

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Kalmanovy filtry

Diplomová práce

Autor: Kateřina Frončková

Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. RNDr. Pavel Pražák, Ph.D.

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracovala samostatně a s použitím uvedené literatury.

V Hradci Králové dne 27. 4. 2018

Kateřina Frončková

Poděkování

Ráda bych poděkovala doc. RNDr. Pavlu Pražákovi, Ph.D. za odborné vedení, ochotu a rady poskytnuté při vypracovávání této práce. Dále děkuji Ing. Pavlu Křížovi, Ph.D. za poskytnutí testovacích dat z výzkumu indoor lokalizace.

Anotace

Diplomová práce se zabývá představením Kalmanova filtru a možností jeho využití v různých aplikačních doménách. Pozornost je zaměřena na podrobný teoretický výklad algoritmu i na praktické aspekty implementace. Kromě původního Kalmanova filtru jsou zmíněny také různé další algoritmy vycházející z jeho podstaty, které bývají souhrnně označovány jako Kalmanovy filtry. Aplikační část práce se věnuje zejména použití Kalmanova filtru při řešení úlohy indoor lokalizace a dále ukazuje rovněž možnosti uplatnění v analýze časových řad a ve zpracování obrazu. Součástí práce je také implementace představených algoritmů a aplikačních úloh v prostředí MATLAB.

Annotation

Title: **Kalman Filters**

The diploma thesis deals with the presentation of the Kalman filter and its applications in different areas. The attention is focused on the detailed theoretical description of the algorithm and on the practical aspects of its implementation. Besides the standard Kalman filter, other algorithms developing its fundamental idea are mentioned. These algorithms are known as the Kalman filters. The practical part of the thesis is devoted mainly to the use of the Kalman filter in indoor localization. Other applications from the field of time series analysis and image processing are shown at the end. The presented algorithms and the solutions of practical problems are implemented in MATLAB.

Obsah

1	Úvod	1
2	Základní pojmy a teoretická východiska	3
2.1	Uvedení pojmů z oblasti lineární algebry	3
2.1.1	Definice matice a vektoru	3
2.1.2	Vybrané typy matic se speciální strukturou	4
2.1.3	Regulární matice a matice inverzní	4
2.1.4	Pozitivně definitní a pozitivně semidefinitní matice	4
2.2	Základy teorie pravděpodobnosti a statistiky	5
2.2.1	Definice pojmu náhodné veličiny	5
2.2.2	Distribuční funkce a funkce hustoty pravděpodobnosti	5
2.2.3	Některé charakteristiky náhodné veličiny	5
2.2.4	Náhodná veličina s normálním rozdělením	6
2.2.5	Náhodný proces	7
2.3	Dynamické systémy a jejich modely	8
2.3.1	Deterministické dynamické systémy	8
2.3.2	Stochastické dynamické systémy	11
3	Kalmanova filtrace	13
3.1	Kalmanův filtr	13
3.1.1	Představení a historie Kalmanova filtru	13
3.1.2	Algoritmus Kalmanova filtru a jeho odvození	15
3.1.3	Vlastnosti algoritmu a poznámky	20
3.1.4	Demonstrační příklad	24
3.1.5	Praktické aspekty implementace algoritmu	29
3.2	Další algoritmy Kalmanovy filtrace	34
3.2.1	Nelineární filtrace	34
3.2.2	Filtrace ve spojitém čase	39

4	Aplikace Kalmanových filtrů	40
4.1	Přehled aplikačních domén	40
4.2	Indoor lokalizace	40
4.2.1	Přehled stávajících řešení	41
4.2.2	Představení Kalmanových filtrů v indoor lokalizaci	44
4.2.3	Implementace, testování a zhodnocení popsaných algoritmů . . .	47
4.3	Ukázky dalších aplikací Kalmanových filtrů	52
4.3.1	Analýza časových řad	52
4.3.2	Zpracování obrazu	56
5	Shrnutí výsledků	60
6	Závěry a doporučení	62
	Literatura	63
	Použité značení	68
	Seznam použitých zkratk	69
	Seznam obrázků	70
	Seznam tabulek	70
	Přílohy	71

1 Úvod

Kalmanův filtr je nástroj sloužící k odhadování neznámých stavů stochastických lineárních dynamických systémů na základě naměřených dat zatížených šumem. Algoritmus má svůj původ v 60. letech minulého století, od té doby byla jeho základní myšlenka využita v řadě aplikačních domén nejen technického charakteru. Pro zajímavost, Kalmanův filtr se objevil i v seznamu 10 nejdůležitějších algoritmů 20. a 21. století dle [20].

Cílem práce je podrobně představit algoritmus Kalmanova filtru se zaměřením na jeho teoretický výklad i aspekty praktické implementace. Pozornost je rovněž věnována dalším algoritmům vycházejícím z podstaty Kalmanova filtru, které bývají souhrnně označovány jako algoritmy Kalmanovy filtrace či Kalmanovy filtry. Úkolem práce je také ukázat možnosti využití Kalmanových filtrů v různých aplikačních oblastech. Představené algoritmy a demonstrační úlohy jsou pak implementovány v prostředí MATLAB.

První část práce se věnuje připomenutí některých základních pojmů a východisek z oblasti lineární algebry, teorie pravděpodobnosti, statistiky a teorie dynamických systémů, které budou stěžejní pro celý další text.

Druhá část představuje algoritmus Kalmanova filtru, uvádí jeho odvození, popis vlastností, diskuzi předpokladů a poznámky k praktické implementaci. Princip fungování Kalmanova filtru je ukázán na jednoduchém ilustračním příkladu. Dále se tato část zabývá i přiblížením dalších Kalmanových filtrů, přičemž pozornost je zaměřena zejména na algoritmy umožňující řešit úlohu nelineární filtrace.

Poslední část práce se zabývá aplikacemi Kalmanova filtru a z něj vycházejících algoritmů. Hlavní prezentovanou aplikační doménou je indoor lokalizace, neboť navigace, lokalizace a sledování pohybu jsou typické oblasti, ve kterých Kalmanův filtr nachází své uplatnění. Představeno je několik možností použití principů Kalmanova filtru při řešení úlohy statické lokalizace i lokalizace dynamicky se pohybujícího objektu. Popsané

algoritmy jsou následně otestovány na reálných datech a výsledky srovnány s klasickými lokalizačními technikami. Dále je širší využití Kalmanova filtru demonstrována také na jeho aplikacích v analýze časových řad a ve zpracování obrazu.

2 Základní pojmy a teoretická východiska

Nejprve budou stručně představeny některé pojmy a základní východiska z oblasti lineární algebry, teorie pravděpodobnosti a teorie dynamických systémů, které budou dále v práci používány. Zdefinovány či vysvětleny budou jen stěžejní pojmy a uvedena budou rovněž pouze vybraná nejdůležitější teoretická východiska, se kterými bude bezprostředně pracováno. Více podrobností a definice dalších pojmů v práci zmíněných jen okrajově, stejně jako některých všeobecně známých elementárních pojmů, lze nalézt v odborné literatuře, například [5] a [36] (lineární algebra), [3] (teorie pravděpodobnosti a statistika), [18] a [33] (dynamické systémy a jejich modely).

2.1 Uvedení pojmů z oblasti lineární algebry

2.1.1 Definice matice a vektoru

Obecně bude v práci vše formulováno v oboru reálných čísel. Obdélníkové schéma $\mathbf{A} \in \mathbb{R}^{m \times n}$ sestavené z reálných čísel

$$\mathbf{A} = \begin{pmatrix} a_{[11]} & a_{[12]} & \cdots & a_{[1n]} \\ a_{[21]} & a_{[22]} & \cdots & a_{[2n]} \\ \vdots & \vdots & \ddots & \vdots \\ a_{[m1]} & a_{[m2]} & \cdots & a_{[mn]} \end{pmatrix}$$

se nazývá (reálná) **matice** typu $m \times n$. Matice budou v textu označovány velkými tučnými písmeny, prvek matice \mathbf{A} nacházející se na pozici ij bude zapisován jako $a_{[ij]}$. Symbol $\mathbb{R}^{m \times n}$ značí množinu všech reálných matic typu $m \times n$. Je-li $m = n$, jedná se o čtvercovou matici řádu n .

n -rozměrným aritmetickým sloupcovým **vektorem** se rozumí matice typu $n \times 1$

$$\mathbf{x} = \begin{pmatrix} x_{[1]} \\ x_{[2]} \\ \vdots \\ x_{[n]} \end{pmatrix},$$

řádkovým vektorem je pak matice typu $1 \times n$

$$\mathbf{x}^T = (x_{[1]} \quad x_{[2]} \quad \cdots \quad x_{[n]}).$$

Vektory budou značeny malými tučnými písmeny, i -tý prvek vektoru \mathbf{x} bude označován jako $x_{[i]}$. Pro množinu všech reálných n -rozměrných vektorů je použit symbol \mathbb{R}^n .

2.1.2 Vybrané typy matic se speciální strukturou

Nechť $\mathbf{A} \in \mathbb{R}^{n \times n}$ je čtvercová matice řádu n . Matice \mathbf{A} se nazývá

- **horní trojúhelníková**, jestliže pro každé $i, j = 1, \dots, n$, $i > j$, je $a_{[ij]} = 0$,
- **dolní trojúhelníková**, jestliže pro každé $i, j = 1, \dots, n$, $i < j$, je $a_{[ij]} = 0$,
- **diagonální**, jestliže pro každé $i, j = 1, \dots, n$, $i \neq j$, je $a_{[ij]} = 0$,
- **symetrická**, jestliže pro každé $i, j = 1, \dots, n$, je $a_{[ij]} = a_{[ji]}$ (resp. $\mathbf{A} = \mathbf{A}^T$).

2.1.3 Regulární matice a matice inverzní

Opět necht' $\mathbf{A} \in \mathbb{R}^{n \times n}$ je čtvercová matice řádu n . **Inverzní maticí** k matici \mathbf{A} se rozumí matice $\mathbf{A}^{-1} \in \mathbb{R}^{n \times n}$, pro kterou platí $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$, kde \mathbf{I} je jednotková matice řádu n . Matice \mathbf{A} , k níž inverzní matice existuje, se nazývá **regulární**, v opačném případě se hovoří o matici **singulární**.

2.1.4 Pozitivně definitní a pozitivně semidefinitní matice

Nyní necht' $\mathbf{A} \in \mathbb{R}^{n \times n}$ je symetrická matice řádu n . Matice \mathbf{A} se nazývá

- **pozitivně definitní**, jestliže pro každý vektor $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} \neq \mathbf{0}$, je $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$,
- **pozitivně semidefinitní**, jestliže pro každý vektor $\mathbf{x} \in \mathbb{R}^n$, je $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$.

Budiž poznamenáno, že analogicky by bylo možné definovat také matici negativně definitní, negativně semidefinitní a dále matici indefinitní.

Věta 2.1. *Každá pozitivně definitní matice je regulární.*

Pro důkaz věty viz např. [36, s. 94].

2.2 Základy teorie pravděpodobnosti a statistiky

2.2.1 Definice pojmu náhodné veličiny

Náhodnou veličinu lze zjednodušeně popsat jako měřitelnou funkci definovanou na prostoru elementárních jevů Ω , jejíž každou hodnotou je reálné číslo (podrobněji viz např. [3, s. 27]). Pojem náhodné veličiny je možné zobecnit také na případ n -rozměrného vektoru. Náhodné veličiny bývají v literatuře obvykle značeny velkými písmeny a jejich konkrétní realizace potom písmeny malými, v dalších částech práce však toto nebude dodrženo a bude upřednostněno značení typicky používané v souvislosti s Kalmanovým filtrem a modely dynamických systémů.

2.2.2 Distribuční funkce a funkce hustoty pravděpodobnosti

K popisu rozdělení náhodné veličiny X slouží **distribuční funkce** F , která je definována vzorcem $F(x) = P(X \leq x)$, tzn. distribuční funkce každému reálnému číslu x přiřazuje pravděpodobnost, že náhodná veličina X nabude hodnoty menší či rovné tomuto číslu x . Derivace distribuční funkce, pokud existuje, se nazývá **hustota pravděpodobnosti** a značí se f . Zmíněné pojmy lze analogicky definovat rovněž pro vektorový případ, více viz např. [3, s. 37].

V textu se bude možné setkat také s pojmem **podmíněné hustoty pravděpodobnosti** náhodné veličiny X za podmínky, že náhodná veličina Y nabyla konkrétní realizace y , toto lze vyjádřit vzorcem

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)},$$

kde $f_{X,Y}(x,y)$ je sdružená hustota pravděpodobnosti veličin X, Y a $f_Y(y)$ je marginální hustota pravděpodobnosti veličiny Y .

2.2.3 Některé charakteristiky náhodné veličiny

Pro střední hodnotu náhodné veličiny bude používán ustálený symbol E . V textu práce bude obecně pracováno převážně s vektorovými náhodnými veličinami, tedy necht

$\mathbf{X} = (X_1, X_2, \dots, X_n)^T$ je n -rozměrná vektorová náhodná veličina, **střední hodnota** $E\langle \mathbf{X} \rangle$ je vektor středních hodnot

$$E\langle \mathbf{X} \rangle = (E\langle X_1 \rangle, E\langle X_2 \rangle, \dots, E\langle X_n \rangle)^T.$$

Kovarianční matice vektorové náhodné veličiny \mathbf{X} je definována jako

$$\mathbf{P}_{XX} = E\langle (\mathbf{X} - E\langle \mathbf{X} \rangle)(\mathbf{X} - E\langle \mathbf{X} \rangle)^T \rangle.$$

Věta 2.2. *Kovarianční matice je vždy symetrická a pozitivně semidefinitní.*

Symetrie kovarianční matice je přímo patrná ze vzorce, pro důkaz pozitivní semidefinitnosti viz např. [3, s. 39].

Dále necht' je navíc $\mathbf{Y} = (Y_1, Y_2, \dots, Y_m)^T$ m -rozměrná vektorová náhodná veličina. Pro **vzájemnou kovarianční matici veličin \mathbf{X} a \mathbf{Y}** lze zavést vztah

$$\mathbf{P}_{XY} = E\langle (\mathbf{X} - E\langle \mathbf{X} \rangle)(\mathbf{Y} - E\langle \mathbf{Y} \rangle)^T \rangle.$$

Budiž poznamenáno, že v případě obou těchto matic se lze v literatuře setkat s různou terminologií, například první z uvedených matic bývá někdy nazývána maticí varianční.

2.2.4 Náhodná veličina s normálním rozdělením

Náhodná veličina X má (Gaussovo) normální rozdělení, má-li hustota pravděpodobnosti tvar

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

kde střední hodnota μ a rozptyl σ^2 jsou parametry rozdělení (toto bude značeno $X \sim N(\mu, \sigma^2)$).

Je-li \mathbf{X} n -rozměrná vektorová náhodná veličina, má \mathbf{X} n -rozměrné normální rozdělení se střední hodnotou $\boldsymbol{\mu}$ a kovarianční maticí $\boldsymbol{\Sigma}$, pokud pro (sdruženou) hustotu pravděpodobnosti vektoru platí

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}.$$

Výše uvedená definice vyžaduje, aby matice $\boldsymbol{\Sigma}$ byla pozitivně definitní (aby existovala její inverze $\boldsymbol{\Sigma}^{-1}$), obecně však lze vícerozměrné normální rozdělení definovat i jiným způsobem, při kterém postačuje, aby tato matice byla pozitivně semidefinitní.

Předpoklad normálního rozdělení veličin bývá častý, proto budiž zmíněn speciální vztah tohoto rozdělení k lineárním transformacím (obecně pro vektorové náhodné veličiny). Důkazy obou vět uvádí např. [33, s. 112].

Věta 2.3. *Nechť \mathbf{X} je n -rozměrná vektorová náhodná veličina mající normální rozdělení se střední hodnotou $\boldsymbol{\mu}_X$ a kovarianční maticí $\boldsymbol{\Sigma}_{XX}$, dále necht \mathbf{A} je nenáhodná matice typu $m \times n$ a \mathbf{b} je nenáhodný m -rozměrný vektor. Potom náhodná veličina $\mathbf{Z} = \mathbf{A}\mathbf{X} + \mathbf{b}$ má také normální rozdělení se střední hodnotou $\boldsymbol{\mu}_Z = \mathbf{A}\boldsymbol{\mu}_X + \mathbf{b}$ a kovarianční maticí $\boldsymbol{\Sigma}_{ZZ} = \mathbf{A}\boldsymbol{\Sigma}_{XX}\mathbf{A}^T$.*

Věta 2.4. *Nechť \mathbf{X} a \mathbf{Y} jsou vektorové náhodné veličiny rozměrů n resp. m , které mají sdružené normální rozdělení, a dále \mathbf{A} a \mathbf{B} jsou nenáhodné matice typu $p \times n$ resp. $p \times m$. Pak náhodná veličina $\mathbf{W} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{Y}$ má normální rozdělení se střední hodnotou $\boldsymbol{\mu}_W = \mathbf{A}\boldsymbol{\mu}_X + \mathbf{B}\boldsymbol{\mu}_Y$ a kovarianční maticí $\boldsymbol{\Sigma}_{WW} = \mathbf{A}\boldsymbol{\Sigma}_{XX}\mathbf{A}^T + \mathbf{A}\boldsymbol{\Sigma}_{XY}\mathbf{B}^T + \mathbf{B}\boldsymbol{\Sigma}_{YX}\mathbf{A}^T + \mathbf{B}\boldsymbol{\Sigma}_{YY}\mathbf{B}^T$.*

2.2.5 Náhodný proces

Náhodným (nebo také stochastickým) **procesem** se rozumí množina $\{X(t), t \in T\}$, jejíž prvky $X(t)$ jsou náhodné veličiny. Prvky množiny $T \subset \mathbb{R}$ obvykle reprezentují čas, přičemž čas může být chápán spojitě či diskrétně (v diskrétním případě se někdy místo o náhodném procesu hovoří o náhodné posloupnosti). Na náhodný proces lze pohlížet také jako na funkci, jejímiž hodnotami jsou náhodné veličiny.

Náhodný proces může být obecně n -rozměrný. V souvislosti s náhodným procesem lze podobně zadefinovat a použít i předchozí uvedené pojmy a charakteristiky (viz např. [33, kapitola 4]). Gaussovský náhodný proces je množina náhodných veličin $X(t)$, které mají sdružené normální rozdělení ($t \in F$, kde $F \subset T$ je konečnou podmnožinou množiny T).

Navíc budiž zavedeny pojmy nekorelovanosti procesu v čase a vzájemné nekorelovanosti dvou procesů.

Náhodný proces $\{\mathbf{X}(t)\}$ (obecně n -rozměrný) je nekorelovaný (v čase), jestliže pro všechna $t_i, t_j \in T, t_i \neq t_j$, platí

$$\mathbf{E}\langle(\mathbf{X}(t_i) - \mathbf{E}\langle\mathbf{X}(t_i)\rangle)(\mathbf{X}(t_j) - \mathbf{E}\langle\mathbf{X}(t_j)\rangle)^T\rangle = \mathbf{0}.$$

Náhodné procesy $\{\mathbf{X}(t)\}$ a $\{\mathbf{Y}(t)\}$ jsou vzájemně nekorelované, platí-li pro všechna $t_i, t_j \in T$

$$\mathbf{E}\langle(\mathbf{X}(t_i) - \mathbf{E}\langle\mathbf{X}(t_i)\rangle)(\mathbf{Y}(t_j) - \mathbf{E}\langle\mathbf{Y}(t_j)\rangle)^T\rangle = \mathbf{0}.$$

Jinými pojmy jsou nezávislost procesu v čase a vzájemná nezávislost procesů. Nezávislost implikuje nekorelovanost, obrácená implikace obecně neplatí, platí pouze v případě

gaussovských náhodných procesů, které však budou v práci v popředí zájmu (oba pojmy tak často budou používány ekvivalentně). Budiž ještě poznamenáno, že proces nezávislý v čase bývá nazýván „bílým“.

2.3 Dynamické systémy a jejich modely

Kalmanův filtr je úzce spojen s pojmem lineární dynamický systém. Algoritmus Kalmanova filtru umožňuje odhadnout stav takového systému v určitém čase za využití jeho modelu a naměřených dat. Proto je sestavení příslušného modelu nutným prvním krokem předcházejícím vlastní implementaci algoritmu.

2.3.1 Deterministické dynamické systémy

Systém je komplex vazbami propojených a interagujících prvků, na který lze nahlížet jako na celek. Pokud se atributy systému, které jsou předmětem sledování, mění v čase, jedná se o systém dynamický. Vývoj dynamického systému v čase se nazývá proces. Hodnoty sledovaných atributů v určitém časovém okamžiku charakterizují stav dynamického systému.

Již od dob Newtona a Leibnize slouží jako přirozený nástroj popisu dynamických systémů diferenciální rovnice. Pro potřeby Kalmanova filtru bude využit tzv. stavový model systému (*state-space model*). Při použití faktu, že libovolnou diferenciální rovnici vyššího řádu (či jejich soustavu) lze ekvivalentně vyjádřit v podobě soustavy diferenciálních rovnic prvního řádu, je možné dynamiku systému obecně popsat pomocí následující soustavy stavových rovnic

$$\begin{aligned}\dot{x}_1(t) &= f_1(t, x_1(t), x_2(t), x_3(t), \dots, x_n(t), u_1(t), u_2(t), u_3(t), \dots, u_r(t)) \\ \dot{x}_2(t) &= f_2(t, x_1(t), x_2(t), x_3(t), \dots, x_n(t), u_1(t), u_2(t), u_3(t), \dots, u_r(t)) \\ \dot{x}_3(t) &= f_3(t, x_1(t), x_2(t), x_3(t), \dots, x_n(t), u_1(t), u_2(t), u_3(t), \dots, u_r(t)) \\ &\vdots \\ \dot{x}_n(t) &= f_n(t, x_1(t), x_2(t), x_3(t), \dots, x_n(t), u_1(t), u_2(t), u_3(t), \dots, u_r(t)),\end{aligned}$$

kde t je nezávisle proměnná (čas), $x_i(t)$, $i = 1, \dots, n$ jsou závisle proměnné, přičemž symbol „tečky“ označuje jejich derivace podle času $\dot{x}_i(t) = \frac{d}{dt}x_i(t)$, a $u_i(t)$, $i = 1, \dots, r$ jsou (známé) vstupy systému (někdy též označované jako řízení).

Zkráceně lze uvedenou soustavu zapsat vektorovým zápisem

$$\dot{\mathbf{x}}(t) = \frac{d}{dt}\mathbf{x}(t) = f(t, \mathbf{x}(t), \mathbf{u}(t)).$$

Vektor $\mathbf{x}(t)$ se nazývá stavový vektor, prvky tohoto vektoru $x_{[i]}(t) = x_i(t)$ jsou stavové proměnné, které popisují stav systému v čase t , mimoto může stavový vektor obsahovat také derivace stavových proměnných.

Je-li funkce f v předchozím zápisu lineární vzhledem ke svým proměnným, jedná se o lineární dynamický systém, v opačném případě o nelineární dynamický systém. Jak již bylo zmíněno, Kalmanův filtr (ve své základní podobě) předpokládá lineární dynamický systém, tedy systém, jehož dynamiku lze popsat pomocí soustavy lineárních diferenciálních rovnic prvního řádu, kterou lze reprezentovat maticovým zápisem

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{C}(t)\mathbf{u}(t). \quad (2.1)$$

Matice $\mathbf{F}(t)$ se nazývá matice koeficientů dynamiky, matice $\mathbf{C}(t)$ je maticí popisující vliv vstupů systému (vektoru $\mathbf{u}(t)$). Vektor $\mathbf{x}(t)$ má rozměr n a vektor $\mathbf{u}(t)$ má rozměr r , matice $\mathbf{F}(t)$ je pak typu $n \times n$ a matice $\mathbf{C}(t)$ typu $n \times r$.

Rovnice (2.1), označovaná jako stavová rovnice (*state-space equation*), modeluje dynamiku lineárního systému, samotné stavové proměnné $x_i(t)$ však v praxi obvykle nejsou měřitelné a jejich hodnoty musejí být odvozovány. K tomu jsou využívány výstupy systému $z_i(t)$, $i = 1, \dots, l$, neboli hodnoty získané měřením například za využití různých senzorů. Vztah (lineární) těchto hodnot ke stavovým proměnným a vstupům systému lze popsat následující soustavou lineárních rovnic

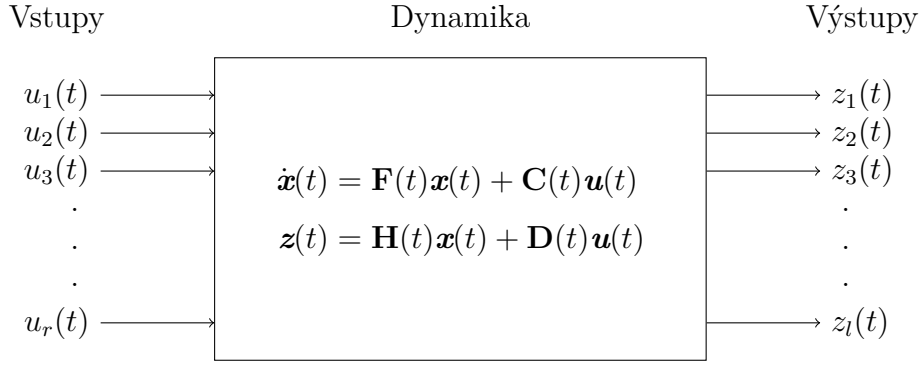
$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t). \quad (2.2)$$

Vektor $\mathbf{z}(t)$ se nazývá vektor výstupů systému, vektor měření či vektor pozorování, matice $\mathbf{H}(t)$ vyjadřuje citlivost měření a matice $\mathbf{D}(t)$ reprezentuje vliv vstupů systému na naměřené výstupy. Vektor $\mathbf{z}(t)$ má rozměr l , matice $\mathbf{H}(t)$ je typu $l \times n$ a matice $\mathbf{D}(t)$ typu $l \times r$. Rovnice (2.2) bývá označována jako rovnice měření (*measurement equation*).

Rovnice (2.1) a (2.2) společně vytvářejí model deterministického lineárního dynamického systému ve spojitém čase. Takovýto systém je znázorněn na obrázku 1.

Předchozí rovnice jsou modelem lineárního dynamického systému ve spojitém čase, neboť nezávisle proměnná t spojitě nabývá hodnot nějakého intervalu $t \in [t_0, t_f]$, $t_0 < t_f$ (t_0 je počáteční čas, t_f koncový čas). V řadě praktických úloh je však stav systému předmětem zájmu pouze v konkrétní množině diskrétních časových okamžiků $t \in \{t_0, t_1, t_2, \dots\}$. Tyto okamžiky mohou odpovídat například časům, ve kterých jsou měřeny výstupy systému. Pro tyto úlohy lze vhodně rekurentně vyjádřit stav dynamického systému (v obecném případě) jako funkci stavu předcházejícího

$$\mathbf{x}(t_k) = f(\mathbf{x}(t_{k-1}), t_{k-1}, t_k), \quad k = 1, 2, 3, \dots$$



Obrázek 1: Schematické znázornění deterministického lineárního dynamického systému ve spojitém čase (zpracováno podle [18])

K modelu lineárního dynamického systému popsanému rovnicemi (2.1) a (2.2) lze potom odvodit (více např. [18, s. 53]) následující odpovídající diskrétní (diskretizovaný) model

$$\begin{aligned}\mathbf{x}(t_k) &= \mathbf{\Phi}(t_k, t_{k-1})\mathbf{x}(t_{k-1}) + \mathbf{\Gamma}(t_{k-1})\mathbf{u}(t_{k-1}) \\ \mathbf{z}(t_k) &= \mathbf{H}(t_k)\mathbf{x}(t_k) + \mathbf{D}(t_k)\mathbf{u}(t_k).\end{aligned}$$

Matice $\mathbf{\Phi}(t_k, t_{k-1})$ je stavová matice přechodu a matice $\mathbf{\Gamma}(t_{k-1})$ je odpovídající matice popisující vliv vstupů systému.

Při zavedení zjednodušeného značení $\mathbf{x}_k = \mathbf{x}(t_k)$, $\mathbf{\Phi}_{k-1} = \mathbf{\Phi}(t_k, t_{k-1})$ atd. je možné předchozí dvě rovnice zapsat takto (toto značení bude v textu práce preferováno)

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{\Gamma}_{k-1}\mathbf{u}_{k-1} \quad (2.3)$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{D}_k\mathbf{u}_k. \quad (2.4)$$

V prozatím uvedených modelech je předpokládáno, že matice \mathbf{F} a \mathbf{C} resp. $\mathbf{\Phi}$ a $\mathbf{\Gamma}$ závisejí na t resp. k . Takovýmto modelům se říká v čase proměnlivé. V opačném případě se jedná o modely v čase neměnné (invariantní).

Podrobnější pojednání o diferenciálních rovnicích (a v diskrétním případě o diferenčních rovnicích), jejich řešení a vztahu ke Kalmanově filtru nabízí například [18, kapitola 2].

Budiž však poznamenáno, že existují i dynamické systémy, které nelze vhodně modelovat pomocí diferenciálních (resp. diferenčních) rovnic, tyto však v práci nebudou uvažovány.

2.3.2 Stochastické dynamické systémy

Doposud bylo předpokládáno, že současný stav systému a vstupy jednoznačně určují budoucí stav systému, tedy že uvedený model, reprezentovaný například rovnicí (2.1) v případě lineárního dynamického systému ve spojitém čase, přesně a úplně popisuje dynamiku daného systému. Tento předpoklad je charakteristický pro deterministické systémy, pro řadu reálných systémů však nemusí být vhodný a dostačující. Dynamiku reálných systémů ovlivňuje množství vlivů, které nejsou pod ničí kontrolou a které není možné (a ani praktické) modelovat deterministicky. Navíc ani senzory neposkytují o systému naprosto přesná a úplná data, ale naopak jsou naměřené údaje vždy zatíženy určitou chybou. Pro tyto účely jsou vhodné stochastické modely, které chápou dynamiku systému jako náhodný proces a pomocí statistických vlastností reprezentují znalosti o systému.

S využitím stochastických diferenciálních rovnic a stochastického kalkulu lze odvodit (více např. [18, kapitola 3]) následující model stochastického lineárního dynamického systému ve spojitém čase

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{F}(t)\mathbf{x}(t) + \mathbf{C}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{H}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) + \mathbf{v}(t),\end{aligned}$$

kde navíc oproti deterministické variantě jsou $\{\mathbf{w}(t)\}$ a $\{\mathbf{v}(t)\}$ nekorelované gaussovské procesy s nulovými středními hodnotami a kovariančními maticemi $\mathbf{Q}(t)$ resp. $\mathbf{R}(t)$ v čase t (procesy mají vlastnosti gaussovského bílého šumu), neboli platí tyto předpoklady

$$\begin{aligned}E\langle \mathbf{w}(t) \rangle &= \mathbf{0}, \\ E\langle \mathbf{v}(t) \rangle &= \mathbf{0}, \\ E\langle \mathbf{w}(t_1)\mathbf{w}^T(t_2) \rangle &= \mathbf{Q}(t_1)\delta(t_2 - t_1), \\ E\langle \mathbf{v}(t_1)\mathbf{v}^T(t_2) \rangle &= \mathbf{R}(t_1)\delta(t_2 - t_1), \\ \delta(t) &= \begin{cases} 1, & t = 0, \\ 0, & t \neq 0. \end{cases}\end{aligned}$$

Symbol $\delta(t)$ značí Diracovo delta. Matice $\mathbf{Q}(t)$ a $\mathbf{R}(t)$ jsou matice typu $m \times m$ a $l \times l$. Vektor šumu $\mathbf{w}(t)$ (nazývaný systémový šum) má rozměr m a reprezentuje náhodné vlivy působící na dynamiku systému a neurčitost jejího modelu. Vektor šumu $\mathbf{v}(t)$ (označovaný jako šum měření) je rozměru l a vyjadřuje chyby v naměřených hodnotách. Matice $\mathbf{G}(t)$ typu $n \times m$ popisuje vliv šumu $\mathbf{w}(t)$ v modelu.

