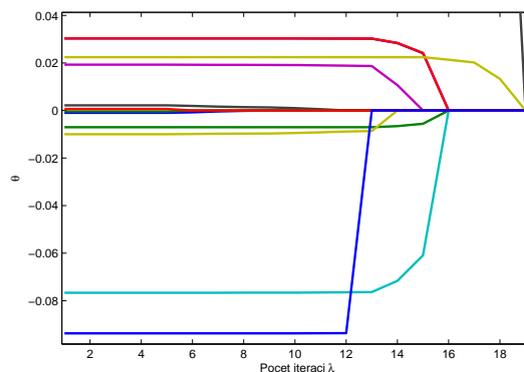
Z uvedených výsledků plyne, že použitý algoritmus pro redukcí řádu ARX modelu opravdu zajišťuje postupné nulování koeficientů polynomů A a B od nejvyšších řádů, čímž se stává velmi dobře použitelným v identifikaci LTI systémů. Jedná-li se o srovnání s konkurenčním SITB a jeho funkcí `selstruc` pro výběr optimálního řádu ARX modelu, lze konstatovat, že pokud se zaměříme na kvalitu odhadu měřenou fit faktorem (2.9), výsledek použití `selstruc` je téměř totožný s NNG metodou - viz. Obr. 3.5. Velikost fit faktoru zde hraje pouze mírně ve prospěch použití NNG metody. Zaměříme-li se však na určení skutečného řádu modelu, zjišťujeme,

že funkce `selstruc` stanovila řád ARX modelu (6,5) oproti skutečnému (4,4), zatímco NNG metoda našla skutečný řád (4,4).

V případě, že NNG metodu použijeme pouze pro stanovení skutečného řádu modelu a následně provedeme odhad parametrů modelu pomocí funkce `arx` pro stanovený řád, dosáhneme nejlepších výsledků ve smyslu fit faktoru (viz. Obr. 3.5) i ve smyslu numerické shody odhadnutých parametrů se skutečnými (Tab. 3.1).



OBRÁZEK 3.5: ARX: Porovnání NNG a `selstruc`



OBRÁZEK 3.6: ARX: Konvergence parametrů θ k nule s rostoucím počtem iterací λ v detailním pohledu na nejdůležitější koeficienty

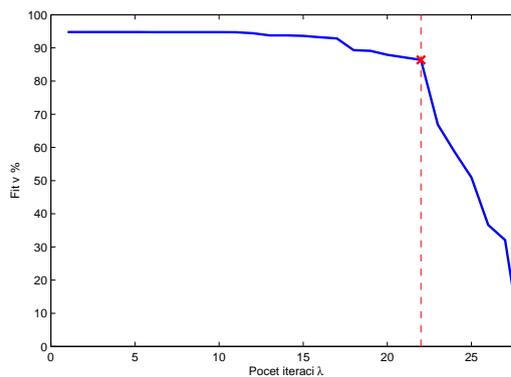
3.4.3 Redukce řádu ARMAX modelu

Využití NNG pro redukci řádu ARMAX modelu je podobné jako v případě ARX, pouze s tím rozdílem, že bereme v úvahu omezení (3.30) pro polynom $C(q)$. Víme, že vektor dat (4.29) závisí na neznámých hodnotách θ . Pokud spočteme odhad $\hat{\theta}$, můžeme zkonstruovat (4.29) využitím rovnosti (2.8). Mějme náhodně zvolený ARMAX model řádu (4,4,3) s polynomy $A(q)$, $B(q)$ a $C(q)$ (Tab. 3.2), na jehož vstup byl přiveden bílý šum $\mathcal{N}(0,1)$ délky 6000 a byl měřen signál na jeho výstupu, který byl následně zkreslen aditivním bílým šumem $\mathcal{N}(0,0.1)$. Simulační data byla rozdělena na poloviny. První část byla použita jako data pro následnou identifikaci, druhá část jako validační data. Z identifikačních dat byl proveden odhad parametrů ARMAX

	Skutečný	NNG (λ_{22})	balred	armax
$A(q)$	0.1	-0.1532	0.1819	0.1129
	-0.2	-0.0991	-0.3348	-0.2006
	0.04	-0.000012	0.00298	0.03734
	-0.03	0.000003	0	-0.02975
$B(q)$	1	0.9634	1.001	1.001
	2	1.681	2.082	2.012
	- 0.04	-0.5074	-0.02161	-0.01635
	- 0.009	-0.0000037	-0.3062	-0.01509
$C(q)$	- 0.08	-0.1438	-	0.1084
	- 0.001	-0.1599	-	-0.1946
fit		86.98%	94.64%	95.01%

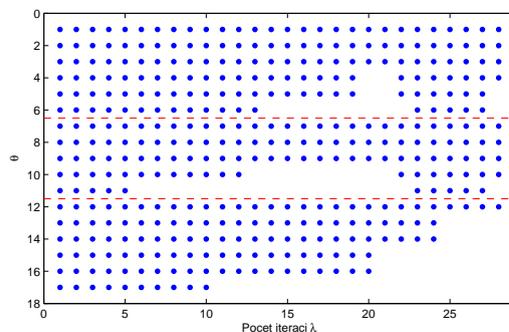
TABULKA 3.2: ARMAX: Porovnání NNG a balred

modelu pomocí funkce `armax` pro řád (5, 5, 5). Vzniklý model byl použit jako výchozí model pro NNG. Aplikováním NNG na výchozí model byl postupně redukován řád, jak zachycuje Obr. 3.8. Vidíme, že pro λ_{22} bylo dosaženo správného řádu modelu. Průběh fit faktoru v Obr. 3.7 příliš neklesá do chvíle, než je dosažena λ_{22} a poté prudce klesá⁶. Parametry θ ARMAX modelu stanovené NNG metodou pro λ_{22} zachycuje Tab. 3.2. V Obr. 3.9 je názorně zobrazeno, jak parametry θ z počátečních hodnot konvergují k nule s rostoucím λ . Provedením `armax(4, 4, 2)` získáváme poměrně kvalitní odhad parametrů (viz. poslední sloupec Tab. 3.2). Pro srovnání se SITB zde byla použita funkce `balred`⁷ ([30, 31]). Tato funkce automaticky provedla redukcí původního ARMAX modelu řádu (5, 5, 5) na ARX řádu (3, 4). Dosažené výsledky výše uvedených metod srovnává Tab. 3.2 a Obr. 3.10.

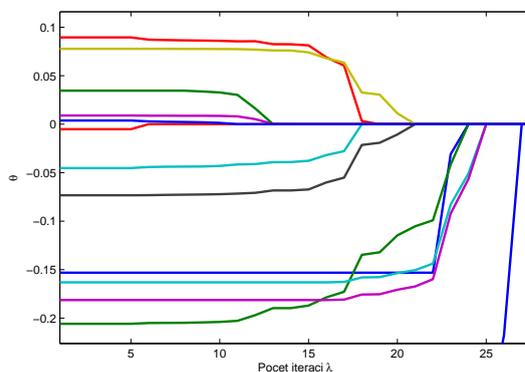
OBRÁZEK 3.7: ARMAX: Závislost fit faktoru na počtu iterací λ

⁶Otázka automatického stanovení řádu modelu na základě průběhu fit faktoru je předmětem dalšího výzkumu.

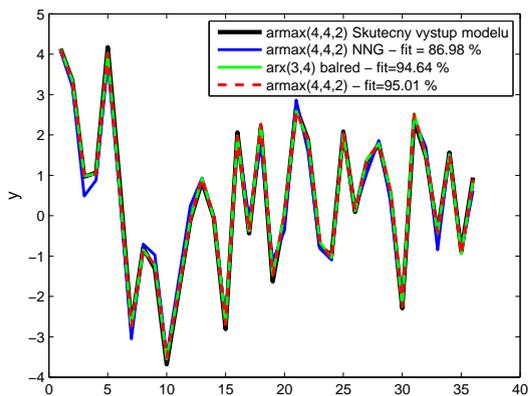
⁷V tomto případě nelze použít funkci `selstruc`, neboť je určena pouze pro ARX modely.



OBRÁZEK 3.8: ARMAX: Vizualizace obsazenosti koeficientů v θ v závislosti na počtu iterací λ (plné body označují nenulové koeficienty)



OBRÁZEK 3.9: ARMAX: Konvergence parametrů θ k nule s rostoucím počtem iterací λ v detailním pohledu na nejdůležitější koeficienty



OBRÁZEK 3.10: ARMAX: Porovnání NNG a balred

Kapitola 4

Systemy a modely

Tato kapitola se zabývá popisem lineárních, časově invariantních (LTI) modelů systémů, které charakterizuje zejména skutečnost, že jejich výstup je vždy lineární kombinací vstupů, přičemž se tato vlastnost s časem nemění. LTI model uvažujeme ve tvaru Obr. 4.1(a) a charakterizuje ho rovnice

$$y(k) = G(q)u(k) + H(q)v(k). \quad (4.1)$$

4.1 Struktura obecného LTI modelu

V této sekci bude popsána obecná struktura LTI modelů, používaných v identifikaci lineárních dynamických systémů, a základní terminologie. Vybrané modely budou následně popsány podrobněji. Obecný tvar lineárního modelu je znázorněn v Obr. 4.1. Tento model představuje obecný framework, od kterého se odvozují další zjednodušené modely. Výstup $y(k)$ deterministického lineárního procesu v čase k lze získat filtrací vstupního signálu přes lineární filtr $G(q)$:

$$y(k) = G(q)u(k) = \frac{\tilde{B}(q)}{\tilde{A}(q)}u(k). \quad (4.2)$$

Obecně může lineární přenosová funkce $G(q)$ obsahovat čítelel $B(q)$ a jmenovatel $A(q)$. Kromě deterministické části lze modelovat i stochastickou část, filtrací bílého šumu $v(k)$ přes lineární filtr $H(q)$. Lze tedy generovat libovolný šumový signál $n(k)$:

$$n(k) = H(q)v(k) = \frac{\tilde{C}(q)}{\tilde{D}(q)}v(k). \quad (4.3)$$

Obecný lineární model v Obr. 4.1(a) zahrnující deterministické a stochastické vlivy má (uvažujeme-li tvar přenosové funkce) podobu (4.1). Filtr $G(q)$ se nazývá *vstupní přenosová funkce*, neboť

popisuje vztah mezi vstupem $u(k)$ a výstupem $y(k)$. Filtr $H(q)$ popisuje vztah mezi vstupujícím šumem $v(k)$ a výstupem $y(k)$ a nazývá se *přenosová funkce šumu*. Přenosové funkce lze rozepsat na čítelel a jmenovatel a obecný tvar lineárního modelu pak dostává podobu Obr. 4.1(b). Pro pozdější analýzu je užitečné separovat případný společný jmenovatel $A(q)$ z $G(q)$ a $H(q)$, což názorně popisuje Obr. 4.1(c) a Obr. 4.1(d). Potom $F(q)A(q) = \tilde{A}(q)$ a $D(q)A(q) = \tilde{D}$. V případě, že $\tilde{A}(q)$ a $\tilde{D}(q)$ nemají společného jmenovatele, pak jednoduše $A(q) = 1$. Více podrobností lze nalézt v [1, 3]. Obecný lineární model má finální podobu

$$y(k) = \frac{B(q)}{F(q)A(q)}u(k) + \frac{C(q)}{D(q)A(q)}v(k), \quad (4.4)$$

neboli

$$A(q)y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}v(k). \quad (4.5)$$

Od obecného tvaru lineárního modelu se odvozují nejrůznější, prakticky používané modely na základě toho, které z polynomů A , B , C , D a F jsou použity. Vybrané z nich jsou popsány v Kap. 4.3.

4.1.1 Terminologie

V této sekci bude stručně popsána aktuálně používaná terminologie lineárních dynamických modelů dle [1].

Model časových řad, který má pouze jmenovatel (Obr. 4.2(a))

$$y(k) = \frac{1}{D(q)}v(k), \quad (4.6)$$

se nazývá *autoregresní (AR) model*.

Model, který uvažuje pouze čítelel (Obr. 4.2(b))

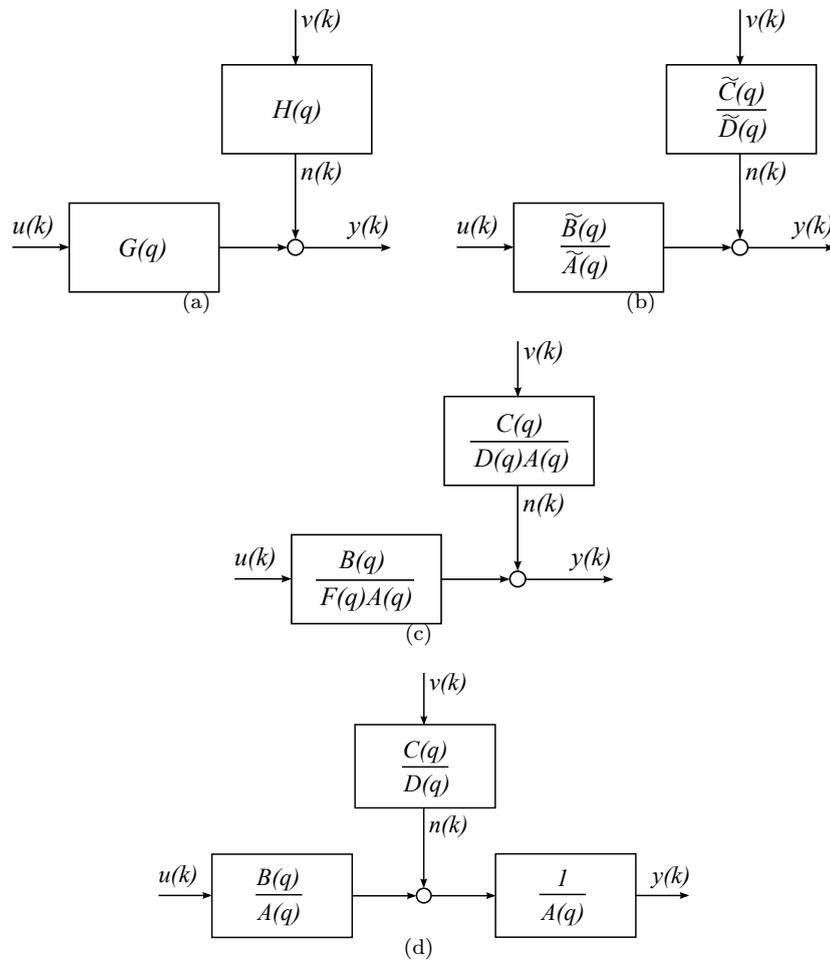
$$y(k) = C(q)v(k), \quad (4.7)$$

se nazývá *model klouzavý průměr* neboli *moving average (MA)*.

V případě, že model využívá čítelel i jmenovatel (Obr. 4.2(c))

$$y(k) = \frac{C(q)}{D(q)}v(k), \quad (4.8)$$

jedná se o *model autoregresní klouzavý průměr (ARMA)*.

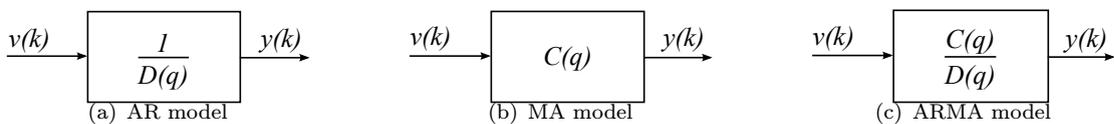


OBRÁZEK 4.1: Struktura obecného LTI modelu [3]

Výše uvedené modely neuvažují žádný vstup $u(k)$. Vedle takových modelů se zavádí modely s jedním nebo více vstupy. Vstup $u(k)$ se nazývá *exogenní* vstup a pokud je uvažován, v označení modelu se to projeví symbolem "X" pro exogenní vstup. Např. model

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{A(q)}v(k) \tag{4.9}$$

se nazývá *autoregresní s exogenním vstupem* (ARX).



OBRÁZEK 4.2: Terminologie LTI modelů

4.2 Optimální prediktor

Existují dva přístupy, jak lze na základě minulých dat odhadovat budoucí chování nějakého procesu. Pokud je výstup modelu vypočítáván na základě znalosti vstupního signálu, aniž bychom znali výstup skutečného procesu, problém se nazývá *simulace* (Obr. 4.3(a)). Naopak, jestliže jsme v situaci, kdy známe historii vstupů, zároveň měříme procesní výstup např. do kroku $k - 1$ a požadujeme výpočet odezvy modelu l kroků do budoucnosti, hovoříme o problému *predikce* (Obr. 4.3(b)). Ve většině případů se používá tzv. jednokroková predikce, kdy $l = 1$. Existují i metody vícekové predikce, těmi se však práce nebude zabývat.

