BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF MATHEMATICS

ÚSTAV MATEMATIKY

ESTIMATING OF MOTION MODELS AND ITS PARAMETERS TO IDENTIFY TARGET TRAJECTORY

HLEDANÍ MODELŮ POHYBU A JEJICH PARAMETRŮ PRO IDENTIFIKACI TRAJEKTORIE CÍLŮ

MASTER'S THESIS

DIPLOMOVÁ PRÁCE

AUTHOR — **Bc. MATEJ BENKO**

AUTOR PRÁCE

SUPERVISOR — **doc. RNDr. LIBOR ŽÁK, Ph.D.**

VEDOUCÍ PRÁCE

BRNO 2021

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

Assignment Master's Thesis

Institut:	Institute of Mathematics
Student:	**Bc. Matej Benko**
Degree programm:	Applied Sciences in Engineering
Branch:	Mathematical Engineering
Supervisor:	**doc. RNDr. Libor Žák, Ph.D.**
Academic year:	2020/21

As provided for by the Act No. 111/98 Coll. on higher education institutions and the BUT Study and Examination Regulations, the director of the Institute hereby assigns the following topic of Master's Thesis:

Estimating of motion models and its parameters to identify target trajectory

Recommended bibliography:

ANDĚL, Jiří. Základy matematické statistiky. 3. opr. vyd. Praha: Matfyzpress, 2011. ISBN 978-8--7378-001-2

CONGDON, Peter. Applied Bayesian modelling. 2nd ed. Wiley, 2014. ISBN 978-1119951513

Deadline for submission Master's Thesis is given by the Schedule of the Academic year 2020/21

In Brno,

L. S.

prof. RNDr. Josef Šlapal, CSc.
Director of the Institute

doc. Ing. Jaroslav Katolický, Ph.D.
FME dean

Summary

This text deals with removing noise from inaccurate multilateration measurements. It is used Bayesian estimation theory to find the posterior density of the real position (or, moreover, velocity) of an airplane. Together with true position, we estimate on Bayesian principle the geometry of a maneuver that an airplane obeys and so-called process noise, which describes how much an airplane's trajectory differs from the geometry. The estimation of the process noise is the essential part of the work. It is derived Bayesian approach together with the maximum likelihood approach. Then, improvements to these algorithms are introduced. They provide better results in particular cases, such as a maneuver change of the target or initial uncertainty of the maximum likelihood estimation. At the end of the text, the possibility of a combination of geometry and process noise estimation is described.

Abstrakt

Táto práca sa zaoberá odstraňovaním šumu, ktorý vzniká z tzv. multilateračných meraní leteckých cieľov. Na tento účel bude využitá najmä teória Bayesovských odhadov. Odvodí sa aposteriórna hustota skutočnej (presnej) polohy lietadla. Spolu s polohou (alebo aj rýchlosťou) lietadla bude odhadovaná tiež geometria trajektórie lietadla, ktorú lietadlo v aktuálnom čase sleduje a tzv. procesný šum, ktorý charakterizuje ako moc sa skutočná trajektória môže od tejto líšiť. Odhad spomínaného procesného šumu je najdôležitejšou časťou tejto práce. Je odvodený prístup maximálnej vierohodnosti a Bayesovský prístup a ďalšie rôzne vylepšenia a úpravy týchto prístupov. Tie zlepšujú odhad pri napr. zmene manévru cieľa alebo riešia problém počiatočnej nepresnosti odhadu maximálnej vierohodnosti. Na záver je ukázaná možnosť kombinácie prístupov, t.j. odhad spolu aj geometrie aj procesného šumu.

Keywords

target tracking, Bayesian estimate, maximum likelihood estimate, Kalman Filter, Interactive Multiple Model algorithm, estimation of the process noise, adaptive filtering

Klíčová slova

sledovanie leteckých cieľov, Bayesovské odhady, maximálne vierohodný odhad, Kálmánov filter, Interactive Multiple Model algoritmus, odhad procesného šumu, adaptívna filtrácia

BENKO, M. *Estimating of Motion Models and its Parameters to Identify Target Trajectory.* Master's Thesis. Brno University of Technology, Faculty of Mechanical Engineering, 2021. 103 s. Supervisor doc. RNDr. Libor Žák, Ph.D.