Dále lze také odvodit odpovídající diskrétní model stochastického lineárního dynamického systému

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1} \mathbf{x}_{k-1} + \mathbf{\Gamma}_{k-1} \mathbf{u}_{k-1} + \mathbf{G}_{k-1} \mathbf{w}_{k-1} \quad (2.5)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{D}_k \mathbf{u}_k + \mathbf{v}_k, \quad (2.6)$$

s předpoklady

$$\mathbb{E}\langle \mathbf{w}_k \rangle = \mathbf{0},$$

$$\mathbb{E}\langle \mathbf{v}_k \rangle = \mathbf{0},$$

$$\mathbb{E}\langle \mathbf{w}_{k_1} \mathbf{w}_{k_2}^T \rangle = \mathbf{Q}_{k_1} \Delta(k_2 - k_1),$$

$$\mathbb{E}\langle \mathbf{v}_{k_1} \mathbf{v}_{k_2}^T \rangle = \mathbf{R}_{k_1} \Delta(k_2 - k_1),$$

$$\Delta(k) = \begin{cases} 1, & k = 0, \\ 0, & k \neq 0, \end{cases}$$

symbol Δ značí Kroneckerovo delta.

Na závěr budiž poznamenáno, že někdy se obecněji povoluje i nesplnění některého z uvedených předpokladů či se uvažují mírně odlišné podoby rovnic modelu například nezahrnující vstupy systému do rovnice měření apod. Kromě modelů ve spojitém a v diskrétním čase existují také modely kombinované, které obvykle popisují dynamiku systému spojitě, zatímco měření jsou vnímána jako diskrétní.

3 Kalmanova filtrace

Kapitola se zabývá komplexním představením Kalmanova filtru, vysvětluje jeho základní myšlenku a uvádí odvození algoritmu, které je následně doplněno popisem vlastností, diskuzí předpokladů a poznámkami k praktické implementaci. Princip fungování Kalmanova filtru je ukázán na jednoduchém ilustračním příkladu. Druhá část kapitoly se věnuje algoritmům rozšiřujícím standardní Kalmanův filtr, přičemž pozornost je zaměřena zejména na řešení úlohy nelineární filtrace. Hlavními zdroji pro tuto kapitolu jsou [6], [8], [11], [18], [23], [31], [33], [39] a mnohé další.

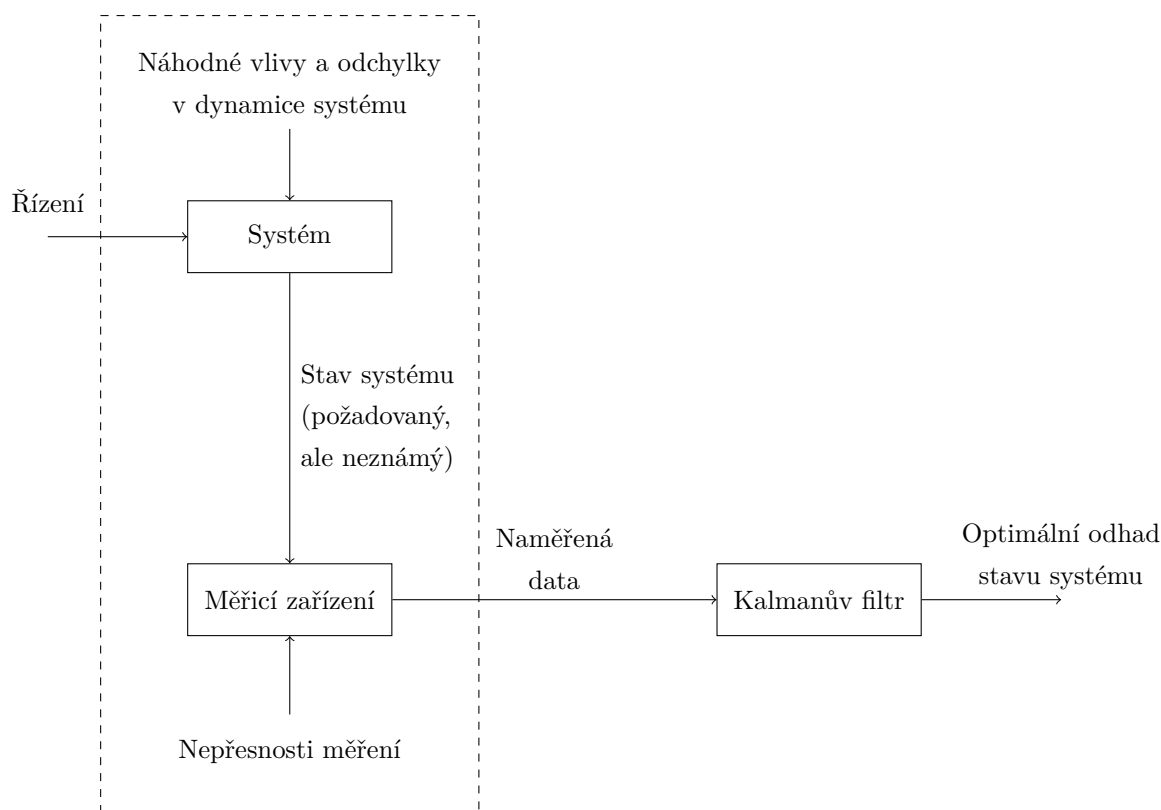
3.1 Kalmanův filtr

Kalmanův filtr je nástroj umožňující odhadnout stav stochastického lineárního dynamického systému za využití naměřených dat zatížených šumem. Odhad, který Kalmanův filtr produkuje, je v jistém smyslu statisticky optimální (viz dále). Princip použití Kalmanova filtru znázorňuje obrázek 2.

Kalmanův filtr pracuje se všemi informacemi, které jsou k dispozici, tj. k odhadu stavu systému využívá všechna dostupná naměřená data, model systému spolu se statistickým popisem jeho nepřesností, šumu a chyb měření a dále informace o počátečních podmínkách a počátečním stavu systému.

3.1.1 Představení a historie Kalmanova filtru

Filtr nese jméno Rudolfa Emila Kalmana (maď. Rudolf Emil Kálmán; 1930 – 2016), maďarského matematika žijícího ve Spojených státech amerických, který jej publikoval v roce 1960 v článku [23]. Jak uvádí [18], psal se rok 1958 a Kalman se již nějakou dobu věnoval výzkumu v oblasti teorie řízení. Jednoho listopadového večera se vracel vlakem do Baltimoru z návštěvy Princetonu, bylo asi jedenáct hodin a vlak musel na hodinu



Obrázek 2: Schéma použití Kalmanova filtru (zpracováno podle [33])

zastavit těsně před Baltimorem. Jak tak seděl uvězněný ve vlaku, napadla ho myšlenka: *Proč nepoužít pojem stavových proměnných při řešení Wiener-Kolmogorovova filtračního problému?* (V té době byly při řešení filtračního problému preferovány metody pracující ve frekvenční oblasti a nikoli přístupy využívající časovou oblast a stavové modely jako Kalmanův filtr, viz [18, s. 12].) Bylo již pozdě a Kalman byl příliš unavený na to, aby se myšlenkou toho večera více zabýval. Jednalo se ale o prvopočátek k odvození Kalmanova filtru.

Kalmanova myšlenka se však nejprve setkala se skeptickým přístupem jeho kolegů, a tak se Kalman rozhodl článek s výsledky své práce publikovat v časopise zaměřeném na strojírenství a nikoli v takovém z oblasti řízení systémů či zpracování signálu. Protože, jak řekl, *„When you fear stepping on hallowed ground with entrenched interests, it is best to go sideways.“* Druhý článek týkající se filtru ve spojitém čase, na kterém se jako spoluautor podílel Richard S. Bucy, byl nejprve časopisem odmítnut, neboť jeden z kroků důkazu prý nemohl být správný (ve skutečnosti však správný byl). Kalman přesto nevzdal své snažení a postupem času začal být o výsledky jeho práce čím dál větší zájem. Jako příklad budiž uvedeno, že Kalmanův filtr byl využit ve vesmírném projektu Apollo a v mnohých dalších projektech z oblasti navigace i jiných.

3.1.2 Algoritmus Kalmanova filtru a jeho odvození

Nyní již bude odvozen vlastní algoritmus Kalmanova filtru, k čemuž bude využit tzv. princip ortogonality. Tento přístup použil Kalman ve svém původním článku [23]. Postupem času byly dalšími autory k odvození využity i jiné přístupy založené například na tzv. inovacích či bayesovské statistice, jejich ukázkou lze nalézt například v [6], [33]. Zde uvedené odvození Kalmanova filtru je zpracováno podle [18], jiné postupy mohou vyžadovat mírně odlišné předpoklady.

Budiž uvažován model stochastického lineárního dynamického systému v diskrétním čase (představený v části 2.3.2), navíc pro zjednodušení budiž předpokládáno, že tento systém nemá žádné vstupy ($\mathbf{u}_k = \mathbf{0}$ pro všechna t_k), tj.

$$\mathbf{x}_k = \Phi_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1}, \quad (3.1)$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \quad (3.2)$$

$\{\mathbf{w}_k\}$ a $\{\mathbf{v}_k\}$ jsou nekorelované gaussovské procesy šumu s nulovými středními hodnotami a kovariančními maticemi \mathbf{Q}_k resp. \mathbf{R}_k v čase t_k . Dále necht \mathbf{x}_0 je náhodná veličina s normálním rozdělením se známou střední hodnotou \mathbf{x}_0 a známou kovarianční maticí \mathbf{P}_0 . Navíc necht jsou \mathbf{x}_0 a oba šumy vždy vzájemně nekorelované. Tedy souhrnně pro všechna t_k platí

$$\begin{aligned} \mathbb{E}\langle \mathbf{w}_k \rangle &= \mathbf{0}, \\ \mathbb{E}\langle \mathbf{v}_k \rangle &= \mathbf{0}, \\ \mathbb{E}\langle \mathbf{w}_{k_1} \mathbf{w}_{k_2}^T \rangle &= \mathbf{Q}_{k_1} \Delta(k_2 - k_1), \\ \mathbb{E}\langle \mathbf{v}_{k_1} \mathbf{v}_{k_2}^T \rangle &= \mathbf{R}_{k_1} \Delta(k_2 - k_1), \\ \mathbb{E}\langle \mathbf{w}_{k_1} \mathbf{v}_{k_2}^T \rangle &= \mathbf{0}, \\ \mathbb{E}\langle \mathbf{x}_0 \mathbf{w}_k^T \rangle &= \mathbf{0}, \\ \mathbb{E}\langle \mathbf{x}_0 \mathbf{v}_k^T \rangle &= \mathbf{0}. \end{aligned}$$

Pro účely odvození budiž ještě předpokládáno, že vektor \mathbf{w}_{k-1} je vždy zkombinován s maticí \mathbf{G}_{k-1} , neboli že stavová rovnice modelu má tvar

$$\mathbf{x}_k = \Phi_{k-1}\mathbf{x}_{k-1} + \acute{\mathbf{w}}_{k-1}, \quad (3.3)$$

kde $\acute{\mathbf{w}}_{k-1} = \mathbf{G}_{k-1}\mathbf{w}_{k-1}$ a $\acute{\mathbf{Q}}_{k-1} = \mathbb{E}\langle \acute{\mathbf{w}}_{k-1} \acute{\mathbf{w}}_{k-1}^T \rangle = \mathbf{G}_{k-1}\mathbb{E}\langle \mathbf{w}_{k-1} \mathbf{w}_{k-1}^T \rangle \mathbf{G}_{k-1}^T = \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T$.

Cílem je nalézt odhad stavového vektoru \mathbf{x}_k v čase t_k , značený $\hat{\mathbf{x}}_k$, a to tak, aby byla minimalizována vážená střední kvadratická chyba (*weighted mean square error*, *weighted*

MSE)

$$\mathbf{E}\langle(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{M}(\mathbf{x}_k - \hat{\mathbf{x}}_k)\rangle,$$

kde \mathbf{M} je pozitivně semidefinitní váhová matice.

Algoritmus Kalmanova filtru je rekurzivní, výpočet v obecném čase t_k se skládá ze dvou hlavních kroků – nejprve je vypočítán apriorní odhad $\hat{\mathbf{x}}_{k(-)}$ v čase t_k dosazením aposteriorního odhadu z času t_{k-1} do deterministické části stavové rovnice modelu a následně je tento odhad upraven za využití měření pořízeného v čase t_k , čímž je získán aposteriorní odhad $\hat{\mathbf{x}}_{k(+)}$ v čase t_k .

Pro apriorní odhad $\hat{\mathbf{x}}_{k(-)}$ lze tedy psát jednoduchý vztah

$$\hat{\mathbf{x}}_{k(-)} = \Phi_{k-1} \hat{\mathbf{x}}_{k-1(+)} \quad (3.4)$$

Odvození vztahu pro aposteriorní odhad $\hat{\mathbf{x}}_{k(+)}$ tak, aby byl odhadem optimálním (tj. minimalizoval střední kvadratickou chybu dle úvodního požadavku), je poněkud obtížnější. Pokud bude předpokládáno, že \mathbf{x} a \mathbf{z} mají sdružené normální rozdělení, je možné vyjádřit $\hat{\mathbf{x}}_{k(+)}$ jako lineární kombinaci apriorního odhadu a měření (důkaz viz [18, s. 115])

$$\hat{\mathbf{x}}_{k(+)} = \mathbf{K}_k^1 \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k \mathbf{z}_k \quad (3.5)$$

Matice \mathbf{K}_k^1 a \mathbf{K}_k jsou zatím neznámé, snahou je určit jejich hodnoty tak, aby $\hat{\mathbf{x}}_{k(+)}$ splňoval princip ortogonality. Tento princip zjednodušeně říká, že odhad $\hat{\mathbf{x}}$ (který je lineární funkcí \mathbf{z}) minimalizuje střední kvadratickou chybu právě tehdy, když je chyba odhadu $(\mathbf{x} - \hat{\mathbf{x}})$ kolmá na \mathbf{z} (podrobněji viz [18, s. 116]), neboli

$$\mathbf{E}\langle[\mathbf{x}_k - \hat{\mathbf{x}}_{k(+)}] \mathbf{z}_i^T\rangle = \mathbf{0}, \quad i = 1, 2, \dots, k-1, \quad (3.6)$$

$$\mathbf{E}\langle[\mathbf{x}_k - \hat{\mathbf{x}}_{k(+)}] \mathbf{z}_k^T\rangle = \mathbf{0}. \quad (3.7)$$

Do rovnice (3.6) lze dosadit vztah (3.3) za \mathbf{x}_k a (3.5) za $\hat{\mathbf{x}}_{k(+)}$. Dále je z rovnic (3.3) a (3.2) patrné, že naměřená data $\mathbf{z}_1, \dots, \mathbf{z}_k$ neobsahují šum $\hat{\mathbf{w}}_k$, a jelikož $\hat{\mathbf{w}}_k$ a \mathbf{v}_k jsou nekorelované, lze odvodit, že $\mathbf{E}\langle\hat{\mathbf{w}}_k \mathbf{z}_i^T\rangle = \mathbf{0}$, pro $1 \leq i \leq k$. Rovnici (3.6) je tedy možné přepsat takto

$$\mathbf{E}\langle[\Phi_{k-1} \mathbf{x}_{k-1} - \mathbf{K}_k^1 \hat{\mathbf{x}}_{k(-)} - \mathbf{K}_k \mathbf{z}_k] \mathbf{z}_i^T\rangle = \mathbf{0}, \quad i = 1, 2, \dots, k-1. \quad (3.8)$$

Dosazením za \mathbf{z}_k dle (3.2) je předchozí rovnice upravena

$$\mathbf{E}\langle[\Phi_{k-1} \mathbf{x}_{k-1} - \mathbf{K}_k^1 \hat{\mathbf{x}}_{k(-)} - \mathbf{K}_k \mathbf{H}_k \mathbf{x}_k - \mathbf{K}_k \mathbf{v}_k] \mathbf{z}_i^T\rangle = \mathbf{0}, \quad i = 1, 2, \dots, k-1. \quad (3.9)$$

Princip ortogonality (rovnice (3.6) a (3.7)) platí také pro apriorní odhad $\hat{\mathbf{x}}_{k(-)}$, tedy

$$\mathbf{E}\langle[\mathbf{x}_k - \hat{\mathbf{x}}_{k(-)}] \mathbf{z}_i^T\rangle = \mathbf{0}, \quad i = 1, 2, \dots, k-1,$$

a dále platí

$$\mathbf{E}\langle \mathbf{v}_k \mathbf{z}_i^T \rangle = \mathbf{0}, \quad i = 1, 2, \dots, k-1.$$

S využitím těchto faktů a vlastností střední hodnoty lze rovnici (3.9) postupně upravit následujícím způsobem

$$\begin{aligned} \Phi_{k-1} \mathbf{E}\langle \mathbf{x}_{k-1} \mathbf{z}_i^T \rangle - \mathbf{K}_k^1 \mathbf{E}\langle \hat{\mathbf{x}}_{k(-)} \mathbf{z}_i^T \rangle - \mathbf{K}_k \mathbf{H}_k \Phi_{k-1} \mathbf{E}\langle \mathbf{x}_{k-1} \mathbf{z}_i^T \rangle - \mathbf{K}_k \mathbf{E}\langle \mathbf{v}_k \mathbf{z}_i^T \rangle &= \mathbf{0}, \\ \Phi_{k-1} \mathbf{E}\langle \mathbf{x}_{k-1} \mathbf{z}_i^T \rangle - \mathbf{K}_k^1 \mathbf{E}\langle \hat{\mathbf{x}}_{k(-)} \mathbf{z}_i^T \rangle - \mathbf{K}_k \mathbf{H}_k \Phi_{k-1} \mathbf{E}\langle \mathbf{x}_{k-1} \mathbf{z}_i^T \rangle &= \mathbf{0}, \\ \mathbf{E}\langle [(\mathbf{x}_k - \mathbf{K}_k \mathbf{H}_k \mathbf{x}_k - \mathbf{K}_k^1 \mathbf{x}_k) - \mathbf{K}_k^1 (\hat{\mathbf{x}}_{k(-)} - \mathbf{x}_k)] \mathbf{z}_i^T \rangle &= \mathbf{0}, \\ (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k - \mathbf{K}_k^1) \mathbf{E}\langle \mathbf{x}_k \mathbf{z}_i^T \rangle &= \mathbf{0}. \end{aligned} \quad (3.10)$$

\mathbf{I} značí jednotkovou matici. Rovnost (3.10) bude splněna pro libovolné \mathbf{x}_k , pokud

$$\mathbf{K}_k^1 = \mathbf{I} - \mathbf{K}_k \mathbf{H}_k. \quad (3.11)$$

Nyní je třeba nalézt matici \mathbf{K}_k tak, aby byla splněna i druhá rovnice principu ortogonality (3.7).

Budiž zavedeno následující značení chyb

$$\tilde{\mathbf{x}}_{k(+)} = \hat{\mathbf{x}}_{k(+)} - \mathbf{x}_k, \quad (3.12)$$

$$\tilde{\mathbf{x}}_{k(-)} = \hat{\mathbf{x}}_{k(-)} - \mathbf{x}_k, \quad (3.13)$$

$$\tilde{\mathbf{z}}_k = \hat{\mathbf{z}}_{k(-)} - \mathbf{z}_k = \mathbf{H}_k \hat{\mathbf{x}}_{k(-)} - \mathbf{z}_k. \quad (3.14)$$

Rovnici (3.7) lze psát také pro $\hat{\mathbf{z}}_{k(-)}$, tj.

$$\mathbf{E}\langle [\mathbf{x}_k - \hat{\mathbf{x}}_{k(+)}] \hat{\mathbf{z}}_{k(-)}^T \rangle = \mathbf{0}, \quad (3.15)$$

a dále je odečtením rovnice (3.7) od (3.15) získáno

$$\mathbf{E}\langle [\mathbf{x}_k - \hat{\mathbf{x}}_{k(+)}] \tilde{\mathbf{z}}_k^T \rangle = \mathbf{0}. \quad (3.16)$$

Do rovnice (3.16) je možné dosadit za \mathbf{x}_k , $\hat{\mathbf{x}}_{k(+)}$ a $\tilde{\mathbf{z}}_k^T$ vztahy (3.3), (3.5) a (3.14), navíc platí $\mathbf{E}\langle \hat{\mathbf{w}}_k \mathbf{z}_k^T \rangle = \mathbf{E}\langle \hat{\mathbf{w}}_k \hat{\mathbf{x}}_{k(-)}^T \rangle = \mathbf{0}$, tedy

$$\mathbf{E}\langle [\Phi_{k-1} \mathbf{x}_{k-1} - \mathbf{K}_k^1 \hat{\mathbf{x}}_{k(-)} - \mathbf{K}_k \mathbf{z}_k] [\mathbf{H}_k \hat{\mathbf{x}}_{k(-)} - \mathbf{z}_k]^T \rangle = \mathbf{0}. \quad (3.17)$$

Dále lze rovnici (3.17) upravit dosazením vztahu (3.11) za \mathbf{K}_k^1 a (3.2) za \mathbf{z}_k a následně nahrazením $(\hat{\mathbf{x}}_{k(-)} - \mathbf{x}_k)$ dle zavedeného značení za $\tilde{\mathbf{x}}_{k(-)}$

$$\begin{aligned} \mathbf{E}\langle [\Phi_{k-1} \mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k \mathbf{H}_k \hat{\mathbf{x}}_{k(-)} - \mathbf{K}_k \mathbf{H}_k \mathbf{x}_k - \mathbf{K}_k \mathbf{v}_k] [\mathbf{H}_k \hat{\mathbf{x}}_{k(-)} - \mathbf{H}_k \mathbf{x}_k - \mathbf{v}_k]^T \rangle &= \mathbf{0}, \\ \mathbf{E}\langle [(\mathbf{x}_k - \hat{\mathbf{x}}_{k(-)}) + \mathbf{K}_k \mathbf{H}_k (\hat{\mathbf{x}}_{k(-)} - \mathbf{x}_k) - \mathbf{K}_k \mathbf{v}_k] [\mathbf{H}_k (\hat{\mathbf{x}}_{k(-)} - \mathbf{x}_k) - \mathbf{v}_k]^T \rangle &= \mathbf{0}, \\ \mathbf{E}\langle [-\tilde{\mathbf{x}}_{k(-)} + \mathbf{K}_k \mathbf{H}_k \tilde{\mathbf{x}}_{k(-)} - \mathbf{K}_k \mathbf{v}_k] [\mathbf{H}_k \tilde{\mathbf{x}}_{k(-)} - \mathbf{v}_k]^T \rangle &= \mathbf{0}. \end{aligned} \quad (3.18)$$

Dle definice je apriorní chybová kovarianční matice

$$\mathbf{P}_{k(-)} = \mathbb{E}\langle \tilde{\mathbf{x}}_{k(-)} \tilde{\mathbf{x}}_{k(-)}^T \rangle.$$

Nahrazením za $\mathbf{P}_{k(-)}$ dle předchozího vztahu a využitím faktu, že $\mathbb{E}\langle \tilde{\mathbf{x}}_{k(-)} \mathbf{v}_k^T \rangle = \mathbf{0}$ vede rovnice (3.18) postupnými úpravami na

$$[\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_{k(-)} \mathbf{H}_k^T - \mathbf{K}_k \mathbf{R}_k = \mathbf{0}.$$

Matici \mathbf{K}_k , nazývanou jako Kalmanův zisk (*Kalman gain*), lze pak získat vyřešením výše uvedené maticové rovnice, výsledkem je

$$\mathbf{K}_k = \mathbf{P}_{k(-)} \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_{k(-)} \mathbf{H}_k^T + \mathbf{R}_k]^{-1}. \quad (3.19)$$

Další část odvození spočívá v nalezení vztahu pro aposteriorní chybovou kovarianční matici, která je definována jako

$$\mathbf{P}_{k(+)} = \mathbb{E}\langle \tilde{\mathbf{x}}_{k(+)} \tilde{\mathbf{x}}_{k(+)}^T \rangle. \quad (3.20)$$

Dosazením (3.11) do (3.5) se získává

$$\begin{aligned} \hat{\mathbf{x}}_{k(+)} &= [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k \mathbf{z}_k, \\ \hat{\mathbf{x}}_{k(+)} &= \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k [\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k(-)}]. \end{aligned} \quad (3.21)$$

Dále je možné do rovnice (3.21) dosadit za \mathbf{z}_k a od obou stran odečíst \mathbf{x}_k

$$\begin{aligned} \hat{\mathbf{x}}_{k(+)} - \mathbf{x}_k &= \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k \mathbf{H}_k \mathbf{x}_k + \mathbf{K}_k \mathbf{v}_k - \mathbf{K}_k \mathbf{H}_k \hat{\mathbf{x}}_{k(-)} - \mathbf{x}_k, \\ \tilde{\mathbf{x}}_{k(+)} &= \tilde{\mathbf{x}}_{k(-)} - \mathbf{K}_k \mathbf{H}_k \tilde{\mathbf{x}}_{k(-)} + \mathbf{K}_k \mathbf{v}_k, \\ \tilde{\mathbf{x}}_{k(+)} &= [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \tilde{\mathbf{x}}_{k(-)} + \mathbf{K}_k \mathbf{v}_k. \end{aligned} \quad (3.22)$$

Dosazením takto upraveného vztahu (3.22) do rovnice (3.20) a využitím již zmíněného faktu, že $\mathbb{E}\langle \tilde{\mathbf{x}}_{k(-)} \mathbf{v}_k^T \rangle = \mathbf{0}$, konečně vyplývá

$$\begin{aligned} \mathbf{P}_{k(+)} &= \mathbb{E}\langle [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \tilde{\mathbf{x}}_{k(-)} \tilde{\mathbf{x}}_{k(-)}^T [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k]^T + \mathbf{K}_k \mathbf{v}_k \mathbf{v}_k^T \mathbf{K}_k^T \rangle, \\ \mathbf{P}_{k(+)} &= [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_{k(-)} [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k]^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T. \end{aligned} \quad (3.23)$$

Rovnice (3.23) je tzv. Josephův tvar vztahu pro aposteriorní chybovou kovarianční matici. Roznásobením a uplatněním vzorce (3.19) pro \mathbf{K}_k , ze kterého plyne, že $\mathbf{K}_k [\mathbf{H}_k \mathbf{P}_{k(-)} \mathbf{H}_k^T + \mathbf{R}_k] = \mathbf{P}_{k(-)} \mathbf{H}_k^T$, lze odvodit další tvar tohoto vztahu

$$\begin{aligned} \mathbf{P}_{k(+)} &= \mathbf{P}_{k(-)} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k(-)} - \mathbf{P}_{k(-)} \mathbf{H}_k^T \mathbf{K}_k^T + \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k(-)} \mathbf{H}_k^T \mathbf{K}_k^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T, \\ \mathbf{P}_{k(+)} &= [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_{k(-)} - \mathbf{P}_{k(-)} \mathbf{H}_k^T \mathbf{K}_k^T + \mathbf{K}_k [\mathbf{H}_k \mathbf{P}_{k(-)} \mathbf{H}_k^T + \mathbf{R}_k] \mathbf{K}_k^T, \\ \mathbf{P}_{k(+)} &= [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_{k(-)}. \end{aligned} \quad (3.24)$$

Tvar (3.24) vyžaduje nižší výpočetní nároky, oproti tomu Josephův tvar (3.23) bývá obecně výhodnější z hlediska numerické stability.

Poslední částí odvození je určení vztahu pro apriorní chybovou kovarianční matici, kterou je vhodné vyjádřit jako funkci aposteriorní chybové kovarianční matice z předchozího časového okamžiku. Dle definice je

$$\mathbf{P}_{k(-)} = \mathbf{E}\langle \tilde{\mathbf{x}}_{k(-)} \tilde{\mathbf{x}}_{k(-)}^T \rangle. \quad (3.25)$$

Budiž využito rovnice (3.4) pro $\hat{\mathbf{x}}_{k(-)}$, od jejíž obou stran se odečte \mathbf{x}_k

$$\begin{aligned} \hat{\mathbf{x}}_{k(-)} - \mathbf{x}_k &= \Phi_{k-1} \hat{\mathbf{x}}_{k-1(+)} - \mathbf{x}_k, \\ \tilde{\mathbf{x}}_{k(-)} &= \Phi_{k-1} [\hat{\mathbf{x}}_{k-1(+)} - \mathbf{x}_{k-1}] - \mathbf{w}_{k-1}, \\ \tilde{\mathbf{x}}_{k(-)} &= \Phi_{k-1} \tilde{\mathbf{x}}_{k-1(+)} - \mathbf{w}_{k-1}. \end{aligned} \quad (3.26)$$

Kombinací (3.25), (3.26) a $\mathbf{E}\langle \tilde{\mathbf{x}}_{k-1(+)} \mathbf{w}_{k-1}^T \rangle = \mathbf{0}$ se získává výsledný vztah

$$\begin{aligned} \mathbf{P}_{k(-)} &= \Phi_{k-1} \mathbf{E}\langle \tilde{\mathbf{x}}_{k-1(+)} \tilde{\mathbf{x}}_{k-1(+)}^T \rangle \Phi_{k-1}^T + \mathbf{E}\langle \mathbf{w}_{k-1} \mathbf{w}_{k-1}^T \rangle, \\ \mathbf{P}_{k(-)} &= \Phi_{k-1} \mathbf{P}_{k-1(+)} \Phi_{k-1}^T + \mathbf{Q}_{k-1}. \end{aligned}$$

Shrnutí algoritmu Kalmanova filtru

Následuje shrnutí odvozených rovnic Kalmanova filtru, předpokládá se diskretní model stochastického lineárního dynamického systému bez vstupů tak, jak byl uveden na začátku této části textu (včetně zmíněných předpokladů)

$$\begin{aligned} \mathbf{x}_k &= \Phi_{k-1} \mathbf{x}_{k-1} + \mathbf{G}_{k-1} \mathbf{w}_{k-1}, \\ \mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \end{aligned}$$

a známé počáteční hodnoty

$$\begin{aligned} \hat{\mathbf{x}}_0 &= \mathbf{E}\langle \mathbf{x}_0 \rangle, \\ \mathbf{P}_0 &= \mathbf{E}\langle (\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T \rangle. \end{aligned}$$

V obecném časovém okamžiku t_k je nejprve vypočítán apriorní odhad stavového vektoru $\hat{\mathbf{x}}_{k(-)}$ a apriorní chybová kovarianční matice $\mathbf{P}_{k(-)}$, která vyjadřuje nejistotu tohoto odhadu

$$\begin{aligned} \hat{\mathbf{x}}_{k(-)} &= \Phi_{k-1} \hat{\mathbf{x}}_{k-1(+)}, \\ \mathbf{P}_{k(-)} &= \Phi_{k-1} \mathbf{P}_{k-1(+)} \Phi_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T. \end{aligned}$$

Následně po získání naměřených dat \mathbf{z}_k je kombinací apriorního odhadu a rozdílu mezi naměřenou a očekávanou hodnotou váženého maticí \mathbf{K}_k zjištěn aposteriorní odhad stavového vektoru $\hat{\mathbf{x}}_{k(+)}$, jeho nejistota je vyjádřena aposteriorní chybovou kovarianční maticí $\mathbf{P}_{k(+)}$

$$\begin{aligned}\hat{\mathbf{x}}_{k(+)} &= \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k(-)}], \\ \mathbf{P}_{k(+)} &= \mathbf{P}_{k(-)} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k(-)}, \\ \mathbf{K}_k &= \mathbf{P}_{k(-)}\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_{k(-)}\mathbf{H}_k^T + \mathbf{R}_k]^{-1}.\end{aligned}$$

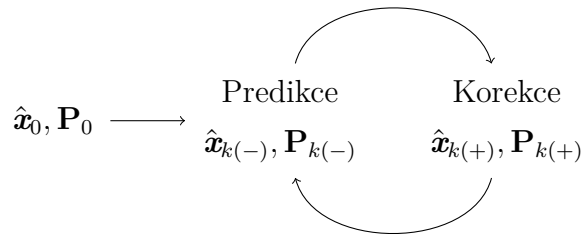
Kalmanův filtr lze zobecnit i pro model systému se vstupy, viz například [11, s. 28].