Z Obr. 4.3(a) je patrné, že problém simulace charakterizuje vztah $\hat{y} = G(q)u(k)$. Nefiguruje zde filtr šumu $H(q)$. Někdy se pro zlepšení simulace uměle vytváří bílý šum $w(k)$ a pomocí nějakého $H(q)$ se aditivně přičítá k výstupu jako $\hat{y} = G(q)u(k) + H(q)w(k)$. V případě predikce uvažujme, že známe $y(s)$ a $u(s)$ pro $s \leq k - 1$. Pak lze psát pro chybu predikce

$$\varepsilon(s) = y(s) - G(q)u(s) \quad (4.10)$$

a cílem je, na základě této informace, nalézt vztah pro predikci hodnoty výstupu

$$\hat{y}(k) = G(q)u(k) + \hat{\varepsilon}(k). \quad (4.11)$$

Na základě výše uvedených skutečností lze predikci výstupu zapsat jako

$$\begin{aligned} \hat{y}(k|k-1) &= G(q)u(k) + \hat{\varepsilon}(k|k-1) \\ &= G(q)u(k) + [1 - H(q)^{-1}(q)] \varepsilon(k) \\ &= G(q)u(k) + [1 - H(q)^{-1}(q)] [y(k) - G(q)u(k)]. \end{aligned} \quad (4.12)$$

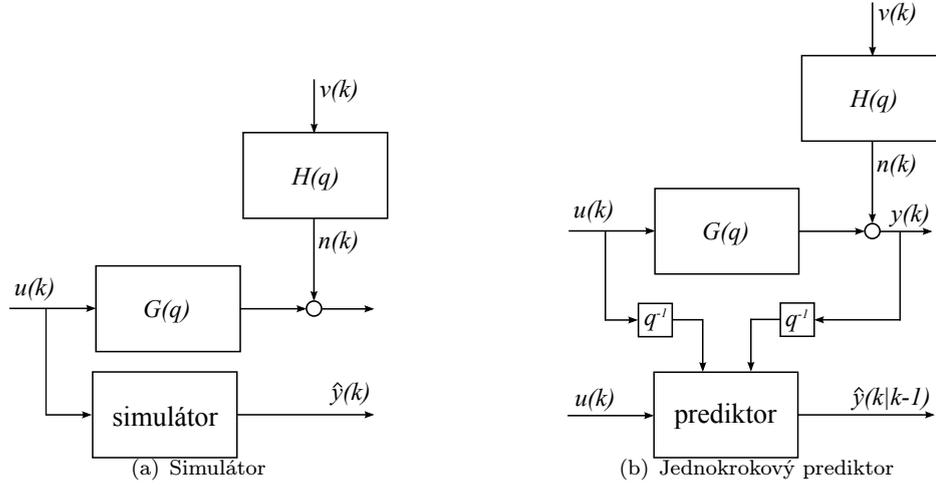
Výsledný tvar jednokrokového prediktoru, optimálního ve smyslu minimalizace kvadrátů chyb predikce, který odvozuje [1] má podobu

$$\hat{y}(k|k-1) = H^{-1}(q)G(q)u(k) + [1 - H^{-1}(q)] y(k). \quad (4.13)$$

4.3 Používané LTI modely

4.3.1 ARX

Model ARX (*Autoregressive with Exogenous Input*) je bezesporu nejpoužívanějším lineárním dynamickým modelem. Je to zejména z toho důvodu, že výpočet jeho parametrů je triviální matematickou úlohou - parametry lze odhadnout pomocí lineárních nejmenších čtverců neboli

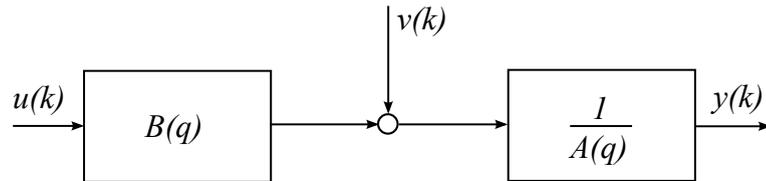


OBRÁZEK 4.3: Simulace a predikce

lineární regresi (Kap. 3.1). Blokové schéma ARX modelu je znázorněno v Obr. 4.4 a koresponduje s rovnicí

$$\begin{aligned} y(k) + a_1 y(k-1) + \dots + a_{n_a} y(k-n_a) \\ = b_0 u(k) + b_1 u(k-1) + \dots + b_{n_b} u(k-n_b) + e(k), \end{aligned} \quad (4.14)$$

přičemž $u(k)$ je posloupnost vstupů a $y(k)$ posloupnost výstupů a obě jsou známé, $v(k)$ představuje šum. A a B jsou odhadované parametry ARX modelu, tzn. polynomy pro SISO resp. polynomiální matice pro MIMO systém.



OBRÁZEK 4.4: ARX model

Definováním vektoru neznámých parametrů

$$\theta = [a_1 \dots a_{n_a} b_0 b_1 \dots b_{n_b}]^T \quad (4.15)$$

a vektoru dat

$$\varphi(k) = [-y(k-1) \dots -y(k-n_a) u(k) \dots u(k-n_b)]^T, \quad (4.16)$$

lze (4.14) zapsat ve tvaru lineární regrese

$$y(k) = \varphi^T(k)\theta + e(k). \quad (4.17)$$

Odhad neznámých parametrů θ se v nejjednodušším případě provede metodou LS podle (3.8) [11], kde

$$Y = [y(1) \ y(2) \ \dots \ y(N)] \quad (4.18)$$

$$\Phi = [\varphi(1) \ \varphi(2) \ \dots \ \varphi(N)]. \quad (4.19)$$

Optimální ARX prediktor popisuje rovnice

$$\hat{y}(k|k-1) = B(q)u(k) + (1 - A(q))y(k), \quad (4.20)$$

kterou lze rozepsat do tvaru

$$\begin{aligned} \hat{y}(k|k-1) = & b_1u(k-1) + \dots + b_mu(k-m) \\ & - a_1y(k-1) - \dots - a_my(k-m) \end{aligned} \quad (4.21)$$

za předpokladu, že stupeň A i B je roven m . Chyba predikce je potom dána

$$\varepsilon(k) = A(q)y(k) - B(q)u(k). \quad (4.22)$$

ARX prediktor je vždy stabilní, a to i v případě, že $A(q)$ (a tím i celý ARX model) je nestabilní. Tato vlastnost umožňuje modelovat nestabilní procesy pomocí ARX modelu.

Odhad parametrů ARX modelu lze provést kromě metody LS i několika dalšími způsoby, jako metodou *instrumentálních proměnných (IV)* (*instrumental variables (IV)*) a mnoha dalšími. Podrobnosti o používaných metodách jsou popsány v [3].

4.3.2 ARMAX

Model ARMAX (*Autoregressive Moving Average with Exogenous Input*) ve srovnání s ARX modelem nabízí mnohem větší flexibilitu, protože umožňuje samostatné modelování šumu. Blokové schéma ARMAX modelu je znázorněno v Obr. 4.5 a je popsáno rovnicí

$$\begin{aligned} & y(k) + a_1y(k-1) + \dots + a_{n_a}y(k-n_a) \\ = & b_0u(k) + b_1u(k-1) + \dots + b_{n_b}u(k-n_b) \\ & + e(k) + c_1e(k-1) + \dots + c_{n_c}e(k-n_c). \end{aligned} \quad (4.23)$$

Jak lze pozorovat v Obr. 4.5, položením $C(q) = 1$, model ARMAX přechází v ARX. Položením $C(q) = A(q)$, ARMAX přechází v OE model (viz. Kap. 4.3.3).

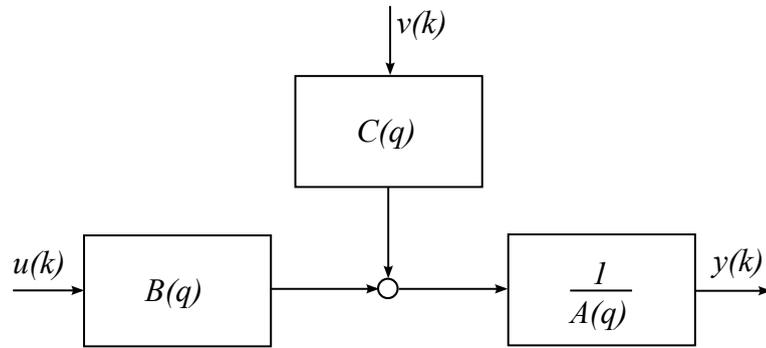
Definujeme vektor neznámých parametrů ve tvaru

$$\theta = [a_1 \dots a_{n_a} b_0 b_1 \dots b_{n_b} c_1 \dots c_{n_c}]^T \quad (4.24)$$

a vektor dat

$$\begin{aligned} \varphi(k) &= [-y(k-1) \dots -y(k-n_a) u(k) \dots u(k-n_b) \\ &\quad \varepsilon(k-1|k-2) \dots \varepsilon(k-n_c|k-n_c-1)]^T, \end{aligned} \quad (4.25)$$

kde $\varepsilon(k|k-1) = y(k) - \hat{y}(k|k-1)$ je chyba predikce. Flexibilita ARMAX modelu, umožňující samostatné modelování šumu, je vykoupena tím, že model je nelineární v parametrech, neboť chyby predikce jsou závislé na parametrech θ a (4.17) je zde problémem tzv. pseudolineární regrese.



OBRÁZEK 4.5: ARMAX model

Optimální ARMAX prediktor je

$$\hat{y}(k|k-1) = \frac{B(q)}{C(q)} u(k) + \left(1 - \frac{A(q)}{C(q)}\right) y(k). \quad (4.26)$$

ARMAX prediktor je stabilní, a to i v případě, že polynom $A(q)$ (a tím i celý ARMAX model) je nestabilní. Polynom $C(q)$ je však požadován stabilní.

Parametry ARMAX modelu lze odhadovat pomocí vícestupňových LS algoritmů a obejít tak použití nelineárních optimalizačních technik. Používá se např. metoda RELS (*Recursive Extended Least Squares*), v některé literatuře označovaná jenom jako ELS (*Extended Least Squares*), která je detailněji popsána v [3].

4.3.3 Output Error (OE)

Output Error model neboli model s *chybou výstupu* uvažuje aditivní přičítání šumu k výstupu procesu, což může být výhodou z hlediska korespondence s většinou reálných procesů (zdrojem stochastické složky je zde chyba měření). OE model je ekvivalentní ARMAX modelu s parametry

$C(q) = A(q)$. Schéma OE modelu je znázorněno v Obr. 4.6 a je popsáno rovnicí

$$\begin{aligned} x(k) + a_1x(k-1) + \dots + a_{n_a}x(k-n_a) \\ = b_0u(k) + b_1u(k-1) + \dots + b_{n_b}u(k-n_b) \\ y(k) = x(k) + e(k). \end{aligned} \quad (4.27)$$

Po definování vektoru neznámých parametrů

$$\theta = [a_1 \dots a_{n_a} b_0 b_1 \dots b_{n_b}]^T \quad (4.28)$$

a vektoru dat

$$\varphi(k) = [\hat{y}(t-1|t-2) \dots \hat{y}(t-n_a|t-n_a-1) u(k) u(k-1) \dots u(t-n_b)]^T \quad (4.29)$$

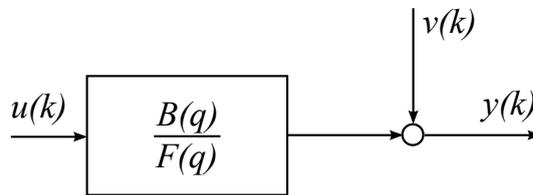
je (4.17) opět pseudolineární regrese, jako v případě ARMAX modelu.

Optimální OE prediktor je¹

$$\hat{y}(k|k-1) = \hat{y}(k) = \frac{B(q)}{F(q)}u(k), \quad (4.30)$$

který je ale ve skutečnosti simulátorem, protože predikce výstupu závisí pouze na minulých měřených vstupech. Co se týká stability prediktoru, je nestabilní, pokud je nestabilní polynom $F(q)$. Z toho důvodu nelze OE model použít pro modelování nestabilních procesů. Chyba predikce OE modelu je dána

$$\varepsilon(k) = y(k) - \frac{B(q)}{F(q)}u(k). \quad (4.31)$$



OBRÁZEK 4.6: OE model

4.3.4 Box - Jenkins (BJ)

Box - Jenkins model představuje rozšíření OE a ARMAX modelu, které umožňuje modelovat libovolný barevný šum na výstupu procesu. Barevný šum lze generovat filtrací vstupujícího bílého šumu přes lineární filtr s libovolným čitatelem $C(q)$ a jmenovatelem $D(q)$. Je zřejmé, že

¹Pozn. Notaci " $|k-1$ " lze odstranit, neboť optimální predikce není závislá na předchozích výstupech procesu.

položením $C(q) = D(q)$, se BJ model zjednoduší na OE². Struktura BJ modelu je nejobecnější ze všech používaných lineárních modelů a nabízí tak dostatečnou flexibilitu. Umožňuje separátně odhadovat přenosové funkce (s libovolnými čitateli i jmenovateli) identifikovaného procesu ze vstupu na výstup a ze vstupujícího šumu na výstup procesu. Flexibilita modelu je však vykoupěna složitostí odhadování, neboť je třeba odhadovat značné množství parametrů. BJ modely nejsou vhodné pro případy, kde je kritickým bodem rychlost odhadování, a pro případy, kdy je k dispozici jenom malé množství (zkreslených) dat.

BJ model je znázorněn v Obr. 4.7 a popisuje ho rovnice

$$y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}v(k). \quad (4.32)$$

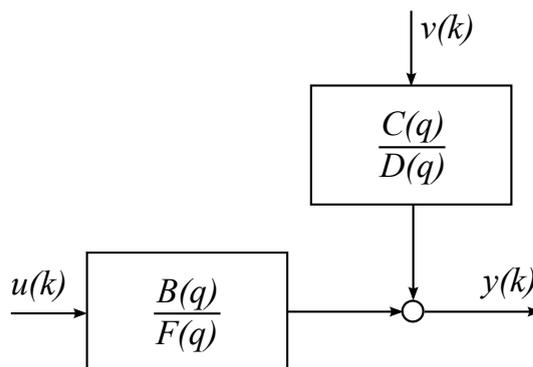
Optimální BJ prediktor má tvar

$$\hat{y}(k|k-1) = \frac{B(q)D(q)}{F(q)C(q)}u(k) + \frac{C(q) - D(q)}{C(q)}y(k). \quad (4.33)$$

Chybu predikce BJ modelu lze vyjádřit ve tvaru

$$\varepsilon(k) = \frac{D(q)}{C(q)}y(k) - \frac{B(q)D(q)}{F(q)C(q)}u(k). \quad (4.34)$$

Odhad parametrů BJ modelu se obvykle provádí pomocí nelineární optimalizace. Jednou z variant je prvotní odhad parametrů ARX modelu za účelem stanovení počátečních hodnot parametrů b_i a f_i . Následuje proces nelineární optimalizace s výpočtem predikce. Celý postup je popsán v [3].



OBRÁZEK 4.7: BJ model

²Další speciální případy BJ modelu: $D(q) = 1 \dots$ ARMAX; $C(q) = 1 \dots$ ARARX

4.3.5 State space

State space, neboli stavový popis, je běžně používaným vnitřním popisem LTI systému a je určen rovnicemi (1.2), kde $w(k)$ a $v(k)$ jsou bílé posloupnosti s kovarianční maticí

$$E \left\{ \begin{bmatrix} v(k) \\ w(k) \end{bmatrix} \cdot \begin{bmatrix} v(j)^T & w(j)^T \end{bmatrix} \right\} = \begin{bmatrix} R(k) & S(k)^T \\ S(k) & Q(k) \end{bmatrix} \delta(k-j) \geq 0. \quad (4.35)$$

kde $\delta(k-j)$ je jednotkový puls v čase $(k-j)$. Jednou z forem stavového popisu, která bude dále v práci používána, je tzv. kanonická forma pozorovatelnosti (OCF). Tato forma má přímou souvislost s ARMAX modelem (4.23)³

$$\begin{aligned} x(k+1) &= \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n-1} & 0 & 0 & \dots & 1 \\ -a_n & 0 & 0 & \dots & 0 \end{bmatrix} x(k) + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} u(k) + \begin{bmatrix} c_1 - a_1 \\ c_2 - a_2 \\ \vdots \\ c_{n-1} - a_{n-1} \\ c_n - a_n \end{bmatrix} v(k) \quad (4.36) \\ y(k) &= \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix} x(k) + v(k). \end{aligned}$$

4.4 Kalmanův filtr

Kalmanův filtr je speciální případ modelu, umožňující odhadovat stav systému na základě pozorování vstupů a výstupů systému. V deterministické formulaci KF se k odhadování stavu používá pozorovatel stavu. Ve stochastické formulaci lze úlohu odhadu stavu formulovat ve smyslu optimálního LMS odhadu (Kap. 5.1) stavu a výsledný optimální pozorovatel stavu se nazývá Kalmanův filtr [5]. Pro názornost pochopení principu Kalmanova filtru (KF) bude nejprve zaveden pojem asymptotický pozorovatel stavu, poté bude definován KF problém. V další části bude popsán mechanismus Kalmanova filtru v reprezentaci pro stochastické procesy. V závěru této sekce bude popsán algoritmus tzv. square-root kovariančního filtru pro efektivní řešení KF problému.

4.4.1 Asymptotický pozorovatel stavu

Asymptotický pozorovatel stavu je filtr, který na základě modelu dynamického systému a naměřených vstupních a výstupních datových sekvencí, aproximuje stavový vektor systému. Pokud

³Uvažujeme ARMAX model se zpožděním $n_k = 1$.

uvažujeme stavový model ve tvaru

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) \\y(k) &= Cx(k) + Du(k),\end{aligned}\tag{4.37}$$

lze stavový vektor aproximovat rovnicí

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k).\tag{4.38}$$

Pokud je počáteční stav $\hat{x}(0)$ roven $x(0)$, průběhy stavových veličin $\hat{x}(k)$ a $x(k)$ jsou si rovny. V případě, že jsou počáteční podmínky rozdílné, sekvence $\hat{x}(k)$ a $x(k)$ se vyrovnají až po určitém čase (za předpokladu, že je matice A asymptoticky stabilní). Rozdíl mezi odhadem stavového vektoru $\hat{x}(k)$ a skutečným stavovým vektorem $x(k)$ je dán

$$x_e(k) = \hat{x}(k) - x(k),\tag{4.39}$$

zároveň platí $x_e(k+1) = Ax_e(k)$. Uvedený případ neposkytuje kontrolu nad časem, za který $\hat{x}(k)$ aproximuje $x(k)$. Proto se zavádí korekce v podobě rozdílu mezi měřeným výstupem $y(k)$ a odhadovaným výstupem $\hat{y}(k) = C\hat{x}(k) + Du(k)$, a rovnice (4.38) přechází do tvaru

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K(y(k) - C\hat{x}(k) - Du(k)).\tag{4.40}$$

Rozdíl mezi odhadovaným a měřeným stavem je dán:

$$x_e(k+1) = (A - KC)x_e(k).\tag{4.41}$$

K představuje matici zesílení a systém reprezentovaný rovnicí (4.40) se nazývá asymptotický pozorovatel. Volba velikosti zesílení K takového, že matice $A - KC$ je asymptoticky stabilní⁴, zajistí, že rozdíl mezi odhadovaným $\hat{x}(k)$ a reálným $x(k)$ konverguje k nule pro $k \rightarrow \infty$:

$$\lim_{k \rightarrow \infty} x_e(k) = \lim_{k \rightarrow \infty} (\hat{x}(k) - x(k)) = 0.\tag{4.42}$$

4.4.2 Definice KF

Úlohou KF je aproximovat stavový vektor dynamického systému z naměřených vstupních a výstupních sekvencí - stejně jako u pozorovatele - s tím rozdílem, že KF uvažuje vstupující šum a rovnice (4.37) přecházejí do podoby (1.2), kde $w(k)$ je šum procesu a $v(k)$ šum měření, přičemž je definována kovarianční matice (4.35). S použitím pozorovatele (4.40) pro rekonstrukci stavového vektoru systému (1.2), je rozdíl $x_e(k) = \hat{x}(k) - x(k)$ mezi odhadovaným $\hat{x}(k)$ a reálným

⁴Pokud je (A, C) pozorovatelná, existuje matice K zajišťující asymptotickou stabilitu $A - KC$.

stavovým vektorem $x(k)$

$$x_e(k+1) = (A - KC)x_e(k) - w(k) + Kv(k). \quad (4.43)$$

Je zřejmé, že v tomto případě již asymptotická stabilita $A - KC$ nezajistí nulový rozdíl $x_e(k)$ pro $k \rightarrow \infty$, kvůli přítomnosti šumů $w(k)$ a $v(k)$. Úkolem KF je rozdíl mezi odhadem stavu a skutečným stavem $x_e(k)$ minimalizovat. Konkrétně požadujeme, aby střední hodnota $E[x_e(k)]$ byla rovna nule, tzn., že hodnota x_e osciluje kolem nuly. Kovarianční matice $E[x_e(k)x_e(k)^T]$ musí být minimální možná. Jinými slovy hledáme nevychýlený odhad stavu $\hat{x}(k)$ s minimálním rozptylem.