Rozšírený abstrakt

Táto práca sa zaoberá odstraňovaním šumu, resp. nepresnosti, meraní vznikajúcom pri tzv. multilateračných meraniach. Je to technológia, kedy objekt vyšle signál, ktorý zachytia 4 vysielače v rôznych časoch aby následne z toho odhadol systém polohu leteckého cieľa. Spolu s odhadom tejto polohy je systém schopný vydávať aj variančnú maticu šumu týchto meraní. Jej výpočet je založený na fyzikálnych vlastnostiach signálu, ktorý systém zachytil.

Ďalej je v práci predstavený Kálmánov filter. Je podrobne popísaný tzv. lineárny stavový model, ktorý popisuje pohyb leteckého cieľa. Na základe tohoto modelu je podrobne odvodený spomínaný Kálmánov filter. Tento algoritmus predstavuje Bayesovský odhad neznámej polohy cieľa z nepresných meraní. Funguje na princípe, že apriórnu hustotu odvodí z lineárneho stavového modelu a vierohodnosť z aktuálnych meraní. Ich vynásobením dostávame aposteriórnu hustotu, ktorej strednú hodnotu považujeme za odhad neznámej polohy cieľa. V prípade lineárneho stavového modelu je možné túto hustotu spočítať jednoducho pomocou maticových operácii, čo je princíp Kálmánovho filtra.

Je popísaný problém využitia Kálmánovho filtra. Je dokázané, že je to najlepší nestranný odhad neznámej polohy, ale za predpokladu, že lineárny stavový model dobre popisuje sledovaný letecký cieľ. Čo žiaľ nie je možné vždy zaručiť. V prípade, že Kálmánov filter neodhaduje neznámu polohu správne, tzn. odhad nie je nestranný (v takom prípade model nepopisuje správne sledovaný cieľ), je možné model upraviť, aby Kálmánov filter poskytol nestranný odhad. Takéto úpravy môžu byť dvoch typov – zmena geometrie trajektórie modelu alebo zmena procesného šumu. Zmena geometrie predstavuje napríklad zmenu z predpokladaného pohybu po priamke na predpokladaný pohyb po parabole alebo kružnici. Toto je údaj, ktorý popisuje spomenutý lineárny stavový model. Procesný šum prestavuje náhodnú veličinu, ktorá hovorí o samotnom modeli ako moc sa môže predpokladaná geometria trajektórie leteckého cieľa líšiť od skutočnej.

Samozrejme, je nepraktické tieto zmeny nastavovať ručne a cieľom tejto práce je odhadnúť okrem neznámej polohy aj trajektóriu a procesný šum leteckého cieľa. Na odhad trajektórie sa dnes už bežne využíva algoritmus IMM (Interacting Multiple Model). Tento je v práci podrobne odvodený. Spočíva v tom, že je potrebné vybrať niekoľko možných trajektórii, z ktorých potom algoritmus odhadne ten najbližší k skutočnej trajektórii. Problémom je, že každému modelu je potrebné nastaviť spomenutý procesný šum ako konštantu.

Najpodstatnejšiu a originálnu časť tvorí kapitola o odhadovaní procesného šumu. Na rozdiel od dostupnej literatúry je tu potrebné predpokladať, že merania sú ďaleko nepresnejšie než trajektória. Teda šum meraní je rádovo vyšší než procesný šum. To vyraďuje takmer všetky spôsoby dostupné v literatúre, obzvlášť korelačné a kovariačné metódy. V práci je odvodený Bayesovský prístup, prístup cez maximálnu vierohodnosť. Následne úpravy ako vylepšený prístup cez maximálnu vierohodnosť alebo riešenie problému zmien procesného šumu počas trajektórie. Čo popisuje manévrujúci cieľ.