Ukázka zdrojového kódu implementace algoritmu v softwaru MATLAB je uvedena v příloze A.1.1.

3.1.3 Vlastnosti algoritmu a poznámky

Cyklus predikce – korekce

Algoritmus Kalmanova filtru má predikčně-korekční charakter. Výpočet apriorního odhadu lze chápat jako predikční krok, který je po získání naměřené hodnoty následován krokem korekčním – výpočtem aposteriorního odhadu. Toto znázorňuje obrázek 3.



Obrázek 3: Cyklus predikce – korekce (zpracováno podle [43])

Predikování, filtrování, vyrovnávání

Kalmanův filtr je nástrojem k řešení tzv. lineárně-kvadratického gaussovského problému teorie odhadu (*linear-quadratic Gaussian (LQG) estimation problem*) – problému spočívajícím v odhadu stavu lineárního dynamického systému s gaussovskými náhodnými procesy při použití kvadratické ztrátové funkce (podrobněji viz např. [18] nebo ukázka v části 4.3.1). Tento problém zahrnuje tři úlohy, k jejichž řešení lze Kalmanův filtr použít:

- **Predikování** spočívá v odhadu stavu systému v určitém čase za využití naměřených pozorování z časů předcházejících času odhadu ($t_{\text{pozorování}} < t_{\text{odhad}}$).
- **Filtrování** spočívá v odhadu stavu systému v určitém čase za využití naměřených pozorování z tohoto času a z časů předcházejících ($t_{\text{pozorování}} \leq t_{\text{odhad}}$).
- **Vyrovnávání** spočívá v odhadu stavu systému v určitém čase za využití naměřených pozorování z časů následujících po čase odhadu ($t_{\text{pozorování}} > t_{\text{odhad}}$).

Statistické vlastnosti algoritmu a odhadů

Z pohledu bayesovské statistiky algoritmus Kalmanova filtru pracuje s apriorním a aposteriorním rozdělením stavového vektoru v určitém čase, přičemž aposteriorní rozdělení je podmíněno informacemi získanými z provedeného měření. Rozdělení jsou specifikována středními hodnotami a kovariančními maticemi (což v případě normálního rozdělení stačí k úplnému popisu), které jsou propagovány mezi jednotlivými časovými okamžiky. Optimálním odhadem hodnoty stavu systému je podmíněná střední hodnota aposteriorního rozdělení (která je za uvedených předpokladů také jeho modem a mediánem).

Odhady získané Kalmanovým filtrem (při splnění uvedených předpokladů) jsou optimální vzhledem k různým kritériím. Jedná se o odhady konzistentní, nevyčýlené a s minimálním rozptylem (*minimum variance unbiased (MVU) estimates*). Pohledem bayesovské statistiky odhady minimalizují střední kvadratickou chybu MSE (*minimum mean square error (MMSE) estimates*); resp. obecně minimalizují kvadratickou ztrátovou funkci) a jedná se také o odhady ve smyslu MAP (*maximum a posteriori probability estimates*), neboť jsou rovněž modem aposteriorního rozdělení. Dokonce i pokud není splněn některý z předpokladů, mohou být odhady v jistém smyslu optimální, například, není-li šum v rovnicích modelu gaussovský, poskytuje Kalmanův filtr nejlepší (tj. mající minimální rozptyl) nevyčýlené odhady ze všech lineárních filtrů (ne však již obecně nejlepší v porovnání s nelineárními filtry). O kritériích optimality podrobněji pojednává například [33].

Poznámky k maticím \mathbf{K}_k , $\mathbf{P}_{k(-)}$ a $\mathbf{P}_{k(+)}$

Budiž připomenut vztah pro aposteriorní odhad $\hat{\mathbf{x}}_{k(+)}$ a matici Kalmanova zisku \mathbf{K}_k

$$\hat{\mathbf{x}}_{k(+)} = \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k [\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k(-)}],$$

$$\mathbf{K}_k = \mathbf{P}_{k(-)} \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_{k(-)} \mathbf{H}_k^T + \mathbf{R}_k]^{-1}.$$

Aposteriorní odhad je kombinací apriorního odhadu a rozdílu mezi skutečnou a očekávanou hodnotou měření (tento rozdíl $[z_k - \mathbf{H}_k \hat{\mathbf{x}}_{k(-)}]$ bývá označován jako inovace či někdy též reziduum měření). Matice \mathbf{K}_k je váhou v této kombinaci a zajišťuje, že apriorní odhad je upraven tak, aby výsledný aposteriorní odhad splňoval požadované kritérium optimality. Je-li nejistota apriorního odhadu malá a zdá-li se naopak měření nepřesné, bude aposteriorní odhad odpovídat apriornímu a měření se téměř zanedbá. V opačném případě, pokud je apriorní odhad nejistý a měření lze považovat za přesné, bude mít na aposteriorní odhad naměřená hodnota výraznější vliv. Tuto funkci matice \mathbf{K}_k podrobněji přibližuje příklad v části 3.1.4.

Chybové kovarianční matice $\mathbf{P}_{k(-)}$ a $\mathbf{P}_{k(+)}$ vyjadřují nejistotu apriorního resp. aposteriorního odhadu

$$\begin{aligned}\mathbf{P}_{k(-)} &= E\langle \tilde{\mathbf{x}}_{k(-)} \tilde{\mathbf{x}}_{k(-)}^T \rangle = \Phi_{k-1} \mathbf{P}_{k-1(+)} \Phi_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T, \\ \mathbf{P}_{k(+)} &= E\langle \tilde{\mathbf{x}}_{k(+)} \tilde{\mathbf{x}}_{k(+)}^T \rangle = \mathbf{P}_{k(-)} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k(-)}.\end{aligned}$$

Za předpokladu, že jsou tyto kovarianční matice pozitivně definitní, lze nalézt matice k nim inverzní ($\mathbf{Y} = \mathbf{P}^{-1}$), které se nazývají informační matice. S pomocí informačních matic a informačních vektorů lze odvodit algoritmus, tzv. informační filtr, který může být v určitých situacích výhodnější než filtr Kalmanův (podrobněji o informačních filtrech např. v [18]).

Matice $\mathbf{P}_{k(-)}$ a $\mathbf{P}_{k(+)}$, stejně jako matice \mathbf{K}_k , nezávisí na naměřených datech a lze je tedy předpočítat. Tohoto faktu lze využít při prvotním posuzování kvality navrženého modelu systému a algoritmu Kalmanova filtru ještě před vlastní implementací a to i ve fázi, když ještě nejsou dostupná žádná naměřená data. Předpočítání může být výhodné také z hlediska ušetření výpočetního výkonu, navíc v případě modelů systémů invariantních v čase zmíněné matice často brzy dosáhnou určitých ustálených hodnot, které lze využít, a vyhnout se tak nutnosti jejich opakovaného výpočtu v každém časovém okamžiku.

Na závěr ještě drobná poznámka, jak již bylo naznačeno při odvozování, pro matice \mathbf{K}_k , $\mathbf{P}_{k(-)}$ a $\mathbf{P}_{k(+)}$ existují kromě zde uvedených i další tvary vztahů k jejich výpočtu, které mají rozdílné numerické charakteristiky.

Nesplnění předpokladů

Vzájemně korelované šумы Kalmanův filtr lze rozšířit i na situace, kdy existuje korelace mezi šumem ve stavové rovnici modelu a šumem v rovnici měření (za předpo-

kladu sdruženého normálního rozdělání těchto šumů). Necht je

$$E\langle \mathbf{w}_{k_1} \mathbf{v}_{k_2}^T \rangle = \mathbf{C}_{k_1} \Delta(k_2 - k_1).$$

V tomto případě postačí upravit pouze rovnice pro výpočet apriorního odhadu a apriorní chybové kovarianční matice, tedy

$$\begin{aligned} \hat{\mathbf{x}}_{k(-)} &= \Phi_{k-1} \hat{\mathbf{x}}_{k-1(+)} + \mathbf{G}_{k-1} \mathbf{C}_{k-1} [\mathbf{H}_{k-1} \mathbf{P}_{k-1(-)} \mathbf{H}_{k-1}^T + \mathbf{R}_{k-1}]^{-1} [\mathbf{z}_{k-1} - \mathbf{H}_{k-1} \hat{\mathbf{x}}_{k-1(-)}], \\ \mathbf{P}_{k(-)} &= \Phi_{k-1} \mathbf{P}_{k-1(+)} \Phi_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T \\ &\quad - \mathbf{G}_{k-1} \mathbf{C}_{k-1} [\mathbf{H}_{k-1} \mathbf{P}_{k-1(-)} \mathbf{H}_{k-1}^T + \mathbf{R}_{k-1}]^{-1} \mathbf{C}_{k-1}^T \mathbf{G}_{k-1}^T \\ &\quad - \Phi_{k-1} \mathbf{K}_{k-1} \mathbf{C}_{k-1}^T \mathbf{G}_{k-1}^T - \mathbf{G}_{k-1} \mathbf{C}_{k-1} \mathbf{K}_{k-1}^T \Phi_{k-1}^T. \end{aligned}$$

Rovněž ale může nastat i situace, kdy

$$E\langle \mathbf{w}_{k_1-1} \mathbf{v}_{k_2}^T \rangle = \mathbf{C}_{k_2} \Delta(k_2 - k_1).$$

Nyní je naopak třeba upravit rovnice pro matici Kalmanova zisku a aposteriorní chybovou kovarianční matici, neboli nově

$$\begin{aligned} \mathbf{K}_k &= [\mathbf{P}_{k(-)} \mathbf{H}_k^T + \mathbf{G}_{k-1} \mathbf{C}_k] [\mathbf{H}_k \mathbf{P}_{k(-)} \mathbf{H}_k^T + \mathbf{R}_k + \mathbf{H}_k \mathbf{G}_{k-1} \mathbf{C}_k + \mathbf{C}_k^T \mathbf{G}_{k-1}^T \mathbf{H}_k^T]^{-1}, \\ \mathbf{P}_{k(+)} &= \mathbf{P}_{k(-)} - \mathbf{K}_k [\mathbf{H}_k \mathbf{P}_{k(-)} + \mathbf{C}_k^T \mathbf{G}_{k-1}^T]. \end{aligned}$$

Pokud by se stalo, že by \mathbf{v}_k bylo korelované s \mathbf{w}_k i s \mathbf{w}_{k-1} , lze předchozí uvedené zkombinovat. Uvedené vztahy předpokládají model systému dle rovnic (3.1) a (3.2), bylo by je však možné zobecnit i na systémy se vstupem. Podrobněji o použití Kalmanova filtru v situacích se vzájemně korelovanými šumy pojednává například [33].

V čase korelované (barevné) šumy K porušení předpokladů dojde také, pokud je šum ve stavové rovnici modelu či šum v rovnici měření korelovaný v čase, tedy nesplňuje požadovanou „bílou“ vlastnost. Nastane-li toto v případě šumu stavové rovnice, řešením může být použití tzv. tvarovacího filtru. Tvarovací filtr umožňuje vyjádřit barevný šum jako výstup lineárního dynamického systému, jehož vstupem je bílý šum. Původní model systému je rozšířen o tvarovací filtr a na takto nově vytvořený model je již možné použít Kalmanův filtr. Tvarovací filtr pro šum $\{\mathbf{w}_k\}$ může vypadat například takto

$$\begin{aligned} \mathbf{x}_k^{(f)} &= \Phi_{k-1}^{(f)} \mathbf{x}_{k-1}^{(f)} + \mathbf{G}_{k-1}^{(f)} \mathbf{n}_{k-1}^{(f)}, \\ \mathbf{w}_k &= \mathbf{H}_k^{(f)} \mathbf{x}_k^{(f)}, \end{aligned}$$

kde $\{\mathbf{n}_k^{(f)}\}$ je gaussovský bílý šum. Rozšířený model systému (za předpokladu, že je původní model reprezentován rovnicemi (3.1) a (3.2)) má potom následující podobu

(přičemž je zřejmé, že se v něm vyskytuje již jen gaussovský bílý šum)

$$\begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_k^{(f)} \end{pmatrix} = \begin{pmatrix} \Phi_{k-1} & \mathbf{G}_{k-1} \mathbf{H}_{k-1}^{(f)} \\ \mathbf{0} & \Phi_{k-1}^{(f)} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_{k-1}^{(f)} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{G}_{k-1}^{(f)} \end{pmatrix} \mathbf{n}_{k-1}^{(f)},$$

$$\mathbf{z}_k = \begin{pmatrix} \mathbf{H}_k & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_k^{(f)} \end{pmatrix} + \mathbf{v}_k.$$

Pomocí tvarovacího filtru lze podobně řešit také případy, kdy je v čase korelovaný šum rovnice měření. Tvarovací filtr pro šum $\{\mathbf{v}_k\}$ lze analogicky navrhnout například takto

$$\begin{aligned} \mathbf{x}_k^{(f)} &= \Phi_{k-1}^{(f)} \mathbf{x}_{k-1}^{(f)} + \mathbf{G}_{k-1}^{(f)} \mathbf{n}_{k-1}^{(f)}, \\ \mathbf{v}_k &= \mathbf{H}_k^{(f)} \mathbf{x}_k^{(f)}. \end{aligned}$$

Rozšířený model systému je nyní následující

$$\begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_k^{(f)} \end{pmatrix} = \begin{pmatrix} \Phi_{k-1} & \mathbf{0} \\ \mathbf{0} & \Phi_{k-1}^{(f)} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_{k-1}^{(f)} \end{pmatrix} + \begin{pmatrix} \mathbf{G}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{k-1}^{(f)} \end{pmatrix} \begin{pmatrix} \mathbf{w}_{k-1} \\ \mathbf{n}_{k-1}^{(f)} \end{pmatrix},$$

$$\mathbf{z}_k = \begin{pmatrix} \mathbf{H}_k & \mathbf{H}_k^{(f)} \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_k^{(f)} \end{pmatrix}.$$

Z rozšířeného modelu je patrné, že se v rovnici měření již explicitně nevyskytuje šum, což lze vyjádřit také tak, že $\mathbf{R}_k = \mathbf{0}$ pro všechna t_k . Tato situace, kdy je \mathbf{R}_k nulová (singulární), může však mít negativní dopad na numerickou stabilitu algoritmu, více viz část 3.1.5 a [33, s. 249]. Jako nevýhodu uvedeného přístupu lze vnímat také zvýšení rozměru stavového vektoru a z něj plynoucí nárůst výpočetní náročnosti. V souvislosti s tímto budiž zmíněno, že existuje i jiný alternativní přístup k řešení problému v čase korelovaného šumu rovnice měření, který nevyžaduje rozšířený model systému. Tento přístup je založen na výpočtu diferencí měření a je vhodný zejména při úlohách vyrovnávání.

Popsané přístupy lze zobecnit také na systémy se vstupy či přizpůsobit situacím, kdy jsou v čase korelované oba šумы. Rovněž je možné se setkat i s jinými formami vyjádření tvarovacích filtrů. Podrobnější pojednání nabízí například [33], [31].

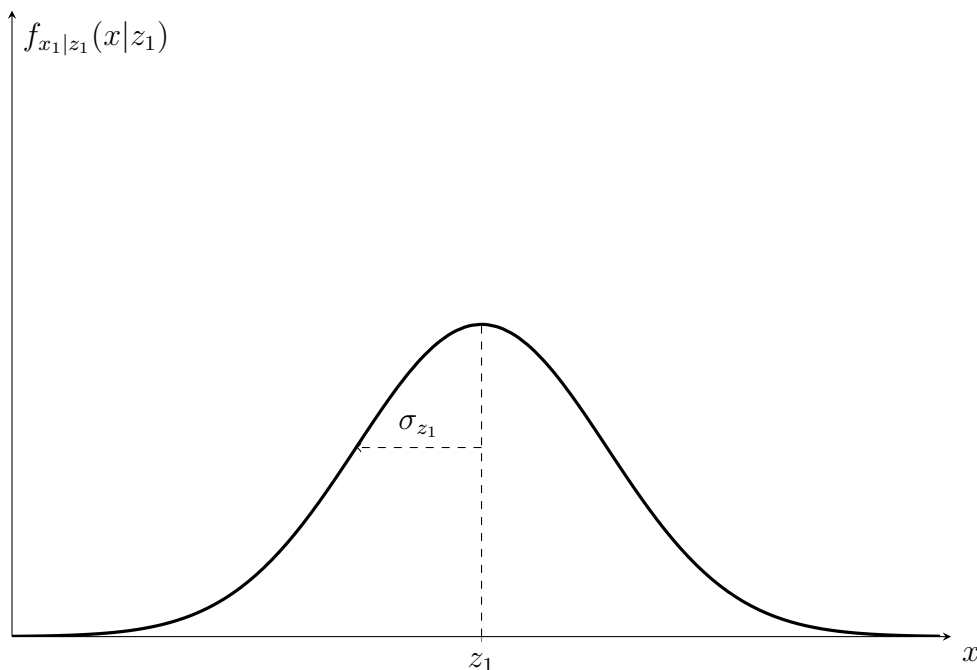
Někdy nemusí být nutné, i když jsou šумы vzájemně korelované či v čase korelované, volit některý z popsaných přístupů a lze se spokojit s výsledky, které dává standardní algoritmus Kalmanova filtru, i když nejsou optimální.

3.1.4 Demonstrační příklad

Nyní bude ukázán princip fungování Kalmanova filtru na jednoduchém ilustračním příkladu. Jako téma příkladu je zvoleno určování zeměpisné polohy objektu, neboť se

jedná o úlohu, při jejímž řešení je Kalmanův filtr v praxi velice často využíván. První část příkladu ukazuje, jak je možné v jednom časovém okamžiku zkombinovat více naměřených dat (z více zdrojů) tak, aby byl zpřesněn výsledný odhad polohy, druhá část se zabývá odhadem polohy v aktuálním časovém okamžiku na základě všech doposud naměřených dat (tj. dat z aktuálního a ze všech předchozích časových okamžiků – tedy jde o úlohu filtrování). Řešení je zároveň ukázkou toho, jak lze Kalmanův filtr vnímat z pohledu bayesovské statistiky. Příklad je převzat z [33].

Představme si, že jsme ztraceni uprostřed moře a nemáme ponětí o tom, kde se nacházíme. Jelikož je noc, rozhodneme se k určení polohy využít hvězdy (pro zjednodušení zde budiž uvažováno, že je poloha pouze jednorozměrná a rovněž celá situace je mírně zjednodušená). V určitém časovém okamžiku t_1 určíme naši polohu jako z_1 . V důsledku nepřesnosti měřicího zařízení, naší chybou apod. je tato hodnota do určité míry nepřesná. Řekněme, že naši nejistotu vyjádříme pomocí směrodatné odchylky σ_{z_1} (nebo ekvivalentně pomocí rozptylu $\sigma_{z_1}^2$). Nyní je možné stanovit podmíněnou pravděpodobnost veličiny x_1 , pozice v čase t_1 , která je podmíněná měřením z_1 . Příslušnou podmíněnou hustotu pravděpodobnosti $f_{x_1|z_1}(x|z_1)$ znázorňuje obrázek 4 (v celém příkladu budiž předpokládáno, že všechny veličiny mají normální rozdělení).



Obrázek 4: Podmíněná hustota pravděpodobnosti polohy založená na měření z_1

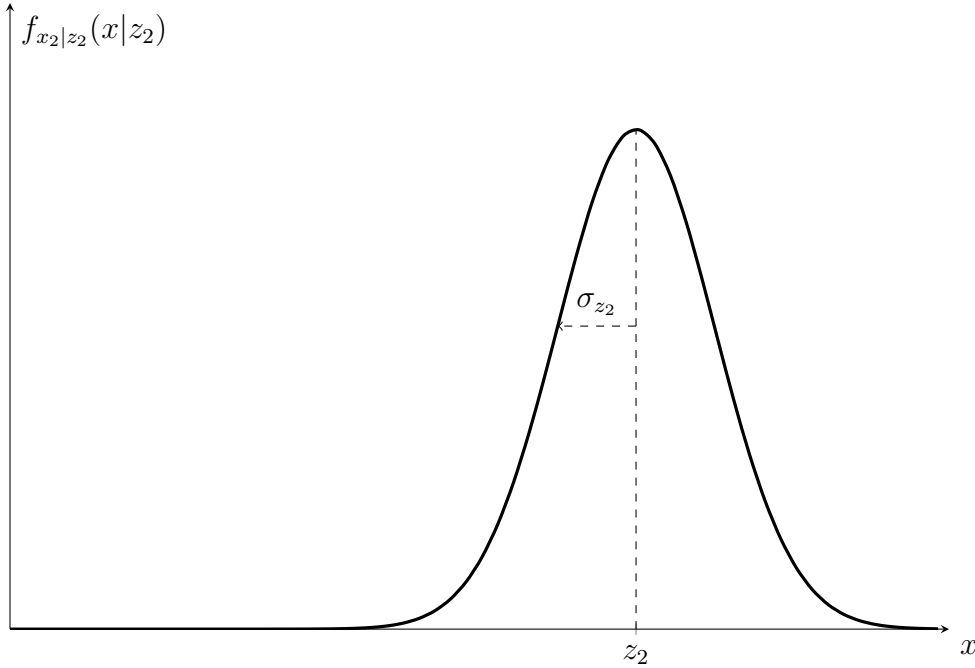
Nyní je nejlepším odhadem naší polohy střední hodnota tohoto podmíněného rozdělení (která je zde zároveň jeho modem i mediánem)

$$\hat{x}_1 = z_1$$

a příslušný rozptyl je

$$\sigma_{x_1}^2 = \sigma_{z_1}^2.$$

Dále si představme, že ve stejném časovém okamžiku $t_2 \cong t_1$ (takže zatím nedošlo ke změně polohy) provede měření také náš kolega. Výsledkem jeho snažení je hodnota z_2 s rozptylem $\sigma_{z_2}^2$, jelikož je ale kolega zkušenější, je nejistota (rozptyl) jeho měření menší, tedy $\sigma_{z_2}^2 < \sigma_{z_1}^2$. Odpovídající hustota pravděpodobnosti, podmíněná pouze měřením z_2 , je na obrázku 5.



Obrázek 5: Podmíněná hustota pravděpodobnosti polohy založená na měření z_2

K určení naší polohy v čase $t_2 \cong t_1$ jsou nyní k dispozici dvě provedená měření. Zkombinováním obou informací lze popsat nové podmíněné rozdělení veličiny x_2 , pro jehož střední hodnotu μ a rozptyl σ^2 platí

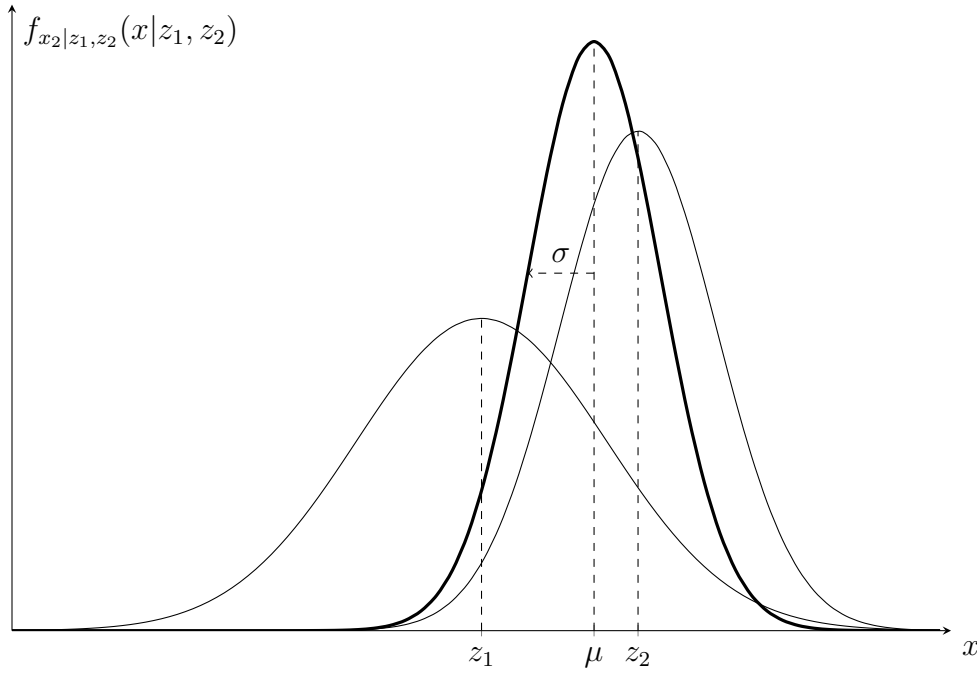
$$\mu = \frac{\sigma_{z_2}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} z_1 + \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} z_2,$$

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z_2}^2}.$$

Hustotu pravděpodobnosti tohoto podmíněného rozdělení $f_{x_2|z_1, z_2}(x|z_1, z_2)$ ukazuje obrázek 6.

Nejlepším (dle všech zmíněných kritérií optimality) odhadem naší polohy je střední hodnota (resp. modus i medián) nového podmíněného rozdělení

$$\hat{x}_2 = \mu$$



Obrázek 6: Podmíněná hustota pravděpodobnosti polohy založená na měření z_1 i z_2

s rozptylem

$$\sigma_{x_2}^2 = \sigma^2.$$

Je zřejmé, že tento rozptyl je menší než σ_{z_1} i σ_{z_2} , neboť využitím obou měření se snížila nejistota našeho odhadu.

Vztah pro \hat{x}_2 lze upravit

$$\hat{x}_2 = \frac{\sigma_{z_2}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} z_1 + \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} z_2 = z_1 + \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} (z_2 - z_1)$$

a dále, jelikož je $z_1 = \hat{x}_1$, získává se výsledná podoba vztahu tak, jak je známá z Kalmanova filtru

$$\hat{x}_2 = \hat{x}_1 + K_2(z_2 - \hat{x}_1),$$

kde

$$K_2 = \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}.$$

Z uvedeného vztahu je patrné, že optimální odhad \hat{x}_2 je roven jeho nejlepší predikci před obdržetím měření z_2 , tj. \hat{x}_1 , sečtené s korekčním členem, kterým je rozdíl skutečné hodnoty měření z_2 a její predikce násobený optimální vahou K_2 . Toto ukazuje zmíněný predikčně-korekční charakter algoritmu Kalmanova filtru.

S využitím K_2 lze ještě přepsat vztah pro $\sigma_{x_2}^2$

$$\sigma_{x_2}^2 = \sigma_{x_1}^2 - K_2 \sigma_{x_1}^2.$$

Doposud byla úloha pouze statická, nyní si ale představme, že od předchozího měření uběhl určitý čas a naše poloha se mezitím změnila. Předpokládejme, že model našeho pohybu vychází z jednoduché rovnice

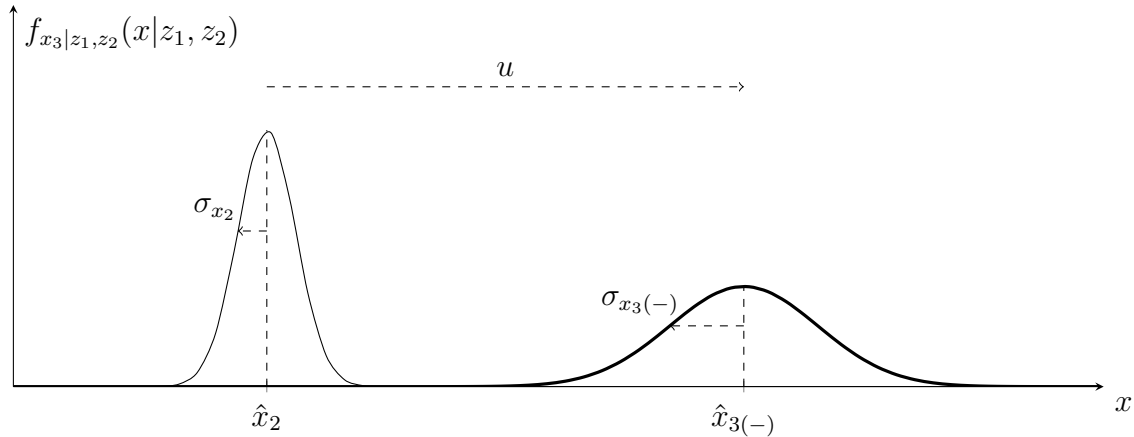
$$\frac{dx}{dt} = u + w,$$

kde u je předpokládaná rychlost našeho pohybu a $\{w\}$ je gaussovský bílý šum s nulovou střední hodnotou a rozptylem σ_w^2 , který reprezentuje nepřesnosti v určení naší rychlosti, vlivy nezahrnuté do modelu apod.

V čase t_3 , těsně předtím, než je provedeno měření, můžeme odhadnout naši polohu na základě předchozích informací a modelu, tedy

$$\begin{aligned}\hat{x}_{3(-)} &= \hat{x}_2 + u(t_3 - t_2), \\ \sigma_{x_{3(-)}}^2 &= \sigma_{x_2}^2 + \sigma_w^2(t_3 - t_2).\end{aligned}$$

Obrázek 7 zachycuje časový vývoj hustoty pravděpodobnosti, střední hodnota (odhad polohy) se posunuje podél osy x dle modelu a zároveň s přibývajícím časem roste rozptyl, neboť se zvyšuje nejistota odhadu.



Obrázek 7: Propagace podmíněné hustoty pravděpodobnosti v čase

Nyní je provedeno měření, jehož výsledkem je hodnota z_3 s rozptylem $\sigma_{z_3}^2$. Stejně jako v předchozí části tohoto příkladu, jsou opět k dispozici dvě rozdělení (dvě hustoty pravděpodobnosti), apriorní rozdělení polohy, které v sobě zahrnuje všechny informace známé před provedením měření, a rozdělení obsahující informaci získanou aktuálním měřením. Zkombinováním obou rozdělení vzniká nové aposteriorní rozdělení polohy se střední hodnotou \hat{x}_3 a rozptylem $\sigma_{x_3}^2$ (v použitém značení již není symbolem (+)

zdůrazňováno, že se jedná o aposteriorní rozdělení),

$$\begin{aligned}\hat{x}_3 &= \hat{x}_{3(-)} + K_3(z_3 - \hat{x}_{3(-)}), \\ \sigma_{x_3}^2 &= \sigma_{x_3(-)}^2 - K_3\sigma_{x_3(-)}^2, \\ K_3 &= \frac{\sigma_{x_3(-)}^2}{\sigma_{x_3(-)}^2 + \sigma_{z_3}^2},\end{aligned}$$

přičemž platí, že střední hodnota \hat{x}_3 je novým nejlepším odhadem naší polohy v čase t_3 a rozptyl $\sigma_{x_3}^2$ vyjadřuje nejistotu tohoto odhadu.