4.4.3 Implementace KF

Předpokládáme, že v k -tém kroku známe apriorní⁵ odhad stavu $x(k)$, tj. $\hat{x}(k|k-1)$ a kovarianční matici chyby odhadu stavu $P(k|k-1)$. Po změření hodnoty $y(k)$ chceme hodnoty aktualizovat - získat aposteriorní⁶ odhad stavu $\hat{x}(k|k)$ a kovarianční matici chyby odhadu $P(k|k)$. Dále požadujeme na základě aposteriorního odhadu stavu $\hat{x}(k|k)$ a kovarianční matice $P(k|k)$ v čase k nalezení apriorních hodnot těchto veličin $\hat{x}(k+1|k)$ a $P(k+1|k)$ v čase $k+1$. Měření $y(k)$ bylo získáno pomocí lineárního modelu (1.2), a tudíž úloha nalezení optimálního odhadu $\hat{x}(k+1|k)$ koresponduje s LMS odhadem (Kap. 5.1) [5].

Pokud uvažujeme nekorelované šумы procesu a měření ($S = 0$), lze z (5.2) a (5.3) odvodit vztahy pro odhad stavu $\hat{x}(k+1|k)$ a kovarianční matice $P(k+1|k)$, které představují algoritmus Kalmanova filtru a dají se buď rozdělit do dvou nezávislých kroků (datový a časový), nebo oba kroky spojit v jeden, a tím získat Algoritmus 3. Odvození je netriviální a poměrně zdlouhavé, proto ho nebudeme provádět. Podrobnosti o odvození lze nalézt v [5, 6].

4.4.4 Square-root implementace KF

Jako další variantu, vedle klasické implementace KF uvedeme tzv. *square-root* kovarianční filtr (SRKF) [6]. Implementace pracuje s tzv. Choleského odmocninou. Choleského odmocnina pozitivně semi-definitní matice $X^{-1}(k)$ je definována jako trojúhelníková matice, která má nezáporné prvky na hlavní diagonále a splňuje vztah $X^{-1}(k) = Y(k)Y^T(k)$, přičemž transpozice Choleského odmocniny $Y^T(k)$ je horní trojúhelníková matice. Numerická výhoda square-root implementace spočívá v tom, že počítá-li se rekurzivně místo matice $X^{-1}(k)$ její odmocnina $Y(k)$, pak pro jakoukoliv reálnou matici $Y(k)$ je součin $X^{-1}(k) = Y(k)Y^T(k)$ vždy pozitivně semi-definitní matice. Navíc odmocniny malých čísel jsou podstatně větší než tato čísla, a proto

⁵Apriorním odhadem stavu rozumíme, že známe data až do času $k-1$, ale nebereme v úvahu data y .

⁶Aposteriorním odhadem rozumíme odhad zahrnující měření $y(k)$.

Algoritmus 3: KF jako jednokrokový prediktor stavu [6]

Vstup: $A(k)$, $B(k)$, $C(k)$, $\hat{x}(0|-1)$, $P(0|-1)$ pro $k = 0, 1, 2, \dots, N-1$
for $k = 0, 1, 2, \dots, N-1$ **do**

 Urči Kalmanovo zesílení $K(k)$:

$$K(k) = (S(k) + A(k)P(k|k-1)C(k)^T) \quad (4.44)$$

$$\times (R(k) + C(k)P(k|k-1)C(k)^T)^{-1}. \quad (4.45)$$

Vypočti predikci stavu

$$\hat{x}(k+1|k) = A(k)\hat{x}(k|k-1) + B(k)u(k) + K(k)(y(k) - C(k)\hat{x}(k|k-1)). \quad (4.46)$$

 Aktualizuj kovarianční matici vyřešením *Riccatiho rovnice*

$$P(k+1|k) = A(k)P(k|k-1)A(k)^T + Q(k) \quad (4.47)$$

$$- (S(k) + A(k)P(k|k-1)C(k)^T) \quad (4.48)$$

$$\times (R(k) + C(k)P(k|k-1)C(k)^T)^{-1} \quad (4.49)$$

$$\times (S(k) + A(k)P(k|k-1)C(k)^T)^T. \quad (4.50)$$

nejsou numerické operace zatíženy velkou zaokrouhlovací chybou. SRKF má uplatnění v případech, kdy je požadován rychlý výpočet a vysoká numerická přesnost.

Pro odvození square-root řešení návrhu algoritmu odhadu stavu Kalmanovým filtrem nejprve uvažujme úplný problém nejmenších čtverců (3.9), přičemž známe střední hodnotu \bar{x} a kovarianční matici $P = [(x - \bar{x})(x - \bar{x})^T]$. Pro pozdější maticové vyjádření se zavádí pomocná proměnná ξ , reprezentující náhodnou veličinu s nulovou střední hodnotou a kovarianční maticí I_n : $\xi \sim (0, I_n)$ [6]. Pokud je matice P semi-pozitivně-definitní, můžeme vypočítat její odmocninu ($P = P^{1/2}P^{T/2}$). Odhad proměnné x lze potom určit pomocí odmocniny (square-root) kovarianční matice P z rovnice (rovnice je známá pod názvem *generalized covariance representation* z [32])

$$x = \bar{x} - P^{1/2}\xi. \quad (4.51)$$

Jádrem celé implementace je Choleského faktorizace kovarianční matice (viz. [6]).

Dále v celé sekci budeme pracovat s odvozením algoritmu SRKF převzatým z [6]. Uvažujeme stochastický systém (1.2) se známou kovarianční maticí (4.35). Definujeme odmocninu $L(k)$ jako

$$\begin{bmatrix} R(k) & S(k)^T \\ S(k) & Q(k) \end{bmatrix} = L(k)L(k)^T, \quad (4.52)$$

poté lze šum procesu a šum měření vyjádřit namísto (4.35) rovnicí

$$\begin{bmatrix} v(k) \\ w(k) \end{bmatrix} = \underbrace{\begin{bmatrix} R(k)^{1/2} & 0 \\ X(k) & Q_x(k)^{1/2} \end{bmatrix}}_{L(k)} \begin{bmatrix} \tilde{v}(k) \\ \tilde{w}(k) \end{bmatrix}, \quad \begin{bmatrix} \tilde{v}(k) \\ \tilde{w}(k) \end{bmatrix} \sim (0, I_{l+n}), \quad (4.53)$$

kde

$$X(k) = S(k)R(k)^{-T/2} \quad (4.54)$$

$$Q_x(k) = Q(k) - S(k)R(k)^{-1}S(k)^T. \quad (4.55)$$

Navíc podle (4.51) platí, že

$$x(k) = \hat{x}(k|k-1) - P(k|k-1)^{1/2}\tilde{x}(k). \quad (4.56)$$

Uspořádáním do maticového tvaru dostáváme omezení na neznámé stavy $x(k)$ a $x(k+1)$

$$\begin{bmatrix} \hat{x}(k|k-1) \\ y(k) \\ -B(k)u(k) \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ C(k) & 0 \\ A(k) & -I_n \end{bmatrix} \begin{bmatrix} x(k) \\ x(k+1) \end{bmatrix} \quad (4.57)$$

$$+ \begin{bmatrix} P(k|k-1)^{1/2} & 0 & 0 \\ 0 & R(k)^{1/2} & 0 \\ 0 & X(k) & Q_x(k)^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{x}(k) \\ \tilde{v}(k) \\ \tilde{w}(k) \end{bmatrix}. \quad (4.58)$$

Přepíšeme-li výše uvedené rovnice do kompaktního tvaru

$$\bar{y}(k) = \bar{F}(k)\bar{x}(k) + \bar{L}(k)\bar{\mu}(k), \quad (4.59)$$

dostáváme výsledný tvar problému návrhu square-root Kalmanova filtru

$$\min_{\bar{x}(k)} \bar{\mu}(k)^T \bar{\mu}(k) \quad s.t. \quad \bar{y}(k) = \bar{F}(k)\bar{x}(k) + \bar{L}(k)\bar{\mu}(k), \quad (4.60)$$

jehož řešení je odvozeno v [6] a shrnuje ho Algoritmus 4.

4.4.5 Použití Kalmanova filtru k odhadu stavu systému 2. řádu

Zvolme si systém druhého řádu, na kterém budeme ověřovat funkci klasického KF a SRKF

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & 0 \\ 0.1 & 0.95 \end{bmatrix} x(k) + \begin{bmatrix} 0.1 \\ 0.002 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} 0 & 1 \end{bmatrix} x(k). \end{aligned} \quad (4.61)$$

Algoritmus 4: Square-root kovarianční filtr [6]**Vstup:** $A(k)$, $B(k)$, $C(k)$, $\hat{x}(0| - 1)$, $P(0| - 1)^{1/2}$ pro $k = 0, 1, 2, \dots, N - 1$ **for** $k = 0, 1, 2, \dots, N - 1$ **do**Aplikuj transformaci T_r , která splňuje

$$\begin{aligned} & \begin{bmatrix} C(k)P(k|k-1)^{1/2} & R(k)^{1/2} & 0 \\ A(k)P(k|k-1)^{1/2} & X(k) & Q_x(k)^{1/2} \end{bmatrix} T_r \\ &= \begin{bmatrix} R^e(k)^{1/2} & 0 & 0 \\ G(k) & P(k+1|k)^{1/2} & 0 \end{bmatrix}. \end{aligned}$$

Vypočti predikci stavu

$$\begin{aligned} \hat{x}(k+1|k) &= A(k)\hat{x}(k|k-1) + B(k)u(k) \\ &+ G(k)R^e(k)^{-1/2}(y(k) - C(k)\hat{x}(k|k-1)) \end{aligned}$$

a aktualizuj kovarianční matici

$$E \left[(x(k+1) - \hat{x}(k+1|k)) (x(k+1) - \hat{x}(k+1|k))^T \right] = P(k+1|k)^{1/2} P(k+1|k)^{T/2}.$$

Nyní vygenerujme náhodné diskrétní bílé posloupnosti délky 200

$$w \sim \sqrt{Q} \times \mathcal{N}(0, 1) \quad (4.62)$$

$$v \sim \sqrt{R} \times \mathcal{N}(0, 1),$$

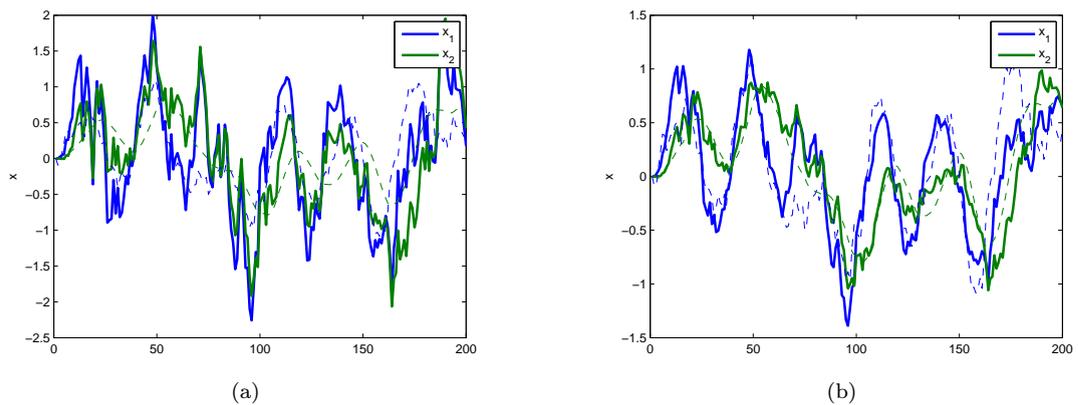
kde $Q = 1.5$ a $R = 0.5$. Vyrobíme vstupní signál $u_w = u + w$, který použijeme jako vstup do systému (4.61). Na systému vygenerujeme výstup y , který následně zkreslíme bílým šumem v a vznikne $y_v = y + v$. Předpokládáme, že nyní známe pouze vstupní signál u_w a výstupní signál y_v . Následně provedeme odhad stavů pomocí KF a SRKF, přičemž obě implementace, pro srovnání, identicky realizujeme jak v `Matlabu`, tak ve `Scilabu`⁷ pro porovnání výpočetních rychlostí.

Porovnání výpočetních rychlostí shrnuje Tab. 4.1. Je zřejmé, že implementace SRKF je výrazně rychlejší, než klasická implementace KF. Dalším bodem srovnání je fakt, že obě varianty implementace jsou rychlejší v případě `Scilabu`. Průběh odhadu stavů v porovnání KF a SRKF je znázorněn Obr. 4.8. Zrekonstruovaný výstup systému na základě odhadu stavů x_1 a x_2 je v Obr. 4.9. Chybu odhadu výstupu y znázorňuje Obr. 4.10. Správnost návrhu Kalmanova filtru ověřuje autokorelační funkce Obr. 4.11, odkud je patrné, že chyba odhadu je bílý šum.

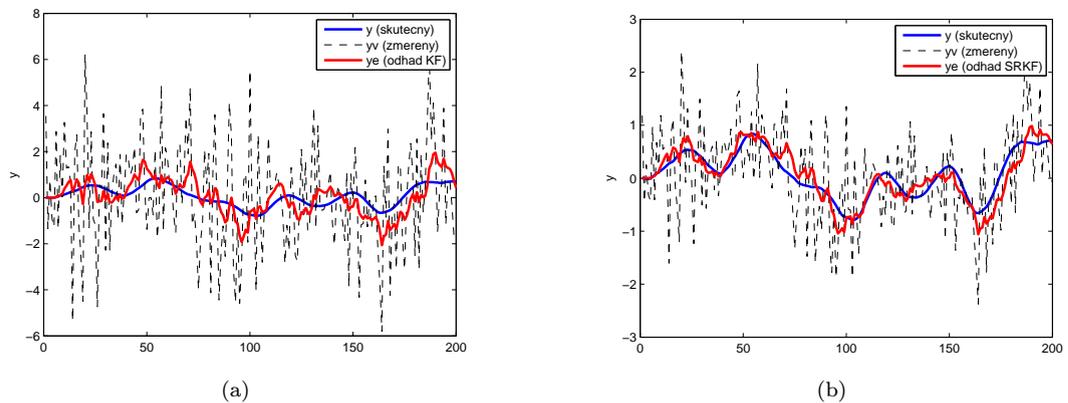
⁷Výpočet byl prováděn na stroji s Intel® Celeron® M 420 1.6 GHz, 533 MHz FSB, 1MB L2 cache, 32-bit; 2GB DDR2. Použit byl výpočetní systém `Matlab` ve verzi R2010a a `Scilab` ve verzi 5.3.

Matlab		Scilab	
KF	SRKF	KF	SRKF
0.125 s	0.098 s	0.124 s	0.067 s

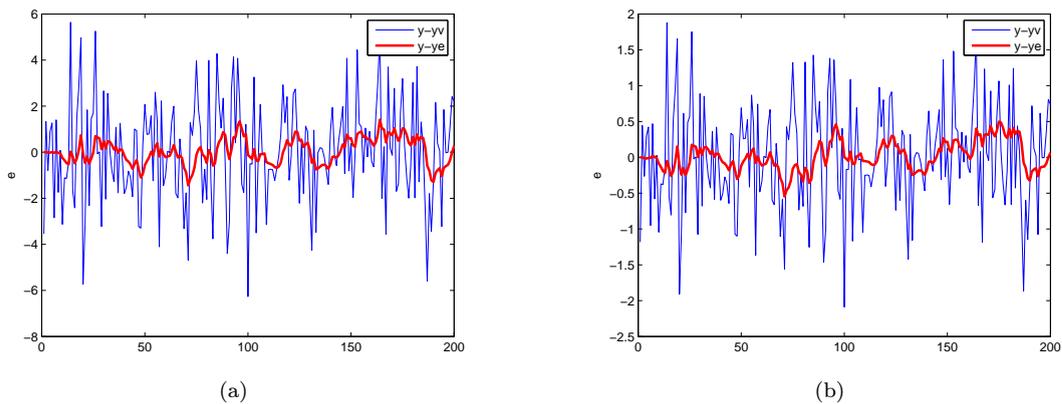
TABULKA 4.1: Porovnání výpočetních rychlostí algoritmů odhadu stavu systému klasickým (KF) a square-root kovariančním filtrem (SRKF) v *Matlabu* a ve *Scilabu*



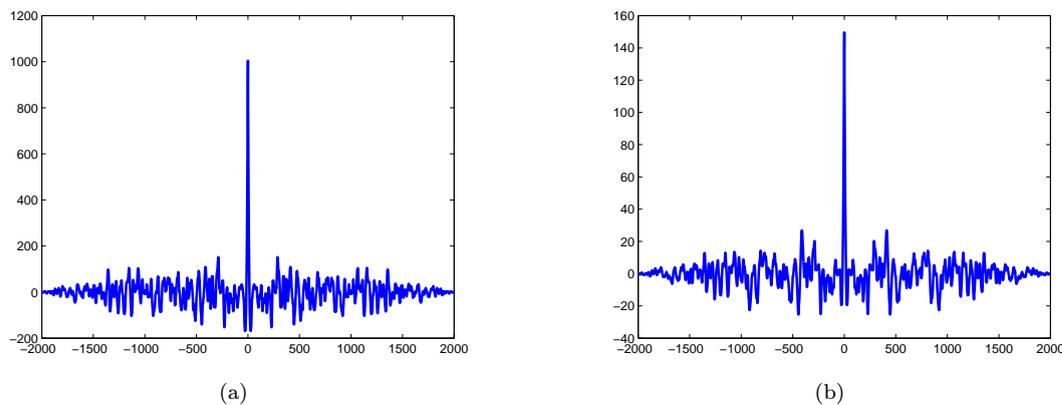
OBRÁZEK 4.8: Průběhy odhadu stavů systému Kalmanovým filtrem a square-root kovariančním filtrem



OBRÁZEK 4.9: Průběhy odhadu výstupů systému Kalmanovým filtrem a square-root kovariančním filtrem



OBRÁZEK 4.10: Průběhy chyb odhadů výstupů systému Kalmanovým filtrem a square-root kovariančním filtrem



OBRÁZEK 4.11: Autokorelační funkce chyb odhadů výstupu systému Kalmanovým filtrem a square-root kovariančním filtrem

Kapitola 5

Identifikační metody

5.1 LMS odhad

LMS (*Linear Mean Square*), neboli lineárně středně kvadratický odhad je lineární odhad minimalizující střední kvadratickou chybu

$$J_{LMS} = E\{\tilde{x}\tilde{x}^T\} = \text{tr } E\{\tilde{x}^T\tilde{x}\} = \text{tr } E\{(x - Ay - b)(x - Ay - b)^T\}, \quad (5.1)$$

přičemž výsledný vztah pro optimální LMS odhad lze odvodit využitím vztahů pro derivaci stopy matice¹ (odvození viz. [5])

$$\hat{x}_{LMS}(y) = \mu_x + P_{xy}P_{yy}^{-1}(y - \mu_y) \quad (5.2)$$

Kovarianční matice chyby odhadu se určí jako

$$P_{\hat{x}_{LMS}} = E\{(x - \hat{x}_{LMS}(y))(x - \hat{x}_{LMS}(y))^T\} = P_{xx} - P_{xy}P_{yy}^{-1}P_{yx}. \quad (5.3)$$

LMS odhad je ortogonální ke všem lineárním funkcím dat $f(y) = Ay + b$.