Na záver je navrhnutá kombinácia aj odhadu trajektórie spolu s odhadom procesného šumu. Nanešťastie sa ukázalo, že nie je možné tieto dva prístupy kombinovať spolu s algoritmom IMM, avšak je navrhnutý prístup, ktorý odhadne geometriu trajektórie cez výšku odhadnutého procesného šumu. T.j. uplatní princíp, že modely s nižším procesným šumom lepšie popisujú geometriu trajektórie leteckého cieľa.

I hereby declare that I have written my Master's Thesis *Estimating of Motion Models and its Parameters to Identify Target Trajectory* independently under the supervision of doc. RNDr. Libor Žák, Ph.D. using literature listed in the bibliography section at the end of the thesis.

Bc. Matej Benko

At this place, I would like to express my gratitude to doc. RNDr. Libor Žák, Ph.D. for his patience and willingness during the consultations and guidance of my Master's Thesis. Moreover, I would like to thank Ing. Pavel Kulmon from ERA a.s. for his friendliness and valuable advice. Last but not least, I thank my family and friends for their support during the study.

Bc. Matej Benko

Contents

1 Introduction

Multilateration is the way of getting some flying object position. Unfortunately, it provides with high inaccuracy. In this text, we will try to develop techniques to suppress this inaccuracy.

We will base our consideration on Bayesian estimation Assume that each flying object (so-called target) flies according to some defined stochastic model. From this model, we deduce the prior distribution of the target position, and from measurement, we deduce its likelihood. The by Bayesian principle, we provide the posterior distribution of the target position. Moreover, we can exclude that, and together with position, we can estimate the target's velocity.

The main problem studied in the text will be to find the proper stochastic model in order to deduce the prior distribution of the state. The most significant part of the developed theory and proofs will be based on Bayesian estimation theory. The author follows the notation in [1] and assumes that the reader has knowledge from Bayesian estimation [7].

This text will be used simulated multilateration measurements corresponding to the real one used in real positioning systems.

2 | Multilateration

Multilateration (MLAT) [27] is a technique for determining the position of a target. The target transmits an energy wave (e.g., radio signal). It can be of two types: cooperative target – transmits voluntary or non-cooperative target – transmits inadvertently. Positioning is based on comparing the time when the signal arrives to individual receivers.

2.1 | A Geometrical Approach to the TDOA Measurements

In this section, we will deal with target positioning from so-called TDOA measurements. It will be defined below. Let us denote $T \in E^3$ the (unknown) position of the target, which we want to determine. So at first, we start with the definition of a positioning system.

Definition 2.1 (Passive positioning system)**.** Let R_0, R_1, R_2, R_3 be four points in E^3. They symbolize the known positions of 4 receivers. We call *passive positioning system* the following ordered 4-tuple

$$\langle R_0, R_1, R_2, R_3 \rangle.$$

Definition 2.2 (Time of arrival (TOA) to the i^{th} receiver)**.** Consider the passive positioning system defined above. The difference between the time of transmission of a signal from the target and the arrival of the signal to the i-th receiver we call the *time of arrival (TOA) to the i^{th} receiver* and denote τ_i.

Remark. In the passive positioning system, we have 4 TOAs at the same time. See it on Figure 2.1.

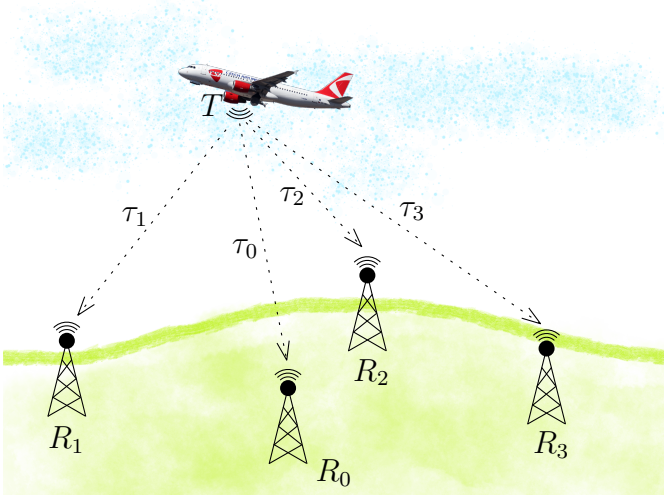


Figure 2.1: Presented passive positioning system with 4 TOAs.