Na závěr příkladu ještě poznámka k významu optimální váhy K . Z výše uvedeného vztahu pro K_3 je patrné, že je-li rozptyl měření $\sigma_{z_3}^2$ velký, bude K_3 malé, v důsledku čehož bude mít toto nepřesné měření jen malý vliv na výsledný odhad \hat{x}_3 . V limitním případě, kdy $\sigma_{z_3}^2 \rightarrow \infty$, bude K_3 nulové a \hat{x}_3 se bude rovnat $\hat{x}_{3(-)}$. Jestliže je velký rozptyl šumu modelu dynamiky systému σ_w^2 , je velký i rozptyl $\sigma_{x_3(-)}^2$ a stejně tak K_3 . Není jistota v predikci dle modelu a bude tedy přikládána větší váha měření z_3 . Opět v limitě $\sigma_w^2 \rightarrow \infty$, $\sigma_{x_3(-)}^2 \rightarrow \infty$, $K_3 \rightarrow 1$ a z rovnice pro \hat{x}_3 plyne, že $\hat{x}_3 = z_3$. Naopak pokud je rozptyl $\sigma_{x_3(-)}^2$ malý, bude malé i K_3 a vliv měření z_3 se zanedbá, neboť je kladena důvěra v odhad získaný dle modelu.

3.1.5 Praktické aspekty implementace algoritmu

Numerická stabilita a vliv zaokrouhlovacích chyb

Jak již bylo několikrát zmíněno, výsledky, které poskytuje Kalmanův filtr, jsou z teoretického pohledu optimální, při praktické implementaci však mohou nastat určité problémy. Algoritmus obecně není numericky stabilní a snadno podléhá vlivu zaokrouhlovacích chyb. Pokud například není možné provádět výpočty v aritmetice s dostatečnou (dvojnásobnou) přesností nebo je úloha špatně podmíněná, mohou se obdržené výsledky výrazně lišit od očekávaných, algoritmus může divergovat či úplně selhat a mohou nastat situace, které jsou v rozporu s teorií – například se může stát, že spočtená chybová kovarianční matice nebude symetrická a pozitivně semidefinitní (toto lze pozorovat, pokud jsou měření velmi přesná relativně vůči nejistotě apriorních odhadů).

Řešením těchto problémů se postupem času zabývala řada autorů. Již v 60. letech přišel s prvními výsledky James E. Potter [34], čímž položil základ pro skupinu alternativních metod implementace Kalmanova filtru nazývaných jako odmocninová filtrace (*square-root filtering*). Základní myšlenkou odmocninové filtrace je nalezení matice \mathbf{S} , pro kterou platí

$$\mathbf{S}\mathbf{S}^T = \mathbf{P} \tag{3.27}$$

(jedná se o jeden z možných i když méně obvyklých způsobů definice odmocniny matice). Místo chybové kovarianční matice \mathbf{P} algoritmus rekurzivně propaguje matici \mathbf{S} , která je rovněž místo matice \mathbf{P} využita při výpočtu Kalmanova zisku. Původní Potterův přístup zdokonalili a rozšířili další autoři, z nichž budiž jmenován například Neal A. Carlson [9]. Matice \mathbf{S} může mít specifickou strukturu – může se jednat o matici horní či dolní trojúhelníkovou (pak rovnice (3.27) představuje Choleského rozklad matice \mathbf{P}). V některých případech může být výhodné rozložit na „odmocniny“ také kovarianční matice \mathbf{Q} a \mathbf{R} .

Další skupinu metod pro implementaci Kalmanova filtru s lepšími numerickými vlastnostmi představuje tzv. UD filtrace (*UD filtering*). Tento přístup představil Gerald J. Bierman [7] a následně rozšířila Catherine L. Thornton [40]. Hlavní myšlenkou je zde opět rozklad chybové kovarianční matice \mathbf{P} , který má ale nyní podobu

$$\mathbf{P} = \mathbf{UDU}^T,$$

kde \mathbf{U} je horní trojúhelníková matice s jednotkovou hlavní diagonálou a \mathbf{D} je matice diagonální.

Lepší numerickou stabilitu mohou mít také již zmíněné informační filtry (neboli inverzní kovarianční filtry), které jsou vhodné zejména v situacích, kdy je vysoká nejistota odhadu počátečního stavu systému, a které jsou efektivnější, také pokud má vektor měření velký rozměr ($l > n$).

Srovnání numerických vlastností různých metod implementace Kalmanova filtru uvádí například [41], [42]. Souhrnný podrobnější popis jmenovaných metod pak nabízí například [8], [18], [39]. Popis algoritmů pro numerický výpočet obou zmíněných maticových rozkladů lze nalézt třeba v [17].

Výpočetní nároky

Kalmanův filtr je často používán v reálném čase – průběžně zpracovává přicházející naměřená data a s jejich využitím produkuje odhady aktuálního stavu systému. Pokud implementace filtru navíc běží na vestavěném systému s omezeným výpočetním výkonem, což může být v praxi časté, má smysl se otázkou výpočetní náročnosti algoritmu zabývat. Kritická může být situace, kdy má stavový vektor či vektor měření velký rozměr.

Řešení tohoto problému se odvíjí od konkrétní úlohy a podmínek. Někdy je možné dosáhnout nižší výpočetní náročnosti pouze vhodnou úpravou implementace algoritmu, jindy však může být nutné pozměnit algoritmus tak, že poskytované výsledky již nebudou optimální, nýbrž suboptimální, přičemž ale stále dostačující potřebám dané úlohy.

Jedna z metod pro ušetření výpočetního času již byla zmíněna, matice Kalmanova zisku nezávisí na naměřených datech, tudíž její výpočet není nutné provádět v reálném čase, ale lze ji pro jednotlivé časové okamžiky předpočítat, uložit a v reálném čase její konkrétní hodnotu pouze vyzvednout. Rovněž byla uvedena skutečnost, že v případě v čase invariantních systémů lze opakovaný výpočet matice Kalmanova zisku nahradit použitím její ustálené hodnoty (*steady-state Kalman gain*) – takovýto filtr však již bude suboptimální.

Pokud by přeci jen matice Kalmanova zisku měla být počítána v každém časovém okamžiku, lze zvážit použití další metody. Při pohledu na vztah pro výpočet této matice je patrné, že obsahuje operaci, které je snaha se v numerických výpočtech vyhnout – inverzi matice. Řešením je zpracovávat vektor měření sekvenčně po jeho jednotlivých prvcích, čímž bude při výpočtu aposteriorního odhadu (resp. Kalmanova zisku) vyžadováno použití pouze skalárních operací. Výsledkem je snížení výpočetních nároků i pozitivní vliv na odolnost vůči šíření zaokrouhlovacích chyb. Aby však bylo možné vnímat prvky vektoru měření jako jednotlivé nezávislé skaláry, musí být příslušná kovarianční matice \mathbf{R} diagonální. Pokud toto není splněno, lze například za využití zmíněného UDU rozkladu této matice naměřená data tzv. „předfiltrovat“ a splnění tohoto požadavku zajistit. Použití uvedené metody má smysl, je-li matice \mathbf{R} ve všech časových okamžicích diagonální nebo pokud sice není diagonální, ale je v čase invariantní, a tedy bude stačit její rozklad počítat pouze jednou.

Někdy se také může stát, že některé prvky stavového vektoru nejsou předmětem zájmu, a tedy lze ušetřit výpočetní výkon, pokud nebudou počítány odhady jejich hodnot. Na této myšlence jsou založeny modifikace klasického Kalmanova filtru označované jako algoritmy redukované filtrace (*reduced-order filtering*). Jedním z nejznámějších zástupců této skupiny algoritmů je Schmidtův-Kalmanův filtr, jehož autorem je Stanley F. Schmidt [37]. Tento přístup lze uplatnit v případě, kdy má model systému následující tvar

$$\begin{pmatrix} \mathbf{x}_{k[1]} \\ \mathbf{x}_{k[2]} \end{pmatrix} = \begin{pmatrix} \Phi_{k-1[11]} & \mathbf{0} \\ \mathbf{0} & \Phi_{k-1[22]} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{k-1[1]} \\ \mathbf{x}_{k-1[2]} \end{pmatrix} + \begin{pmatrix} \mathbf{G}_{k-1[11]} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{k-1[22]} \end{pmatrix} \begin{pmatrix} \mathbf{w}_{k-1[1]} \\ \mathbf{w}_{k-1[2]} \end{pmatrix},$$

$$\mathbf{z}_k = \begin{pmatrix} \mathbf{H}_{k[1]} & \mathbf{H}_{k[2]} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{k[1]} \\ \mathbf{x}_{k[2]} \end{pmatrix} + \mathbf{v}_k,$$

kde $\mathbf{x}_{k[1]}$ jsou prvky stavového vektoru, jejichž hodnoty jsou sledovány (odhadovány) a $\mathbf{x}_{k[2]}$ je část stavového vektoru, která naopak není předmětem zájmu (je tedy vhodný například pro model systému zahrnující tvarovací filtr). Schmidtův-Kalmanův filtr je filtrem suboptimálním.

Mají-li rovnice modelu systému výše naznačenou blokově-diagonální podobu (a je-li pozornost zaměřena na odhad celého stavového vektoru), nabízí se možnost vnímat zadanou úlohu filtrace jako oddělené podúlohy, které lze řešit paralelně.

Podrobnější informace k časové výpočetní náročnosti Kalmanova filtru a detailnější představení zde zmíněných i dalších metod pro její redukci lze nalézt například v [18], [31], [39].

Co se týká paměťových nároků, úspory lze dosáhnout zejména vhodným přepisováním a opakovaným používáním stejných datových struktur v průběhu výpočtu algoritmu či eliminací redundance v některých datových strukturách, která plyne například ze symetrie kovariančních matic apod. Otázka paměťové náročnosti však dnes nebývá tak kritická.

Návrh modelu systému

Zatímco teorie předpokládá, že jsou matice modelu systému Φ , \mathbf{G} , \mathbf{H} , \mathbf{Q} a \mathbf{R} pro všechny časové okamžiky známé a rovněž je známý počáteční stav, v praxi nemusí být lehké jejich hodnoty určit. Navržený model zpravidla odpovídá skutečnému systému pouze přibližně, což musí být při aplikaci Kalmanova filtru bráno v potaz. Filtr je optimální pro použitý model systému, nikoli však pro skutečný systém. Nepřesnosti modelu způsobují, že skutečná chyba odhadů je odlišná od chyby očekávané, vyjádřené chybovou kovarianční maticí, přičemž se může stát, že skutečná chyba bez omezení roste a filtr diverguje (i když vypočítané chybové kovarianční matice toto neindikují).

Poměrně snadným řešením nastíněného problému je přidání falešného šumu do stavové rovnice modelu (resp. zvýšení hodnot příslušné kovarianční matice \mathbf{Q}). V důsledku tohoto filtr méně důvěřuje predikcím vypočítaným dle modelu a dává větší váhu na naměřená data. Podobného efektu lze dosáhnout také exponenciálním vážením dat či tzv. „filtrací se slábnoucí pamětí“, které kladou větší důraz na nedávná data.

Předchozích jmenovaných přístupů by však nemělo být zneužíváno na úkor kvality modelu systému. Kvalitu modelu a příslušného filtru lze zkoumat již ve fázi návrhu pomocí citlivostní analýzy. Citlivostní analýza pracuje se „správným“ modelem skutečného systému a modelem (zjednodušeným), na základě kterého je navrhován Kalmanův filtr, umožňuje tak porovnat výsledky více navržených filtrů založených na různých modelech vzhledem k předem stanoveným požadavkům. Využívá se kovarianční analýzy a Monte Carlo simulací (podrobněji např. [16], [18]).

Dalším způsobem, jak lze indikovat problémy v návrhu modelu systému, je analýza inovací (neboli reziduí měření). Lze dokázat (viz např. [39, s. 298]), že inovace by měly být nekorelovaným náhodným procesem s nulovou střední hodnotou a kovarianční maticí $[\mathbf{H}_k \mathbf{P}_{k(-)} \mathbf{H}_k^T + \mathbf{R}_k]$ v čase t_k . Pokud toto není splněno, může se jednat právě o důsledek chybného modelu systému.

Pokud nelze zajistit dostatečnou přesnost modelu na základě znalosti systému (některé matice modelu jsou známy pouze s určitou nepřesností či nejsou známy vůbec), je také možné využít modifikace klasického Kalmanova filtru – robustní Kalmanův filtr nebo adaptivní Kalmanův filtr, více viz například [31] a [11]. Pro další poznatky o problematice návrhu modelů systémů lze odkázat třeba na [18], [33], [39].

Odlehlá či chybějící pozorování

Problémy se mohou objevit také v souvislosti s naměřenými daty, která jsou Kalmanovu filtru poskytnuta ke zpracování. Jednou z existujících nástrah jsou odlehlá pozorování, která mohou být například důsledkem chyby měřicího zařízení. Takovéto chybné pozorování poškodí momentální odhad a má negativní vliv na odhady následující, z teoretického pohledu by se však měl filtr v konečném čase z jeho následků zotavit. V praxi ale nemusí být tato schopnost dostačující, a proto je obecně lepší podezřelá pozorování filtru vůbec neposkytnout. K posouzení, zdali je nějaké pozorování odlehlé, lze vhodně využít inovaci neboli rozdíl mezi skutečnou a očekávanou hodnotou tohoto pozorování. Když je rozdíl vyšší než nějaká stanovená mez, nebude příslušné pozorování filtrem zpracováno. Budiž podotknuto, že existují i sofistikovanější přístupy pro identifikaci odlehlých pozorování (více např. [18]) a zároveň budiž také upozorněno, že ne každá odlehlá hodnota je nutně chybou. Pokud by přeci jen chybné pozorování do výpočtů filtru zasáhlo, lze podpořit jeho zotavení zvýšením kovariance šumu stavové rovnice modelu, čímž bude filtr dávat větší váhu následujícím pozorováním.

Dalším problémem, který se může při práci s daty objevit, jsou chybějící pozorování. Pokud v určitém časovém okamžiku nejsou k dispozici naměřená data (nebo by tato data mohla být nespolehlivá), bude proveden pouze výpočet apriorního odhadu, který však nebude následován korekčním krokem, neboť není k dispozici žádná nová informace, která by mohla odhad zpřesnit.

3.2 Další algoritmy Kalmanovy filtrace

V souvislosti s praktickou implementací algoritmu byly zmíněny různé modifikace klasického Kalmanova filtru, které lze vhodně využít v určitých specifických situacích. Například algoritmy odmocninové filtrace, UD filtrace či informační filtry řeší problém numerické stability, algoritmy redukované filtrace snižují výpočetní náročnost, adaptivní nebo robustní Kalmanův filtr pomáhá v případech, kdy je problematické navrhnout model systému apod. Specifická rozšíření Kalmanova filtru existují také pro nelineární modely systémů či modely systémů ve spojitém čase. Všechny tyto algoritmy však vycházejí ze společné základní myšlenky filtru navrženého Rudolfem E. Kalmanem, a bývají tedy souhrnně označovány jako algoritmy Kalmanovy filtrace či Kalmanovy filtry.

3.2.1 Nelineární filtrace

Jedním z předpokladů pro použití Kalmanova filtru, tak jak byl popsán v předchozí části textu, je linearita systému. Ve skutečnosti však často může být dynamika systému či vztah mezi jeho stavem a měřeními (nebo obojí) nelineární. Rovnice modelu systému mohou mít níže uvedenou podobu, přičemž pro účely dalšího výkladu budiž uvažováno, že systém nemá žádné vstupy, šum v obou rovnicích je aditivní a stále platí předpoklady uvedené v části 3.1.2

$$\begin{aligned}\mathbf{x}_k &= \phi_{k-1}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}, \\ \mathbf{z}_k &= h_k(\mathbf{x}_k) + \mathbf{v}_k,\end{aligned}$$

kde ϕ_{k-1} a h_k jsou obecně nelineární funkce.

V případě lineárního systému je za předpokladu normality rozdělení počátečního stavu a obou šumů zachována normalita rozdělení sledovaných veličin i při přechodu mezi časovými okamžiky, neboť lineární transformace zachovávají normální rozdělení (viz věty 2.3 a 2.4). Pro popis vývoje systému v čase pak stačí propagovat střední hodnoty a kovarianční matice – a právě na tomto principu funguje Kalmanův filtr. Nelineární transformace normální rozdělení obecně nezachovávají, a Kalmanův filtr tedy nelze takto použít.

Existují přístupy, které vycházejí z myšlenky Kalmanova filtru a zároveň dovolují pracovat i s nelineárními systémy, nejedná se však o nástroje umožňující nalézt optimální řešení úlohy, ale pouze jeho více či méně dobrou aproximaci. Podrobněji budou představeny dva nejznámější a nejčastěji používané nelineární Kalmanovy filtry a zmíněny

i některé další. Volba konkrétního filtru pak záleží na povaze a požadavcích řešené aplikační úlohy.

Extended Kalman filter

Jedním z možných přístupů, jak řešit problém nelinearity modelu systému, je aproxi-
movat příslušné nelineární funkce funkcemi lineárními. Na této myšlence je založen tzv.
rozšířený Kalmanův filtr (*extended Kalman filter*, EKF), který k linearizaci využívá Tay-
lorův rozvoj dané funkce v bodě aktuálního odhadu. V případě funkce stavové rovnice
 ϕ_{k-1} je tímto odhadem $\hat{\mathbf{x}}_{k-1(+)}$, lineární část Taylorova rozvoje má potom podobu

$$\phi_{k-1}(\mathbf{x}_{k-1}) \approx \phi_{k-1}(\hat{\mathbf{x}}_{k-1(+)} + \Phi_{k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1(+)}),$$

kde Φ_{k-1} je Jacobiho matice parciálních derivací

$$\Phi_{k-1} = \left. \frac{\partial \phi_{k-1}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1(+)}}.$$

Taylorův rozvoj funkce rovnice měření h_k je konstruován v bodě $\hat{\mathbf{x}}_{k(-)}$ a obdobně pak

$$h_k(\mathbf{x}_k) \approx h_k(\hat{\mathbf{x}}_{k(-)} + \mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k(-)}),$$

kde \mathbf{H}_k je opět příslušná Jacobiho matice

$$\mathbf{H}_k = \left. \frac{\partial h_k}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k(-)}}.$$

S využitím uvedených aproximací lze model systému přepsat do lineární podoby a ná-
sledně již použít klasický Kalmanův filtr. Výsledkem jsou následující rovnice souhrnně
označované jako rozšířený Kalmanův filtr

$$\begin{aligned} \hat{\mathbf{x}}_{k(-)} &= \phi_{k-1}(\hat{\mathbf{x}}_{k-1(+)}), \\ \mathbf{P}_{k(-)} &= \Phi_{k-1} \mathbf{P}_{k-1(+)} \Phi_{k-1}^T + \mathbf{Q}_{k-1}, \\ \hat{\mathbf{x}}_{k(+)} &= \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k [z_k - h_k(\hat{\mathbf{x}}_{k(-)})], \\ \mathbf{P}_{k(+)} &= \mathbf{P}_{k(-)} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k(-)}, \\ \mathbf{K}_k &= \mathbf{P}_{k(-)} \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_{k(-)} \mathbf{H}_k^T + \mathbf{R}_k]^{-1}. \end{aligned}$$

Uvedené rovnice je možné zobecnit také pro systémy se vstupy a neaditivním šumem,
viz [39, s. 409].

Původní myšlenku rozšířeného Kalmanova filtru publikoval Stanley F. Schmidt [38], od
té doby je filtr v praxi hojně využíván. Nutno však podotknout, že ne vždy přináší jeho

použití požadované výsledky, vhodnost závisí na povaze konkrétní řešené úlohy, obecně však platí, že se hodí pro systémy slabě nelineární. Jako nevýhodu lze kromě nutnosti výpočtu Jacobiho matic považovat také fakt, že matice \mathbf{K}_k , $\mathbf{P}_{k(-)}$ a $\mathbf{P}_{k(+)}$ závisejí na naměřených datech, a nelze je tedy předpočítat jako v případě klasického Kalmanova filtru.

Unscented Kalman filter

Zcela jiný přístup k řešení problému nelinearity modelu systému přináší tzv. *unscented Kalman filter* (UKF). Snahou v tomto případě není problém linearizovat, ale pracovat přímo s jeho nelineární podstatou. Využito je přitom statistické vzorkování, konkrétně technika nazývaná jako *unscented transform*, podle které filtr získal své jméno. Určitým deterministickým způsobem je zvoleno několik bodů označovaných jako *sigma points*, které vhodně charakterizují rozdělení sledované veličiny, tyto body jsou transformovány nelineární funkcí a z transformovaných bodů je poté vypočtena střední hodnota a kovarianční matice popisující nové rozdělení transformované veličiny.

Obecně lze *unscented transform* popsat následovně, přičemž necht $\hat{\mathbf{x}}$ je aktuální odhad stavu systému, který je střední hodnotou veličiny \mathbf{x} , a \mathbf{C}_{xx} je odmocnina resp. Choleského faktor příslušné kovarianční matice \mathbf{P}_{xx} (viz část 3.1.5). Dále necht g je obecně nelineární transformační funkce a $\mathbf{y} = g(\mathbf{x})$ je transformovaná veličina. V případě stavové rovnice představuje g funkci ϕ a \mathbf{y} stavový vektor v následujícím časovém okamžiku. Pro případ rovnice měření vyjadřuje g funkci h a \mathbf{y} vektor měření \mathbf{z} .

Unscented transform spočívá ve výběru bodů $\boldsymbol{\chi}_i$ z rozdělení \mathbf{x} a přiřazení odpovídajících vah W_i tak, že platí

$$\hat{\mathbf{x}} = \sum_i W_i \boldsymbol{\chi}_i,$$

$$\mathbf{P}_{xx} = \sum_i W_i (\boldsymbol{\chi}_i - \hat{\mathbf{x}})(\boldsymbol{\chi}_i - \hat{\mathbf{x}})^T.$$

Následně je každý bod $\boldsymbol{\chi}_i$ transformován funkcí g

$$\boldsymbol{\xi}_i = g(\boldsymbol{\chi}_i).$$

Na závěr je s využitím těchto transformovaných bodů vypočítána aproximace nové střední hodnoty a kovarianční matice transformované veličiny \mathbf{y}

$$\hat{\mathbf{y}} \approx \sum_i W_i \boldsymbol{\xi}_i,$$

$$\mathbf{P}_{yy} \approx \sum_i W_i (\boldsymbol{\xi}_i - \hat{\mathbf{y}})(\boldsymbol{\xi}_i - \hat{\mathbf{y}})^T.$$

Pro volbu bodů $\boldsymbol{\chi}_i$ a příslušných vah W_i existují různé strategie. Strategie, která bude dále popsána, bývá v souvislosti s Kalmanovým filtrem používána nejčastěji, ukázky dalších strategií lze nalézt například v [18].

Strategie nazývaná *scaled unscented transform* spočívá ve výběru celkem $2n + 1$ (n je rozměr stavového vektoru) bodů symetricky rozmístěných kolem střední hodnoty, přičemž přizpůsobení výběru potřebám konkrétní úlohy umožňují tři volitelné parametry α, β, κ . Parametry α a κ lze pro zjednodušení zkombinovat do jednoho parametru označovaného λ

$$\lambda = \alpha^2(n + \kappa) - n.$$

Body a jejich váhy jsou voleny takto ($\mathbf{c}_{[i]}$ zde značí i -tý sloupec matice \mathbf{C}_{xx})

$$\begin{aligned} \boldsymbol{\chi}_0 &= \hat{\boldsymbol{x}}, & W_0^{[\hat{y}]} &= \lambda/(n + \lambda) \\ \boldsymbol{\chi}_i &= \hat{\boldsymbol{x}} + \sqrt{n + \lambda} \mathbf{c}_{[i]}, & W_0^{[P_{yy}]} &= \lambda/(n + \lambda) + 1 - \alpha^2 + \beta \\ \boldsymbol{\chi}_{n+i} &= \hat{\boldsymbol{x}} - \sqrt{n + \lambda} \mathbf{c}_{[i]}, & W_i &= 1/[2(n + \lambda)], & 1 \leq i \leq n \\ & & W_{n+i} &= 1/[2(n + \lambda)], & 1 \leq i \leq n. \end{aligned}$$

Jak již bylo uvedeno, tyto body jsou transformovány funkcí g

$$\boldsymbol{\xi}_i = g(\boldsymbol{\chi}_i), \quad 0 \leq i \leq 2n$$

a dále je vypočítána aproximace $\hat{\boldsymbol{y}}$ a \mathbf{P}_{yy} , přičemž pro bod $\boldsymbol{\xi}_0$ jsou použity dvě různé váhy $W_0^{[\hat{y}]}$ a $W_0^{[P_{yy}]}$

$$\begin{aligned} \hat{\boldsymbol{y}} &\approx W_0^{[\hat{y}]} \boldsymbol{\xi}_0 + \sum_{i=1}^{2n} W_i \boldsymbol{\xi}_i, \\ \mathbf{P}_{yy} &\approx W_0^{[P_{yy}]} (\boldsymbol{\xi}_0 - \hat{\boldsymbol{y}})(\boldsymbol{\xi}_0 - \hat{\boldsymbol{y}})^T + \sum_{i=1}^{2n} W_i (\boldsymbol{\xi}_i - \hat{\boldsymbol{y}})(\boldsymbol{\xi}_i - \hat{\boldsymbol{y}})^T. \end{aligned}$$

Parametry α a κ ovlivňují rozptýlení bodů kolem střední hodnoty (α bývá obvykle volena jako malé kladné číslo, κ bývá často rovna nule), β se podílí na vytváření váhy $W_0^{[P_{yy}]}$, její hodnota vychází z informací o rozdělení veličiny (pro normální rozdělení bývá $\beta = 2$).

Celý algoritmus *unscented Kalman filter* vypadá následovně, $ut()$ značí funkci realizující *unscented transform*, v kulaté závorce jsou uvedeny její vstupní argumenty, v hranaté pak výstupní argumenty, $\mathbf{C}_{k-1(+)}$ resp. $\mathbf{C}_{k(-)}$ je Choleského faktor (odmocnina) matice $\mathbf{P}_{k-1(+)}$ resp. $\mathbf{P}_{k(-)}$

$$\begin{aligned}
[\hat{\mathbf{x}}_{k(-)}, \mathbf{P}_{xx}] &= ut(\hat{\mathbf{x}}_{k-1(+)}, \mathbf{C}_{k-1(+)}, \phi_{k-1}, \alpha, \beta, \kappa), \\
\mathbf{P}_{k(-)} &= \mathbf{P}_{xx} + \mathbf{Q}_{k-1}, \\
[\hat{\mathbf{z}}_k, \mathbf{P}_{zz}] &= ut(\hat{\mathbf{x}}_{k(-)}, \mathbf{C}_{k(-)}, h_k, \alpha, \beta, \kappa), \\
\mathbf{P}_{xz} &= \sum_i W_i (\boldsymbol{\chi}_i - \hat{\mathbf{x}}_{k(-)}) (\boldsymbol{\xi}_i - \hat{\mathbf{z}}_k)^T, \\
\mathbf{K}_k &= \mathbf{P}_{xz} [\mathbf{P}_{zz} + \mathbf{R}_k]^{-1}, \\
\hat{\mathbf{x}}_{k(+)} &= \hat{\mathbf{x}}_{k(-)} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k), \\
\mathbf{P}_{k(+)} &= \mathbf{P}_{k(-)} - \mathbf{K}_k \mathbf{P}_{xz}^T.
\end{aligned}$$

Unscented Kalman filter navrhli Simon J. Julier a Jeffrey K. Uhlmann [22]. V porovnání s rozšířeným Kalmanovým filtrem poskytuje tento filtr obecně lepší výsledky zejména v případě vysoce nelineárních systémů, výpočetní náročnost mají oba filtry podobnou.

Zdrojové kódy implementací algoritmů EKF a UKF v MATLABu jsou k nahlédnutí v příloze A.1.2.

Další nelineární Kalmanovy filtry

Kromě EKF a UKF existují i jiné nelineární filtry, které svým principem bývají obvykle podobné jednomu z těchto dvou uvedených, ale liší se svojí robustností či výpočetní náročností.

Do skupiny filtrů vycházejících z myšlenky linearizace nelineárních funkcí patří například linearizovaný Kalmanův filtr (*linearized Kalman filter*), který provádí linearizaci v bodech tzv. nominální trajektorie, na rozdíl od EKF (provádějícího linearizaci v bodech odhadnuté trajektorie) umožňuje předpočítání, na druhou stranu je však méně robustní vůči nelinearitám. Dalším zástupcem této skupiny je *iterated extended Kalman filter* (IEKF), který zpřesňuje odhad získaný pomocí EKF tím, že iterativně opakuje korekční krok algoritmu, přičemž linearizaci funkce měření už neprovádí v bodě apriorního odhadu, ale v bodě právě získaného posteriorního odhadu. Lepší výsledky z pohledu přesnosti a robustnosti vůči vlivu silnějších nelinearit poskytují také filtry, které pracují nejen s lineárním členem Taylorova rozvoje, ale i s členy vyšších řádů. Mezi tyto filtry patří například *second-order extended Kalman filter*, jejich nevýhodou je však velmi vysoká výpočetní náročnost.

Druhou skupinu tvoří filtry založené na statistickém vzorkování, přičemž jsou využívány různé vzorkovací techniky. Obecně bývají tyto filtry robustnější vůči silným nelinearitám, v některých případech je však nutné zaplatit vyšší časovou náročností. Kromě UKF sem patří například *Monte Carlo Kalman filter*, *ensemble Kalman filter*, *Gauss-Hermite Kalman filter* a mnohé další. Podobné myšlenky využívají také částicové filtry.

Podrobnější pojednání o zmíněných filtrech lze nalézt například v [8], [18], [39].

3.2.2 Filtrace ve spojitém čase

Doposud bylo pracováno pouze s modely systémů v diskrétním čase, Kalmanův filtr lze však analogicky odvodit a použít i ve spojitém čase. Na návrhu filtru se podílel Richard S. Bucy [24], a bývá tedy označován jako Kalmanův-Bucyův filtr.

Budiž uvažován následující model stochastického lineárního dynamického systému ve spojitém čase, jenž byl popsán v části 2.3.2 (stejně jako v předchozím textu bude pro zjednodušení předpokládáno, že systém nemá žádné vstupy)

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t), \\ \mathbf{z}(t) &= \mathbf{H}(t)\mathbf{x}(t) + \mathbf{v}(t).\end{aligned}$$

Dále necht' platí předpoklady analogické jako v diskrétním čase. Rovnice Kalmanova-Bucyova filtru pak mají podobu

$$\begin{aligned}\dot{\hat{\mathbf{x}}}(t) &= \mathbf{F}(t)\hat{\mathbf{x}}(t) + \mathbf{K}(t)[\mathbf{z}(t) - \mathbf{H}(t)\hat{\mathbf{x}}(t)], \\ \dot{\mathbf{P}}(t) &= \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) - \mathbf{K}(t)\mathbf{R}(t)\mathbf{K}^T(t) + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t), \\ \mathbf{K}(t) &= \mathbf{P}(t)\mathbf{H}^T(t)\mathbf{R}^{-1}(t).\end{aligned}$$

Podrobné odvození rovnic i zobecnění pro modely systémů se vstupy lze nalézt například v [18], [33].