5.2 ML odhad

ML odhad (*Maximum likelihood*), neboli metoda *maximální věrohodnosti* [5, 33], je jednou ze základních statistických metod pro odhad neznámých parametrů v závislosti na pozorovaných datech. Metoda je založena na formulaci pravděpodobnostního modelu. Definujme deterministickou veličinu θ a pozorovaná data y a předpokládejme, že nemáme žádnou apriorní informaci o veličině θ , kterou chceme odhadovat. Závislost pozorované veličiny y na odhadované veličině θ

¹Střední hodnoty a kovarianční matice značíme $\mu_x = \varepsilon\{x\}$, $\mu_y = \varepsilon\{y\}$, $P_{xx} = \varepsilon\{(x - \mu_x)(x - \mu_x)^T\}$, $P_{xy} = \varepsilon\{(x - \mu_x)(y - \mu_y)^T\}$, $P_{yy} = \varepsilon\{(y - \mu_y)(y - \mu_y)^T\}$.

je dána podmíněnou hustotou pravděpodobnosti $p(y|\theta)$. Dále definujeme věrohodnostní funkci $l(\theta|y) = p(y|\theta)$, která bude pro dané měření y reprezentovat podmíněnou hustotu pravděpodobnosti $p(y|\theta)$. Maximálně věrohodný odhad $\hat{\theta}_{ML}(y)$ je definován jako hodnota θ , maximalizující věrohodnostní funkci pro pozorovaná data

$$\hat{\theta}_{ML}(y) = \arg \max_{\theta} l(\theta|y). \quad (5.4)$$

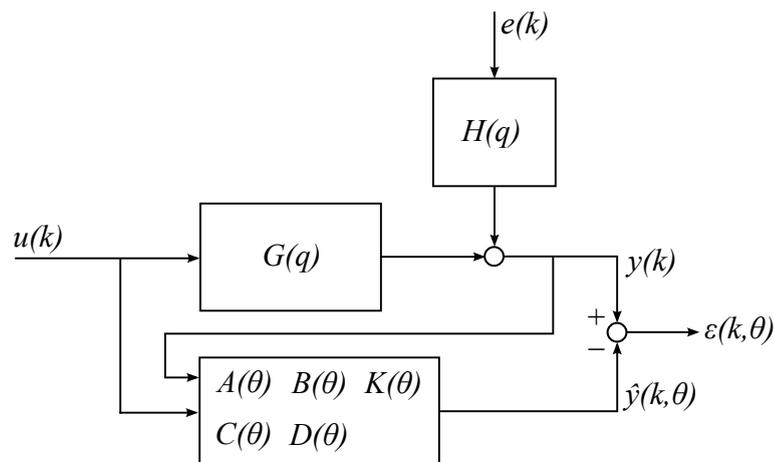
Metoda maximální věrohodnosti pracuje pro výpočetní jednoduchost s předpokladem, že hustoty pravděpodobnosti mají tvar exponenciální funkce. Proto se jako věrohodnostní funkce, volí funkce logaritmická $\ln(\cdot)$. Podmínku maxima věrohodnostní funkce potom vyjadřuje tzv. věrohodnostní rovnice

$$\frac{\partial \ln l(\theta|y)}{\partial \theta} = 0. \quad (5.5)$$

Praktickou ukázkou použití ML metody pro odhad neznámé veličiny θ zatížené normálně rozděleným šumem pro konkrétní model $y = C\theta + e$ poskytně [5]. Lze ukázat, že odhad neznámých parametrů metodou maximální věrohodnosti vede na minimalizaci váženého součtu čtverců chyb predikce výstupu

$$\varepsilon = y - C\tilde{\theta}_{ML}. \quad (5.6)$$

5.3 Prediction error metoda (PEM)



OBRÁZEK 5.1: Prediction-error metoda

Prediction error metoda, neboli metoda *predikčních chyb* uvažuje model ve tvaru (4.1) a hledá takový vektor parametrů θ_N , který nejlépe popisuje pozorovaná data Z^N ve smyslu minimalizace kritéria

$$\hat{\theta}_N = \arg \min_{\theta \in \mathbb{R}^n} J_N(\varepsilon(k, \theta), Z^N), \quad (5.7)$$

příčemž jako ztrátová funkce se obvykle volí kvadratická funkce

$$J_N(\theta, Z^N) = \frac{1}{N} \sum_{k=0}^{N-1} \|y(k) - \hat{y}(k|k-1, \theta)\|_2^2, \quad (5.8)$$

kde $\hat{y}(t|t-1, \theta)$ je predikovaný výstup generovaný libovolným modelem z Kap. 4.3. Pokud uvažujeme některý z LTI modelů jako ARX, ARMAX apod., kritérium má konkrétní podobu (2.4). Odhad parametrů metodou PEM tedy minimalizuje vážený součet čtverců chyb predikce (tím souvisí s ML metodou z Kap. 5.2). Metoda PEM je někdy označována jako zobecněný framework pro systémovou identifikaci, ze kterého se odvozuji další metody pomocí určitých zjednodušení (viz. Kap. 5.4).

5.4 Output error metoda (OEM)

Output error metoda, neboli metoda *chyb výstupu*, je speciálním případem metody PEM (Kap. 5.3), kdy minimalizujeme vážený součet čtverců chyb predikce výstupu, ale do predikce nezahrnujeme informace o výstupu skutečného procesu

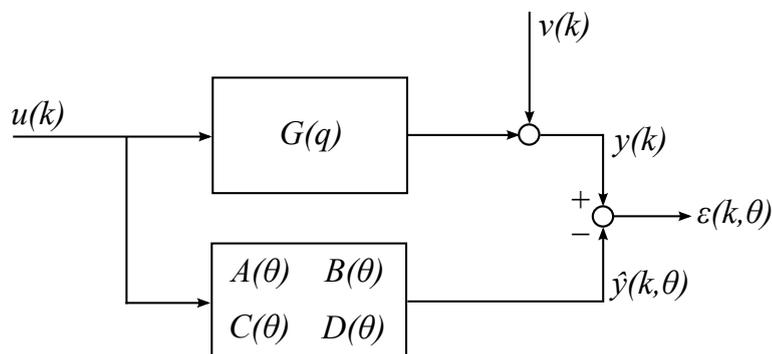
$$\hat{\theta}_N = \arg \min_{\theta \in \mathbb{R}^n} \frac{1}{N} \sum_{k=0}^{N-1} \|y(k) - \hat{y}(k, \theta)\|_2^2 \quad (5.9)$$

a navíc uvažujeme model ve tvaru

$$y(k) = G(q)u(k) + w(k), \quad (5.10)$$

kde $w(k)$ je šum měření, narozdíl od metody PEM, kdy uvažujeme samostatné modelování šumu pomocí filtru $H(q)$.

Další zobecnění metody PEM představují např. LSM, GLSM, ELSM, PLRM, MLM (viz. [1]).



OBRÁZEK 5.2: Output-error metoda

5.5 Subspace

Metody Subspace identifikace, někdy též označované jako 4SID - *Subspace State-Space System Identification*, jsou poměrně novou metodou a poskytují velmi výkonnou alternativu ke klasickým metodám založených na MS, ML odhadech apod. Existuje mnoho literatury zabývající se rozsáhle touto problematikou, ať už z hlediska teorie, či z hlediska numerické implementace, jako je např. [34, 35], stručnější přehledy a vysvětlení principů potom nabídne např. [6, 26]. Podstatnou výhodou subspace metod je fakt, že, na rozdíl od klasických přístupů, které odhadují číselné hodnoty parametrů LTI modelů, jakými jsou např. ARX, ARMAX apod., výsledkem subspace identifikace je přímo model ve formě stavového popisu. Dalšími klady subspace identifikace, které jsou oceňovány zejména v praxi, jsou malý počet ladících parametrů, stejná složitost odhadování pro SISO i MIMO systémy i schopnost odhadu řádu systému. Co se týká predikčních schopností výsledného modelu, tak subspace metody hledají modely optimalizované na více-krokové predikce (jsou vhodné pro aplikace jako MPC), zatímco např. klasické metody PEM hledají modely s jednokrokově optimálními predikcemi. Nevýhodami subspace metod jsou naopak špatná použitelnost pro malé soubory dat a velmi náročná rekurzifikace. Mezi nevýhody lze zařadit i fakt, že subspace metody jsou velmi abstraktní a dílčí kroky jsou obtížně fyzikálně interpretovatelné.

5.5.1 Princip Subspace identifikace

Metoda subspace identifikace představuje explicitní numerický algoritmus pro získání odhadů stavů a matic stavového popisu systému ze vstupně výstupních datových posloupností. Termín *subspace* vyjadřuje skutečnost, že posloupnost stavů systému lze přímo získat z řádkových a sloupcových podprostorů blokově uspořádaných Hankelových matic pouze ze vstupně výstupních dat.

Dále v celé kapitole uvažujme algoritmus stochastické identifikace z [34]. Subspace identifikační algoritmy uvažují model ve tvaru (1.2), který lze ekvivalentně zapsat jako

$$y(t + \tau) = CA^\tau x(t) + \sum_{i=0}^{\tau-1} CA^{\tau-i-1} Bu(t + i) + Du(t + \tau) + \sum_{i=0}^{\tau-1} CA^{\tau-i-1} w(t + i) + v(t + \tau). \quad (5.11)$$

Zavedením horizontu predikce α a

$$y_\alpha(t) \triangleq (y(t)^T \ y(t+1)^T \ \cdots \ y(t+\alpha-1)^T)^T, \quad (5.12)$$

obdobně pro $u_\alpha(t)$, $w_\alpha(t)$, $v_\alpha(t)$, je možno vyjádřit rovnici (5.11) v maticovém tvaru

$$y_\alpha(t) = \Gamma_\alpha x(t) + \Phi_\alpha u_\alpha(t) + \Psi_\alpha w_\alpha(t) + v_\alpha(t), \quad (5.13)$$

kde

$$\Gamma_\alpha \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\alpha-1} \end{bmatrix}, \quad \Phi_\alpha \triangleq \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & & \\ \vdots & & \ddots & \\ CA^{\alpha-2}B & CA^{\alpha-3}B & \cdots & D \end{bmatrix}, \quad (5.14)$$

$$\Psi_\alpha \triangleq \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C & 0 & & \\ CA & C & \ddots & \\ \vdots & & & \\ CA^{\alpha-2} & CA^{\alpha-3} & \cdots & 0 \end{bmatrix}.$$

Matice Γ_α se nazývá rozšířená matice pozorovatelnosti. Uspořádáním dat z rovnice (5.13) do maticového tvaru

$$Y_{\beta,\alpha,N} = \Gamma_\alpha X_{\beta,N} + \Phi_\alpha U_{\beta,\alpha,N} + \Psi_\alpha W_{\beta,\alpha,N} + V_{\beta,\alpha,N}, \quad (5.15)$$

kde

$$Y_{\beta,\alpha,N} \triangleq (y_\alpha(\beta) \ y_\alpha(\beta+1) \ \cdots \ y_\alpha(\beta+N-1)), \quad (5.16)$$

obdobně jako $U_{\beta,\alpha,N}, W_{\beta,\alpha,N}, V_{\beta,\alpha,N}$ a

$$X_{\beta,N} \triangleq (X(\beta) \ X(\beta+1) \ \cdots \ X(\beta+N-1)), \quad (5.17)$$

vznikne tzv. datová rovnice, která je výchozím bodem algoritmu. Parametr β je jedním z ladících parametrů algoritmu a určuje rozdělení matic na tzv. *minulá* data p a *budoucí* data f . Hankelovy matice vstupů a výstupů mají potom podobu

$$\begin{bmatrix} U_p \\ U_f \end{bmatrix} = \begin{bmatrix} u_0 & u_1 & \cdots & u_{j-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{\beta-1} & u_\beta & \cdots & u_{\beta+j-2} \\ u_\beta & u_{\beta+1} & \cdots & u_{\beta+j-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2\beta-1} & u_{2\beta} & \cdots & u_{N-1} \end{bmatrix}, \quad \begin{bmatrix} Y_p \\ Y_f \end{bmatrix} = \begin{bmatrix} y_0 & y_1 & \cdots & y_{j-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{\beta-1} & y_\beta & \cdots & y_{\beta+j-2} \\ y_\beta & y_{\beta+1} & \cdots & y_{\beta+j-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{2\beta-1} & y_{2\beta} & \cdots & y_{N-1} \end{bmatrix}.$$

Jádrem identifikačního algoritmu je optimální lineární víceukový prediktor budoucích výstupů

Y_f ze znalosti U_p, Y_p, U_f , přičemž se minimalizuje střední kvadratická chyba predikce. Řešení lze nalézt metodou nejmenších čtverců a dá se ukázat, že z části predikovaného výstupu, která se získá lineární transformací minulých dat U_p a Y_p se dá extrahovat rozšířená matice pozorovatelnosti z datové rovnice (5.13), na základě které lze snadno určit výsledné matice systému A a C , a v deterministickém případě i stavovou posloupnost. Ve stochastickém případě lze extrahovat stavovou posloupnost predikcí Kalmanových filtrů běžících paralelně po sloupcích Hankelových matic minulých dat, což podrobněji popisuje [34]. Pro osamostatnění rozšířené matice pozorovatelnosti se zavádí operátor projekce řádkového prostoru matice $U_{\beta,\alpha,N}$ na ortogonální doplněk tohoto řádkového prostoru

$$\Pi_{U_{\beta,\alpha,N}}^\perp \triangleq I - U_{\beta,\alpha,N}^T (U_{\beta,\alpha,N} U_{\beta,\alpha,N}^T)^{-1} U_{\beta,\alpha,N}. \quad (5.18)$$

Rozšířenou matici pozorovatelnosti lze odhadnout ze sloupcového prostoru matice

$$Y_{\beta,\alpha,N} \Pi_{U_{\beta,\alpha,N}}^\perp \begin{bmatrix} U_{0,\beta,N} \\ Y_{0,\beta,N} \end{bmatrix} \quad (5.19)$$

pomocí singulárního rozkladu (SVD)

$$Y_{\beta,\alpha,N} \Pi_{U_{\beta,\alpha,N}}^\perp \begin{bmatrix} U_{0,\beta,N} \\ Y_{0,\beta,N} \end{bmatrix} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}. \quad (5.20)$$

Odhad výsledné rozšířené matice pozorovatelnosti je $\hat{\Gamma}_\alpha = U_1^2$. Singulární čísla v matici Σ_1 mohou sloužit k odhadu řádu systému³. Z rovnice (5.14) vyplývají výsledné vztahy pro matice A a C

$$\hat{C} = \hat{\Gamma}_{\alpha,1}, \quad \hat{A} = \begin{bmatrix} \hat{\Gamma}_{\alpha,1} \\ \vdots \\ \hat{\Gamma}_{\alpha,\alpha} \end{bmatrix}^{-1} \begin{bmatrix} \hat{\Gamma}_{\alpha,2} \\ \vdots \\ \hat{\Gamma}_{\alpha,\alpha} \end{bmatrix}; \quad (5.21)$$

$$\hat{\Gamma}_\alpha = \begin{bmatrix} \hat{\Gamma}_{\alpha,1} \\ \vdots \\ \hat{\Gamma}_{\alpha,\alpha} \end{bmatrix}.$$

Uvedený postup nalezení rozšířené matice pozorovatelnosti je odvozen v [26].

Matice B a D lze potom odhadnout zavedením rovnice

$$y(t) = CA^t x(0) + \sum_{i=0}^{t-1} CA^{t-i-1} Bu(i) + Du(t) + n(t), \quad (5.22)$$

²Rozměr U_1 je $\alpha n_y \times n_x$.