Proposition 2.3. *Let $c = 299\,792\,458$ m/s be the speed of light. Then distance between target and i-th receiver $\rho(T, R_i)$ is equal to $c\tau_i$.*

In order to find out the target position from TOAs, it is handy to consider differences between them instead of TOAs themselves.

Definition 2.4 (Time difference of arrival (TDOA) on the j^{th} receiver)**.** Consider the passive positioning system defined above. The difference between the TOA on j-th receiver and the TOA on the 0^{th}-th receiver we call the *time difference of arrival (TDOA) on the j-th receiver* and denote

$$\hat{\tau}_j := \tau_j - \tau_0,\ j \in \{1, 2, 3\}. \tag{2.1}$$

Remark. In passive positioning system we have 3 TDOA measurements. $\hat{\tau}_j \in \mathbb{R} \setminus \{0\}$, because in case $\hat{\tau}_j = 0$ j^{th} and 0^{th} receiver have the same position, which is not feasible. The sign of TDOA signify if target is closer to R_j (negative) or R_0 (positive). From one TDOA, we can generate a surface of potential target positions.

The idea of positioning from TDOA measurements is to generate 3 surfaces of potential target position T from 3 TDOAs. Then the intersection of these surfaces is the demanded position T of the target. We are going to show that surface of potential T from 1 TDOA is one sheet from hyperboloid of two sheets. So, at first, we define what is a hyperboloid of two sheets.

Definition 2.5 (Rigid transformation)**.** Let be two affine frames $\langle A, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\rangle$ and $\langle B, \mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\rangle$ in E^3. The second is given by*rigid transformation* of first one iff its basis vectors are given by rotation of the first frame:

$$\mathbf{e}'_1 = \mathbf{R}(\phi, \theta, \psi)\mathbf{e}_1, \quad \mathbf{e}'_2 = \mathbf{R}(\phi, \theta, \psi)\mathbf{e}_2, \quad \mathbf{e}'_3 = \mathbf{R}(\phi, \theta, \psi)\mathbf{e}_3; \quad \mathbf{R}(\phi, \theta, \psi) \in SO(3).$$

and origin B is given by translation of A:

$$B = A + \mathbf{t}; \quad \mathbf{t} \in \mathbb{R}^3. \tag{2.2}$$

Remark. $SO(3)$ denotes a nonabelian group of all rotations about the origin in $\mathbb{R}^3$ represented by following matrices

$$\mathbf{R}(\phi, \theta, \psi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix}$$

By Euler's rotation theorem, it is composition of 3 rotations: counterclockwise rotation about the positive z-axis by angle ϕ after counterclockwise rotation about the positive y-axis by angle θ after counterclockwise rotation about the positive x-axis by angle ψ. Axis xyz are with respect to original affine frame $\langle A, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e_3}\rangle$.

Definition 2.6 (Hyperboloid of two sheets)**.** We call *hyperboloid of two sheets* a quadratic surface defined with normal form

$$\frac{x'^2}{\alpha^2} - \frac{y'^2}{\beta^2} - \frac{z'^2}{\gamma^2} - 1 = 0,$$

where $\alpha, \beta, \gamma \in \mathbb{R}^+$ and axis $x'y'z'$ are with respect to suitable affine frame $\langle S, \mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\rangle$ given by rigid transformation of cartesian affine frame $\langle O, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e_3}\rangle$.