V praxi se lze setkat také s tzv. hybridními modely, které obvykle popisují dynamiku systému spojitě, zatímco měření jsou chápána jako diskrétní.

Nejen klasický Kalmanův filtr, ale i další Kalmanovy filtry (například filtry nelineární) existují ve variantách pro modely systémů v diskrétním čase, ve spojitém čase i pro modely hybridní.

4 Aplikace Kalmanových filtrů

Kapitola se věnuje představení praktického použití Kalmanova filtru a z něj vycházejících algoritmů. Hlavní pozornost je zaměřena na úlohu indoor lokalizace a na možnosti využití Kalmanova filtru při jejím řešení. Představeno je několik algoritmů založených na myšlenkách Kalmanova filtru, které jsou následně implementovány, otestovány a porovnány s klasickými lokalizačními technikami. Druhá část kapitoly demonstruje širší uplatnění Kalmanova filtru ukázkami jeho aplikací v analýze časových řad a ve zpracování obrazu.

4.1 Přehled aplikačních domén

Kalmanův filtr nachází různorodá uplatnění. Jednou z nejvýznamnějších aplikačních domén je lokalizace pohybujících se objektů a navigace – Kalmanův filtr či obecně Kalmanovy filtry jsou využívány například v globálních družicových polohových systémech (GPS apod.), v radarech, při navigaci a řízení robotů, v autopilotech či autonomních vozidlech, v počítačovém vidění při sledování pohybu objektů v sekvenci video snímků, dále v rozšířené a virtuální realitě atd. Do oblasti navigace a řízení spadá také první aplikace Kalmanova filtru z 60. let minulého století, kterou byl vesmírný projekt Apollo. Z dalších aplikačních domén lze jmenovat třeba analýzu časových řad, ekonometrii, zpracování signálu, předpovídání počasí a mnohé jiné.

4.2 Indoor lokalizace

Rozvoj v oblasti mobilních technologií a SMART řešení podněcuje potřebu zabývat se otázkou lokalizace mobilních zařízení či objektů v prostoru. Inteligentní systém, který má povědomí o své poloze, se pak dokáže adaptovat a poskytnout uživateli informace, které jsou pro něj relevantní vzhledem ke geografickému kontextu.

Na předpokladu znalosti polohy mobilního zařízení, resp. uživatele, jsou založeny tzv. *location-based services* (LBS). Důležitá je přitom schopnost zjistit polohu zařízení nejen ve venkovním otevřeném prostoru, ale velice často také uvnitř budov (indoor lokalizace). Kromě statické lokalizace se může jednat rovněž o lokalizaci dynamicky se pohybujícího objektu, přičemž je možné řešit také úlohu predikce pohybu.

Právě indoor lokalizace nachází své uplatnění v řadě oblastí [44], jako příklad budiž uvedena navigace ve veřejných budovách spolu s inteligentním poskytováním potřebných nebo zajímavých kontextových informací uživateli podle toho, kde se právě nachází. Dále lze jmenovat také záchranářství a řešení krizových situací, bezpečností a vojenské aplikace, chytré domácnosti, cílený marketing, oblast zábavy či turizmu a mnohé další. Na svém významu získává indoor lokalizace také v souvislosti s dnes populárními koncepty jako je například rozšířená realita.

4.2.1 Přehled stávajících řešení

Pro lokalizaci mobilních zařízení či objektů uvnitř budov neexistuje obecné standardizované řešení. Úlohu určení polohy ve venkovním otevřeném prostoru uspokojivě řeší systém GPS (*global positioning system*), ten však uvnitř budov nelze z podstaty použít.

Při indoor lokalizaci jsou často využívány různé rádiové technologie, které mohou vycházet z existujících bezdrátových sítí v budovách, mezi tyto technologie patří například Wi-Fi, Bluetooth (zejména technologie Bluetooth Low Energy, BLE), Zigbee, atd. K lokalizaci lze využít rovněž světelné či zvukové vlnění. Každá z těchto technologií umožňuje dosáhnout určité přesnosti lokalizace a její použití je spojeno s většími či menšími finančními a časovými náklady na realizaci. Kromě těchto technologií bývají využívány také přístupy pracující s daty ze senzorů pohybu a orientace (*inertial measurement unit*, IMU) v mobilních zařízeních.

Obecně existují různé techniky lokalizace [44]. První skupinu tvoří přístupy založené na principech trigonometrie a na znalosti úhlů, pod kterými signál přichází z referenčních bodů se známou polohou, nebo na znalosti vzdáleností lokalizovaného objektu k těmto známým referenčním bodům. Tyto přístupy však nebývají vhodné pro použití ve vnitřních prostorách s množstvím překážek, kde není přímá viditelnost mezi lokalizovaným objektem a referenčními body.

Další lokalizační technikou je *fingerprinting*. Fingerprinting je tvořen dvěma fázemi. V první fázi, tzv. off-line, je vytvořena referenční mapa otisků (fingerprintů) naměřením a uložením informace o silách signálů (*received signal strength indicator*, RSSI)

z jednotlivých vysílačů na předem určených místech se známou polohou. Druhá fáze, tzv. on-line, spočívá v určení polohy objektu na základě porovnání aktuálně naměřených signálů a otisků z referenční mapy. Tato technika bude podrobněji popsána v následující části textu.

Jiné informace využívá pro lokalizaci technika nazývaná *pedestrian dead reckoning* (PDR). Tato technika pracuje s daty o pohybu mobilního zařízení poskytovanými jeho senzory (akcelerometr, gyroskop, magnetometr), s jejich pomocí a při znalosti počáteční polohy pak odhaduje vývoj aktuální pozice v čase. Výhodou PDR je nezávislost na existenci rádiových sítí, nutná je však znalost přesné počáteční polohy, přesnost závisí také na kvalitě senzorů a v delším čase dochází ke kumulaci a nárůstu chyby odhadů.

Technika fingerprintingu

Fingerprinting je v indoor lokalizaci hojně využívanou technikou a často slouží jako srovnávací referenční technika pro nově vyvinuté algoritmy. Jeho realizace nebývá nákladná a obtížná, neboť využívá rádiové bezdrátové sítě, které v budovách již často existují. Obvykle se jedná o technologii Wi-Fi (IEEE 802.11) nebo v poslední době populární technologii BLE v podobě beaconů (vysílacích zařízení a protokolu iBeacon), se kterými přišla na trh v roce 2013 společnost Apple [30]. Nevýhodou fingerprintingu je nutnost opětovného vytváření nové referenční mapy otisků při jakékoli změně síťové infrastruktury.

Určení polohy technikou fingerprintingu spočívá v naměření aktuálních sil signálu z jednotlivých vysílačů, tj. vytvoření aktuálního otisku v daném místě, a následném porovnání tohoto otisku s otisky z referenční mapy. Cílem je nalézt nejpodobnější referenční otisk, přičemž podobnost je měřena vzdáleností otisků v prostoru všech otisků. Při výpočtu vzdálenosti je možné využít různé metriky, obvykle bývá používána metrika manhattanská či euklidovská. Tento postup je označován jako algoritmus nejbližšího souseda (*nearest neighbor*, NN). Poloha asociovaná s nejbližším referenčním otiskem je pak považována za hledanou aktuální polohu.

V obecnějším pojetí lze nalézt několik (k) nejpodobnějších referenčních otisků a hledanou polohu spočítat jako průměr poloh referenčních otisků. V tomto případě se pak jedná o algoritmus k nejbližších sousedů (*k nearest neighbors*, KNN).

Další ještě obecnější modifikací zmíněného postupu je vážený algoritmus k nejbližších sousedů (*weighted k nearest neighbors*, WKNN), který počítá polohu jako vážený průměr poloh k referenčních otisků, přičemž podobnější otisky mají vyšší váhu. Algoritmus

WKNN bývá v indoor lokalizaci používán nejčastěji, teoreticky vede vyšší hodnota k k přesnějším výsledkům ovšem za cenu rostoucí výpočetní náročnosti, v praxi bývá často voleno $k = 3$, $k = 4$.

Jedním z prvních systémů pro indoor lokalizaci založených na rádiových sítích je RADAR [4] navržený výzkumníky společnosti Microsoft.

Motivace pro použití Kalmanových filtrů

Nevýhody využití rádiového signálu v indoor lokalizaci však plynou z fyzikálních zákonitostí jeho šíření a z povahy vnitřního prostředí budov. Signál uvnitř budovy naráží na množství různých překážek z různých materiálů (zdi, nábytek, lidé, ...), a tím dochází k jeho pohlcování, lomu či odrazům. Vlastnosti (síla) signálu se tedy v takto komplexním prostředí mění v závislosti na místě a čase způsobem, který nelze jednoduše popsat.

Naměřené hodnoty síly signálu v určitém místě, které mají být využity při lokalizaci, závisí také na konkrétním mobilním zařízení (přijímači), na jeho momentální orientaci při měření apod. V důsledku všeho výše uvedeného je obtížné dosáhnout ideální přesnosti lokalizace, která by byla uvnitř budovy požadována.

Neustále je tedy snahou nalézt různá rozšíření či vylepšení existujících technik nebo techniky úplně nové, které umožní dosáhnout vyšší přesnosti lokalizace. Velice často se jedná o různé matematické a statistické metody, které pomáhají do určité míry redukovat vliv šumu v rádiových datech vzniklý v důsledku zmíněných fyzikálních principů šíření signálu.

Příkladem takovéto metody jsou právě Kalmanovy filtry. Kotanen et al. [26] představují jako jedni z prvních možnost využití rozšířeného Kalmanova filtru (EKF) v indoor lokalizaci. Jedná se o statickou lokalizaci v rámci jedné místnosti, filtr je aplikován na data o vzdálenostech lokalizovaného objektu k referenčním vysílacím bodům, které jsou vypočteny z naměřené síly Bluetooth signálu.

Dále existují dvě varianty klasického Kalmanova filtru nazvané jako *position Kalman filter* (PKF) [28], [2] a *fingerprint Kalman filter* (FKF) [1]. Oba algoritmy jsou autory původně využity ke statické lokalizaci v budově s Wi-Fi sítí. PKF má na svém vstupu již vypočtené odhady polohy (získané například pomocí techniky fingerprintingu), oproti tomu FKF pracuje přímo s naměřenými silami signálu, a má tedy lepší schopnost filtrovat šum v nich obsažený, následné určení polohy je založeno na principu fingerprintingu a algoritmu WKNN.

Tyto zmíněné přístupy jsou dále aplikovány a rozvíjeny dalšími autory, například Lee et al. [30] využívají první zmíněný přístup spolu s BLE beacony, Fang et al. [15] rozšiřují FKF o adaptivní výpočet kovariance šumu apod.

Kalmanův filtr lze využít také jako nástroj umožňující zkombinovat data různé povahy z různých zdrojů. Tohoto využívají například Coronel et al. [13], Lee et al. [29], Chen et al. [10], při řešení úlohy dynamické lokalizace kombinují rádiová data a data o pohybu z příslušných senzorů mobilních zařízení.

Kromě Kalmanových filtrů bývají v indoor lokalizaci využívány také částicové filtry. Částicové filtry mají obecnější možnosti aplikace, nevýhodou je však vyšší výpočetní náročnost limitující jejich použití v reálném čase na zařízeních s omezeným výkonem. Rovněž dosažená přesnost nebývá výrazně lepší. Setkat se lze také s lokalizačními technikami využívajícími neuronové sítě, genetické algoritmy atd.

4.2.2 Představení Kalmanových filtrů v indoor lokalizaci

V následující části textu budou popsány různé lokalizační algoritmy založené na myšlence Kalmanova filtru. Algoritmy budou vycházet z výše zmíněných článků a budou je různými způsoby upravovat a rozšiřovat. Důraz bude kladen na jejich následné otestování v reálných podmínkách, neboť některé ze zmíněných algoritmů jsou svými autory testovány pouze v přesně připravených laboratorních podmínkách (například lokalizace v jedné prázdné místnosti se čtyřmi vysílacími zařízeními v rozích apod.) nebo na simulovaných datech. Kromě původního použití pro statickou lokalizaci budou některé algoritmy vyzkoušeny rovněž při řešení úlohy lokalizace dynamicky se pohybujícího objektu. Při praktické aplikaci budou použita data z technologie BLE, popsané postupy by však bylo možné aplikovat i na data jiných rádiových technologií.

Prvním nutným krokem předcházejícím implementaci algoritmů je návrh modelu systému. Z pohledu úlohy indoor lokalizace je zřejmé, že hledaná poloha je v modelu systému neznámým vektorem \mathbf{x} . Jako vektor měření \mathbf{z} lze chápat různé údaje. Popsány budou tři algoritmy pro statickou lokalizaci a jeden algoritmus pro dynamickou lokalizaci, přičemž jako měření \mathbf{z} budou chápána vždy data jiné povahy. Pozornost je zaměřena na lokalizaci v rámci jednoho patra budovy, jedná se tedy o určení souřadnic $[x, y]$ objektu ve dvojrozměrném prostoru (v rovině).

Statická lokalizace

Při statické lokalizaci je předpokládáno, že se poloha objektu v čase nemění, z čehož vyplývá, že matice Φ_k je pro všechna t_k rovna matici jednotkové, tedy že stavová rovnice modelu systému má tvar

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix} + \mathbf{w}_{k-1}.$$

Tento předpoklad je společný všem třem dále uvedeným algoritmům.

Algoritmus 1

Původní myšlenku publikovali Kotanen et al. [26], dále tento přístup zkoumali a aplikovali například Yim et al. [49] nebo Lee et al. [30]. Algoritmus je založen na výpočtu vzdáleností k jednotlivým vysílacím zařízením (beaconům) z údajů o síle jejich signálu naměřené na lokalizovaném místě. Pro přepočítání síly signálu na vzdálenost lze použít vztah

$$d = d_0 \cdot 10^{(P_{RX}(d_0) - P_{RX}(d))/10n},$$

kde $P_{RX}(d)$ je naměřená síla signálu v lokalizovaném místě, $P_{RX}(d_0)$ je síla signálu v referenční vzdálenosti d_0 (obvykle 1 m) od beaconu a n je exponent útlumu. Konkrétní hodnoty byly zvoleny $n = 2$ a $P_{RX}(1 \text{ m}) = -62$ (hodnota odpovídá vysílacímu výkonu 0 dBm, který měly nastaveny beacony při testování [27]).

Prvky vektoru měření jsou právě tyto vypočtené vzdálenosti, rovnice měření modelu systému má pak podobu

$$\begin{pmatrix} d_{1\ k} \\ d_{2\ k} \\ \vdots \\ d_{m\ k} \end{pmatrix} = \begin{pmatrix} \sqrt{(x_1 - x_k)^2 + (y_1 - y_k)^2} \\ \sqrt{(x_2 - x_k)^2 + (y_2 - y_k)^2} \\ \vdots \\ \sqrt{(x_m - x_k)^2 + (y_m - y_k)^2} \end{pmatrix} + \mathbf{v}_k,$$

kde d_i je vzdálenost k i -tému beaconu a $[x_i, y_i]$ jsou souřadnice i -tého beaconu, $i = 1, 2, \dots, m$ (m je počet beaconů). Vzhledem k nelinearitě v rovnici měření je vhodné využít algoritmus EKF, UKF či jiný ze zmíněných nelineárních filtrů. Konkrétní hodnoty prvků matic \mathbf{Q}_k a \mathbf{R}_k pro jednotlivé časové okamžiky jsou určovány experimentálně z testovacích dat.

Výhodou tohoto algoritmu je jeho jednoduchost, neboť pro určení polohy nevyžaduje speciální předchozí přípravu, jako je například off-line fáze fingerprintingu. Na druhou stranu však nedosahuje vysoké přesnosti lokalizace, jelikož princip přepočtu naměřené

síly signálu na vzdálenost obecně vede v kontextu indoor lokalizace k významným chybám.

Algoritmus 2

Algoritmus je založen na principu PKF (*position Kalman filter*) [28], [2], neboť jako měření chápe již vypočítané odhady polohy pomocí jiné techniky (např. fingerprinting + WKNN). Na rozdíl od předchozího algoritmu se nejedná o lokalizační techniku v pravém slova smyslu, ale pouze o nástroj, který by měl zpřesnit stávající odhady. Tato schopnost je však do určité míry omezená tím, že nepracuje se surovými naměřenými daty, což umožňuje redukovat vliv šumu pouze zprostředkovaně.

Rovnice měření má jednoduchou podobu

$$\begin{pmatrix} z_{x\ k} \\ z_{y\ k} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \mathbf{v}_k.$$

Celý model systému je lineární, lze tedy použít klasický Kalmanův filtr.

Algoritmus 3

Třetím popsaným algoritmem je FKF (*fingerprint Kalman filter*), který navrhli Ali-Löytty et al. [1]. Algoritmus počítá odhady polohy přímo z dat o síle signálu, která jsou v modelu systému chápána jako měření, přičemž využívá rovněž principů techniky fingerprintingu. Nutným předpokladem jeho použití je tedy předchozí vytvoření referenční mapy otisků.

Stavová rovnice modelu má podobu stejnou jako v případě ostatních algoritmů a rovněž výpočet apriorního odhadu polohy je shodný s klasickým Kalmanovým filtrem. Rovnice měření však není explicitně vyjádřena a vztahy pro výpočet aposteriorního odhadu polohy mají mírně odlišnou podobu, princip Kalmanova filtru je ale zachován. Odlišnosti pramení ze skutečnosti, že odhad měření (který je poté využit při výpočtu aposteriorního odhadu polohy) není konstruován dle rovnice měření, ale pomocí techniky fingerprintingu a algoritmu WKNN. Podrobné odvození nových rovnic FKF je uvedeno v [1].

Dynamická lokalizace

Při dynamické lokalizaci bývá za neznámou považována nejen aktuální poloha, ale také rychlost pohybu \mathbf{u} , stavová rovnice modelu má v tomto případě tvar (rychlost je

předpokládána konstantní)

$$\begin{pmatrix} x_k \\ y_k \\ u_{x\ k} \\ u_{y\ k} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ u_{x\ k-1} \\ u_{y\ k-1} \end{pmatrix} + \mathbf{w}_{k-1}.$$

Algoritmus 4

V algoritmu jsou jako měření chápány již vypočítané odhady poloh v jednotlivých časových okamžicích, rovnice měření modelu systému má tedy podobu

$$\begin{pmatrix} z_{x\ k} \\ z_{y\ k} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ u_{x\ k} \\ u_{y\ k} \end{pmatrix} + \mathbf{v}_k.$$

Stavová rovnice i rovnice měření jsou lineární, takže je možné použít klasický Kalmanův filtr.

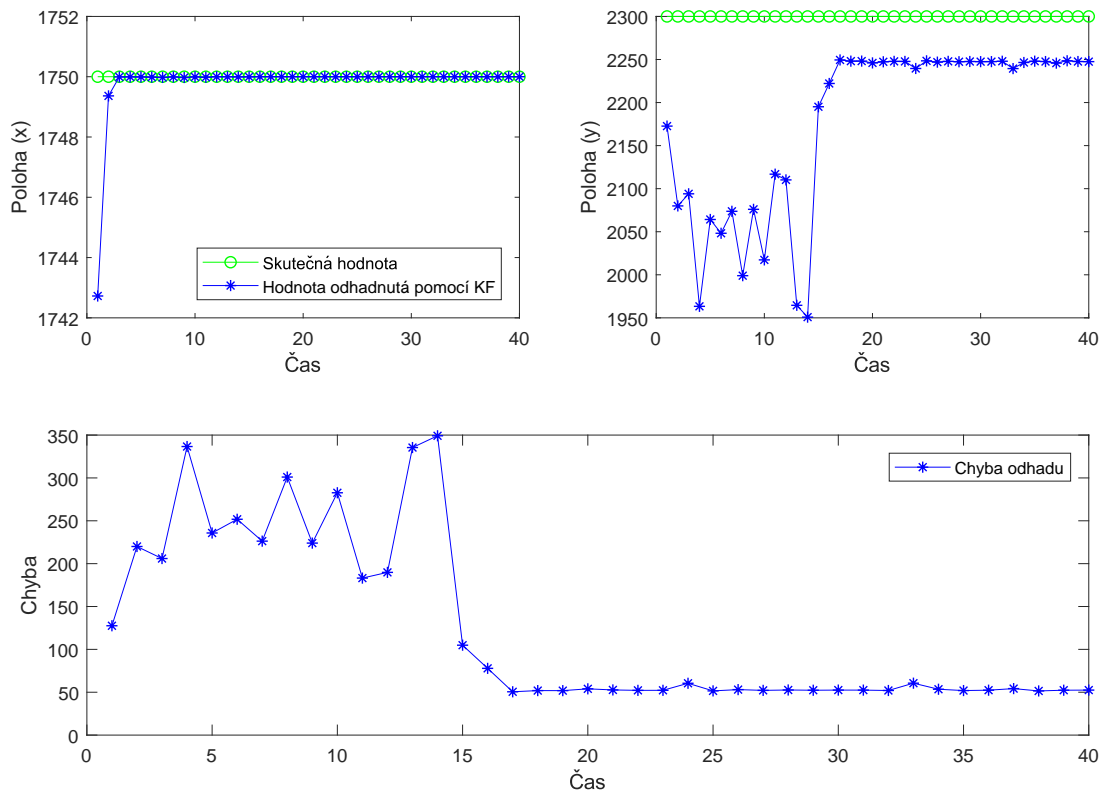
Cílem algoritmu je zpřesnit odhad trajektorie pohybu kombinací apriorní informace o očekávané poloze v novém časovém okamžiku vypočítané na základě modelu pohybu a informace o odhadu polohy získaného z dat rádiové sítě.

Algoritmus lze dále rozšířit o schopnost detekce odlehlých pozorování. Pokud je některý z jednotlivých odhadů polohy poskytnutých Kalmanovu filtru jako měření odlehlý, nebude do výpočtů filtru zahrnut, a nepoškodí tak odhad celé trajektorie pohybu. Kalmanův filtr rovněž umožňuje vyřešit situaci chybějícího měření v některém časovém okamžiku, pokud toto nastane, bude za aktuální odhad polohy považován apriorní odhad získaný dle stavové rovnice modelu systému. Tento postup rovněž umožňuje využít algoritmus k predikci budoucí polohy.

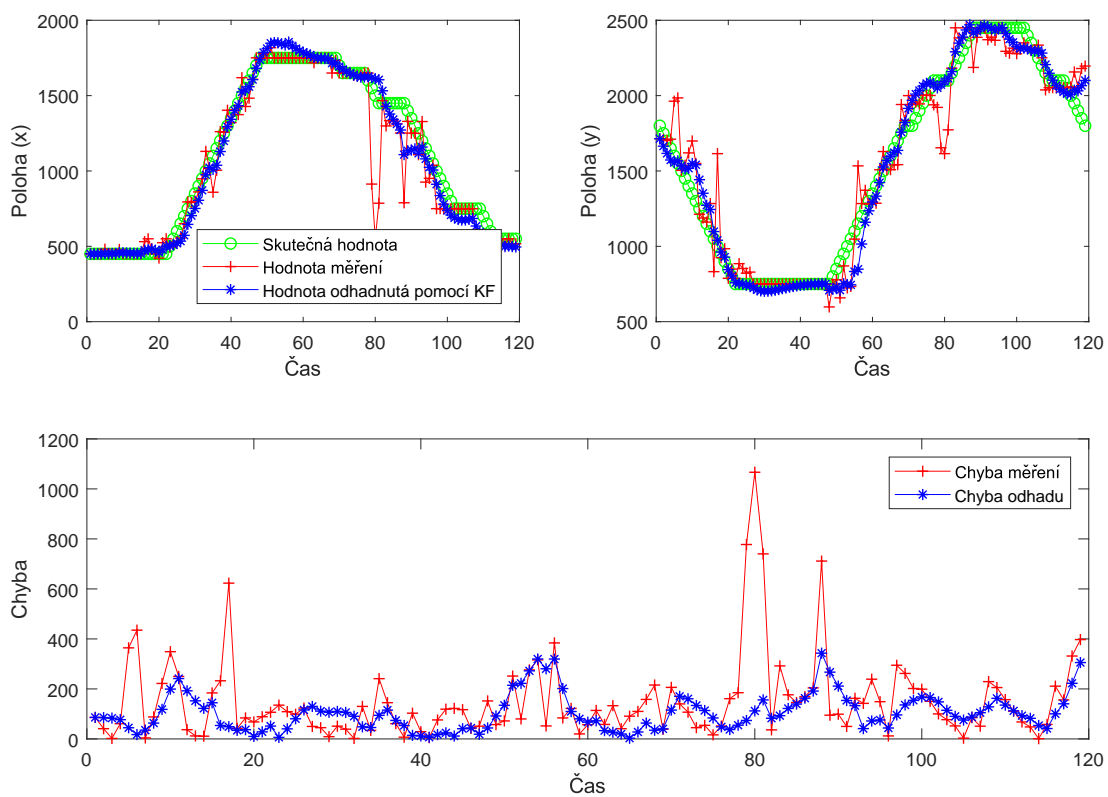
4.2.3 Implementace, testování a zhodnocení popsaných algoritmů

Implementace algoritmů byla provedena v prostředí MATLAB, zdrojové kódy jsou uvedeny v příloze A.2.1.

Pro ilustraci algoritmů statické lokalizace poslouží grafický výstup implementace algoritmu 3, který je znázorněn na obrázku 8(a). Grafy v horní části ukazují vývoj odhadů souřadnic x a y během 40 iterací algoritmu, graf v dolní části pak vývoj chyb odhadů



(a)



(b)

Obrázek 8: Grafický výstup implementace algoritmu 3 (a) a algoritmu 4 (b) v softwaru MATLAB

(měřených v jednotkách použité soustavy souřadnic, přičemž 50 jednotek odpovídá vzdálenosti 1 metr). Výsledky implementace algoritmu 4 pro dynamickou lokalizaci ilustruje obrázek 8(b). Grafy na obrázku nabízejí možnost srovnání hodnot skutečné polohy, původního odhadu polohy získaného z naměřených rádiových dat a odhadu polohy po aplikaci Kalmanova filtru, resp. popsaného algoritmu. Spodní graf pak znázorňuje původní a novou velikost chyby odhadů v jednotlivých časových okamžicích.

Testování všech algoritmů bylo provedeno na reálných datech pocházejících z prostředí budovy Fakulty informatiky a managementu Univerzity Hradec Králové (data byla nasbírána v rámci specifického výzkumu probíhajícího na FIM UHK). Ve druhém patře této budovy je rozmístěno celkem 17 BLE vysílacích zařízení Estimote Beacon (HW revize D3.4). Všechna zařízení byla nastavena na vysílací výkon 0 dBm a interval vysílání 100 ms.

Pro účely testování algoritmů statické lokalizace bylo zvoleno 20 míst, na kterých bylo provedeno souvislé měření síly signálu trvající vždy 20 sekund. Pro testování dynamické lokalizace byly vytvořeny dvě procházkové trasy, první trasa vedla okolo celého patra, druhá trasa byla rovná, vedoucí po chodbě pouze na jedné straně patra. Z důvodu využití techniky fingerprintingu byla vytvořena rovněž referenční mapa sestávající ze 171 otisků.

Na obrázku 9(a) je znázorněn plán patra budovy, vyznačeno je 20 míst, na kterých byly testovány algoritmy statické lokalizace (čtverečky), a také odhady jejich poloh získané pomocí algoritmu 3 (kroužky). Obrázek 9(b) znázorňuje první procházkovou trasu, skutečná trajektorie pohybu je znázorněna zeleně, odhady poloh získané technikou fingerprintingu červeně a výsledný odhad trajektorie dle algoritmu 4 modře.

Na připravených datech byly otestovány všechny popsané algoritmy a pro účely srovnání byla aplikována také technika fingerprintingu s algoritmem WKNN. Z výsledků byly vypočítány tyto popisné statistiky: průměrná chyba odhadu polohy (ME), odmocnina střední kvadratické chyby (RMSE), medián chyb (MDN E), maximální chyba (MAX E).

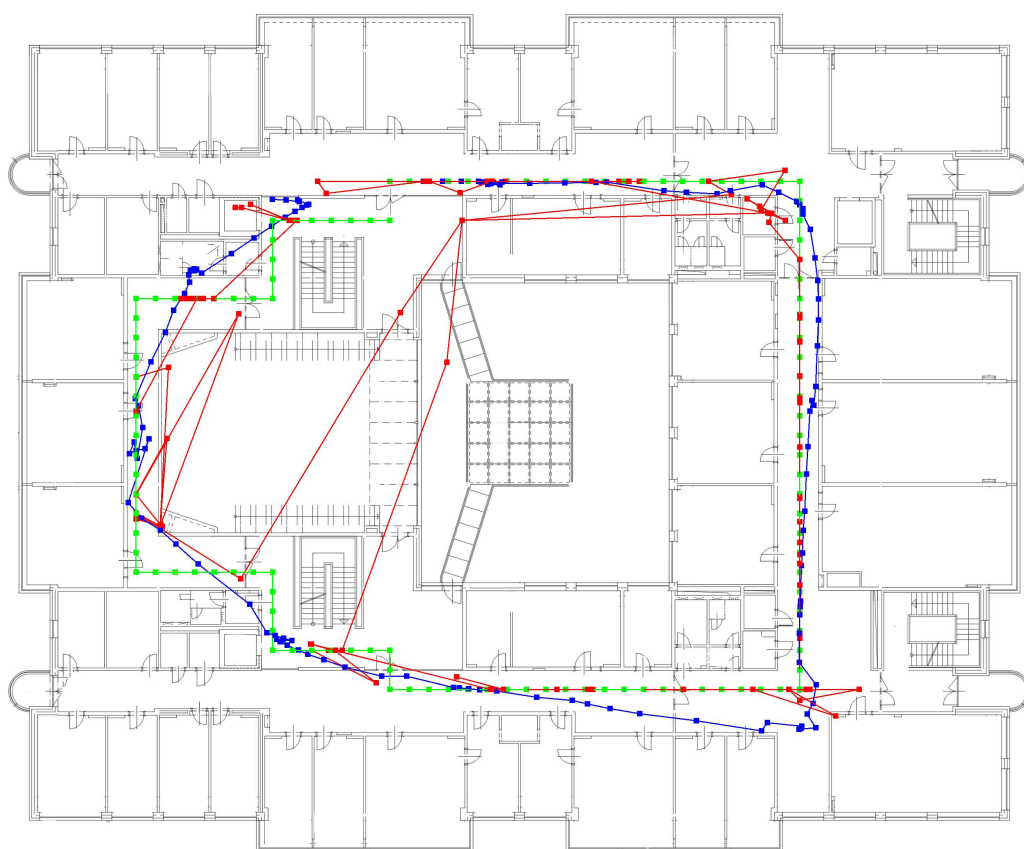
Výsledky testování a jejich zhodnocení

Statická lokalizace

Výsledky testování algoritmů statické lokalizace shrnuje tabulka 1 (hodnoty jsou uvedeny v metrech a zaokrouhleny na 2 desetinná místa).