³Počet nenulových singulárních čísel v matici Σ_1 je roven řádu systému.

která vznikne z (5.11) položením $t = 0$. $n(t)$ sdružuje příspěvky šumu procesu a měření. Rovnice (5.22) má ekvivalentní podobu

$$y(t) = CA^t x(0) + \left(\sum_{i=0}^{t-1} u(i)^T \otimes CA^{t-i-1} \right) \text{vec}(B) + (u(t)^T \otimes I_{n_y}) \text{vec}(D) + n(t). \quad (5.23)$$

Za předpokladu, že šum $n(t)$ je nekorelovaný se vstupním signálem $u(t)$, matice B a D lze, po zavedení vektoru parametrů $\theta = (x(0)^T \text{vec}(B)^T \text{vec}(D)^T)^T$ a

$$\varphi(t)^T = \left(\hat{C} \hat{A}^t \sum_{i=0}^{t-1} u(i)^T \otimes \hat{C} \hat{A}^{t-i-1} u(t)^T \otimes I_{n_y} \right), \quad (5.24)$$

odhadnout pomocí metody nejmenších čtverců (viz. Kap. 3.1). Odhad stavové posloupnosti lze provést dle vzorce

$$\hat{X}_{\beta,N} = \hat{\Gamma}_\alpha^{-1} \left(Y_{\beta,\alpha,N} \Pi_{\tilde{Z}_\beta} - \hat{\Phi}_\alpha U_{\beta,\alpha,N} \right), \quad (5.25)$$

kde

$$\Pi_{\tilde{Z}_\beta} \triangleq \tilde{Z}_\beta^T (\tilde{Z}_\beta \tilde{Z}_\beta^T)^{-1} \tilde{Z}_\beta; \quad \tilde{Z}_\beta \triangleq \begin{bmatrix} U_{\beta,\alpha,N} \\ U_{0,\beta,N} \\ Y_{0,\beta,N} \end{bmatrix} \quad (5.26)$$

K výpočtu Kalmanova zesílení pro stavový model v tzv. inovačním tvaru je třeba určit navíc $X_{\beta+1,N}$. Potom se dá odhadnout šum procesu a šum měření podle

$$\begin{bmatrix} \hat{W}_{\beta,1,N-1} \\ \hat{V}_{\beta,1,N-1} \end{bmatrix} = \begin{bmatrix} \hat{X}_{\beta+1,N} \\ Y_{\beta,1,N-1} \end{bmatrix} - \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} \hat{X}_{\beta,N-1} \\ U_{\beta,1,N-1} \end{bmatrix}, \quad (5.27)$$

ze kterých se stanoví kovariance šumu

$$\begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \hat{W}_{\beta,1,N} \\ \hat{V}_{\beta,1,N} \end{bmatrix} \begin{bmatrix} \hat{W}_{\beta,1,N} \\ \hat{V}_{\beta,1,N} \end{bmatrix}^T. \quad (5.28)$$

Výsledný odhad Kalmanova zesílení je řešením Riccatiho rovnice

$$\hat{P} = \hat{A} \hat{P} \hat{A}^T + \hat{Q} - (\hat{A} \hat{P} \hat{C}^T + \hat{S})(\hat{C} \hat{P} \hat{C}^T + \hat{R})^{-1} (\hat{A} \hat{P} \hat{C}^T + \hat{S})^T, \quad (5.29)$$

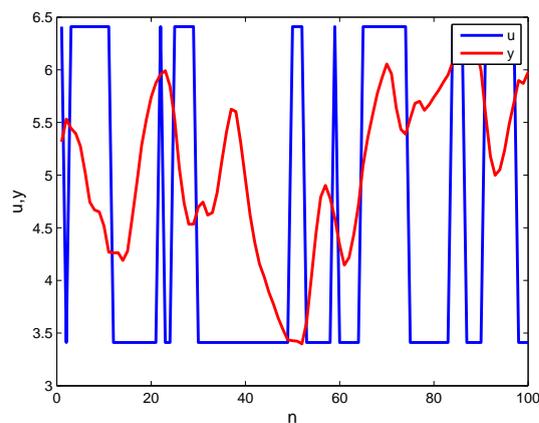
$$\hat{K} = (\hat{A} \hat{P} \hat{C}^T + \hat{S})(\hat{C} \hat{P} \hat{C}^T + \hat{R})^{-1}. \quad (5.30)$$

Problematika subspace identifikace je velmi obsáhlým tématem. Podrobnější popis principů a dílčích kroků je nad rámec této práce. Podrobnější informace nalezne čtenář ve výše uvedené literatuře.

5.5.2 Použití subspace identifikace

Uvažujme testovací data `dryer.mat`, která jsou součástí testovacích příkladů pro SITB v `Matlabu`. Po spuštění GUI SITB příkazem `ident`, jsou data `dryer.mat` dostupná z menu *Import data - Example*. Jedná se o sadu dat s jedním vstupem a jedním výstupem, naměřená na obyčejném vysoušeči vlasů, přičemž vstupem je elektrické napětí na topnou spirálu a výstupem teplota vyfukovaného vzduchu - viz. Obr. 5.3⁴. Na těchto datech budeme porovnávat úspěšnost a výpočetní rychlost různých druhů implementací subspace identifikace. Testování algoritmů bylo prováděno na prvních 700 vzorcích dat (zbylých 300 vzorků bylo použito pro následnou validaci navrženého modelu), přičemž se odhadovaly systémové matice A , B , C , D a K řádu $N = 3$ a horizont predikce byl zvolen jednotně $\alpha = 10$. Identifikačním kritériem byla chyba predikce na validačních datech.

První variantou implementace je ryze analytický výpočet korespondující s postupem popsáným v Kap. 5.5.1. Další variantou je `Matlab` funkce `n4sid`, ve které je implementován algoritmus N4SID, popsáný v Kap. 10.6 v [1]. Poslední variantou implementace je tzv. *robustní kombinovaný algoritmus* uvedený v Kap. 4.5 v [34], jehož jádrem je jedna QR faktorizace, přičemž je počítán pouze R faktor, což přináší značné urychlení. Souhrn výsledků úspěšností jednotlivých algoritmů shrnuje Tab. 5.1⁵. Ukazuje se, že použití ryze analytické implementace je nevhodné z důvodu časové neúspěšnosti a numerické nepřesnosti. Ostatní metody již vykazují srovnatelné úspěšnosti ve smyslu výpočetní rychlosti i fit faktoru. Zajímavým ukazatelem je výpočetní rychlost implementace kombinovaného algoritmu ve `Scilabu`, který je i součástí vyvíjeného identifikačního toolboxu. Tato implementace vykazuje vyšší výpočetní rychlost než `n4sid` komerčního SITB.



OBRÁZEK 5.3: Testovací data `dryer.mat`

⁴Pozn. Data mají délku $n = 1000$, ale pro přehlednost je zobrazeno pouze prvních 100 vzorků.

⁵Pozn. Matice A je v Tab. 5.1 pro úsporu místa vypsána do sloupce pomocí koeficientů, kde $\hat{A} \sim \text{vec}(\hat{A})$.

Algoritmus	Matlab			Scilab
	Analyticky	n4sid	Kombinovaný	Kombinovaný
\hat{A}	0.8814	0.9548	0.8785	0.8895
	0.0569	-0.3141	0.0522	0.1788
	0.0045	0.1086	0.0158	0.0362
	-0.4064	0.1452	-0.4009	-0.1759
	0.6019	0.6664	0.6132	0.6163
	-0.1461	0.6798	-0.1623	-0.3287
	0.1759	0.0448	0.1871	0.0353
	0.7321	-0.2661	0.8373	0.3494
	0.3760	0.1362	0.4737	0.3625
\hat{B}	$\begin{bmatrix} -0.1916 \\ 0.1360 \\ 0.1279 \end{bmatrix}$	$\begin{bmatrix} -0.0001 \\ 0.0100 \\ -0.0272 \end{bmatrix}$	$\begin{bmatrix} -0.2067 \\ 0.1980 \\ 0.0520 \end{bmatrix}$	$\begin{bmatrix} -0.1340 \\ 0.2863 \\ 0.1909 \end{bmatrix}$
\hat{C}	$\begin{bmatrix} -0.4763 \\ -0.6729 \\ 0.4624 \end{bmatrix}^T$	$\begin{bmatrix} 32.9209 \\ 0.0798 \\ -0.1709 \end{bmatrix}^T$	$\begin{bmatrix} -0.4813 \\ -0.6627 \\ 0.5100 \end{bmatrix}^T$	$\begin{bmatrix} -0.7104 \\ -0.4565 \\ 0.1590 \end{bmatrix}^T$
\hat{D}	$[0]$	$[0]$	$[0]$	$[0]$
\hat{K}	$\begin{bmatrix} 0.0059 \\ -0.0066 \\ -0.0204 \end{bmatrix}$	$\begin{bmatrix} 0.0273 \\ -0.0008 \\ -0.0241 \end{bmatrix}$	$\begin{bmatrix} 0.0016 \\ -0.0022 \\ -0.0007 \end{bmatrix}$	$\begin{bmatrix} -0.9101 \\ 0.1918 \\ 0.5247 \end{bmatrix}$
Výpočetní rychlost	7.699 s	0.276 s	0.278 s	0.194 s
Fit faktor	75.65 %	85.75 %	85.42 %	85.88 %

TABULKA 5.1: Porovnání různých implementací subspace identifikace (pozn. matice A je pro úsporu místa vypsána do sloupce; $\hat{A} \sim \text{vec}(\hat{A})$.)

5.6 Inicializace PEM pomocí subspace

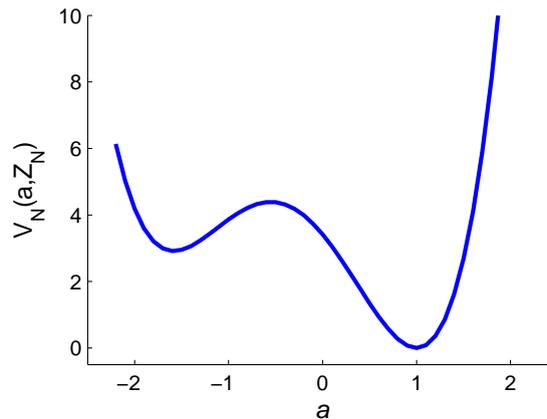
Identifikace neznámých parametrů založená na PEM přístupu v některých případech koresponduje s řešením problému nekonvexní optimalizace, což nese za následek, že není vždy garantováno nalezení globálního optima ztrátové funkce (5.8). Jako příklad takového problému uvažujme jednoduchý model [26] ve tvaru

$$y(t) = au(t) + a^2u(t-1) \quad (5.31)$$

pro $a = 1$. Budícím signálem je $u(t) = \sin(t) + \sin(t/5)$. Vykreslením závislosti hodnoty ztrátové funkce pro různé hodnoty parametrů a lze v Obr. 5.4 pozorovat, že ztrátová funkce má dvě lokální minima. Při identifikaci takových systémů je na začátku třeba zajistit vhodný počáteční odhad parametrů garantující nalezení globálního minima. Princip inicializačních metod spočívá v nalezení nějakého počátečního odhadu, ležícího v blízkosti globálního minima.

Metody PEM se běžně používají v praxi. Úloha nalezení počátečního odhadu pro PEM však již tolik rozšířená není. Používanou variantou inicializace je víceukrokový predikční algoritmus založený na IV metodě (Kap. 3.3), který je popsán v [26] a je implementován v SITB. Vedle

toho se budeme zabývat inicializací pomocí subspace metody, která může v některých případech přinést výrazné zlepšení, neboť algoritmy subspace identifikace jsou efektivní a numericky stabilní. Odhad parametrů lze navíc optimalizovat pro víceukrokové predikce pouze volbou velikosti bloků Hankelových matic vstupů a výstupů.



OBRÁZEK 5.4: Závislost velikosti ztrátové funkce (5.8) pro model (5.31) na parametru a

Zmínka o vhodnosti použití subspace metod jako inicializačních metod pro identifikace PEM se objevuje v [6], další podrobnější studii této problematiky je věnována kapitola z [26], nesoucí název *Utilizing Structure Information in Subspace Identification*. Uvedený název autor použil z toho důvodu, že jím prezentovaná metoda umožňuje, mimo jiné, i určitým způsobem specifikovat strukturu výsledných odhadů matic stavového popisu.

Metodika v [26] umožňuje odhadovat koeficienty ARMAX modelu (4.23), případně OE modelu (4.27). Dále uvažujme pro názornost konkrétní ARMAX model⁶

$$y(t) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1} + a_2 q^{-2}} u(t) + \frac{1 + c_1 q^{-1}}{1 + a_1 q^{-1} + a_2 q^{-2}} e(t) \quad (5.32)$$

s korespondujícím tvarem v kanonické formě pozorovatelnosti (4.36)

$$\begin{aligned} x(t+1) &= \begin{bmatrix} -a_1 & 1 \\ -a_2 & 0 \end{bmatrix} x(t) + \begin{bmatrix} b_1 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} c_1 - a_1 \\ -a_2 \end{bmatrix} e(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) + e(t), \end{aligned} \quad (5.33)$$

odkud $n_a = 2$, $n_b = 1$ a $n_c = 1$. Nejprve provedeme odhad stavového modelu řádu rovnému stupni polynomu A , tj. $n = n_a$, subspace metodou popsanou v Kap. 5.5.1, přičemž odhadujeme pouze matice \hat{A} a \hat{C} . Odhad matic následně převedeme do kanonické formy pozorovatelnosti,

⁶OE je pouze zobrazeným případem ARMAX modelu.

což lze provést např. užitím Matlabu:

$$\begin{aligned}\hat{a} &= \text{poly}(\hat{A}), \quad \hat{A}_{OCF} = [-\hat{a}(2:n_a+1)^T, \text{eye}(n_a, n_a-1)], \\ \hat{C}_{OCF} &= \text{eye}(1, n_a),\end{aligned}\tag{5.34}$$

kde \hat{a} je vektor kořenů charakteristické rovnice matice \hat{A} . Tím získáme počáteční a výchozí odhad pro metodu PEM. V dalším kroku odhadujeme matice \hat{B} a \hat{D} pomocí PEM přístupu. Aby bylo zajištěno $n_b = 1$ a $D = 0$, pro odhad využijeme LS odhad z Kap. 5.5.1 s potřebným omezením

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B \\ D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.\tag{5.35}$$

K odhadu Kalmanova zesílení \hat{K} je třeba zrekonstruovat stavovou posloupnost a nalézt odhady \hat{W} a \hat{V} podle (5.27). Samotný odhad \hat{K} je řešením LS problému $\hat{K}\hat{V} = \hat{W}$ s omezením na požadovanou strukturu K

$$\begin{bmatrix} 0 & 1 \end{bmatrix} K = -\hat{a}_2\tag{5.36}$$

Zobecnění výše uvedeného postupu pro libovolné hodnoty n_a , n_b a n_c ⁷ autor v [26] realizoval zavedením tzv. virtuálních vstupů. Pro ilustraci principu zavádění virtuálních vstupů uvažujme ARMAX model bez tvarovacího filtru šumu, tedy OE model

$$y(t) = \frac{b_1 q^{-n_k} + \dots + b_{n_b} q^{-n_k - n_b + 1}}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}} u(t) + e(t).\tag{5.37}$$

Počet virtuálních vstupů je $n_u = \text{ceil}(n_b/(n_a + 1))$ a celkový počet odhadovaných parametrů v B a D je $n_p = n_u n_a + n_u$, kde n_l parametrů je odhadováno s omezením. OE prediktor má potom tvar

$$y(t|t-1, \theta) \triangleq \frac{b_1 q^{-n_k} + \dots + b_{n_b} q^{-n_k - n_b + 1}}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}} u(t) = \sum_{k=1}^{n_u} G_k(q) u_k(t)\tag{5.38}$$

a operuje s $u_k(t) \triangleq u(t - n_k - (k-1)(n_a + 1))$ virtuálními vstupy. Zavedením $m_k \triangleq (k-1)(n_a + 1)$ má přenosová funkce $G(k)$ v (5.38) tvar

$$G_k(q) = b_{m_k+1} \frac{P_k(q)}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}}, \quad k = 1, \dots, n_u,\tag{5.39}$$

⁷Zřejmým omezujícím faktem je, že $n_c \leq n_a$.

kde pro $k = 1, \dots, n_u - 1$:

$$P_k(q) = \sum_{i=1}^{n_a} (b_{m_k+i+1} - a_i b_{m_k+1}) q^{-i} \quad (5.40)$$

a pro $k = n_u$:

$$P_{n_u}(q) = \sum_{i=1}^{n_a-n_l} (b_{m_{n_u}+i+1} - a_i b_{m_{n_u}+1}) q^{-i} + \sum_{i=n_a-n_l+1}^{n_a} (-a_i b_{m_{n_u}+1}) q^{-i}. \quad (5.41)$$

Ve fázi, kdy odhadneme matice \hat{A} a \hat{C} a následně převedeme do kanonické formy pozorovatelnosti podle (5.34):

$$\hat{A} = \begin{pmatrix} -\hat{a} & I_{n_a \times (n_a-1)} \end{pmatrix}, \quad \hat{C} = I_{1 \times n_a}, \quad (5.42)$$

formulujeme obecně omezení pro odhady matic \hat{B} a \hat{D} následovně:

$$\begin{bmatrix} [0_{n_l \times (n_u n_a - n_l)} & I_{n_l \times n_l}] & [0_{n_l \times (n_u-1)} & -\hat{a}_l] \end{bmatrix} \begin{bmatrix} \text{vec} B(\theta) \\ \text{vec} D(\theta) \end{bmatrix} = 0_{n_l \times 1} \quad (5.43)$$

$$\hat{a}_l \triangleq [\hat{a}_{n_a-n_l+1} \quad \dots \quad \hat{a}_{n_a}]^T.$$

Odhad Kalmanova zesílení \hat{K} se provede výše uvedeným postupem s obecnou formulací omezení na požadovanou strukturu K :

$$\begin{aligned} ((0_{(n_a-n_c) \times n_c} & I_{(n_a-n_c) \times (n_a-n_c)}) K = -\hat{a}_{n_c+1:n_a} \\ -\hat{a}_{n_c+1:n_a} & = \begin{pmatrix} \hat{a}_{n_c+1} & \dots & \hat{a}_{n_a} \end{pmatrix}^T. \end{aligned} \quad (5.44)$$

Algoritmus 5: Odhad parametrů ARMAX modelu s použitím subspace metody jako inicializační metody [26]

1. Zavedení virtuálních výstupů podle (5.38) - (5.41).
2. Nalezení odhadů \hat{A} a \hat{C} řádu n_a .
3. Převod \hat{A} a \hat{C} do kanonické formy pozorovatelnosti užitím (5.42).
4. Odhad \hat{B} a \hat{D} s omezením (5.43).
5. Rekonstrukce stavové posloupnosti $X_{\beta, N}$ a nalezení \hat{W} a \hat{V} podle (5.27).
6. Odhad \hat{K} vyřešením LS problému $\hat{K}\hat{V} = \hat{W}$ s omezením (5.44).
7. Výsledný model ve formě stavového popisu lze převést na korespondující tvar (5.32).

Pozn.: V případě odhadování parametrů OE modelu se aplikuje výše uvedený postup s tím, že se odhadují pouze potřebné matice \hat{A} , \hat{B} , \hat{C} a \hat{D} .

	Skutečný	PEM	IV + PEM	4SID + PEM
$A(q)$	-1.9	-1.939	-1.785	-1.979
	1.3	1.412	1.097	1.432
	-0.35	-0.4241	-0.2585	-0.3571
$B(q)$	-0.04	-0.04	-0.0388	-0.0414
	0.1	0.0991	0.1029	0.1012
$C(q)$	-0.5	-0.6364	-1	- 0.6362
Výpočetní rychlost		0.715 s	1.824 s	1.668 s
Fit faktor		82.73 %	84.94 %	85.19 %

TABULKA 5.2: PEM identifikace ARMAX modelu, PEM identifikace ARMAX modelu inicializovaná IV a 4SID metodami

5.6.1 PEM identifikace ARMAX modelu inicializovaná pomocí subspace

Výše uvedenou metodu inicializace PEM metody pomocí 4SID metody budeme aplikovat na identifikaci konkrétního ARMAX modelu, přičemž provedeme srovnání s obyčejnou PEM identifikací a s PEM identifikací inicializovanou metodou vícekrokové predikce založené na metodě IV, jež je realizována v SITB.