Proposition 2.7. *Let R_0, R_1 be two points in E^3. Consider affine frame $\langle S, \mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\rangle$ given by rigid transformation of cartesian affine frame $\langle O, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e_3}\rangle$, such that*

$$S = \frac{1}{2}(R_0 + R_1)$$

and $R_0 = [-d, 0, 0]_{x'y'z'}$ and $R_1 = [d, 0, 0]_{x'y'z'}$ with respect to $\langle S, \mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\rangle$, where

$$2d = \rho(R_0, R_1).$$

Then the surface of possible points $P \in E^3$, which have constant differences between distances from P to R_0 and R_1

$$|\rho(P, R_0) - \rho(P, R_1)| = 2a, \qquad a \in \mathbb{R}^+ \tag{2.3}$$

is defined by equation

$$\frac{x'^2}{a^2} - \frac{y'^2}{b^2} - \frac{z'^2}{b^2} = 1, \qquad b^2 = d^2 - a^2, \tag{2.4}$$

where x', y', z' are coordinates of X with respect to $\langle S, \mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\rangle$.

Proof. Consider $X = [x', y', z']$ with respect to $\langle S, \mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3\rangle$. By notation introduced in this proposition:

$$\begin{aligned}
2a &= |\rho(P, R_0) - \rho(P, R_1)| \\
2a &= |\sqrt{(x' - (-d))^2 + (y' - 0)^2 + (z' - 0)^2} - \sqrt{(x' - d)^2 + a(y' - 0)^2 + (z' - 0)^2}|; \\
2a &= |\sqrt{(x' + d)^2 + y'^2 + z'^2} - \sqrt{(x' - d)^2 + y'^2 + z'^2}|.
\end{aligned}$$

Due to absolute value there are 2 options. $\rho(P, R_0) - \rho(P, R_1) > 0$ or < 0. At first, consider first option (greater then zero). Move the right square root to the left side of the equation and continue with algebraic rearrangements:

$$\begin{aligned}
2a + \sqrt{(x' - d)^2 + y'^2 + z'^2} &= \sqrt{(x' + d)^2 + y'^2 + z'^2} \\
\left(2a + \sqrt{(x' - d)^2 + y'^2 + z'^2}\right)^2 &= (x' + d)^2 + y'^2 + z'^2 \\
4a^2 + 4a\sqrt{(x' - d)^2 + y'^2 + z'^2} + (x' - d)^2 + y'^2 + z'^2 &= (x' + d)^2 + y'^2 + z'^2 \\
4a^2 + 4a\sqrt{(x' - d)^2 + y'^2 + z'^2} + x'^2 - 2x'd + d^2 &= x'^2 + 2x'd + d^2 \\
4a\sqrt{(x' - d)^2 + y'^2 + z'^2} &= 4x'd - 4a^2 \\
a^2\left((x' - d)^2 + y'^2 + z'^2\right) &= \left(dx' - a^2\right)^2 \\
a^2x'^2 - 2a^2dx' + a^2d^2 + a^2y'^2 + a^2z'^2 &= d^2x'^2 - 2a^2dx' + a^4 \\
(a^2 - d^2)x'^2 + a^2y'^2 + a^2z'^2 &= a^4 - a^2d^2 \qquad / \, b^2 = d^2 - a^2 \\
b^2x'^2 - a^2y'^2 - a^2z'^2 &= a^2b^2 \\
\frac{b^2x'^2}{a^2b^2} - \frac{a^2y'^2}{a^2b^2} - \frac{a^2z'^2}{a^2b^2} &= 1 \\
\frac{x'^2}{a^2} - \frac{y'^2}{b^2} - \frac{z'^2}{b^2} &= 1
\end{aligned}$$

Consider second option $\rho(X, R_0) - \rho(X, R_1) < 0$:

$$|\rho(X, R_0) - \rho(X, R_1)| = -\rho(X, R_0) + \rho(X, R_1)$$

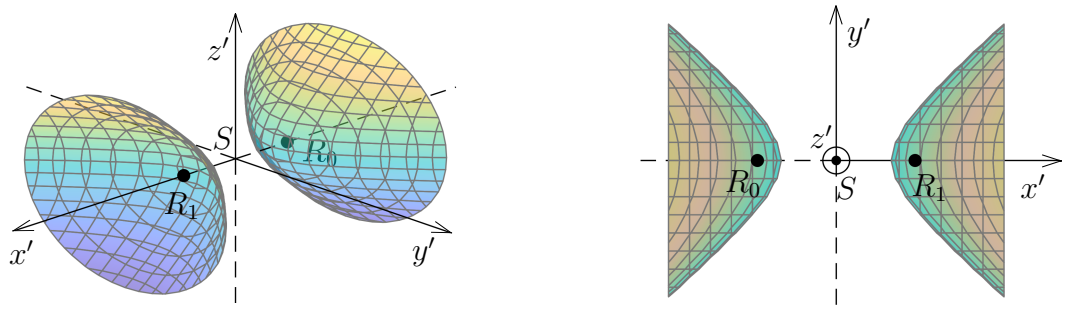
Then

$$\begin{aligned}
2a - \sqrt{(x'-d)^2 + y'^2 + z'^2} &= -\sqrt{(x'+d)^2 + y'^2 + z'^2} \\
\left(2a - \sqrt{(x'-d)^2 + y'^2 + z'^2}\right)^2 &= (x'+d)^2 + y'^2 + z'^2 \\
4a^2 - 4a\sqrt{(x'-d)^2 + y'^2 + z'^2} + (x'-d)^2 + y'^2 + z'^2 &= (x'+d)^2 + y'^2 + z'^2 \\
4a^2 - 4a\sqrt{(x'-d)^2 + y'^2 + z'^2} + x'^2 - 2x'd + d^2 &= x'^2 + 2x'd + d^2 \\
-4a\sqrt{(x'-d)^2 + y'^2 + z'^2} &= 4x'd - 4a^2 \\
a^2\left((x'-d)^2 + y'^2 + z'^2\right) &= \left(dx' - a^2\right)^2
\end{aligned}$$

Other steps are the same as above. □

Remark. The surface in Proposition 2.7 is special case of two-sheets hyperboloid (symmetric about x' axis). Compare with Definition 2.6.

Example 2.8. We have points R_0, R_1 such that $\rho(R_0, R_1) = 2\sqrt{2}$. The surface of points X with differences between distances to these points equal to 2 ($a = 1$ according to (2.3)) is drawn in Figure 2.2.



(a) in three-dimensional $x'y'z'$ space **(b)** projection into $x'y'$ plane

Figure 2.2: Surface with constant difference between distances to points R_0, R_1 (special case of two-sheets hyperboloid).

Proposition 2.9. *From 1 TDOA measurement $\hat{\tau}_i \neq 0$, surface of possible target position T is one sheet of hyperboloid of two sheets in Proposition 2.7. In case of $\hat{\tau}_i > 0$, it is sheet closer to receiver R_0, in case $\hat{\tau}_i < 0$, it is sheet closer to receiver R_1.*

Proof. With no loss of generality, we consider TDOA on the 1^{st} receiver: $\hat{\tau}_1$. So, consider receivers R_0, R_1 with known positions. We identify these receivers with points R_0, R_1 in Proposition 2.7.

Recall the Definition 2.4 of TDOA on the 1^{st} receiver:

$$\hat{\tau}_1 = \tau_1 - \tau_0.$$

By Proposition 2.3:

$$\rho(T, R_0) = c\tau_0, \quad \rho(T, R_1) = c\tau_1$$

$$\Rightarrow \rho(T, R_1) - \rho(T, R_0) = c\tau_1 - c\tau_0 = c(\tau_1 - \tau_0) = c\hat{\tau}_i \tag{2.5}$$

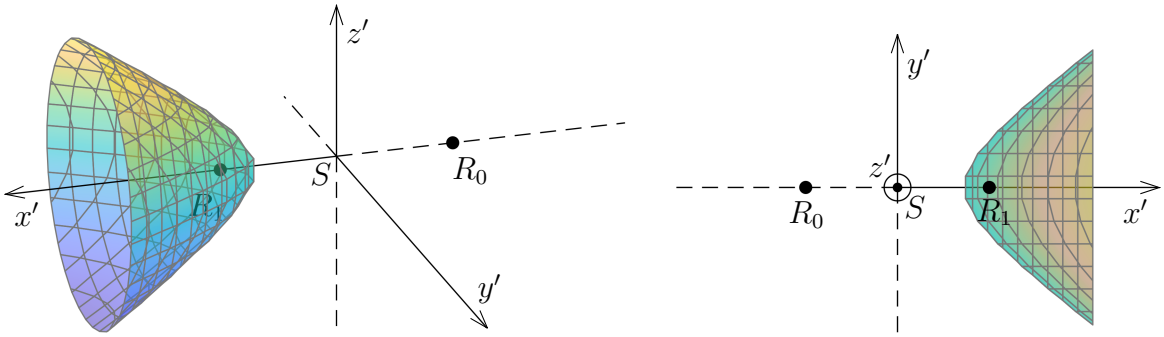
In the case of both sides in (2.5) are in absolute value, the situation would is the same as in Proposition 2.7. The surface of possible target position T would have been two-sheets hyperboloid determined by (2.4).