(a)



(b)

Obrázek 9: Vizualizace výsledků testování algoritmu 3 (a) a algoritmu 4 (b)

	ME	RMSE	MDN E	MAX E
Fingerprinting	3,03	3,77	3,01	8,70
Algoritmus 1	7,85	8,73	8,35	14,60
Algoritmus 2	2,94	3,46	2,97	6,10
Algoritmus 3	2,06	2,64	1,88	6,00

Tabulka 1: Shrnutí výsledků testování algoritmů statické lokalizace

Algoritmus 1 poskytuje dle očekávání nejhorší výsledky, z analýzy dat navíc vyplývá, že rozdělení šumu měření nesplňuje předpoklady požadované Kalmanovým filtrem. Výsledky získané při použití algoritmu 2 jsou mírně lepší než ty, ke kterým vede použití pouze klasické techniky fingerprintingu, zlepšení je patrné zejména ve snížení maximální chyby, pokles průměrné chyby není tak výrazný. Algoritmus 3 dosahuje nejlepších výsledků ve všech kritériích, získaná průměrná chyba je 2 m, což je o jeden metr méně nežli v případě fingerprintingu.

Dynamická lokalizace

Tabulka 2 obsahuje shrnutí výsledků testování algoritmu 4 pro dynamickou lokalizaci a jeho srovnání s technikou fingerprintingu, tj. s původními daty, na které byl aplikován Kalmanův filtr, resp. algoritmus 4.

<i>Trasa 1</i>	ME	RMSE	MDN E	MAX E
Fingerprinting	3,01	4,48	2,13	21,33
Algoritmus 4	2,09	2,56	1,82	6,85

<i>Trasa 2</i>	ME	RMSE	MDN E	MAX E
Fingerprinting	1,63	2,09	1,37	4,68
Algoritmus 4	0,51	0,57	0,55	0,99

Tabulka 2: Shrnutí výsledků testování algoritmů dynamické lokalizace

V případě trasy 1, která vedla okolo celého patra, došlo použitím algoritmu 4 ke snížení průměrné chyby ze 3 m na 2 m, výrazně byla zredukována také maximální chyba. K výraznému snížení chyby došlo na rovných úsecích chodeb, naopak v místě rohů chyba mírně vzrostla. Toto plyne z principu fungování Kalmanova filtru – model systému udržuje trend předchozího pohybu a nepředpokládá okamžitou změnu směru pohybu zahnutím za roh, nějaký čas tedy trvá, než se filtr adaptuje na nový směr pohybu. Toto je patrné také z obrázku 9(b). Rovněž se tímto potvrzuje, že Kalmanův filtr je obecně

vhodnější ke sledování pohybu velkých či těžkých objektů, jejichž trajektorie pohybu bývá hladká.

Trasa 2 byla navržena tak, aby sloužila opět k ilustraci výše popsaného jevu. Pokud se tedy jedná o lokalizaci objektu pohybujícího se rovně neznámou konstantní rychlostí, lze dosáhnout chyby v průměru pouze okolo 0,5 m.

Popsaný problém by bylo možné částečně eliminovat využitím techniky PDR, která by nahradila výpočet apriorního odhadu polohy dle uvedené stavové rovnice modelu systému, podrobněji viz například [10], [13], [29].

Kompletní podrobné výsledky testování všech algoritmů jsou součástí elektronické přílohy práce.

4.3 Ukázky dalších aplikací Kalmanových filtrů

4.3.1 Analýza časových řad

Jak již z textu vyplývá, Kalmanův filtr lze vhodně využít při práci s časovými řadami. Obecně umožňuje řešit nejen úlohu filtrování, ale také úlohy predikování a vyrovnávání, které nabývají na významu například v souvislosti s časovými řadami socio-ekonomické povahy.

Při analýze časových řad bývají často používány tzv. ARMA (*autoregressive moving average*) modely či jejich další obecnější varianty (podrobněji viz např. [19]). Tyto modely lze však transformovat do podoby stavového modelu sestávajícího ze stavové rovnice a rovnice měření, na který lze následně aplikovat Kalmanův filtr. V souvislosti s konstrukcí modelů časových řad nachází Kalmanův filtr také další uplatnění, které spočívá ve výpočtu odhadů neznámých parametrů těchto modelů.

Prozatím budiž uvažován pouze autoregresní model řádu p AR(p)

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t,$$

kde $\{\varepsilon_t\}$ je nekorelovaný gaussovský proces s nulovou střední hodnotou a konstantním rozptylem σ^2 a $\phi_i, i = 1, \dots, p$, jsou parametry modelu. (Použité značení bylo ponecháno v souladu se zavedenou praxí modelů časových řad.)

Cílem nyní může být odhadování parametrů tohoto modelu. Při řešení této úlohy tvoří parametry neznámý stavový vektor a celá stavová rovnice (za předpokladu, že jsou

parametry v čase konstantní) a rovnice měření má potom následující podobu

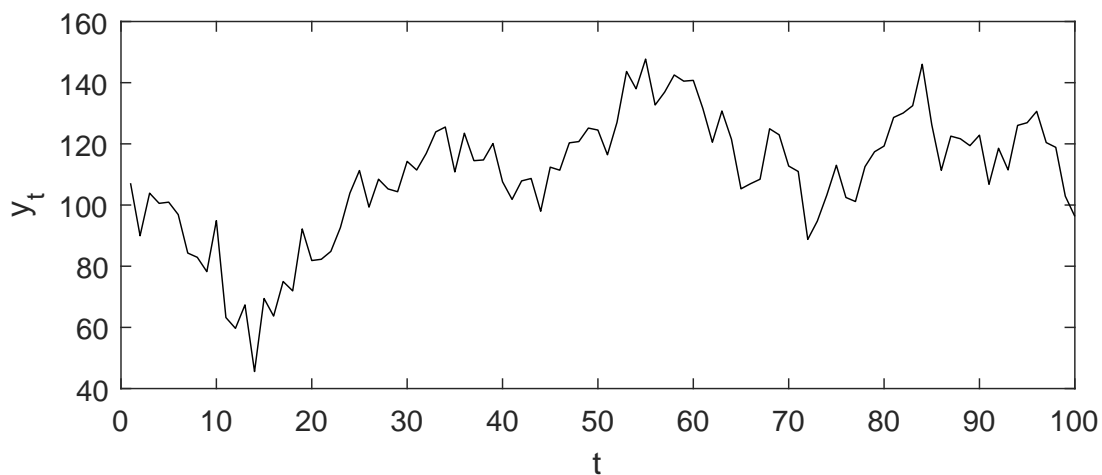
$$\begin{pmatrix} \phi_{1 t} \\ \phi_{2 t} \\ \vdots \\ \phi_{p t} \end{pmatrix} = \begin{pmatrix} \phi_{1 t-1} \\ \phi_{2 t-1} \\ \vdots \\ \phi_{p t-1} \end{pmatrix},$$

$$y_t = (y_{t-1} \quad y_{t-2} \quad \cdots \quad y_{t-p}) \begin{pmatrix} \phi_{1 t} \\ \phi_{2 t} \\ \vdots \\ \phi_{p t} \end{pmatrix} + \varepsilon_t.$$

Pro ukázkou úlohy poslouží simulovaný AR(2) proces

$$y_t = 0,7y_{t-1} + 0,3y_{t-2} + \varepsilon_t, \quad \varepsilon_t \sim N(0, 10), \quad t = 1, \dots, 100,$$

který je znázorněn na obrázku 10.



Obrázek 10: Simulovaný AR(2) proces

Na vstupu úlohy jsou k dispozici pouze hodnoty y_t , parametry ϕ_1 a ϕ_2 jsou neznámé. Dle výše uvedeného byla sestavena stavová rovnice a rovnice měření a následně aplikován Kalmanův filtr. Postupný vývoj získaných odhadů parametrů shrnuje tabulka 3 (počáteční hodnoty odhadů by bylo možné volit libovolně).

t	0	10	20	30	40	50	60	70	80	90	100
$\hat{\phi}_1$	1,000	0,521	0,458	0,541	0,523	0,553	0,572	0,631	0,680	0,706	0,704
$\hat{\phi}_2$	1,000	0,465	0,523	0,465	0,480	0,453	0,437	0,369	0,320	0,295	0,293

Tabulka 3: Odhady parametrů modelu AR(2) procesu získané pomocí Kalmanova filtru

Dále se lze zabývat řešením zmíněných úloh predikování, filtrování a vyrovnávání. Pro tyto účely je možné transformovat uvedený $AR(p)$ model do podoby stavového modelu, který může vypadat například takto

$$\begin{pmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-p+2} \\ y_{t-p+1} \end{pmatrix} = \begin{pmatrix} \phi_1 & \phi_2 & \cdots & \phi_{p-1} & \phi_p \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} y_{t-1} \\ y_{t-2} \\ \vdots \\ y_{t-p+1} \\ y_{t-p} \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \varepsilon_t,$$

$$y_t = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \end{pmatrix} \begin{pmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-p+2} \\ y_{t-p+1} \end{pmatrix}.$$

Do podoby stavového modelu lze převést i obecnější $ARMA(p, q)$ model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}.$$

Výsledkem pak může být například následující stavová rovnice a rovnice měření [14]

$$\boldsymbol{\alpha}_t = \begin{pmatrix} \phi_1 & 1 & 0 & \cdots & 0 \\ \phi_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{n-1} & 0 & \cdots & 0 & 1 \\ \phi_n & 0 & 0 & \cdots & 0 \end{pmatrix} \boldsymbol{\alpha}_{t-1} + \begin{pmatrix} 1 \\ \theta_1 \\ \vdots \\ \theta_{n-2} \\ \theta_{n-1} \end{pmatrix} \varepsilon_t,$$

$$y_t = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \end{pmatrix} \boldsymbol{\alpha}_t,$$

kde

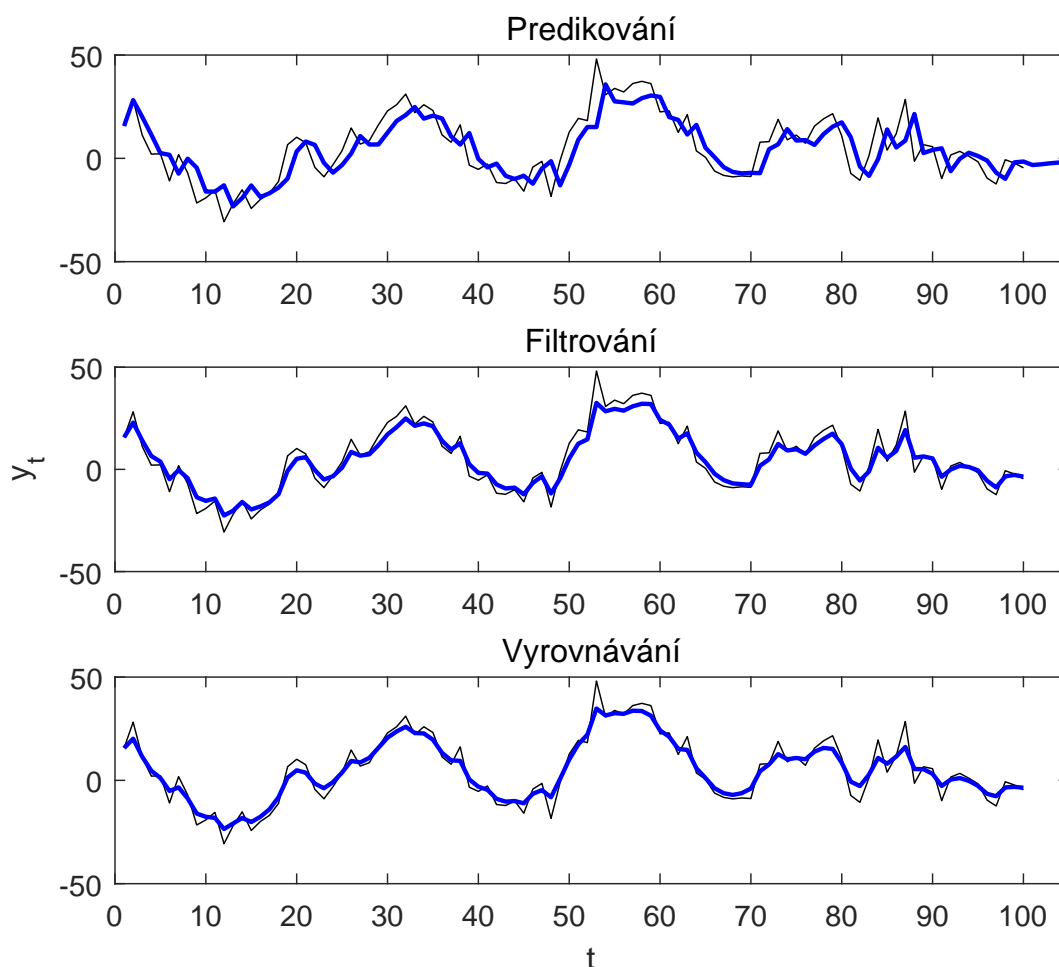
$$\boldsymbol{\alpha}_t = \begin{pmatrix} y_t \\ \phi_2 y_{t-1} + \cdots + \phi_p y_{t-n+1} + \theta_1 \varepsilon_t + \cdots + \theta_{n-1} \varepsilon_{t-n+2} \\ \vdots \\ \phi_n y_{t-1} + \theta_{n-1} \varepsilon_t \end{pmatrix}, \quad n = \max(p, q + 1)$$

Nutné je ale poznamenat, že existuje více alternativních stavových modelů reprezentujících tentýž $ARMA$ model, které se navzájem liší definicí stavového vektoru apod. Tyto různé další přístupy shrnuje například [21]. Obecně však takto definované stavové vektory nemívají věcnou interpretaci.

Úlohy predikování, filtrování a vyrovňování budou demonstrovány na simulovaném ARMA(2,1) procesu

$$y_t = 0,6y_{t-1} + 0,2y_{t-2} + \varepsilon_t + 0,1\varepsilon_{t-1}, \quad \varepsilon_t \sim N(0, 10), \quad t = 1, \dots, 100.$$

Tento ARMA model byl dle výše uvedeného vzoru převeden do podoby stavového modelu. Úloha filtrování byla řešena standardním aplikováním algoritmu Kalmanova filtru tak, jak tomu bylo doposud. Řešení úlohy predikování spočívá ve využití pouze predikčního kroku algoritmu, který však není následován krokem korekčním, neboť pozorování, které by mohlo apriorní odhad zlepšit, ještě není k dispozici (v ukázce byly počítány jednokrokové predikce ve sledovaném období a dále predikce pro pět budoucích časových okamžiků). Úlohu vyrovňování lze řešit různými způsoby, zde byl využit algoritmus nazývaný *Rauch–Tung–Striebel smoother*, který v prvním běhu aplikuje na pozorovaná data standardní Kalmanův filtr a následně v druhém běhu postupuje rekurzivně od konce a kombinací filtrovaných hodnot vytváří hodnoty vyrovnané, podrobněji viz například [18], [35]. Dosažené výsledky shrnuje obrázek 11.



Obrázek 11: Ukázka použití Kalmanova filtru pro predikování, filtrování a vyrovňování v časové řadě reprezentované ARMA(2,1) modelem

Zdrojové kódy implementací obou příkladů jsou uvedeny v příloze A.2.2.

Podrobněji se popsané problematice a souvisejícím možnostem jejího využití (zejména v oblasti ekonomie) věnuje třeba [12], [14], [19]. Kromě analýzy časových řad lze Kalmanův filtr použít také v regresi, více viz například [14].

4.3.2 Zpracování obrazu

Kalmanův filtr nachází své využití také ve zpracování obrazu, i když se nejedná o jeho typickou aplikační oblast. Černobílý (šedotónový) digitální obraz lze přirozeně reprezentovat dvourozměrnou maticí, jejíž prvky vyjadřují hodnoty intenzit jednotlivých pixelů. Použití Kalmanova filtru v úlohách zpracování obrazu tak vyžaduje jeho rozšíření z náhodných procesů na koncept dvourozměrných náhodných polí.

Jednou z modifikací Kalmanova filtru navrženou pro tyto účely je tzv. *reduced update Kalman filter* (RUKF), který publikovali Woods a Radewan [47]. Tento algoritmus byl původně určen k filtraci šumu v obraze, po čase ho ale Woods a Ingle [46] dále rozpracovali a rozšířili i na použití pro rekonstrukci rozmazaného obrazu.

Algoritmus vychází z následujícího dvourozměrného autoregresního (2D AR) modelu obrazu (stavová rovnice)

$$x(i, j) = \sum_{(k, l) \in \mathcal{D}} \phi(k, l)x(i - k, j - l) + w(i, j),$$

kde notace $x(i, j)$ vyjadřuje pixel ideálního nezašuměného obrazu nacházející se na pozici (i, j) , $w(i, j)$ označuje systémový „šum“ odpovídající nepřesnostem modelu a $\phi(k, l)$ reprezentuje příslušný parametr autoregresního modelu, přičemž se předpokládá, že $\mathcal{D} = \{k \geq 0, l \geq 0\} \cup \{k > 0, l < 0\}$. (Použití značení bylo ponecháno v souladu s původními články.)

Zatížení obrazu aditivním šumem měření lze modelovat touto skalární rovnicí (rovnice měření)

$$z(i, j) = x(i, j) + v(i, j),$$

kde $z(i, j)$ značí pixel zašuměného obrazu a $v(i, j)$ je šum vzniklý obvykle v důsledku technických principů pořizování obrazu.

Dle předpokladů vyžadovaných Kalmanovým filtrem mají w i v vlastnosti gaussovského bílého šumu.

Někdy se může stát, že je obraz kromě šumu degradován také rozmazáním, v tomto případě má rovnice měření tvar

$$z(i, j) = \sum_{(k,l)} h(k, l)x(i - k, j - l) + v(i, j),$$

kde h reprezentuje rozmazání obrazu způsobené například pohybem či špatně zaostřenou optickou soustavou fotoaparátu. Uvedenou rovnici lze vnímat také jako vyjádření dvourozměrné diskrétní konvoluce obrazu x s konvolučním jádrem h .

K řešení obou úloh lze využít zmíněný *reduced update Kalman filter*. Tento algoritmus je založen na sekvenčním procházení (skenování) obrazu pixel po pixelu počínaje v levém horním rohu obrazu a pokračuje po jednotlivých řádcích. V každém okamžiku lze tedy aktuálně zpracováváný pixel vnímat jako „přítomnost“, doposud zpracované pixely jako „minulost“ a nadcházející pixely jako „budoucnost“. Tímto přístupem je dvourozměrná úloha transformována na jednorozměrnou a lze na ni použít klasický Kalmanův filtr. Problémem však je, že stavový vektor, který je tvořen jednotlivými pixely obrazu, nabývá velkého počtu prvků, což má za následek vysoké výpočetní nároky. RUKF řeší tuto situaci tím, že nepracuje vždy s celým obrazem, ale Kalmanův filtr aplikuje pouze na určité okolí aktuálně procházeného pixelu. Podrobný popis algoritmu a odvození filtračních rovnic lze nalézt ve zmíněných článcích [47], [46] nebo v [45].

K praktické demonstraci algoritmu RUKF byla využita fotografie Velkého náměstí v Hradci Králové (obrázek 12(a)). V prvním experimentu byl do fotografie přidán aditivní gaussovský bílý šum (obrázek 12(b)). Množství šumu v obraze lze vyjádřit pomocí poměru SNR (*signal-to-noise ratio*), který definuje například [45] jako

$$\text{SNR} = 10 \log \left(\frac{\text{rozptyl obrazu}}{\text{rozptyl šumu}} \right)$$

(vzorec předpokládá, že obraz i šum mají nulovou střední hodnotu; SNR je takto vyjádřeno v decibelech). V případě obrázku 12(b) nabývá SNR hodnoty 10,77 dB. Výsledek použití RUKF znázorňuje obrázek 12(c), SNR vzrostlo na 15,36 dB, zlepšení (někdy též označované jako *improvement in signal-to-noise ratio* ISNR) je tedy 4,59 dB.

Při druhém experimentu byla původní fotografie navíc degradována rozmazáním vzniklým pohybem v horizontálním směru (obrázek 13(b)). Výsledek aplikování RUKF je k vidění na obrázku 13(c).

Příslušné zdrojové kódy skriptů v MATLABu jsou v příloze A.2.3.

Rozšíření představeného algoritmu či jiné přístupy využívající Kalmanův filtr pro filtraci a rekonstrukci obrazu uvádí například [47] (*Kalman strip filter*), [25] (*reduced order model Kalman filter*), [48] (*fast modified RUKF*). Kalmanův filtr lze využít také pro nalezení parametrů dvourozměrného autoregresního modelu, viz třeba [50].



(a)



(b)



(c)

Obrázek 12: Filtrace šumu v obraze pomocí RUKF: (a) původní ideální obraz, (b) obraz zatížený aditivním šumem, (c) výsledek použití RUKF



(a)



(b)



(c)

Obrázek 13: Rekonstrukce rozmazaného obrazu pomocí RUKF: (a) původní ideální obraz, (b) obraz degradovaný rozmazáním, (c) výsledek použití RUKF

5 Shrnutí výsledků

Práce představila Kalmanův filtr a další z něj vycházející algoritmy spolu s možnostmi jejich využití v různých aplikačních doménách. První část byla zaměřena na podrobný teoretický výklad algoritmu Kalmanova filtru a na diskuzi praktických aspektů jeho implementace. Následně byly zmíněny také další Kalmanovy filtry a jejich vhodnost pro řešení určitých specifických typů úloh. Ve druhé části práce bylo popsáno a prakticky ukázáno použití Kalmanova filtru v indoor lokalizaci a dále při analýze časových řad a zpracování obrazu. Představené algoritmy a postupy byly implementovány v prostředí MATLAB, výstupy pak byly prezentovány v textu práce.

Ze zmíněných aplikačních úloh byla největší pozornost věnována indoor lokalizaci, neboť lokalizace, navigace a sledování pohybu objektů jsou typickými oblastmi, ve kterých Kalmanův filtr nachází své uplatnění, zároveň je dnes indoor lokalizace aktuálním a diskutovaným tématem. Ukázáno bylo několik algoritmů, které využívaly Kalmanův filtr jak při řešení úlohy statické lokalizace, tak i při lokalizaci dynamicky se pohybujícího objektu. Důraz byl následně kladen na jejich otestování na datech pocházejících z reálného prostředí.

Výsledky ukázaly, že použitím Kalmanova filtru lze v případě obou těchto úloh dosáhnout průměrné chyby lokalizace dva metry, což je o jeden metr méně než při použití klasické lokalizační techniky fingerprintingu. Výhody využití Kalmanova filtru v indoor lokalizaci lze vnímat také v přirozené schopnosti vyloučit z výpočtů odlehlá pozorování a dále v nepříliš vysoké výpočetní náročnosti umožňující jeho použití v reálném čase.

V kontextu indoor lokalizace je obvykle požadována vysoká přesnost určení polohy, dosažení lepších výsledků je však limitováno použitím dat technologie BLE, která jsou zatížena důsledky fyzikálních principů šíření rádiového signálu uvnitř budov. Zde získané výsledky jsou v souladu s výstupy publikovanými jinými autory.

Kalmanův filtr je ze své podstaty vhodný zejména pro dynamickou lokalizaci, při lokalizaci pohybujících se lidí či objektů v budově se však naráží na situace, kdy sledovaný ostře změni směr pohybu například zahnutím za roh, což způsobí momentální nárůst

chyby odhadů v daném místě odhadované trajektorie pohybu. Jak již bylo poznamenáno, negativní vlivy tohoto jevu by bylo možné redukovat využitím techniky PDR. Zároveň se tímto potvrdilo, že je Kalmanův filtr vhodnější pro lokalizaci větších či těžších objektů, které se pohybují po hladké trajektorii.

Ukázky aplikací Kalmanova filtru v analýze časových řad a ve zpracování obrazu měly za cíl demonstrovat širší uplatnění tohoto algoritmu. Práce s časovými řadami je další typickou oblastí, ve které je Kalmanův filtr v praxi hojně využíván – zde uvedené příklady naznačily jeho využití při konstrukci modelů časových řad a při úlohách predikování, filtrování a vyrovnávání.

Naopak zpracování obrazu se neřadí mezi obvyklé aplikační domény. Jak bylo ukázáno, Kalmanův filtr lze použít pro filtraci šumu v obraze a také pro rekonstrukci rozmazaného obrazu. Dosažené výsledky jsou srovnatelné s použitím jiných obrazových filtrů. Nevýhodou představeného algoritmu je vysoká výpočetní náročnost, kterou je však částečně možné snížit nahrazením opakovaného výpočtu Kalmanova zisku použitím jeho ustálené hodnoty, tzv. *steady-state RUKF*.

6 Závěry a doporučení

Kalmanův filtr měl své nezastupitelné místo již před padesáti lety při řešení vesmírného projektu Apollo a široké možnosti uplatnění nachází i dnes. Snahou práce bylo přiblížit tento algoritmus a princip jeho fungování a zároveň ukázat některé vybrané aplikační oblasti.

Problematika Kalmanovy filtrace je však mnohem širší. Kromě klasického Kalmanova filtru se lze zabývat množstvím dalších Kalmanových filtrů, řešit otázku efektivní implementace těchto algoritmů a posuzovat jejich výhody a nevýhody v konfrontaci s praktickými aplikačními úlohami.

Další pozornost by mohla být věnována i v práci představeným úlohám, například v souvislosti s indoor lokalizací by bylo možné prozkoumat přínos zmíněné techniky PDR při sledování dynamicky se pohybujícího objektu či dále rozvinout téma časových řad a zaměřit se podrobněji zejména na úlohy predikování a vyrovnávání (vyrovnávání by bylo možné aplikovat i v kontextu indoor lokalizace, jednalo by se však o následné zpracování dat, nikoli o určování polohy v reálném čase). Možnosti uplatnění Kalmanova filtru či obecně Kalmanových filtrů však tímto zdaleka nejsou vyčerpány.

Literatura

- [1] ALI-LÖYTTY, Simo, PERALA, Tommi, HONKAVIRTA, Ville, PICHÉ, Robert. Fingerprint Kalman filter in indoor positioning applications. In: *2009 IEEE Control Applications, (CCA) & Intelligent Control, (ISIC), St. Petersburg, Russia, July 8 - 10, 2009*. Los Alamitos, California: IEEE Computer Society Press, 2009, s. 1678 – 1683. ISSN 1085-1992.
- [2] ALI-LÖYTTY, Simo, SIROLA, Niilo, PICHÉ, Robert. Consistency of three Kalman filter extensions in hybrid navigation. *European Journal of Navigation*, 2006, vol. 4, no. 1, s. 33 – 40. ISSN 1571-473X.
- [3] ANDEĚL, Jiří. *Statistické metody*. 4. vyd. Praha: Matfyzpress, 2007. ISBN 80-7378-003-8.
- [4] BAHL, Paramvir, PADMANABHAN, Venkata N. RADAR: An in-building RF-based user location and tracking system. In: *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Tel Aviv, Israel, March 26 – 30, 2000*. Los Alamitos, California: IEEE Computer Society Press, 2000, vol. 2, s. 775 – 784. ISSN 0743-166X.
- [5] BEČVÁŘ, Jindřich. *Lineární algebra*. 4. vyd. Praha: Matfyzpress, 2010. ISBN 978-80-7378-135-4.
- [6] BERTEIN, Jean-Claude, CESCHI, Roger. *Processus stochastiques et filtrage de Kalman*. 1ère éd. Paris: Hermès, 1998. Collection traitement du signal. ISBN 978-2-86601-699-8.
- [7] BIERMAN, Gerald J. *Factorization methods for discrete sequential estimation*. 1st ed. New York: Academic Press, 1977. Mathematics in Science and Engineering. ISBN 0-12-097350-2.
- [8] BROWN, Robert G., HWANG, Patrick Y. C. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises*. 4th ed. Hoboken, N.J.: Wiley, 2012. ISBN 978-0-470-60969-9.

- [9] CARLSON, Neal A. Fast triangular formulation of the square root filter. *AIAA Journal*, 1973, vol. 11, no. 9, s. 1259 – 1265. ISSN 0001-1452.
- [10] CHEN, Zhenghua, ZOU, Han, JIANG, Hao, ZHU, Qingchang, SOH, Yeng, XIE, Lihua. Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization. *Sensors*, 2015, vol. 15, no. 1, s. 715 – 732. ISSN 1424-8220.
- [11] CHUI, Charles K., CHEN, Guanrong. *Kalman filtering with real-time applications*. 4th ed. Berlin: Springer, 2009. ISBN 978-3-540-87848-3.
- [12] CIPRA, Tomáš. *Finanční ekonometrie*. 2. vyd. Praha: Ekopress, 2013. ISBN 978-80-86929-93-4.
- [13] CORONEL, Pedro, FURRER, Simeon, SCHOTT, Wolfgang, WEISS, Beat. Indoor location tracking using inertial navigation sensors and radio beacons. In: *The Internet of Things: First International Conference, IOT 2008, Zurich, Switzerland, March 26 - 26, 2008*. Berlin, Heidelberg: Springer, 2008, s. 325 – 340. ISSN 0302-9743.
- [14] DURBIN, James, KOOPMAN, Siem J. *Time series analysis by state space methods*. 2nd ed. Oxford: Oxford University Press, 2012. ISBN 978-0-19-964117-8.
- [15] FANG, Xuming, NAN, Lei, JIANG, Zonghua, CHEN, Lijun. Noise-aware fingerprint localization algorithm for wireless sensor network based on adaptive fingerprint Kalman filter. *Computer Networks*, 2017, vol. 124, s. 97 – 107. ISSN 1389-1286.
- [16] GELB, Arthur. *Applied optimal estimation*. 1st ed. Cambridge, MA: M.I.T. Press, 1974. ISBN 0-262-20027-9.
- [17] GOLUB, Gene H., VAN LOAN, Charles F. *Matrix computations*. 4th ed. Baltimore, Md.: Johns Hopkins University Press, 2013. Johns Hopkins Studies in the Mathematical Sciences. ISBN 978-1-4214-0794-4.
- [18] GREWAL, Mohinder S., ANDREWS Angus P. *Kalman filtering: theory and practice using MATLAB*. 3rd ed. Hoboken, N.J.: Wiley, 2008. ISBN 978-0-470-17366-4.
- [19] HARVEY, Andrew C. *Forecasting, structural time series models and the Kalman filter*. 1st ed. (Reprinted). Cambridge: Cambridge University Press, 1990. ISBN 0-521-32196-4.
- [20] HIGHAM, Nick. *The Top 10 Algorithms in Applied Mathematics* [online]. 2016-03-29 [cit. 2017-11-18]. Dostupné z <http://nickhigham.wordpress.com/2016/03/29/the-top-10-algorithms-in-applied-mathematics>

- [21] JONG, Piet de, PENZER, Jeremy. The ARMA model in state space form. *Statistics & probability letters*, 2004, vol. 70, no. 1, s. 119 – 125. ISSN 0167-7152.
- [22] JULIER, Simon J., UHLMANN, Jeffrey K. A new extension of the Kalman filter to nonlinear systems. In: *Signal Processing, Sensor Fusion, and Target Recognition VI*. International Society for Optics and Photonics, 1997, s. 182 – 193. ISBN 978-0-8194-2483-9.
- [23] KALMAN, Rudolf E. A new approach to linear filtering and prediction problems. *Transactions of the American Society of Mechanical Engineers, Series D: Journal of Basic Engineering*, 1960, vol. 82, no. 1, s. 35 – 45. ISSN 0021-9223.
- [24] KALMAN, Rudolf E., BUCY, Richard S. New results in linear filtering and prediction theory. *Transactions of the American Society of Mechanical Engineers, Series D: Journal of Basic Engineering*, 1961, vol. 83, no. 1, s. 95 – 108. ISSN 0021-9223.
- [25] KIM, Jaemin, WOODS, John W. A new interpretation of ROMKF. *IEEE Transactions on Image Processing*, 1997, vol. 6, no. 4, s. 599 – 601. ISSN 1057-7149.
- [26] KOTANEN, Antti, HÄNNIKÄINEN, Marko, LEPPÄKOSKI, Helena, HÄMÄLÄINEN, Timo D. Experiments on local positioning with Bluetooth. In: *Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, USA, April 28 – 30, 2003*. Los Alamitos, California: IEEE Computer Society Press, 2003, s. 297 – 303. ISBN 0-7695-1916-4.
- [27] KRAWIEC, Piotr. Measured power values. *Estimote, Inc.* [online] 2016-01-16. [cit. 2018-02-01]. Dostupné z <http://forums.estimote.com/t/measured-power-values/2977>
- [28] KWON, Jaimyoung, DUNDAR, Baris, VARAIYA, Pravin. Hybrid algorithm for indoor positioning using wireless LAN. In: *Proceedings of the IEEE 60th Vehicular Technology Conference, Los Angeles, USA, Sept. 26 – 29, 2004*. Los Alamitos, California: IEEE Computer Society Press, 2004, s. 4625 – 4629. ISBN 0-7803-8521-7.
- [29] LEE, Sangwoo, CHO, Bongkwan, KOO, Bonhyun, RYU, Sanghwan, CHOI, Jaehoon, KIM, Sunwoo. Kalman filter-based indoor position tracking with self-calibration for RSS variation mitigation. *International Journal of Distributed Sensor Networks*, 2015, vol. 11, no. 8, s. 1 – 10. ISSN 1550-1477.