Uvažujme konkrétní ARMAX model řádu $(3, 2, 1)$ s polynomy $A(q)$, $B(q)$ a $C(q)$ (Tab. 5.2), na jehož vstup byl přiveden bílý šum $\mathcal{N}(0, 1)$ délky 1000 a byl měřen signál na jeho výstupu, který byl následně zkreslen aditivním bílým šumem $\mathcal{N}(0, 0.1)$. Simulační data byla rozdělena na první část o délce 700, která byla použita jako data pro následnou identifikaci, zbylá část pro validaci navrženého modelu, přičemž identifikačním kritériem byla chyba simulace na validačních datech. Z identifikačních dat byl proveden odhad parametrů ARMAX modelu pro řád $(3, 2, 1)$ pomocí trojice různých metod z Tab. 5.2.

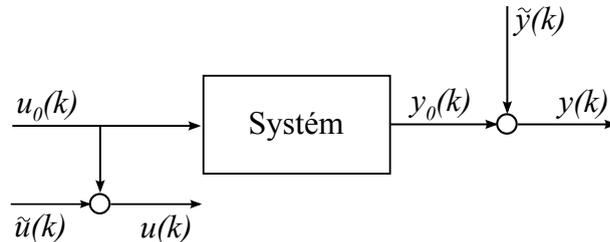
Provedené simulace ukázaly, že inicializace PEM pomocí 4SID metody je prakticky použitelná jako další varianta inicializace vedle vícekrokové IV predikce. Navíc se ukázalo, že výpočet pomocí 4SID vykazuje vyšší výpočetní rychlost. Lze konstatovat, že 4SID přístup má do budoucna vysoký potenciál, vzhledem k možnostem numericky stabilní a robustní implementace.

5.7 Errors-in-variables (EIV) identifikace

Klasický identifikační problém řešitelný metodou LS (Kap. 3.1) vždy předpokládá, že pouze výstupní data jsou zatížena šumem. Oproti tomu identifikace tzv. typu *errors-in-variables* (EIV) uvažuje zatížení jak vstupního, tak výstupního signálu aditivním šumem (Obr. 5.5):

$$\begin{aligned}
 u(k) &= u_0(k) + \tilde{u}(k) \\
 y(k) &= y_0(k) + \tilde{y}(k).
 \end{aligned}
 \tag{5.45}$$

Nástrojem pro řešení EIV problémů je metoda úplných nejmenších čtverců (Kap. 3.2). S problematikou identifikace EIV systémů se často setkáváme v praktických aplikacích (viz. např. [17]). Existuje řada různých metod, jak problém identifikace EIV systémů řešit, jejichž přehled nabídne např. [36]. V rámci této práce byly pro EIV identifikaci implementovány algoritmy TLS (Kap. 3.2.1) a RTLS (Kap. 3.2.2). Použití těchto algoritmů na příkladu ARX modelu demonstruje Kap. 5.7.1.



OBRÁZEK 5.5: EIV identifikační problém

5.7.1 EIV identifikace ARX modelu

Uvažujme konkrétní ARX model (4.14) řádu (4, 3) s polynomy

$$\begin{aligned} A(q) &= 1 + 0.1q^{-1} - 0.3q^{-2} + 0.4q^{-3} - 0.1q^{-4} \\ B(q) &= 1 + 5q^{-1} - 0.5q^{-2}, \end{aligned} \quad (5.46)$$

na jehož vstup byl přiveden náhodný signál délky $N = 1000$ a byl změřen výstupní signál y . Naměřená data byla rozdělena na poloviny, přičemž první polovina byla použita pro identifikaci, a druhá polovina pro následnou validaci odhadnutého modelu. Vstupní a výstupní signály u a y byly postupně zatěžovány bílým šumem s nulovou střední hodnotou podle Tab. 5.3 a byla na nich prováděna identifikace neznámých parametrů ARX modelu řádu (4, 3) pomocí metody LS (Kap. 3.1), TLS (Kap. 3.2) a RTLS (Kap. 3.2.2). Kvalita odhadu byla měřena fit faktorem (2.9). Výsledky úspěšnosti identifikace pomocí výše uvedených metod, měřené fit faktorem jsou zaneseny v Tab. 5.3, odkud je patrné, že při klasickém problému, kdy je šumem zatížen pouze výstupní signál, přístup pomocí EIV identifikace nepřináší žádné zlepšení, naopak v některých případech vykazuje horší odhad ve smyslu fit faktoru. Naopak s postupně se zvyšujícím šumem na měřeném vstupním signálu je výhoda použití EIV identifikace zřejmá.

5.8 Odhad fyzikálních parametrů systému s využitím Black-Box (BB) identifikace

V situacích, kdy modelujeme vstupně-výstupní dynamiku systému pomocí BB modelování, využíváme pouze naměřených vstupně-výstupních datových sekvencí, bez jakýchkoli apriorních

u	y	LS	TLS	RTLS
-	-	95.0086	95.0086	95.0086
-	$\mathcal{N}(0, 0.01)$	91.9568	91.9522	91.9524
-	$\mathcal{N}(0, 0.1)$	89.6696	87.3748	89.2775
-	$\mathcal{N}(0, 1)$	76.0691	63.7773	69.2217
$\mathcal{N}(0, 0.01)$	-	89.0522	89.0442	89.0544
$\mathcal{N}(0, 0.1)$	-	70.6190	70.4518	70.5752
$\mathcal{N}(0, 0.01)$	$\mathcal{N}(0, 0.01)$	85.8615	88.9362	88.9669
$\mathcal{N}(0, 0.1)$	$\mathcal{N}(0, 0.01)$	69.4906	73.2538	74.2723
$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 0.01)$	38.8612	41.3185	40.9305
$\mathcal{N}(0, 0.01)$	$\mathcal{N}(0, 0.1)$	64.2572	64.0677	67.5203
$\mathcal{N}(0, 0.1)$	$\mathcal{N}(0, 0.1)$	68.3320	68.4539	69.1585
$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 0.1)$	54.4960	54.8968	56.1031
$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	48.9562	47.1207	49.0410

TABULKA 5.3: EIV: Porovnání úspěšnosti identifikace ARX modelu s použitím LS, TLS a RTLS při různých úrovních vstupního a výstupního šumu, měřenou fit faktorem

znalostí o dynamice systému. Výsledkem takové identifikace je model, jehož matice stavového popisu obsahují číselné hodnoty, které nemají praktický fyzikální význam. Při identifikaci však můžeme někdy požadovat specifickou strukturu stavových matic, přičemž hodnota některých prvků v maticích je dokonce přesně dána, neboť reprezentuje známou fyzikální konstantu systému apod. Jednou z variant, jak realizovat takovou identifikaci je vhodně využít právě BB modelování ve spojení s nalezením vhodné transformace, která stavový popis systému přivede do požadovaného tvaru [37].

Pro jednotnost budeme stále, jako v celé práci, uvažovat diskrétní model dynamického systému. Metodiku popsanou níže lze však stejným způsobem aplikovat i na spojitý systém - a zejména zde nalézá své opodstatnění (viz. Kap. 5.8.2). Předpokládejme model systému

$$\begin{aligned} x(k+1) &= A_0(\theta)x(k) + B_0(\theta)u(k) \\ y(k) &= C_0(\theta)x(k) + D_0(\theta)u(k), \end{aligned} \quad (5.47)$$

kde prvky vektoru θ , obsažené v maticích A_0 , B_0 , C_0 nebo D_0 , jsou neznámé a ostatní prvky v maticích známe. Dále uvažujme model

$$\begin{aligned} x'(k+1) &= Ax'(k) + Bu(k) \\ y(k) &= Cx'(k) + Du(k), \end{aligned} \quad (5.48)$$

který byl získán použitím libovolné BB identifikace na daném systému. Stav x' tedy nyní neodpovídá reálnému fyzikálnímu stavu a tvary matic A , B , C , D nenesou praktický význam. V případě, že je daný systém říditelný a zároveň pozorovatelný, existuje transformační matice T ,

zajišťující

$$T.A = A_0(\theta).T; \quad T.B = B_0(\theta); \quad C = C_0(\theta).T; \quad D = D_0(\theta). \quad (5.49)$$

Tento známý vztah je prakticky využitelný pro nalezení neznámých parametrů θ v A_0 , B_0 , C_0 , D_0 , které mají skutečný fyzikální význam. Existuje více možností, jak nalézt transformační matici T a řešení (5.49) navíc není unikátní. Různé možnosti přístupu k nalezení T popisuje a srovnává [37], zároveň je zde také prezentován nový efektivní přístup, založený na minimalizaci jednoduchého kritéria.

Přirozenou cestou při hledání transformační matice T je minimalizovat chyby diferencí jednotlivých rovnic (5.49), a to užitím Frobeniovy normy⁸:

$$e = \arg \min_{T, \theta} (\|TA - A_0(\theta)T\|_F^2 + \|TB - B_0(\theta)\|_F^2 + \|C - C_0(\theta)T\|_F^2). \quad (5.50)$$

Minimalizace uvedeného kritéria je potom řešitelná metodou LS (viz. Kap. 3.1). V praxi je vhodné ji řešit iterativně a v každé iteraci střídatě minimalizovat přes T a přes θ , což zajistí postupné přizpůsobování transformační matice T odhadovaným parametrům θ se zvyšujícím se počtem iterací. S nekonečným počtem iterací se bude chyba odhadu (5.50) blížit nule. Uvedený postup hledání vhodné transformační matice T shrnuje Algoritmus 6.

Algoritmus 6: Řešení minimalizace normy (5.50) [37]

1. Urči počáteční hodnoty neznámých parametrů θ_0 v A_0 , B_0 , C_0 .
2. Pro počáteční hodnoty θ_0 minimalizuj kritérium (5.50) vyřešením pro T využitím LS metody:

$$\arg \min_T (\|TA - A_0(\theta_0)T\|_F^2 + \|TB - B_0(\theta_0)\|_F^2 + \|C - C_0(\theta_0)T\|_F^2). \quad (5.51)$$

3. Pro T , získané v kroku 2, minimalizuj kritérium (5.50) vyřešením pro θ využitím LS metody:

$$\arg \min_{\theta} (\|\hat{T}A - A_0(\theta)\hat{T}\|_F^2 + \|\hat{T}B - B_0(\theta)\|_F^2 + \|C - C_0(\theta)\hat{T}\|_F^2). \quad (5.52)$$

4. Urči hodnotu chyby odhadu T :

$$e = \|\hat{T}A - A_0(\hat{\theta})\hat{T}\|_F^2 + \|\hat{T}B - B_0(\hat{\theta})\|_F^2 + \|C - C_0(\hat{\theta})\hat{T}\|_F^2. \quad (5.53)$$

Pokud je e menší, než předem definovaná maximální chyba odhadu ε_0 , akceptuj dvojici θ , T jako jedno z vyhovujících řešení; V opačném případě diskutuj vhodnost volby počátečních parametrů θ_0 a pokračuj krokem 1, jinak $\theta_0 = \hat{\theta}$ a pokračuj krokem 2.

⁸ $\|\cdot\|_F$ je Frobeniova norma: $\|X\|_F = \sqrt{\text{tr}(X.X^T)}$

5.8.1 Zahnutí strukturální informace do black-box identifikace

Uvažujme systém čtvrtého řádu se stavovým popisem

$$\begin{aligned} x(k+1) &= \begin{bmatrix} -0.2 & 1 & 0 & 0 \\ 0 & -0.4 & -1.5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 5 & -1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0.2 \\ 0.1 \\ -0.01 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x(k), \end{aligned} \quad (5.54)$$

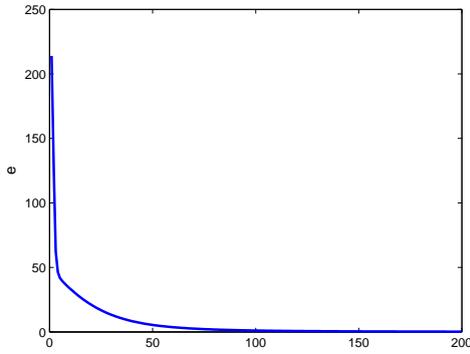
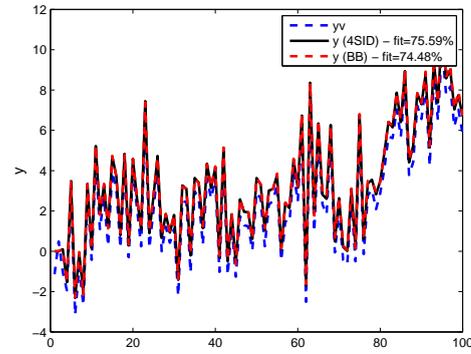
na kterém byl pomocí náhodného vstupního signálu $u(k)$ délky $N = 200$ vygenerován výstupní signál $y(k)$. Výstupní signál byl následně zkreslen aditivním bílým šumem $\mathcal{N}(0, 0.1)$. Výsledná data byla rozdělena na poloviny, přičemž první část slouží jako data identifikační a druhá část jako data validační. Nyní požadujeme identifikovat daný systém, přičemž máme částečnou představu o výsledné podobě stavového popisu, který by měl být výsledkem identifikace

$$\begin{aligned} A &= \begin{bmatrix} \theta(1) & 1 & 0 & 0 \\ 0 & \theta(2) & -1.5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \theta(3) & -1 \end{bmatrix} & B &= \begin{bmatrix} 0 \\ \theta(4) \\ 0.1 \\ \theta(5) \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} & D &= [0]. \end{aligned} \quad (5.55)$$

Hodnoty θ tedy budeme odhadovat. Z naměřených identifikačních dat provedeme identifikaci pomocí metody 4SID (Kap. 5.5.1) a dostáváme následující matice stavového popisu

$$\begin{aligned} A &= \begin{bmatrix} 0.9981 & -0.0989 & 0.0516 & 0.0005 \\ -0.0522 & -0.7494 & -0.3580 & 0.0055 \\ 0.0035 & 0.5439 & 0.1536 & 0.0426 \\ 0.0206 & 0.0787 & -0.0663 & -0.0680 \end{bmatrix} & B &= \begin{bmatrix} -0.0012 \\ -0.0105 \\ -0.0173 \\ -0.0004 \end{bmatrix} \\ C &= \begin{bmatrix} 41.5831 & 11.3775 & -9.7643 & -0.0610 \end{bmatrix} & D &= [0]. \end{aligned} \quad (5.56)$$

Pomocí Algoritmu 6 nalezneme transformační matici T (5.49), která zajistí převod stavového popisu (5.56) do podoby (5.55). Počáteční hodnoty parametrů nevolíme nikterak sofistikovaně, neboť tím výsledný výpočet téměř neovlivníme, tedy $\theta = [1 \ 1 \ 1 \ 1 \ 1]^T$. Provedením výpočtu transformační matice T pomocí Algoritmu 6 aplikovaném ve 200 iteracích bylo dosaženo určení T s chybou $e = 0.0019$ (viz. Obr. 5.6) a výsledný odhad neznámých parametrů je $\hat{\theta}(200) = [-0.1060 \ -0.4763 \ 1.0181 \ 0.1985 \ 0.0492]^T$. Porovnání úspěšnosti identifikace ve smyslu fit faktoru zachycuje Obr. 5.6.

(a) Chyba odhadu neznámých parametrů θ v závislosti na počtu iterací

(b) Skutečná odezva systému v porovnání s odezvou modelu systému identifikovaným pomocí 4SID a odezvou modelu systému identifikovaným pomocí 4SID a následně restrukturalizovaným do požadované podoby pomocí Algoritmu 6

OBRÁZEK 5.6: Zahnutí strukturální informace do black-box identifikace

5.8.2 Použití black-box identifikace pro odhad fyzikálních parametrů systému

Pro názornost uvažujme známý, a z důvodu své jednoduchosti hojně používaný při výuce modelování dynamických systémů a jejich řízení, model systému kuličky na tyči **CE106**, který je umístěn např. v laboratoři K26 katedry řídicí techniky FEL ČVUT

(http://support.dce.felk.cvut.cz/lab26/index.php?page=BB_TQ). Výukový model **CE106** je většinou využíván pro návrh základního řízení polohy kuličky (výstup y) na tyči pomocí rotačního servopohonu (vstupem do systému je napětí na servopohon u), který prostřednictvím převodního mechanismu působí na tyč.

Lze odvodit, že systém **CE106** popisují ve spojitém případě rovnice ([38])

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \\ \dot{\alpha} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mg}{R^2+m} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \\ \alpha \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u \quad (5.57)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \\ \alpha \\ \dot{\alpha} \end{bmatrix},$$

kde $g = 9.8\text{m}\cdot\text{s}^{-2}$ je gravitační zrychlení, $J = 9.99 \times 10^{-6}\text{kg}\cdot\text{m}^2$ moment setrvačnosti kuličky, $R = 0.015\text{m}$ poloměr kuličky a m je neznámá hmotnost kuličky. Uvažujme jednoduchý problém, kdy bude naším úkolem stanovit jediný neznámý parametr, a to hmotnost kuličky, za

předpokladu, že známe konstanty g , J a R a k dispozici máme pouze naměřené posloupnosti vstupně-výstupních dat na systému **CE106**.

Pro názornou ukázkou byl zvolen jednoduchý případ, kdy odhadujeme pouze jediný parametr, a proto je daný případ spíše ilustrativní. Postup lze však aplikovat i na sofistikovanější problémy odhadu více neznámých fyzikálních parametrů (viz. např. [37]).