- [30] LEE, Seoung-Hyeon, LIM, Il-Kwan, LEE, Jae-Kwang. Method for improving indoor positioning accuracy using extended Kalman filter. *Mobile Information Systems*, 2016, vol. 2016, s. 1 – 15. ISSN 1574-017X.
- [31] LEWIS, Frank L., XIE, Lihua, POPA, Dan. *Optimal and robust estimation: with an introduction to stochastic control theory*. 2nd ed. Boca Raton: CRC Press, 2008. ISBN 978-0-8493-9008-1.
- [32] The MathWorks, Inc. *MATLAB R2017b Documentation: Functions* [online]. 2017 [cit. 2018-02-08]. Dostupné z <http://www.mathworks.com/help/matlab/functionlist.html>
- [33] MAYBECK, Peter S. *Stochastic models, estimation and control, Volume I*. 1st ed. New York: Academic Press, 1979. ISBN 0-12-480701-1.
- [34] POTTER, James E. *W Matrix Augmentation*. M.I.T. Instrumentation Laboratory Memo SGA 5-64. Cambridge, MA: Massachusetts Institute of Technology, 1964.
- [35] RAUCH, Harry E., STRIEBEL, Charlotte T., TUNG, Frank. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 1965, vol. 3, no. 8, s. 1445 – 1450. ISSN 0001-1452.
- [36] ROHN, Jiří. *Lineární algebra a optimalizace*. 1. vyd. Praha: Karolinum, 2004. ISBN 80-246-0932-0.
- [37] SCHMIDT, Stanley F. Application of state-space methods to navigation problems. In: *Advances in control systems: theory and applications, Volume 3*. C. T. Leondes (ed.). 1st ed. New York: Academic Press, 1966, s. 293 – 340. ISBN 978-1-4831-6716-9.
- [38] SCHMIDT, Stanley F. Computational techniques in Kalman filtering. In: *Theory and Applications of Kalman Filtering*. C. T. Leondes (ed.). London: AGARDograph 139, NATO Advisory Group for Aerospace Research and Development, 1970.
- [39] SIMON, Dan. *Optimal state estimation: Kalman, H_∞ and nonlinear approaches*. 1st ed. Hoboken, N.J.: Wiley, 2006. ISBN 978-0-471-70858-2.
- [40] THORNTON, Catherine L. *Triangular covariance factorizations for Kalman filtering*. Pasadena, CA, 1976. Ph.D. thesis, University of California.
- [41] THORNTON, Catherine L., BIERMAN, Gerald J. *A numerical comparison of discrete Kalman filtering algorithms: An orbit determination case study*. JPL Technical Memorandum 33-771, NASA/JPL. Pasadena, CA: California Institute of Technology, 1976.

- [42] VERHAEGEN, Michel, VAN DOOREN, Paul. Numerical aspects of different Kalman filter implementations. *IEEE Transactions on Automatic Control*, 1986, vol. 31, no. 10, s. 907 – 917. ISSN 0018-9286.
- [43] WELCH, Greg, BISHOP, Gary. *SIGGRAPH 2001, Course 8: An introduction to the Kalman filter* [online]. Chapel Hill: University of North Carolina, 2001 [cit. 2017-07-08]. Dostupné z http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf
- [44] WERNER, Martin. *Indoor location-based services: prerequisites and foundations*. 1st ed. Cham: Springer, 2014. ISBN 978-3-319-10698-4.
- [45] WOODS, John W. *Multidimensional signal, image, and video processing and coding*. 2nd ed. Boston: Academic Press, 2012. ISBN 978-0-12-381420-3.
- [46] WOODS, John W., INGLE, Vinay K. Kalman filtering in two dimensions: further results. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1981, vol. 29, no. 2, s. 188 – 197. ISSN 0096-3518.
- [47] WOODS, John W., RADEWAN, Clark H. Kalman filtering in two dimensions. *IEEE Transactions on Information Theory*, 1977, vol. 23, no. 4, s. 473 – 482. ISSN 0018-9448.
- [48] WU, Wen-Rong, KUNDU, Amlan. Image estimation using fast modified reduced update Kalman filter. *IEEE Transactions on Signal Processing*, 1992, vol. 40, no. 4, s. 915 – 926. ISSN 1053-587X.
- [49] YIM, Jaegeol, PARK, Chansik, JOO, Jaehun, JEONG, Seunghwan. Extended Kalman filter for wireless LAN based indoor positioning. *Decision Support Systems*, 2008, vol. 45, no. 4, s. 960 – 971. ISSN 0167-9236.
- [50] ZEINALI, Mahdi, SHAFIEE, Masoud. A new Kalman filter based 2D AR model parameter estimation method. *IETE Journal of Research*, 2017, vol. 63, no. 2, s. 151 – 159. ISSN 0377-2063.

Použité značení

\mathbb{R}	Množina reálných čísel
\mathbf{A}	Matice
$a_{[ij]}$	Prvek matice \mathbf{A} na pozici ij
$\mathbf{a}_{[j]}$	j -tý sloupec matice \mathbf{A}
\mathbf{A}^T	Transponovaná matice k matici \mathbf{A}
\mathbf{A}^{-1}	Inverzní matice k regulární matici \mathbf{A}
\mathbf{I}	Jednotková matice
$\det \mathbf{A}$	Determinant matice \mathbf{A}
\mathbf{x}	Sloupcový vektor
\mathbf{x}^T	Transponovaný vektor k vektoru \mathbf{x} (řádkový vektor)
$x_{[i]}$	Prvek vektoru \mathbf{x} na pozici i
X, x resp. \mathbf{X}, \mathbf{x}	Náhodná veličina resp. vektorová náhodná veličina (nebo dle kontextu též jejich realizace)
$F(x)$	Distribuční funkce náhodné veličiny X
$f_X(x), f(x)$	Hustota pravděpodobnosti náhodné veličiny X
$f_{X,Y}(x, y)$	Sdružená hustota pravděpodobnosti náhodných veličin X, Y
$f_{X Y}(x y)$	Podmíněná hustota pravděpodobnosti náhodné veličiny X za podmínky, že náhodná veličina Y nabyla realizace y
$E\langle X \rangle$	Střední hodnota náhodné veličiny X
$\{X(t), t \in T\}, \{X(t)\}$	Náhodný proces
$\frac{d}{dt}x(t), \dot{x}(t)$	Derivace x podle času t
$\mathbf{x}(t_k), \mathbf{x}_k$	Vektor \mathbf{x} v čase t_k
$\hat{\mathbf{x}}_{k(-)}$	Apriorní odhad vektoru \mathbf{x} v čase t_k
$\hat{\mathbf{x}}_{k(+)}$	Aposteriorní odhad vektoru \mathbf{x} v čase t_k

Seznam použitých zkratek

ARMA	Autoregressive moving average
BLE	Bluetooth Low Energy
EKF	Extended Kalman filter
FKF	Fingerprint Kalman filter
GPS	Global positioning system
IEKF	Iterated extended Kalman filter
IMU	Inertial measurement unit
ISNR	Improvement in signal-to-noise ratio
KNN	k nearest neighbors
LBS	Location-based services
LQG	Linear-quadratic Gaussian
MAP	Maximum a posteriori probability
MMSE	Minimum mean square error
MSE	Mean square error
MVU	Minimum variance unbiased
NN	Nearest neighbor
PDR	Pedestrian dead reckoning
PKF	Position Kalman filter
RSSI	Received signal strength indicator
RUKF	Reduced update Kalman filter
SNR	Signal-to-noise ratio
UKF	Unscented Kalman filter
WKNN	Weighted k nearest neighbors

Seznam obrázků

1	Schematické znázornění deterministického lineárního dynamického systému ve spojitém čase	10
2	Schéma použití Kalmanova filtru	14
3	Cyklus predikce – korekce	20
4	Podmíněná hustota pravděpodobnosti polohy založená na měření z_1 . .	25
5	Podmíněná hustota pravděpodobnosti polohy založená na měření z_2 . .	26
6	Podmíněná hustota pravděpodobnosti polohy založená na měření z_1 i z_2	27
7	Propagace podmíněné hustoty pravděpodobnosti v čase	28
8	Grafický výstup implementace algoritmu 3 a algoritmu 4 v softwaru MATLAB	48
9	Vizualizace výsledků testování algoritmu 3 a algoritmu 4	50
10	Simulovaný AR(2) proces	53
11	Ukázka použití Kalmanova filtru pro predikování, filtrování a vyrovnávání v časové řadě reprezentované ARMA(2,1) modelem	55
12	Filtrace šumu v obraze pomocí RUKF	58
13	Rekonstrukce rozmazaného obrazu pomocí RUKF	59

Seznam tabulek

1	Shrnutí výsledků testování algoritmů statické lokalizace	51
2	Shrnutí výsledků testování algoritmů dynamické lokalizace	51
3	Odhady parametrů modelu AR(2) procesu získané pomocí Kalmanova filtru	53

Přílohy

A Zdrojové kódy funkcí a skriptů v MATLABu

Příloha obsahuje zdrojové kódy implementací v práci představených algoritmů a aplikačních úloh. Jedná se o vlastní implementace, všechny kódy byly vytvořeny a otestovány v prostředí MATLAB R2017b a nevyžadují přítomnost žádného toolboxu. Použité datové soubory jsou součástí elektronické přílohy práce.

Ve zdrojových kódech jsou využity následující předdefinované funkce MATLABu, více viz [32].

<code>abs(x)</code>	Vrací absolutní hodnotu prvků vektoru \mathbf{x}
<code>all(x)</code>	Zjišťuje, zdali mají všechny prvky logického vektoru \mathbf{x} hodnotu 1 (pravda)
<code>class(object)</code>	Určuje třídu objektu
<code>csvread(filename)</code>	Načítá data z CSV (comma-separated values) souboru
<code>diag(x)</code>	Vytváří diagonální matici, jejíž hlavní diagonála je tvořena prvky vektoru \mathbf{x}
<code>det(A)</code>	Počítá determinant čtvercové matice \mathbf{A}
<code>double(A)</code>	Provádí konverzi na datový typ double
<code>exp(x)</code>	Vrací hodnotu exponenciální funkce e^x
<code>eye(n)</code>	Vytváří jednotkovou matici řádu n

<code>find(x)</code>	Vrací indexy nenulových prvků vektoru \boldsymbol{x}
<code>imread(filename)</code>	Načítá obraz ze souboru
<code>imshow(X)</code>	Zobrazuje obraz reprezentovaný maticí \mathbf{X}
<code>intmax('classname')</code>	Vrací nejvyšší použitelnou hodnotu v zadané celočíselné třídě
<code>inv(A)</code>	Počítá inverzní matici k regulární matici \mathbf{A}
<code>isfield(S, fieldname)</code>	Zjišťuje, zdali struktura \mathbf{S} obsahuje atribut <code>fieldname</code>
<code>isnan(x)</code>	Zjišťuje, zdali mají prvky vektoru \boldsymbol{x} hodnotu NaN
<code>length(x)</code>	Vrací délku vektoru \boldsymbol{x}
<code>mean(A), mean(x)</code>	Počítá průměr prvků v jednotlivých sloupcích matice \mathbf{A} , počítá průměr prvků vektoru \boldsymbol{x}
<code>ones(m, n)</code>	Vytváří matici o rozměrech $m \times n$, jejíž všechny prvky jsou 1
<code>plot(x, y)</code>	Vykresluje body se zadanými souřadnicemi do 2D grafu
<code>randn(m, n)</code>	Vytváří matici o rozměrech $m \times n$, jejíž prvky jsou náhodným výběrem z normovaného normálního rozdělení
<code>repmat(x, n, 1)</code>	Vytváří matici s n řádky, které jsou tvořeny opakováním vektoru \boldsymbol{x}
<code>sign(x)</code>	Vrací hodnotu funkce signum
<code>size(A)</code>	Vrací rozměry matice \mathbf{A}
<code>sqrt(x)</code>	Vrací druhou odmocninu prvků vektoru \boldsymbol{x}
<code>squeeze(A)</code>	Odstraňuje z vícerozměrného pole \mathbf{A} „prázdné“ dimenze
<code>struct(field, value)</code>	Vytváří strukturu obsahující atribut <code>field</code> s hodnotou <code>value</code>
<code>sum(A), sum(x)</code>	Počítá součet prvků v jednotlivých sloupcích matice \mathbf{A} , počítá součet prvků vektoru \boldsymbol{x}
<code>svd(A)</code>	Počítá SVD rozklad matice \mathbf{A}
<code>zeros(m, n)</code>	Vytváří nulovou matici o rozměrech $m \times n$

A.1 Kalmanova filtrace

A.1.1 Kalmanův filtr

```
function [est] = kf(sys, init, Z, U)
% Kalmanův filtr
%
% Předpokládá se následující model systému (podrobněji viz dále):
%   x = Phi * x + Gamma * u + G * w,
%   z = H * x + D * u + v,
% kde w ~ N(0, Q), v ~ N(0, R)
%
%
% [est] = kf(sys, init, Z, U)
%
% Vstupy:
% sys .... struktura reprezentující model systému
%   sys.Phi ..... matice dynamiky systému, n x n
%   sys.Gamma ... matice vlivu vstupu u na stav systému x,
%                   n x r,
%                   (nepovinná, výchozí hodnota eye(n, r))
%   sys.G ..... matice vlivu šumu w na stav systému x,
%                   n x m,
%                   (nepovinná, výchozí hodnota eye(n, m))
%   sys.H ..... matice popisující vztah mezi stavem systému
%                   x a měřením z, l x n
%   sys.D ..... matice vlivu vstupu u na měření z, l x r,
%                   (nepovinná, výchozí hodnota zeros(l, r))
%   sys.Q ..... kovarianční matice systémového šumu w,
%                   m x m
%   sys.R ..... kovarianční matice šumu měření v, l x l
%
% init ... struktura reprezentující počáteční podmínky
%   init.x ..... počáteční stav systému, n x 1
%   init.P ..... kovarianční matice počátečního stavu, n x n
%
% Z ..... matice, jejíž sloupce jsou vektory měření z pro
%   jednotlivé časové okamžiky 1 : f, l x f
%
% U ..... matice, jejíž sloupce jsou vektory vstupů u pro
%   jednotlivé časové okamžiky 1 : f, r x f,
%   (nepovinná, výchozí hodnota zeros(r, f))
%
% Výstup:
% est .... pole struktur obsahující vypočítané odhady pro jednotlivé
%   časové okamžiky 1 : f
%   est(k).x .... odhad stavu systému v čase k
%   est(k).P .... chybová kovarianční matice odhadu v čase k
%
%
% Datum poslední úpravy: 18. 11. 2017, KF
%
% INICIALIZAČNÍ ČÁST
```

```

[l, f] = size(Z);
n = size(init.x, 1);
m = size(sys.Q, 1);

% Nastavení výchozích hodnot nezadaných argumentů
if nargin == 3
    U = zeros(1, f);
end
r = size(U, 1);
if ~isfield(sys, 'Gamma')
    sys.Gamma = eye(n, r);
end
if ~isfield(sys, 'G')
    sys.G = eye(n, m);
end
if ~isfield(sys, 'D')
    sys.D = zeros(1, r);
end

x = init.x;
P = init.P;
est(1 : f) = struct('x', zeros(n, 1), 'P', zeros(n, n));

% VLASTNÍ ALGORITMUS KALMANOVA FILTRU

for k = 1 : f

    % Predikční krok (apriorní odhad)
    x = sys.Phi * x + sys.Gamma * U(:, k);
    P = sys.Phi * P * sys.Phi' + sys.G * sys.Q * sys.G';

    % Korekční krok (aposteriorní odhad)
    K = P * sys.H' * inv(sys.H * P * sys.H' + sys.R);
    x = x + K * (Z(:, k) - sys.D * U(:, k) - sys.H * x);
    P = P - K * sys.H * P;

    est(k).x = x;
    est(k).P = P;

end

end

```

A.1.2 Další algoritmy Kalmanovy filtrace

```

function [est] = ekf(sys, init, Z)
% Extended Kalman filter
%
% Předpokládá se následující model systému (podrobněji viz dále):
%     x = phi(x) + w,
%     z = h(x) + v,
% kde w ~ N(0, Q), v ~ N(0, R) a phi(x), h(x) jsou obecně nelineární
% funkce
%
%

```



```

% [est] = ekf(sys, init, Z)
%
% Vstupy:
% sys .... struktura reprezentující model systému
% sys.phi ..... funkce stavové rovnice modelu phi(x)
% sys.Phi ..... Jacobiho matice parciálních derivací funkce
% phi podle x
% sys.h ..... funkce rovnice měření modelu h(x)
% sys.H ..... Jacobiho matice parciálních derivací funkce
% h podle x
% sys.Q ..... kovarianční matice systémového šumu w,
% n x n
% sys.R ..... kovarianční matice šumu měření v, l x l
%
% init ... struktura reprezentující počáteční podmínky
% init.x ..... počáteční stav systému, n x 1
% init.P ..... kovarianční matice počátečního stavu, n x n
%
% Z ..... matice, jejíž sloupce jsou vektory měření z pro
% jednotlivé časové okamžiky 1 : f, l x f
%
% Výstup:
% est .... pole struktur obsahující vypočítané odhady pro jednotlivé
% časové okamžiky 1 : f
% est(k).x .... odhad stavu systému v čase k
% est(k).P .... chybová kovarianční matice odhadu v čase k
%
% Datum poslední úpravy: 14. 1. 2018, KF

% INICIALIZAČNÍ ČÁST

f = size(Z, 2);
n = size(init.x, 1);

x = init.x;
P = init.P;
est(1 : f) = struct('x', zeros(n, 1), 'P', zeros(n, n));

% VLASTNÍ ALGORITMUS EKF

for k = 1 : f

    % Predikční krok (apriorní odhad)
    Phi = sys.Phi(x);
    x = sys.phi(x);
    P = Phi * P * Phi' + sys.Q;

    % Korekční krok (aposteriorní odhad)
    H = sys.H(x);
    K = P * H' * inv(H * P * H' + sys.R);
    x = x + K * (Z(:, k) - sys.h(x));
    P = P - K * H * P;

    est(k).x = x;
    est(k).P = P;

```

end

end

```
function [est] = ukf(sys, init, Z, alpha, beta, kappa)
% Unscented Kalman filter
%
% Předpokládá se následující model systému (podrobněji viz dále):
%   x = phi(x) + w,
%   z = h(x) + v,
% kde w ~ N(0, Q), v ~ N(0, R) a phi(x), h(x) jsou obecně nelineární
% funkce
%
%
% [est] = ukf(sys, init, Z, alpha, beta, kappa)
%
% Vstupy:
% sys ..... struktura reprezentující model systému
%   sys.phi ..... funkce stavové rovnice modelu phi(x)
%   sys.h ..... funkce rovnice měření modelu h(x)
%   sys.Q ..... kovarianční matice systémového šumu w,
%               n x n
%   sys.R ..... kovarianční matice šumu měření v, l x l
%
% init .... struktura reprezentující počáteční podmínky
%   init.x ..... počáteční stav systému, n x 1
%   init.P ..... kovarianční matice počátečního stavu,
%               n x n
%
% Z ..... matice, jejíž sloupce jsou vektory měření z pro
%          jednotlivé časové okamžiky 1 : f, l x f
%
% alpha,
% beta,
% kappa ... parametry pro unscented transform (nepovinné, výchozí
%          hodnoty: alpha = 1e-3, beta = 2, kappa = 0)
%
% Výstup:
% est ..... pole struktur obsahující vypočítané odhady pro
%          jednotlivé časové okamžiky 1 : f
%   est(k).x .... odhad stavu systému v čase k
%   est(k).P .... chybová kovarianční matice odhadu v čase k
%
%
% Datum poslední úpravy: 3. 3. 2018, KF

% INICIALIZAČNÍ ČÁST

f = size(Z, 2);
n = size(init.x, 1);

% Nastavení výchozích hodnot nezadaných argumentů
if nargin < 6
    kappa = 0;
end
if nargin < 5
    beta = 2;
```

```

end
if nargin < 4
    alpha = 1e-3;
end

x = init.x;
P = init.P;
est(1 : f) = struct('x', zeros(n, 1), 'P', zeros(n, n));

% VLASTNÍ ALGORITMUS UKF

for k = 1 : f

    % Predikční krok (apriorní odhad)
    [x, Pxx] = ut(x, P, sys.phi, alpha, beta, kappa);
    P = Pxx + sys.Q;

    % Korekční krok (aposteriorní odhad)
    [z, Pzz, Pxz] = ut(x, P, sys.h, alpha, beta, kappa);
    K = Pxz * inv(Pzz + sys.R);
    x = x + K * (Z(:, k) - z);
    P = P - K * Pxz';

    est(k).x = x;
    est(k).P = P;

end

end



---


function [y, Pyy, Pxy] = ut(x, Pxx, g, alpha, beta, kappa)
% Scaled unscented transform
%
% Technika umožňující aproximovat střední hodnotu a kovarianční
% matici veličiny Y, která vznikne transformací veličiny X
% nelineární funkcí,  $Y = g(X)$ 
%
%
% [y, Pyy] = ut(x, Pxx, g, alpha, beta, kappa)
% [y, Pyy, Pxy] = ut(x, Pxx, g, alpha, beta, kappa)
%
% Vstupy:
% x ..... střední hodnota veličiny X
% Pxx ..... kovarianční matice veličiny X
% g ..... obecně nelineární transformační funkce
% alpha,
% beta,
% kappa ... parametry ovlivňující výběr sigma bodů a stanovení
%           jejich vah
%
% Výstupy:
% y ..... aproximace střední hodnoty veličiny Y
% Pyy ..... aproximace kovarianční matice veličiny Y
% Pxy ..... aproximace vzájemné kovarianční matice veličin X a Y
%
% Datum poslední úpravy: 3. 3. 2018, KF

```

```

% Inicializační část
n = length(x);
p = 2 * n + 1;
Chi = zeros(n, p);
x = x(:);

lambda = alpha^2 * (n + kappa) - n;

[U, S, ~] = svd(Pxx);
Cxx = U * sqrt(S) * U';

% Výpočet sigma bodů
Chi(:, 1) = x;
c = sqrt(n + lambda);
for i = 1 : n
    Chi(:, i + 1) = x + c * Cxx(:, i);
    Chi(:, n + i + 1) = x - c * Cxx(:, i);
end

% Stanovení příslušných vah bodů
wOM = lambda / (n + lambda);
wOC = wOM + 1 - alpha^2 + beta;
w = 1 / (2 * (n + lambda));

% Transformace bodů funkcí g
for i = 1 : p
    Xi(:, i) = g(Chi(:, i));
end

% Výpočet střední hodnoty a kovarianční matice transformované veličiny
y = wOM * Xi(:, 1) + w * sum(Xi(:, 2 : p), 2);
Pyy = wOC * (Xi(:, 1) - y) * (Xi(:, 1) - y)';
for i = 2 : p
    Pyy = Pyy + w * (Xi(:, i) - y) * (Xi(:, i) - y)';
end

% Výpočet vzájemné kovarianční matice veličin X a Y
if nargin == 3
    Pxy = wOC * (Chi(:, 1) - x) * (Xi(:, 1) - y)';
    for i = 2 : p
        Pxy = Pxy + w * (Chi(:, i) - x) * (Xi(:, i) - y)';
    end
end
end

```

A.2 Aplikace Kalmanových filtrů

A.2.1 Indoor lokalizace

V úlohách indoor lokalizace je pracováno se soustavou souřadnic vytvořenou dle plánu druhého patra budovy FIM UHK, viz obrázek 9. Souřadnice lokalizovaného objektu

reprezentují souřadnice příslušného pixelu tohoto obrázku, přičemž 50 pixelů odpovídá vzdálenosti 1 metr.

Při dynamické lokalizaci je předpokládáno, že se objekt pohybuje konstantní rychlostí 1 m/s.

Hodnoty kovariančních matic šumů jsou ve všech úlohách určovány experimentálně. Počáteční hodnoty stavových vektorů a odpovídající chybové kovarianční matice lze volit do určité míry libovolně resp. lze využít různé strategie (například za výchozí odhad neznámé polohy může být považována známá poloha beaconu, jehož signál je v daném místě dle měření nejsilnější apod.).

```
% Algoritmus 1
%
% Jedná se o úlohu statické lokalizace, přičemž jako měření jsou
% chápány vzdálenosti k referenčním bodům (beaconům), které byly
% získány přepočtem z naměřených sil signálu
%
% Předpokládají se datové soubory s umístěním 'data\exp1\exp1_X.csv',
% kde X = 1, 2, ..., 20 je číslo testovacího bodu,
% a soubor 'data\exp1\beacons.csv' obsahující souřadnice referenčních
% bodů

clear

filename = 'data\exp1\exp1_1.csv';

beaconsAll = csvread('data\exp1\beacons.csv');
positionTrue = csvread(filename, 0, 0, [0 0 0 1]);
measurementsAll = csvread(filename, 1, 0);

notNaNIndices = find(~isnan(measurementsAll(1, :)));
beacons = beaconsAll(notNaNIndices, :);
measurements = measurementsAll(:, notNaNIndices);

sys.phi = @(x) x;
sys.Phi = @(x) [1 0; 0 1];
sys.h = @(x) sqrt((x(1) - beacons(:, 1)).^2 ...
+ (x(2) - beacons(:, 2)).^2);
sys.H = @(x) [(x(1) - beacons(:, 1)) ./ sqrt((x(1) ...
- beacons(:, 1)).^2 + (x(2) - beacons(:, 2)).^2) ...
(x(2) - beacons(:, 2)) ./ sqrt((x(1) - beacons(:, 1)).^2 ...
+ (x(2) - beacons(:, 2)).^2)];
sys.Q = [10 0; 0 10];
sys.R = 500 * eye(size(beacons, 1));
init.x = [1100; 1500];
init.P = [500 0; 0 500];

[est] = ekf(sys, init, measurements');
% [est] = ukf(sys, init, measurements');

positionEst = [est.x]';

f = size(measurements, 1);
```

```

k = 1 : f;
positionTrueRep = repmat(positionTrue, f, 1);

e_xhat = sqrt(sum((positionTrueRep - positionEst).^2, 2));

figure
subplot(2, 2, 1)
plot(k, positionTrueRep(:, 1), 'g-o', k, positionEst(:, 1), 'b-*')
xlabel('Čas'), ylabel('Poloha (x)')
subplot(2, 2, 2)
plot(k, positionTrueRep(:, 2), 'g-o', k, positionEst(:, 2), 'b-*')
xlabel('Čas'), ylabel('Poloha (y)')
legend({'Skutečná hodnota', 'Hodnota odhadnutá pomocí KF'}, ...
       'Location', 'southeast')
subplot(2, 2, [3, 4])
plot(k, e_xhat, 'b-*', k, 0, 'k')
xlabel('Čas'), ylabel('Chyba')
legend({'Chyba odhadu'}, 'Location', 'northeast')

```

```

% Algoritmus 2
%
% Jedná se o úlohu statické lokalizace, přičemž jako měření jsou
% chápány odhady polohy získané jinou lokalizační technikou
%
% Předpokládají se datové soubory s umístěním 'data\exp2\exp2_X.csv',
% kde X = 1, 2, ..., 20 je číslo testovacího bodu

clear

filename = 'data\exp2\exp2_1.csv';

positionTrue = csvread(filename, 0, 0, [0 0 0 1]);
measurements = csvread(filename, 1, 0);

sys.Phi = [1 0; 0 1];
sys.H = [1 0; 0 1];
sys.Q = [10 0; 0 10];
sys.R = [16000 0; 0 16000];
init.x = [measurements(1, 1); measurements(1, 2)];
init.P = [100 0; 0 100];

[est] = kf(sys, init, measurements(2 : end, :));

positionEst = [est.x]';
positionEst = [measurements(1, :); positionEst];

f = size(measurements, 1);
k = 1 : f;
positionTrueRep = repmat(positionTrue, f, 1);

e_z = sqrt(sum((positionTrueRep - measurements).^2, 2));
e_xhat = sqrt(sum((positionTrueRep - positionEst).^2, 2));

figure
subplot(2, 2, 1)
plot(k, positionTrueRep(:, 1), 'g-o', ...
     k, measurements(:, 1), 'r+', k, positionEst(:, 1), 'b-*')
xlabel('Čas'), ylabel('Poloha (x)')

```

```

subplot(2, 2, 2)
plot(k, positionTrueRep(:, 2), 'g-o', ...
     k, measurements(:, 2), 'r-+', k, positionEst(:, 2), 'b-*')
xlabel('Čas'), ylabel('Poloha (y)')
legend({'Skutečná hodnota', 'Hodnota měření', ...
       'Hodnota odhadnutá pomocí KF'}, 'Location', 'southeast')
subplot(2, 2, [3, 4])
plot(k, e_z, 'r-+', k, e_xhat, 'b-*', k, 0, 'k')
xlabel('Čas'), ylabel('Chyba')
legend({'Chyba měření', 'Chyba odhadu'}, 'Location', 'northeast')

```

```

% Algoritmus 3
%
% Jedná se o úlohu statické lokalizace, přičemž jako měření jsou
% chápány naměřené síly signálů z jednotlivých vysílacích zařízení
% (beaconů)
%
% Předpokládají se datové soubory s umístěním 'data\exp3\exp3_X.csv',
% kde X = 1, 2, ..., 20 je číslo testovacího bodu,
% a soubor 'data\exp3\calibrated.csv' reprezentující referenční mapu
% fingerprintů

clear

filename = 'data\exp3\exp3_1.csv';

calibratedRead = csvread('data\exp3\calibrated.csv');
positionTrue = csvread(filename, 0, 0, [0 0 0 1]);
measurements = csvread(filename, 1, 0);

n = size(calibratedRead, 1) / 3;
m = size(calibratedRead, 2);
calibrated(1 : n) = struct('position', zeros(1, 2), ...
                          'means', zeros(1, m), 'variances', zeros(1, m));
for i = 1 : n
    calibrated(i).position = calibratedRead(3 * (i - 1) + 1, 1 : 2)';
    calibrated(i).means = calibratedRead(3 * (i - 1) + 2, :)';
    calibrated(i).variances = calibratedRead(3 * (i - 1) + 3, :)';
end

Phi = [1 0; 0 1];
Q = [100 0; 0 100];

diffMin = Inf;
for i = 1 : n
    diff = sum(abs(measurements(1, :) - calibrated(i).means));
    if diff < diffMin
        diffMin = diff;
        x0 = calibrated(i).position;
    end
end

x = x0;
P = [10000 0; 0 10000];

Z = measurements';

f = size(measurements, 1);