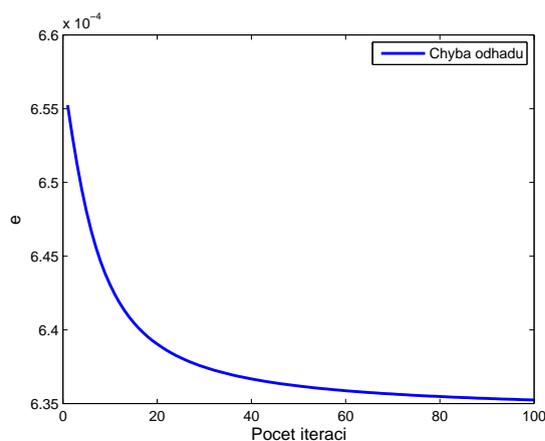
Z naměřených vstupně-výstupních dat byl pomocí BB identifikace (konkrétně SITB funkce `n4sid` pro spojitou subspace identifikaci) stanoven stavový model 4. řádu, kde

$$A = \begin{bmatrix} 0.0008 & -0.0001 & 7.8750 \times 10^{-7} & 2.6883 \times 10^{-7} \\ 0.0019 & 0.0006 & 0.0029 & 0.0009 \\ -0.0017 & -0.0036 & -0.6686 & -0.1450 \\ 0.0001 & 0.0002 & 0.0498 & -0.2035 \end{bmatrix}, \quad B = \begin{bmatrix} 1.4502 \times 10^{-7} \\ 0.0001 \\ -0.0396 \\ 0.0029 \end{bmatrix}$$

$$C = \begin{bmatrix} 8.5414 \times 10^6 & 54.7593 & -4.1077 & 0.9119 \end{bmatrix} \quad (5.58)$$

Zavedením vektoru odhadovaných parametrů $\theta = \frac{-mg}{\frac{J}{R^2} + m}$ a aplikováním Algoritmu 6 ve 100 iteracích (100 iterací stačilo k zajištění dostatečně malé chyby odhadu e) pro počáteční odhad $\theta_0 = 1$ ⁹ bylo vypočteno $\theta(100) = 7.5923$ s chybou $e = 0.000635$. Nyní je snadným úkolem stanovit hmotnost kuličky vyjádřením

$$m = \frac{\theta J}{R^2(\theta + g)} = 0.0194 \text{ kg.} \quad (5.59)$$



OBRÁZEK 5.7: Chyba odhadu neznámých parametrů θ_0 v závislosti na počtu iterací

⁹Volba velikosti θ_0 zásadně neovlivňuje výsledný výpočet, neboť střídavě minimalizujeme přes T a přes θ a chyba odhadu e se pro nekonečný počet iterací blíží nule.

Kapitola 6

Dokumentace k identifikačnímu toolboxu a navrženým algoritmům

6.1 Struktura toolboxu

Návrh struktury toolboxu Tab. 6.1 byl proveden dle http://wiki.scilab.org/howto/Create_a_toolbox. Jedná se o standardní adresářovou strukturu pro Scilab toolboxy.

6.2 Obslužné skripty

Obslužné skripty slouží k zavedení nové instalace toolboxu. Při instalaci je nejprve spuštěn skript `buildmacros`, který zkompileje všechny funkce. Funkce jsou poté načteny do prostředí

adresář	adresář (soubor)	popis
název toolboxu	demos	příklady, testovací data
	etc	inicializační soubory toolboxu
	help	help (html dokumentace)
	html	html soubory
	macros	jednotlivé funkce toolboxu
	sci_gateway	gateway soubor (c, cpp, ...)
	tests	testy funkcí
	builder.sce	builder funkcí
	loader.sce	loader funkcí
	license.txt	licence
	readme.txt	informace

TABULKA 6.1: Struktura toolboxu

Scilabu skriptem `loadmacros`. Skript `cleanmacros` se spouští při odinstalaci toolboxu a zajišťuje smazání zkompileovaných funkcí. Skript `allexec` je určen pro ladění při vývoji a slouží k hromadné kompilaci funkcí v daném adresáři.

- `allexec.sce` - hromadný kompilátor všech skriptů z daného adresáře.
- `buildmacros.sce` - builder všech funkcí toolboxu.
- `loadmacros.sce` - loader všech funkcí toolboxu.
- `cleanmacros.sce` - cleaner všech funkcí toolboxu.

6.3 Utility

- `idmodel.sci` - unifikovaná struktura modelu, používaná všemi funkcemi z toolboxu.
- `mydisp.sci` - vylepšení funkce `disp`, které slouží ke zobrazení textu. Vstupními parametry funkce jsou text ke zobrazení a parametr *flag*, na základě kterého se určí, zda bude nebo nebude text zobrazen. Funkce se používá pro ladění nebo pro uživatelské povolování či zablokování generování příslušného výstupu.
- `hold.sci` - ekvivalent k Matlab funkci `hold on/off`.
- `spy.sci` - ekvivalent k Matlab funkce `spy`, který slouží k vizualizaci obsazenosti (nulových/nenulových) prvků dané matice. Vstupem funkce je libovolná matice a výstupem graf obsazenosti jejich prvků.

6.4 Matematické nástroje

V této kapitole budou popsány funkce, realizující pomocné matematické instrumenty, které se dále používají v různých identifikačních algoritmech.

- `nng.sci` - funkce realizující algoritmus redukce řádu modelu pomocí metody Nonnegative Garrote (NNG) (Kap. 3.4). Vstupními parametry funkce jsou počáteční odhad parametrů, matice regresorů pro příslušný model, matice omezení (3.29), případně (3.30) a vektor příslušných výstupních dat. Výstupem funkce je struktura, obsahující sadu redukovaných modelů. Funkce je prioritně určena jako pomocná funkce pro funkci `selstruc`, která vybírá optimální řád ARX/ARMAX, ale lze ji použít i samostatně. Ve funkci je realizován Algoritmus 2.

- `tls.sci` - funkce pro výpočet optimálního řešení úplného problému nejmenších čtverců (TLS) podle Kap. 3.2.1. Vstupem je matice regresorů a vektor pravých stran, výstupem funkce je optimální řešení odhadu neznámých parametrů ve smyslu TLS. Funkci lze používat samostatně, ale prioritně je určena jako pomocná funkce pro funkce ARX nebo ARMAX.
- `rtls.sci` - funkce pro výpočet optimálního řešení úplného problému nejmenších čtverců pomocí Tikhonovovy regularizace (RTLS) podle Kap. 3.2.2. Vstupem je matice regresorů, vektor pravých stran, počáteční řešení odhadu (obvykle lze jako počáteční odhad použít např. LS řešení) a ladící parametr algoritmu - regularizační parametr δ , určující míru penalizace počátečního řešení. Výstupem funkce je optimální řešení odhadu neznámých parametrů ve smyslu RTLS. Funkci lze používat samostatně, ale prioritně je určena jako pomocná funkce pro funkce ARX nebo ARMAX. Ve funkci je realizován Algoritmus 1.

6.5 Předzpracování dat

V této kapitole jsou popsány skripty sloužící k předzpracování dat.

6.5.1 Úprava signálu obsahujícího NaN hodnoty

Chybějící data se objevují v případě, že pro dané pozorování není v proměnné uložena žádná hodnota. V programu Scilab jsou tyto chybějící hodnoty reprezentovány hodnotou NaN (Not a Number). Úlohou je tedy data upravit tak, aby se hodnoty NaN zaplnily jinou hodnotou nebo vynechaly, opravila se konzistence dat, popřípadě se vyřešila nadbytečnost dat. Úpravy signálu obsahující NaN hodnoty tedy zahrnují následující řešení:

- `nanhandle.sci` - vstupní parametry funkce `nanhandle` jsou data určená k úpravě a volba metody úpravy. Výstup skriptu pak jsou data s doplněnými chybějícími hodnotami. Jsou poskytnuty dvě metody doplnění chybějících hodnot. První realizuje doplnění hodnot jako zero-order hold (ZOH), tudíž chybějící hodnotu nahradí nejbližší předchozí hodnotou. Druhá pak realizuje doplnění hodnot pomocí lineární interpolace.
- `nanredundant.sci` - tento skript byl vytvořen pro data, která jsou neekvidistantně vzorkovaná a u kterých proběhl sběr dat z jednotlivých senzorů v krátkých, avšak ne přesně stejných intervalech. Je tedy potřeba, aby v datech byla i časová základna s časy odebrání vzorku. Vstupní parametry funkce jsou pak data, vzorkovací perioda (která v naměřených datech nemusí být zcela přesná) a časový interval mezi vzorky. Tento časový interval musí být menší nebo roven rozdílu vzorkovací periody a pravděpodobné doby sběru dat ze všech senzorů. Výstupem skriptu jsou pak data zbavená nadbytečných hodnot NaN.

- `repairtime.sci` - skript `repairtime` navazuje na předchozí `nanredundant`. Vstupními parametry je výstup skriptu `nanredundant` a vzorkovací perioda. Výstupem jsou pak původní data, ekvidistantně vzorkovaná a bez výpadků v časové ose.

Předpokládá se, že nejčastěji bude pro úpravu signálu obsahujícího hodnoty NaN využíván skript `nanhandle`. Při využití všech tří skriptů se pak předpokládá, že uživatel nejdříve odstraní nadbytečné hodnoty NaN, dále použije skript `repairtime` a nakonec doplní chybějící hodnoty skriptem `nanhandle`.

6.5.2 Převzorkování signálů

- `rsmp.sci` - funkce realizující převzorkování, nová vzorkovací frekvence je vůči původní frekvenci v poměru m/c , kde m a c jsou vstupními parametry funkce. Nejprve je vstupní signál m -násobně nadvzorkovaný (realizuje se lineárním prokládáním mezi jednotlivými hodnotami dat uložených ve vstupním signálu) a následně se vybírá každý c -tý vzorek z nadvzorkovaného signálu. V poslední fázi dojde k vytvoření časového vektoru výstupního signálu tak, aby měli vstupní a výstupní signál stejnou délku. Při poměru m/c menším než 1 se doporučuje použít na předzpracování původního signálu antialiasingový filtr (funkce `alias_k`). Vstupem funkce `rsmp` je vstupní signál (matice), parametr m (nadvzorkování), parametr c (podvzorkování). Výstupem je časový vektor převzorkovaného signálu, převzorkovaný signál, původní časový vektor a původní signál.

6.5.3 Odstranění středních hodnot a trendů

- `notrend.sci` - funkce odstraňující v závislosti na vstupních parametrech střední hodnotu nebo trendy. Odstraňování střední hodnoty signálu zajišťuje vstupní parametr 1. V případě, že vstupní parametr má hodnotu 2, podle třetího vstupního parametru (délka lineárního trendu) je vytvořen vektor zlomových bodů (body, ve kterých začíná nový úsek postihnutý trendem). Dále se vytvoří prázdná matice, která se v cyklu naplní koeficienty zachycujícími lineární trend. Nakonec je od vstupního vektoru odečtena matice trendů. Vstupem funkce je vstupní signál (matice), parametr výběru střední hodnoty/trendu, parametr popisující délku trendu. Výstupem je signál zbavený střední hodnoty/trendu.

6.5.4 Filtry

Implementované jsou ideální filtry (dokonale strmá frekvenční charakteristika s nulovým zvlněním) následujících typů:

- `d.k.sci` (dolní propust) - signál je zpracován algoritmem rychlé Fourierovy transformace (FFT) s využitím funkce `fft`. Získané spektrum je posunuté funkcí `fftshift`. Frekvence vyšší, než zlomová frekvence jsou nulované. Spektrum je posunuté zpět a zpracované inverzní rychlou Fourierovou transformací (IFFT). Použité funkce `fft`, `ifft` a `fftshift` jsou součástí `Scilab Signal Processing` toolboxu. Požadovanými vstupními parametry funkce jsou vstupní signál, vzorkovací frekvence a zlomové frekvence. Výstupem je časový vektor, původní signál (nefiltrovaný) a filtrovaný signál, vektor frekvencí (pro vykreslování frekvenčního spektra), frekvenční spektrum původního signálu, frekvenční spektrum filtrovaného signálu (stejně u všech typů filtrů).
- `h.k.sci` (horní propust) - využívá modifikovaný algoritmus dolnopropustního filtru s tím rozdílem, že filtruje (nuluje) frekvence nižší, než žádaná zlomová frekvence.
- `pp.k.sci` (pásmová propust) - algoritmus jako u předcházejících filtrů. Nulované jsou frekvence nižší než dolní zlomová frekvence a vyšší než horní zlomová frekvence.
- `pz.k.sci` (pásmová zádrž) - nulované jsou frekvence vyšší než dolní zlomová frekvence a nižší než horní zlomová frekvence.
- `ampl.k.sci` (amplitudový filtr) - v případě amplitudového filtru jsou nulované frekvence, u kterých je amplituda ve frekvenčním spektru nižší než b -násobek největší špičky. Vstupními parametry funkce jsou vstupní signál, vzorkovací frekvence a filtrační parametr $b \in (0, 1)$.
- `alias.k.sci` (antialiasing filtr) - jedná se o filtr zabráňující aliasingu - jevu vznikajícímu v případě nedodržení Shannonova teoremu při vzorkování. Filtr je realizován dolní propustí, jejíž zlomová frekvence je rovna Nyquistově frekvenci (polovině vzorkovací frekvence). Tento filtr je vhodné použít před vzorkováním v případě, že se předpokládá nezanedbatelná energie signálu nad Nyquistovou frekvencí. Vstupními parametry funkce jsou vstupní signál, původní vzorkovací frekvence a nová vzorkovací frekvence.

6.6 Identifikační algoritmy

V této sekci budou stručně popsány implementované identifikační algoritmy, které se dají rozdělit do dvou tříd: klasické vstupně výstupní identifikační algoritmy (ARX, ARMAX) a subspace metody, kde je výsledkem identifikace přímo model ve formě stavového popisu.

- `arx.sci` - funkce pro výpočet odhadu neznámých parametrů ARX modelu. Vstupními parametry jsou vstupní posloupnost $u(k)$, výstupní posloupnost $y(k)$, řády polynomů n_a , n_b , zpoždění n_k a parametr `method`, který specifikuje, jakou z metod LS, TLS, případě RTLS, bude výpočet neznámých parametrů ARX modelu proveden. Při použití metody LS se jedná o klasickou implementaci (Kap. 3.1), naopak použití metod TLS nebo RTLS

je vhodné v případě tzv. *Errors in variables* (EIV) identifikačních problémů (Kap. 5.7). Výstupem funkce jsou odhadované polynomy $a(d)$, $b(d)$.

- `armax.sci` - pro odhadování neznámých parametrů ARMAX modelu bylo použito rekurzivní odhadování podle [39], kde je využito vyjádření prediktoru jako

$$\hat{y}(t|t-1) = b(d)u(k) + (1 - a(d))y(k) + (c(d) - 1)\varepsilon(k|k-1),$$

kde $\varepsilon(k|k-1) = y(k) - \hat{y}(k|k-1)$. Následuje sestavení pseudolineární regrese

$$\hat{y}(t|t-1) = z^T(t, \theta)\theta,$$

kde z obsahuje známá vstupně výstupní data a chyby predikce. Rekurzivní identifikace pak probíhá podle vztahů

$$\begin{aligned} \varepsilon(k|k-1) &= y(k) - z^T(k)\hat{\theta}(k-1) \\ \hat{\theta}(k) &= \hat{\theta}(k-1) + \frac{P(k-1)z(k)}{1 + z^T(k)P(k-1)z(k)}\varepsilon(k|k-1) \\ P(k) &= P(k-1) - \frac{P(k-1)z(k)z^T(k)P(k-1)}{1 + z^T(k)P(k-1)z(k)}, \end{aligned}$$

kde $\hat{\theta}(k)$ je odhad koeficientů polynomů a , b , c v čase k a $P(k)$ je jeho normovaná (vydělená rozptylem šumu $e(k)$) kovarianční matice. Vstupem *method* lze volit metodu identifikace. Defaultně se používá rekurzivní odhadování uvedené výše, další možností je TLS a RTLS algoritmus. Oba jsou vhodné pro tzv. *Errors in variables* identifikační problémy (Kap. 5.7).

- `kalman.sci` - funkce realizující výpočet aktualizace odhadu stavu systému kalmanovým filtrem (Kap. 4.4.3). Vstupem funkce jsou systémové matice A , B , C , D , kovariance šumu procesu Q a šumu měření R a aktuální stav. Výstupem je aktualizovaný stav systému. Ve funkci je realizován Algoritmus 3.
- `physical.sci` - funkce slouží pro transformaci zadaného tvaru stavového popisu systému do požadovaného tvaru nalezením příslušné transformační matice T , přičemž lze konkrétně specifikovat některé prvky výsledných matic. Vstupem funkce jsou systémové matice A , B , C , D , získané libovolnou BB identifikací (obvykle subspace), dále požadovaná podoba výsledných matic $A0$, $B0$, $C0$, $D0$, přičemž některé prvky můžeme numericky specifikovat a prvky, které požadujeme odhadovat označíme jako `%inf`, parametrem *initial* lze udat počáteční hodnoty těchto `%inf` hodnot. Pro dobrý odhad není vhodné pevně specifikovat více než polovinu prvků v každé jednotlivé matici. Výstupem funkce jsou matice stavového popisu v požadovaném tvaru a příslušná transformační matice. Ve funkci je realizován Algoritmus 6.

- `rarx.sci` - koeficienty polynomů $a(d)$, $b(d)$ lze odhadovat též rekurzivně, čehož lze například využít ke sledování časově proměnných parametrů nebo v situaci, kdy je objem identifikačních dat příliš velký a nelze provést jednorázovou identifikaci. Funkce realizuje rekurzivní výpočet odhadu neznámých parametrů ARX modelu podobně jako v případě ARMAX modelu ([39]). Vstupními parametry jsou vstupní posloupnost $u(k)$, výstupní posloupnost $y(k)$, řády polynomů n_a , n_b , zpoždění n_k . Výstupem funkce jsou odhadované polynomy $a(d)$, $b(d)$.
- `srcf.sci` - funkce realizující výpočet aktualizace odhadu stavu systému square-root kovariančním filtrem (Kap. 4.4.4). Vstupem funkce jsou systémové matice A , B , C , D , kovariance šumu procesu Q a šumu měření R a aktuální stav. Výstupem je aktualizovaný stav systému. Ve funkci je realizován Algoritmus 4.
- `subid.sci` - funkce realizující algoritmus subspace identifikace. Implementace je založená na jediné QR faktorizaci matice dat s tím, že je počítán pouze R faktor, což přináší výrazné urychlení. Implementace je převzatá z [34], kde je známá jako tzv. robustní kombinovaná subspace identifikace. Vstupními parametry jsou vstupní posloupnost $u(k)$, výstupní posloupnost $y(k)$, počet bloků i Hankelových matic, parametr $dmat$, kterým specifikujeme, zda požadujeme nulovou či nenulovou matici D , případně parametr n , kterým můžeme specifikovat řád modelu. V opačném případě je řád stanoven automaticky. Výstupem funkce jsou systémové matice A , B , C , D a K .
- `substruc.sci` - funkce pro odhad parametrů ARMAX modelu metodou PEM inicializovanou subspace metodou. Vstupními parametry jsou vstupní posloupnost $u(k)$, výstupní posloupnost $y(k)$, počet bloků i Hankelových matic, řády polynomů n_a , n_b , n_c a zpoždění n_k . Výstupem funkce jsou odhadované polynomy $a(d)$, $b(d)$, $c(d)$, případně systémové matice A , B , C , D a K v korespondujícím kanonickém tvaru pozorovatelnosti (4.36). Ve funkci je realizován Algoritmus 5.

6.7 Algoritmy pro redukci řádu modelu

V toolboxu je implementována funkce pro redukci řádu ARX/ARMAX modelu založená na metodě Nonnegative Garrote (NNG) (Kap. 3.4).

- `selstruc.sci` - funkce slouží k redukci řádu ARX nebo ARMAX modelu. Jejím vstupem je libovolný ARX resp. ARMAX model společně se vstupně - výstupními identifikačními a validačními daty. Funkce redukuje řád modelu s využitím funkce `nng`. Během výpočtu je zobrazena uživateli sada redukovaných modelů s příslušnými fit faktory počítanými na validačních datech a uživatel si vybere požadovaný model. Výstupem je potom vybraný model, případně celá sada vypočtených modelů. Redukci řádu modelu tedy realizuje Algoritmus 2.

6.8 Volba identifikačního kritéria a verifikace modelu

Současná verze toolboxu umožňuje zvolit identifikační kritérium jako chybu predikce nebo chybu simulace na validačních datech.

6.8.1 Volba identifikačního kritéria

- `arIdent.sci` - v případě ARX a ARMAX funkce nalezne model minimalizující chybu predikce nebo chybu simulace ze zadaného rozsahu parametrů n_a, n_b, n_k, n_c z (4.14) a (4.23). Tzn., že funkce slouží pro odhad neznámých parametrů ARX modelu pro zadaný rozsah parametrů n_a, n_b, n_k , nebo ARMAX modelu pro zadaný rozsah parametrů n_a, n_b, n_c, n_k . Řád modelu je vybírán na základě nejlepšího fitu pro všechny možné kombinace ze zadaného rozsahu parametrů n_a, n_b, n_k resp. n_a, n_b, n_c, n_k . Dalšími vstupy funkce jsou vstupní posloupnost $u(k)$, výstupní posloupnost $y(k)$, validační data $uv(k)$ a $yv(k)$ a parametr sp , který určuje, zda bude jako identifikační kritérium zvolena chyba predikce nebo chyba simulace na validačních datech. Výstupem funkce jsou odhadované polynomy $a(d), b(d), (c(d))$, normovaná (vydělená rozptylem šumu $e(k)$) kovarianční matice $P(k)$ a dosažený fit faktor.
- `subspaceIdent.sci` - u subspace identifikačního algoritmu jsou ladícími parametry řád modelu n a velikost bloku minulých resp. budoucích dat i . Funkce hledá model minimalizující chybu predikce nebo chybu simulace v zadaném rozsahu parametrů, přičemž algoritmus subspace identifikace vyžaduje aby $pi \geq n$, kde p je počet výstupů systému.

6.8.2 Verifikace modelu

- `estimatex0.sci` - pro zadané systémové matice A, B, C, D, K a vstupní a výstupní signály uv, yv funkce spočítá odhad počátečního stavu systému.
- `fitFactor.sci` - funkce, která určuje míru shody posloupnosti odhadnutých dat vzhledem k posloupnosti dat validačních podle (2.9).
- `lsim.sci` - funkce pro zadané systémové matice A, B, C, D spočítá odezvu systému na zadaný vstupní signál u . Výstupem funkce je výstup systému y a stavová posloupnost x .
- `residues.sci` - funkce počítá rezidua pro dvojici posloupností. Výstupem jsou vypočtená rezidua, střední hodnota jejich kvadrátů a maximální hodnota.
- `valSim.sci` - vstupními parametry funkce jsou koeficienty polynomů ARX modelu a, b a validační data y_v, u_v . Z načtených vstupů a výstupů funkce vypočte simulovaný výstup y_s a fit faktor. Funkce je chápána jako pomocná funkce pro funkci `validate`, ale lze používat i samostatně.

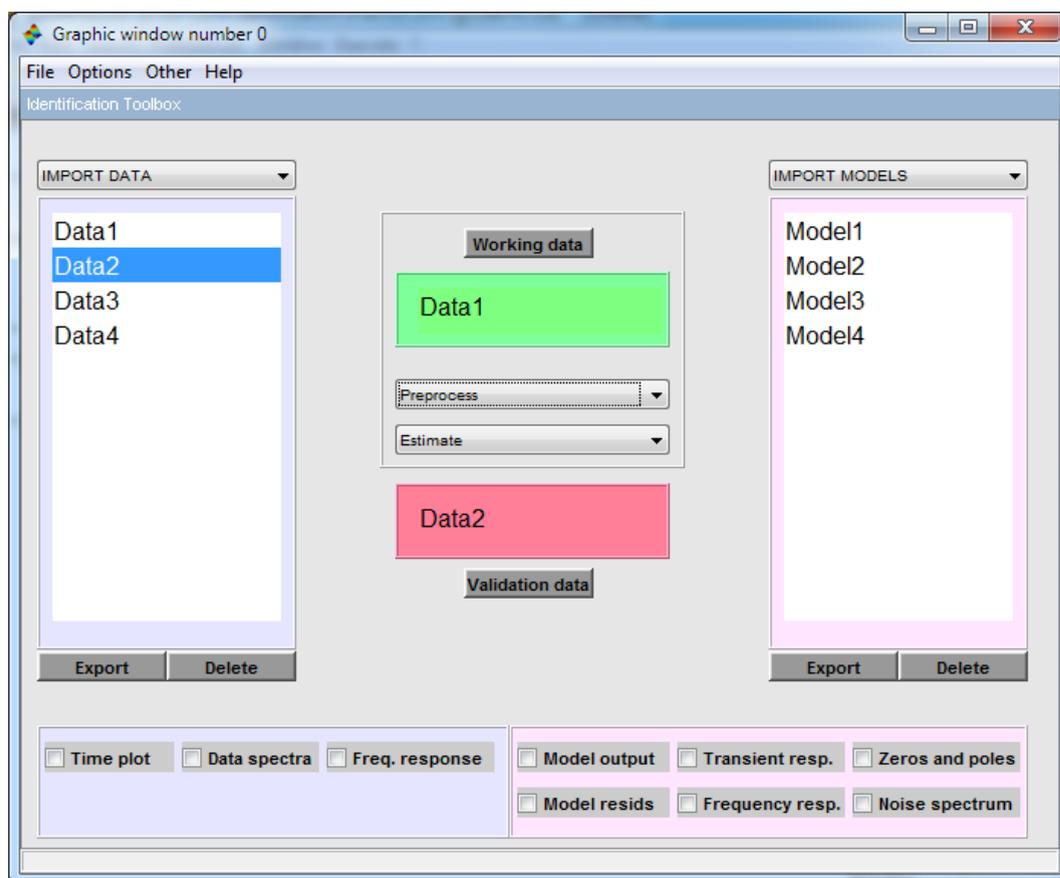
- `valPred.sci` - funguje obdobně jako funkce `valSim`, navíc má vstupní parametr c . Výstupem však není simulovaný, ale predikovaný výstup modelu y_p . Funkce je použita jako pomocná funkce pro funkci `validate` s tím, že ji lze používat i samostatně.
- `validate.sci` - vstupními parametry funkce jsou buď systémové matice A , B , C , D , K nebo koeficienty polynomů a , b , c (v případě ARX nebo ARMAX identifikace) získané některou z identifikačních metod a parametr sp , podle kterého se určuje, zda bude identifikačním kritériem chyba predikce nebo chyba simulace. Vstupy uv , yv představují validační data. Pro polynomiálně zadané systémy se na základě hodnoty parametru sp volá jedna z dvojice funkcí `valPred` nebo `valSim`, která vypočte predikovaný resp. simulovaný výstup identifikovaného systému a fit faktor. V případě, že jsou vstupními parametry systémové matice, nejprve proběhne odhad počátečního stavu funkcí `estimatem0` a dále následuje výpočet simulovaného resp. predikovaného výstupu pomocí funkce `lsim`. Výstupem funkce jsou tedy simulovaný resp. predikovaný výstup modelu spolu s fit faktorem počítaným vůči y_v .

6.9 Dokumentace

Dokumentace k identifikačnímu toolboxu byla pojata formou nápovědy k implementovaným funkcím. Uživatel má tedy k dispozici popis zahrnující volací sekvenci funkce, definici vstupních a výstupních parametrů, krátký popis funkčnosti nebo použitých teoretických základů a jednoduchý příklad použití, vhodný pro bližší pochopení, či ověření funkčnosti. Stejnou strukturou nápovědy disponují také moduly a funkce implementované ve `Scilabu`. Dokumentace byla vytvořena ve standardu `DocBook`. Tento systém je založený na XML (Extensible Markup Language) a určený především pro tvorbu dokumentace k softwaru a hardwaru. Jednou z předností systému `DocBook` je možnost snadné konverze dokumentů do dalších formátů. Bylo tedy využito konverze do HTML (Hypertext Markup Language) pro snadné prohlížení uživatelem. Po začlenění toolboxu do systému `ATOMS` bylo použito automatické generování nápovědy na základě standardizované struktury komentářů v záhlaví jednotlivých funkcí pomocí nástroje `help_from_sci` (http://help.scilab.org/docs/5.3.0/en_US/help_from_sci.html).

6.10 Grafické uživatelské rozhraní

V současnosti je rozpracované grafické uživatelské rozhraní, které bude v budoucnu rozšířeno a začleněno do identifikačního toolboxu. Grafické uživatelské rozhraní bude koncepčně podobné jako v `SITB` pro `Matlab` a uživateli bude přístupné klasickým příkazem `ident`.



OBRÁZEK 6.1: Grafické uživatelské rozhraní

Kapitola 7

Závěr

Cílem této práce bylo seznámit se s používanými lineárními identifikačními metodami a strukturami modelů a následně je implementovat v rámci identifikačního toolboxu pro programové prostředí Scilab. V další fázi bylo úkolem diskutovat možnosti inicializačních metod pro identifikaci systémů a implementovat *Prediction Error Method* (PEM) pro vybrané struktury modelů inicializovanou pomocí *Subspace State-Space System IDentification* (4SID) metody.

Úvod práce předkládá vysvětlení základních pojmů a principů z oblasti identifikace lineárních dynamických systémů, následně jsou rozebrány vybrané matematické nástroje používané v identifikacích. Další kapitoly se zabývají popisem vybraných struktur modelů a identifikačních metod. Vedle standardních modelů a metod se práce zabývá zejména novinkami a nezvyklými implementacemi. První z nich je implementace *Prediction Error Method* (PEM) inicializovaná *Subspace State-Space System IDentification* (4SID) metodou. Jako další byla implementována metoda *NonNegative Garrote* (NNG), která představuje nový přístup k redukci řádu lineárních modelů. Konkrétně byl implementován algoritmus NNG metody pro redukci řádu modelu ARX a odvozeny obecné formulace omezení, umožňující použití NNG metody pro redukci řádu libovolného ARX modelu. Algoritmus byl v rámci práce rozšířen i pro ARMAX model. Vedle toho byla implementována *Errors-In-Variables* (EIV) identifikace. Dále potom tzv. *square-root* kovarianční filtr (SRKF), který představuje rychlou numerickou implementaci Kalmanova filtru. Závěrem byl navržen a implementován algoritmus, představující specifický přístup k odhadu fyzikálních parametrů systému *Black-Box* (BB) modelováním, který určitým způsobem propojuje identifikaci neznámého systému ze vstupně-výstupních posloupností naměřených dat s matematicko-fyzikální analýzou. Všechny výše uvedené metody byly začleněny do Identifikačního Toolboxu pro Scilab.

Stávající podoba identifikačního toolboxu představuje ucelený nástroj, který umožňuje uživateli pomocí jednoduchých příkazů předzpracovávat data, volit typ identifikace a verifikovat navržené modely. V současné verzi je rozpracované grafické uživatelské rozhraní, které v budoucnu práci zkomfortní.

Literatura

- [1] LJUNG, L. *System Identification: Theory for the User (2nd Edition)*. Prentice Hall, 1999. ISBN 978-0136566953.
- [2] LJUNG, L. System identification toolbox for use with matlab: User's guide. 1995. URL <http://infoscience.epfl.ch/record/24813>.
- [3] NELLES, O. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Berlin: Springer, 2001. ISBN 3-540-67369-5.
- [4] ANTSAKLIS, P. J. AND MICHEL, A. N. *A Linear Systems Primer*. Birkhäuser, 2007.
- [5] HAVLENA, V. and ŠTECHA, J. *Moderní teorie řízení*. Praha: Vydavatelství ČVUT, 2000. ISBN 80-01-02095-9.
- [6] VERHAEGEN, M. AND VERDULT, V. *Filtering and system identification: A Least Squares Approach*. Cambridge University Press, 2007. ISBN 978-0-521-87512-7.
- [7] MIRTA, S. K. AND KAISER, J. F. *Handbook for Digital Signal Processing*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1993. ISBN 0471619957.
- [8] AKAIKE, H. Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21:243–247, 1969. ISSN 0020-3157. 10.1007/BF02532251.
- [9] RISSANEN, J. Modeling by shortest data description. *Automatica*, 14(5):465 – 471, 1978. ISSN 0005-1098. doi: DOI:10.1016/0005-1098(78)90005-5.
- [10] ZHU, Y. *Multivariable system identification for process control*. Pergamon, 2001. ISBN 0-08-043985-3.
- [11] LAWSON, C.L. AND HANSON, R.J. *Solving least squares problems*. Classics in applied mathematics. SIAM, 1995. ISBN 9780898713565.
- [12] BJÖRCK, Å. *Numerical methods for least squares problems*. Miscellaneous Bks. SIAM, 1996. ISBN 9780898713602.
- [13] SAMUELSON, P.A. *The collected scientific papers of Paul A. Samuelson*. Number sv. 5 in The Collected Scientific Papers of Paul A. Samuelson. M.I.T. Press, 1986. ISBN 9780262192514.

-
- [14] GOLUB, G. H. AND VAN LOAN, CH. F. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0-8018-5414-8.
- [15] TIKHONOV, A. N. AND ARSENIN, V. I. A. *Solutions of ill-posed problems*. Scripta series in mathematics. Winston, 1977. ISBN 9780470991244.
- [16] GOLUB, G. H. AND HANSEN, P. CH. AND O'LEARY, D. P. Tikhonov regularization and total least squares. *SIAM J. MATRIX ANAL. APPL*, 21:185–194, 1997.
- [17] HUFFEL, S. AND LEMMERLING, P. *Total least squares and errors-in-variables modeling: analysis, algorithms and applications*. Kluwer Academic, 2002. ISBN 9781402004766.
- [18] REIERSØL, O. Confluence Analysis by Means of Lag Moments and Other Methods of Confluence Analysis. *Econometrica*, 9(1):1–24, 1941.
- [19] SÖDERSTRÖM, T. AND STOICA, P. Instrumental variable methods for system identification. *Circuits, Systems, and Signal Processing*, 21:1–9, 2002. ISSN 0278-081X. 10.1007/BF01211647.
- [20] AL-SMADI, A. AND WILKES, D.M. Robust and accurate arx and arma model order estimation of non-gaussian processes. *Signal Processing, IEEE Transactions on*, 50(3): 759–763, March 2002. ISSN 1053-587X. doi: 10.1109/78.984778.
- [21] LIANG, G. AND WILKES, D.M. AND CADZOW, J.A. Arma model order estimation based on the eigenvalues of the covariance matrix. *Signal Processing, IEEE Transactions on*, 41(10):3003–3009, October 1993. ISSN 1053-587X. doi: 10.1109/78.277805.
- [22] WAHLBERG, B. Model reductions of high-order estimated models: the asymptotic ml approach. *International Journal of Control*, 49(1):169–192, 1989.
- [23] TIBSHIRANI, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.
- [24] BREIMAN, L. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995. ISSN 0040-1706.
- [25] YUAN, M. AND LIN, Y. On the non-negative garrote estimator. *Journal Of The Royal Statistical Society Series B*, 69(2):143–161, 2007.
- [26] LYZELL, CH. *Initialization Methods for System Identification*. PhD thesis, Linköping studies in science and technology, 2009.
- [27] LYZELL, C. AND ROLL, J. AND LJUNG, L. The use of nonnegative garrote for order selection of ARX models. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 1974–1979. IEEE, 2009.

- [28] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Statist. Soc. B*, 68(1):49–67, 2006.
- [29] BOYD, S. AND VANDENBERGHE, L. *Convex Optimization*. Cambridge University Press, 2004.
- [30] LAUB, A. AND HEATH, M. AND PAIGE, C. AND WARD, R. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *Automatic Control, IEEE Transactions on*, 32(2):115 – 122, February 1987. ISSN 0018-9286. doi: 10.1109/TAC.1987.1104549.
- [31] MOORE, B. Principal component analysis in linear systems: Controllability, observability, and model reduction. *Automatic Control, IEEE Transactions on*, 26(1):17 – 32, February 1981. ISSN 0018-9286. doi: 10.1109/TAC.1981.1102568.
- [32] DUNCAN, D. B. AND HORN, S. D. Linear Dynamic Recursive Estimation from the Viewpoint of Regression Analysis. *Journal of the American Statistical Association*, 67(340): 815–821, 1972. ISSN 01621459. doi: 10.2307/2284643.
- [33] ELIASON, S. R. *Maximum likelihood estimation: logic and practice*. Number sv. 96 in Coll. principale : Sage university papers. Sage, 1993. ISBN 9780803941076.
- [34] OVERSCHEE, V. P. and DE MOOR, B. *Subspace Identification for Linear Systems*. Boston/London/Dordrecht: Kluwer Academic Publishers, 1996. ISBN 978-0792397175.
- [35] TRNKA, P. *Subspace Identification Methods*. 2008.
- [36] SÖDERSTRÖM, T. Errors-in-variables methods in system identification. *Automatica*, 43(6): 939 – 958, 2007. ISSN 0005-1098. doi: DOI:10.1016/j.automatica.2006.11.025.
- [37] XIE, L. and LJUNG, L. *Estimate Physical parameters by Black-Box Modeling*. Technical report from Automatic Control at Linköpings universitet, 2003.
- [38] Control Systems Principles. URL <http://www.control-systems-principles.co.uk/whitepapers/ball-and-beam1.pdf>.
- [39] PEKAŘ, J. *Odhad parametrů ARMAX modelu*. 2004.

Příloha A

Obsah priloženého CD

K této práci je přiložené CD, na kterém jsou uloženy zdrojové kódy a elektronická podoba práce.

- Adresář **codes**: zdrojové kódy identifikačního toolboxu pro Scilab a zdrojové kódy příkladů pro Matlab.
- Adresář **thesis**: elektronická podoba diplomové práce.