```

```

est(1 : f) = struct('x', zeros(2, 1), 'P', zeros(2, 2));

% Fingerprint Kalman filter
for k = 1 : f

    x = Phi * x;
    P = Phi * P * Phi' + Q;

    beta = zeros(n, 1);
    pEst = zeros(2, 1);
    zEst = zeros(m, 1);
    Pxx = zeros(2, 2);
    Pxz = zeros(2, m);
    Pzz = zeros(m, m);
    Pp = diag([(50 ^ 2) / 12 (50 ^ 2) / 12]);

    sumPdf = 0;
    for i = 1 : n
        sumPdf = sumPdf + pdfnorm(calibrated(i).position, x, P);
    end

    for i = 1 : n
        beta(i) = pdfnorm(calibrated(i).position, x, P) / sumPdf;
        pEst = pEst + beta(i) * calibrated(i).position;
        zEst = zEst + beta(i) * calibrated(i).means;
    end

    for i = 1 : n
        Pxx = Pxx + beta(i) * (Pp + ...
            (calibrated(i).position - pEst) ...
            * (calibrated(i).position - pEst)');
        Pxz = Pxz + beta(i) * (calibrated(i).position - pEst) ...
            * (calibrated(i).means - zEst)';
        Pzz = Pzz + beta(i) * (diag(calibrated(i).variances) ...
            + (calibrated(i).means - zEst) ...
            * (calibrated(i).means - zEst)');
    end

    x = pEst + Pxz * inv(Pzz) * (Z(:, k) - zEst);
    P = Pxx - Pxz * inv(Pzz) * Pxz';

    est(k).x = x;
    est(k).P = P;
end

positionEst = [est.x]';

k = 1 : f;
positionTrueRep = repmat(positionTrue, f, 1);

e_xhat = sqrt(sum((positionTrueRep - positionEst).^2, 2));

figure
subplot(2, 2, 1)
plot(k, positionTrueRep(:, 1), 'g-o', k, positionEst(:, 1), 'b-*')
xlabel('Čas'), ylabel('Poloha (x)')
subplot(2, 2, 2)
plot(k, positionTrueRep(:, 2), 'g-o', k, positionEst(:, 2), 'b-*')

```



```

xlabel('Čas'), ylabel('Poloha (y)')
legend({'Skutečná hodnota', 'Hodnota odhadnutá pomocí KF'}, ...
       'Location', 'southeast')
subplot(2, 2, [3, 4])
plot(k, e_xhat, 'b-*', k, 0, 'k')
xlabel('Čas'), ylabel('Chyba')
legend({'Chyba odhadu'}, 'Location', 'northeast')

```

```

function [y] = pdfnorm(x, mu, Sigma)
% Funkce hustoty pravděpodobnosti n-rozměrného normálního rozdělení
%
% [y] = pdfnorm(x, mu, Sigma)
%
% Vstupy:
% x ..... vektor hodnot náhodné veličiny X, n x 1
% mu ..... vektor středních hodnot, n x 1
% Sigma ... kovarianční matice, n x n
%
% Výstup:
% y ..... hodnota funkce hustoty pravděpodobnosti v bodě x
%
% Datum poslední úpravy: 28. 1. 2018, KF

y = exp(-0.5 * (x - mu)' * inv(Sigma) * (x - mu)) ...
    / sqrt(det(2 * pi * Sigma));

end

```

```

% Algoritmus 4
%
% Jedná se o úlohu dynamické lokalizace, přičemž jako měření jsou
% chápány odhady poloh v jednotlivých časových okamžicích získané
% jinou lokalizační technikou
%
% Předpokládá se datový soubor s umístěním 'data\exp4\exp4.csv',
% obsahující testovací data procházkové trasy 1

clear

filename = 'data\exp4\exp4.csv';

measurements = csvread(filename, 0, 2);
f = size(measurements, 1);
positionsTrue = csvread(filename, 0, 0, [0 0 f-1 1]);

sys.Phi = [1 0 1 0; 0 1 0 1; 0 0 1 0; 0 0 0 1];
sys.H = [1 0 0 0; 0 1 0 0];
sys.Q = [50 0 0 0; 0 50 0 0; 0 0 50 0; 0 0 0 50];
sys.R = [16000 0; 0 16000];
init.x = [measurements(1, 1);
          measurements(1, 2);
          sign(measurements(1, 1) - measurements(2, 1)) * -50;
          sign(measurements(1, 2) - measurements(2, 2)) * -50];
init.P = [200 0 0 0; 0 200 0 0; 0 0 10 0; 0 0 0 10];

```

```

% [est] = kf(sys, init, measurements(2 : end, :));
[est] = kf_outliers(sys, init, measurements(2 : end, :)', 500);

stateEst = [est.x];
positionsEst = stateEst(1 : 2, :);
positionsEst = [measurements(1, :); positionsEst];

e_z = sqrt(sum((positionsTrue - measurements).^2, 2));
mse_z = mean(e_z.^2)
me_z = mean(e_z)
e_xhat = sqrt(sum((positionsTrue - positionsEst).^2, 2));
mse_xhat = mean(e_xhat.^2)
me_xhat = mean(e_xhat)

k = 1 : f;

figure
subplot(2, 2, 1)
plot(k, positionsTrue(:, 1), 'g-o', k, measurements(:, 1), 'r--', ...
     k, positionsEst(:, 1), 'b-*')
xlabel('Čas'), ylabel('Poloha (x)')
subplot(2, 2, 2)
plot(k, positionsTrue(:, 2), 'g-o', k, measurements(:, 2), 'r--', ...
     k, positionsEst(:, 2), 'b-*')
xlabel('Čas'), ylabel('Poloha (y)')
legend({'Skutečná hodnota', 'Hodnota měření', ...
       'Hodnota odhadnutá pomocí KF'}, 'Location', 'southeast')
subplot(2, 2, [3, 4])
plot(k, e_z, 'r--', k, e_xhat, 'b-*', k, 0, 'k')
xlabel('Čas'), ylabel('Chyba')
legend({'Chyba měření', 'Chyba odhadu'}, 'Location', 'northeast')

```

```

function [est] = kf_outliers(sys, init, Z, tol, U)
% Kalmanův filtr s detekcí odlehlých hodnot měření
%
% Předpokládá se následující model systému (podrobněji viz dále):
%   x = Phi * x + Gamma * u + G * w,
%   z = H * x + D * u + v,
% kde w ~ N(0, Q), v ~ N(0, R)
%
%
% [est] = kf_outliers(sys, init, Z, tol, U)
%
% Vstupy:
% sys .... struktura reprezentující model systému
%         sys.Phi ..... matice dynamiky systému, n x n
%         sys.Gamma ... matice vlivu vstupu u na stav systému x,
%                       n x r,
%                       (nepovinná, výchozí hodnota eye(n, r))
%         sys.G ..... matice vlivu šumu w na stav systému x,
%                       n x m,
%                       (nepovinná, výchozí hodnota eye(n, m))
%         sys.H ..... matice popisující vztah mezi stavem systému
%                       x a měřením z, l x n
%         sys.D ..... matice vlivu vstupu u na měření z, l x r,
%                       (nepovinná, výchozí hodnota zeros(l, r))
%         sys.Q ..... kovarianční matice systémového šumu w,
%                       m x m

```

```

%          sys.R ..... kovarianční matice šumu měření v, l x l
%
%  init ... struktura reprezentující počáteční podmínky
%          init.x ..... počáteční stav systému, n x 1
%          init.P ..... kovarianční matice počátečního stavu, n x n
%
%  Z ..... matice, jejíž sloupce jsou vektory měření z pro
%          jednotlivé časové okamžiky 1 : f, l x f
%
%  tol .... tolerance odlehlých měření, je-li rozdíl mezi odhadem
%          měření a skutečnou naměřenou hodnotou větší než tato
%          tolerance, nebude dané měření zahrnuto do výpočtu
%
%  U ..... matice, jejíž sloupce jsou vektory vstupů u pro
%          jednotlivé časové okamžiky 1 : f, r x f,
%          (nepovinná, výchozí hodnota zeros(r, f))
%
%  Výstup:
%  est .... pole struktur obsahující vypočítané odhady pro jednotlivé
%          časové okamžiky 1 : f
%          est(k).x .... odhad stavu systému v čase k
%          est(k).P .... chybová kovarianční matice odhadu v čase k
%
%  Datum poslední úpravy: 28. 1. 2018, KF

```

```

% INICIALIZAČNÍ ČÁST

```

```

[l, f] = size(Z);
n = size(init.x, 1);
m = size(sys.Q, 1);

% Nastavení výchozích hodnot nezadaných argumentů
if nargin == 4
    U = zeros(1, f);
end
r = size(U, 1);
if ~isfield(sys, 'Gamma')
    sys.Gamma = eye(n, r);
end
if ~isfield(sys, 'G')
    sys.G = eye(n, m);
end
if ~isfield(sys, 'D')
    sys.D = zeros(l, r);
end

x = init.x;
P = init.P;
est(1 : f) = struct('x', zeros(n, 1), 'P', zeros(n, n));

```

```

% VLASTNÍ ALGORITMUS KALMANOVA FILTRU

```

```

for k = 1 : f

    % Predikční krok (apriorní odhad)
    x = sys.Phi * x + sys.Gamma * U(:, k);

```

```

P = sys.Phi * P * sys.Phi' + sys.G * sys.Q * sys.G';

% Korekční krok (aposteriorní odhad)
innovation = Z(:, k) - sys.D * U(:, k) - sys.H * x;
if all(abs(innovation) < tol)
    K = P * sys.H' * inv(sys.H * P * sys.H' + sys.R);
    x = x + K * innovation;
    P = P - K * sys.H * P;
end

est(k).x = x;
est(k).P = P;

end

end



---


% Algoritmus 4
%
% Jedná se o úlohu dynamické lokalizace, přičemž jako měření jsou
% chápány odhady poloh v jednotlivých časových okamžicích získané
% jinou lokalizační technikou
%
% Předpokládá se datový soubor s umístěním 'data\exp4\exp4_2.csv',
% obsahující testovací data procházkové trasy 2

clear

filename = 'data\exp4\exp4_2.csv';

measurements = csvread(filename, 0, 2);
f = size(measurements, 1);
positionsTrue = csvread(filename, 0, 0, [0 0 f-1 1]);

sys.Phi = [1 0 1 0; 0 1 0 1; 0 0 1 0; 0 0 0 1];
sys.H = [1 0 0 0; 0 1 0 0];
sys.Q = [10 0 0 0; 0 10 0 0; 0 0 10 0; 0 0 0 10];
sys.R = [16000 0; 0 16000];
init.x = [measurements(1, 1);
          measurements(1, 2);
          sign(measurements(1, 1) - measurements(2, 1)) * -50
          sign(measurements(1, 2) - measurements(2, 2)) * -50];
init.P = [200 0 0 0; 0 200 0 0; 0 0 10 0; 0 0 0 10];

% [est] = kf(sys, init, measurements(2 : end, :));
[est] = kf_outliers(sys, init, measurements(2 : end, :)', 200);

stateEst = [est.x];
positionsEst = stateEst(1 : 2, :);
positionsEst = [measurements(1, :); positionsEst];

e_z = sqrt(sum((positionsTrue - measurements).^2, 2));
mse_z = mean(e_z.^2)
me_z = mean(e_z)
e_xhat = sqrt(sum((positionsTrue - positionsEst).^2, 2));
mse_xhat = mean(e_xhat.^2)
me_xhat = mean(e_xhat)

```

```

k = 1 : f;

figure
subplot(2, 2, 1)
plot(k, positionsTrue(:, 1), 'g-o', k, measurements(:, 1), 'r-+', ...
     k, positionsEst(:, 1), 'b-*')
xlabel('Čas'), ylabel('Poloha (x)')
subplot(2, 2, 2)
plot(k, positionsTrue(:, 2), 'g-o', k, measurements(:, 2), 'r-+', ...
     k, positionsEst(:, 2), 'b-*')
xlabel('Čas'), ylabel('Poloha (y)')
legend({'Skutečná hodnota', 'Hodnota měření', ...
       'Hodnota odhadnutá pomocí KF'}, 'Location', 'northeast')
subplot(2, 2, [3, 4])
plot(k, e_z, 'r-+', k, e_xhat, 'b-*', k, 0, 'k')
xlabel('Čas'), ylabel('Chyba')
legend({'Chyba měření', 'Chyba odhadu'}, 'Location', 'northeast')

```

A.2.2 Analýza časových řad

```

% Ukázka použití Kalmanova filtru pro odhadování parametrů
% AR(2) modelu

```

```

clear

y = csvread('data_AR(2).csv');
n = length(y);
sigma2_e = 10;

figure
plot(1 : n, y, 'k')
xlabel('t')
ylabel('y_t')

sys.Phi = eye(2);
sys.Q = zeros(2);
sys.R = sigma2_e;
est.x = [1; 1];
est.P = [1 0; 0 1];

for t = 3 : n
    sys.H = [y(t-1) y(t-2)];
    est = kf(sys, est, y(t));
end

phi_1 = est.x(1)
phi_2 = est.x(2)

```

```

% Ukázka použití Kalmanova filtru pro predikování, filtrování
% a vyrovnávání v časové řadě reprezentované ARMA(2,1) modelem

```

```

clear

y = csvread('data_ARMA(2,1).csv');
n = length(y);

```

```

phi_1 = 0.6;
phi_2 = 0.2;
theta_1 = 0.1;
sigma2_e = 10;

sys.Phi = [phi_1 1; phi_2 0];
sys.H = [1 0];
sys.G = [1; theta_1];
sys.Q = sigma2_e;
sys.R = sigma2_e;

% Predikování
y_predicted = zeros(1, n+5);
y_predicted(1) = y(1);
y_predicted(2) = y(2);
e = zeros(1, n);
for t = 3 : n % jednokrokové predikce ve sledovaném období
    x = sys.Phi * [y(t-1); phi_2 * y(t-2) + theta_1 * e(t-1)];
    y_predicted(t) = x(1);
    e(t) = y(t) - y_predicted(t);
end
x = [y(n); phi_2 * y(n-1) + theta_1 * e(n-1)];
for t = n+1 : n+5 % predikce pro 5 budoucích časových okamžiků
    x = sys.Phi * x;
    y_predicted(t) = x(1);
end

% Filtrování
% (ekvivalentně lze použít funkci kf(sys, init, y))
est_f(1 : n) = struct('x_', zeros(2, 1), 'P_', zeros(2, 2), ...
    'x', zeros(2, 1), 'P', zeros(2, 2));
est_f(1).x = [y(1); 1];
est_f(1).P = [10 0; 0 10];
for t = 2 : n
    est_f(t).x_ = sys.Phi * est_f(t-1).x;
    est_f(t).P_ = sys.Phi * est_f(t-1).P * sys.Phi' ...
        + sys.G * sys.Q * sys.G';
    K = est_f(t).P_ * sys.H' * inv(sys.H * est_f(t).P_ * sys.H' ...
        + sys.R);
    est_f(t).x = est_f(t).x_ + K * (y(t) - sys.H * est_f(t).x_);
    est_f(t).P = est_f(t).P_ - K * sys.H * est_f(t).P_;
end
estx = [est_f.x];
y_filtered = estx(1, :);

% Vyrovnávání (Rauch-Tung-Striebel smoother)
est_s(1 : n) = struct('x', zeros(2, 1), 'P', zeros(2, 2));
est_s(n).x = est_f(n).x;
est_s(n).P = est_f(n).P;
for t = n-1 : -1 : 2
    A = est_f(t).P * sys.Phi' * inv(est_f(t+1).P_);
    est_s(t).x = est_f(t).x + A * (est_s(t+1).x - est_f(t+1).x_);
    est_s(t).P = est_f(t).P + A * (est_s(t+1).P - est_f(t+1).P_) * A';
end
estx = [est_s.x];
y_smoothed = estx(1, :);
y_smoothed(1) = y(1);

```

```

t = 1 : n;
figure
subplot(3, 1, 1)
p = plot(t, y, 'k', 1 : n+5, y_predicted, 'b');
p(2).LineWidth = 1.5;
xlim([0 105])
title('Predikování')
subplot(3, 1, 2)
p = plot(t, y, 'k', t, y_filtered, 'b');
p(2).LineWidth = 1.5;
xlim([0 105])
ylabel('y_t')
title('Filtrování')
subplot(3, 1, 3)
p = plot(t, y, 'k', t, y_smoothed, 'b');
p(2).LineWidth = 1.5;
xlim([0 105])
xlabel('t')
title('Vyrovňávání')

```

A.2.3 Zpracování obrazu

V úlohách se předpokládá znalost modelu obrazu, modelu rozmazání a rozptylu šumu.

```

% Filtrace šumu v obraze pomocí algoritmu reduced update Kalman filter
% (steady-state approximate RUKF)
%
% dle Woods, Radewan: Kalman filtering in two dimensions

clear

% Načtení původního ideálního obrazu
X = imread('HK.jpg');
X = 0.2989 * X(:, :, 1) + 0.5870 * X(:, :, 2) + 0.1140 * X(:, :, 3);
X = double(X) ./ double(intmax(class(X)));
[s1, s2] = size(X);

figure
imshow(X, [0 1])

% Parametry modelu obrazu
n = 3;
c = [0.020380445 -0.002564352 -0.000146032
      0.009573138  0.048825988  0.031307254
      0.148352208  0.738287199  0];

% Přidání gaussovského bílého šumu do obrazu
Z = X + randn(s1, s2) ./ 10;

figure
imshow(Z, [0 1])

% Centrování obrazu
imgmean = mean(Z(:));

```

```

Z = Z - imgmean;

% Rozptyl systémového šumu a šumu měření
q = 1e-4;
r = 1e-2;

% Velikosti oblastí (okolí zpracovávaného pixelu), kde je algoritmus
% aplikován
% (hodnoty lze volit experimentálně)
u1 = n;
u2 = n;
u3 = n;
t = u3 + n;
ZZ = extendimage(Z, t);

Xest = ZZ;
P = 0.1 * ones(s1+2*t, s2+2*t, u1, u3);

% Zjištění ustálené hodnoty Kalmanova zisku (steady-state Kalman gain)
% (zde pro 100 iterací algoritmu)
K = zeros(u1, u3);
for i = 1 + t : 1 + t
    for j = 1 + t : 100 + t

        % Apriorní chybová kovarianční "matice"
        Ptmp = P;
        for k = 1 : u1
            for l = 1 : u3
                Ptmp(i, j, k, l) = computesum(...
                    P(i-n+1 : i, j-n+1 : j, k, l), c);
            end
        end
        Ptmp(i, j, end, end) = computesum(squeeze(...
            P(i, j, end-n+1 : end, end-n+1 : end)), c) + q;
        P = Ptmp;

        % Kalmanův zisk
        K = squeeze(P(i, j, :, :)) ./ (P(i, j, end, end) + r);

        % Aposteriorní chybová kovarianční "matice"
        Ptmp = P;
        for k = 1 : u1
            for l = 1 : u3
                ii = i-u1+k;
                jj = j-u3+l;
                for o = 1 : u1
                    for p = 1 : u3
                        Ptmp(ii, jj, o, p) = P(ii, jj, o, p) ...
                            - K(k, l) * P(i, j, o, p);
                    end
                end
            end
        end
        P = Ptmp;

    end
end
K(end, end) = 0;

```



```

% Vlastní aplikování RUKF
for i = 1 + t : s1 + t
    for j = 1 + t : s2 + t

        % Apriorní odhad (predikce)
        Xest(i, j) = computesum(Xest(i-n+1 : i, j-n+1 : j), c);

        % Aposteriorní odhad (korekce)
        Xtmp = Xest;
        for k = 1 : u1
            for l = 1 : u2
                ii = i-u1+k;
                jj = j-u2+1;
                Xtmp(ii, jj) = Xest(ii, jj) ...
                    + K(k, l+u3-u2) * (ZZ(i, j) - Xest(i, j));
            end
        end
        Xest = Xtmp;

    end
end

% Výsledný filtrovaný obraz
Xestimated = Xest(1+t : s1+t, 1+t : s2+t) + imgmean;

figure
imshow(Xestimated, [0 1])

function [S] = computesum(A, B)
    AB = A .* B;
    S = sum(AB(:));
end



---


function [Y] = extendimage(X, n)
% Rozšíření obrazu na specifikovanou velikost zrcadlením okrajů
%
% [Y] = extendimage(X, n)
%
% Vstupy:
% X ... vstupní obraz
% n ... počet pixelů, o které má být obraz z každé strany rozšířen
%
% Výstup:
% Y ... rozšířený obraz
%
% Datum poslední úpravy: 17. 3. 2018, KF

Y = zeros(size(X) + 2 * n);
Y(n+1 : end-n, n+1 : end-n) = X;
Y(:, 1 : n) = fliplr(Y(:, n+1 : 2*n));
Y(:, end-n+1 : end) = fliplr(Y(:, end-2*n+1 : end-n));
Y(1 : n, :) = flipud(Y(n+1 : 2*n, :));
Y(end-n+1 : end, :) = flipud(Y(end-2*n+1 : end-n, :));

```

```
end
```

```
% Rekonstrukce rozmazaného obrazu pomocí algoritmu reduced update
% Kalman filter (steady-state approximate RUKF)
%
% dle Woods, Ingle: Kalman filtering in two dimensions: further
  results

clear

% Načtení původního ideálního obrazu
X = imread('HK.jpg');
X = 0.2989 * X(:, :, 1) + 0.5870 * X(:, :, 2) + 0.1140 * X(:, :, 3);
X = double(X) ./ double(intmax(class(X)));
[s1, s2] = size(X);

figure
imshow(X, [0 1])

% Parametry modelu obrazu
n = 3;
c = [0.020380445 -0.002564352 -0.000146032
      0.009573138  0.048825988  0.031307254
      0.148352208  0.738287199  0];

% Rozmazání obrazu
% (simulace rozmazání vzniklého pohybem v horizontálním směru)
m = 7;
h = 1/m * ones(1, m);
XX = extendimage(X, m);
Z = zeros(s1, s2);
for i = 1 : s1
    for j = 1 : s2
        Z(i, j) = computesum(XX(i+m, j+1 : j+m), h);
    end
end

% Přidání gaussovského bílého šumu do rozmazaného obrazu
Z = Z + randn(s1, s2) ./ 1000;

figure
imshow(Z, [0 1])

% Centrování obrazu
imgmean = mean(Z(:));
Z = Z - imgmean;

% Rozptyl systémového šumu a šumu měření
q = 1e-4;
r = 1e-6;

% Velikosti oblastí (okolí zpracovávaného pixelu), kde je algoritmus
% aplikován
% (hodnoty lze volit experimentálně)
u1 = m;
u2 = m;
u3 = m;
t = u3 + m;
```

```

ZZ = extendimage(Z, t);

Xest = ZZ;
P = 0.1 * ones(s1+2*t, s2+2*t, u1, u3);

% Zjištění ustálené hodnoty Kalmanova zisku (steady-state Kalman gain)
% (zde pro 100 iterací algoritmu)
K = zeros(u1, u3);
for i = 1 + t : 1 + t
    for j = 1 + t : 100 + t

        % Apriorní chybová kovarianční "matice"
        Ptmp = P;
        for k = 1 : u1
            for l = 1 : u3
                Ptmp(i, j, k, l) = computesum(...
                    P(i-n+1 : i, j-n+1 : j, k, l), c);
            end
        end
        Ptmp(i, j, end, end) = computesum(squeeze(...
            P(i, j, end-n+1 : end, end-n+1 : end)), c) + q;
        P = Ptmp;

        % Kalmanův zisk
        K = zeros(u1, u3);
        k2 = 0;
        for l = 1 : m
            k2 = k2 + h(l) * computesum(squeeze(...
                P(i, j-m+1, end, end-m+1 : end)), h);
        end
        k2 = k2 + r;
        for k = 1 : u1
            for l = 1 : u3
                k1 = computesum(P(i, j-m+1 : j, k, l), h);
                K(k, l) = k1 / k2;
            end
        end

        % Aposteriorní chybová kovarianční "matice"
        Ptmp = P;
        for k = 1 : u1
            for l = 1 : u3
                ii = i-u1+k;
                jj = j-u3+l;
                for o = 1 : u1
                    for p = 1 : u3
                        Ptmp(ii, jj, o, p) = P(ii, jj, o, p) ...
                            - K(k, l) ...
                            * computesum(P(ii, jj-m+1 : jj, o, p), h);
                    end
                end
            end
        end
        P = Ptmp;

    end
end
end

```

```

% Vlastní aplikování RUKF
for i = 1 + t : s1 + t
    for j = 1 + t : s2 + t

        % Apriorní odhad (predikce)
        Xest(i, j) = computesum(Xest(i-n+1 : i, j-n+1 : j), c);

        % Aposteriorní odhad (korekce)
        Xtmp = Xest;
        for k = 1 : u1
            for l = 1 : u2
                ii = i-u1+k;
                jj = j-u2+1;
                Xtmp(ii, jj) = Xest(ii, jj) ...
                    + K(k, l+u3-u2) * (ZZ(ii, jj) ...
                        - computesum(Xest(ii, jj-m+1 : jj), h));
            end
        end
        Xest = Xtmp;

    end
end

% Výsledný rekonstruovaný obraz
Xestimated = Xest(1+t : s1+t, 1+t : s2+t) + imgmean;

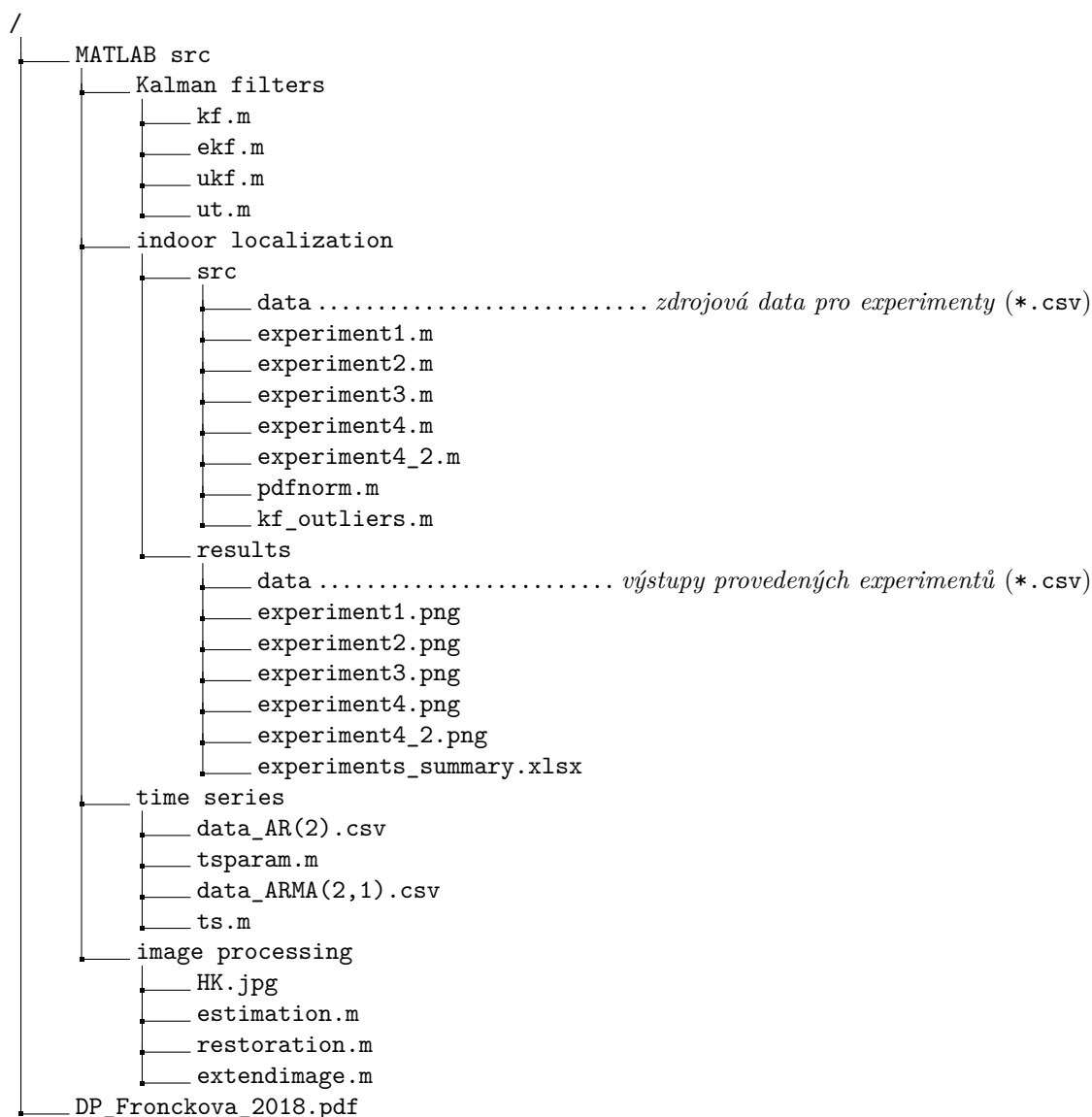
figure
imshow(Xestimated, [0 1])

function [S] = computesum(A, B)
    AB = A .* B;
    S = sum(AB(:));
end

```

B Obsah přiloženého CD

Přiložené CD obsahuje zdrojové kódy z přílohy A v podobě souborů prostředí MATLAB (*.m) včetně použitých datových souborů, dále výsledky testování algoritmů indoor lokalizace a elektronickou verzi této práce.



Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Bc. Frončková Kateřina	Edvarda Beneše 1433/1A, Hradec Králové - Nový Hradec Králové	I1600283

TÉMA ČESKY:

Kalmanovy filtry

TÉMA ANGLICKY:

Kalman Filters

VEDOUCÍ PRÁCE:

doc. RNDr. Pavel Pražák, Ph.D. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Cíl práce:

Teoretický popis Kalmanova filtru a dalších odvozených filtrů, ukázka použití, implementace v prostředí MATLAB

Osnova práce:

Základní pojmy a teoretická východiska

Kalmanova filtrace

Kalmanův filtr

Další algoritmy Kalmanovy filtrace

Použití a implementace v prostředí MATLAB

SEZNAM DOPORUČENÉ LITERATURY:

CHUI, Charles K., CHEN, Guanrong. Kalman filtering with real-time applications. 4th ed. Berlin: Springer, 2009. ISBN 978-3-540-87848-3.

GREWAL, Mohinder S., ANDREWS Angus P. Kalman filtering: theory and practice using MATLAB. 3rd ed. Hoboken, N.J.: Wiley, 2008. ISBN 978-0-470-17366-4.

MAYBECK, Peter S. Stochastic models, estimation and control, Volume I. 1st ed. New York: Academic Press, 1979. ISBN 0-12-480701-1.

SIMON, Dan. Optimal state estimation: Kalman, H infinity and nonlinear approaches. 1st ed. Hoboken, N.J.: Wiley, 2006. ISBN 978-0-471-70858-2.

Podpis studenta:

Frončková

Datum:

9.10.2017

Podpis vedoucího práce:

P. Pražák

Datum:

9.10.17