But in this case, we have one more piece of information. We know, if $\rho(T, R_0)$ is greater or not than $\rho(T, R_1)$ (recall that equality is not feasible). By that, only one sheet of two-sheets hyperboloid would satisfy this additional condition. It is clear to state, that in case of $\hat{\tau}_i > 0$, it is sheet closer to receiver R_0, in case of $\hat{\tau}_i < 0$, it is sheet closer to receiver R_1. □

Example 2.10. We have positioning system. $\rho(R_0, R_1) = 2\sqrt{2}$ km. Consider 1 TDOA measurement on 1^{st} receiver. We obtained its value: $\hat{\tau}_1 = 6,6713 \cdot 10^{-6}$ s. By that, we can evaluate

$$\rho(T, R_1) - \rho(T, R_0) = c\hat{\tau}_1 = 299\,792\,458 \text{ m/s} \cdot (-6,6713 \cdot 10^{-6} \text{ s}) = -2 \text{ km}.$$

Together knowledge of $|\rho(T, R_1) - \rho(T, R_0)|$ and $\rho(R_0, R_1)$ determines the hyperboloid of two sheets from Proposition 2.7 and the sign of $\hat{\tau}_1$ determines the sheet. In this case the sheet closer to R_1. See on Figure 2.3.



(a) in three-dimensional $x'y'z'$ space **(b)** projection into $x'y'$ plane

Figure 2.3: The surface of possible target position T from 1 negative TDOA measurement.

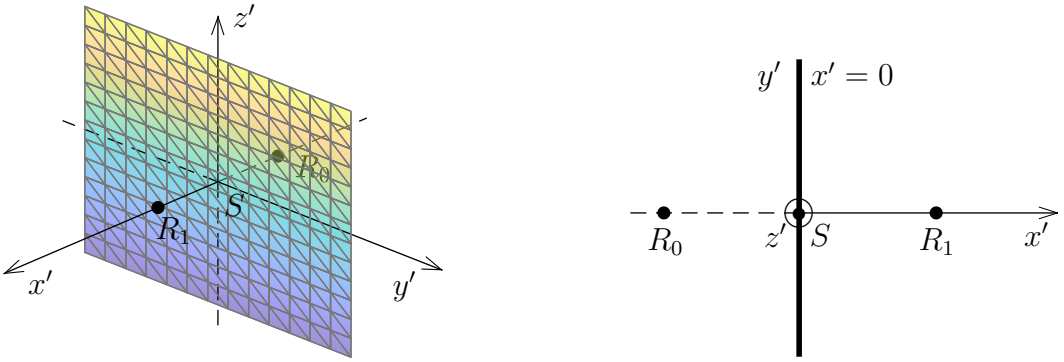
Proposition 2.11. *Let be $\hat{\tau}_1 = 0$. Then, surface of possible target positions T is plane $x' = 0$ with respect to local affine frame $\langle S, \mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3 \rangle$.*

Proof. $\hat{\tau}_1 = 0 \Rightarrow \rho(T, R_0) = \rho(T, R_1)$. Then

$$\begin{aligned}
\sqrt{(x'-d)^2 + y'^2 + z'^2} &= \sqrt{(x'+d)^2 + y'^2 + z'^2} \\
(x'-d)^2 + y'^2 + z'^2 &= (x'+d)^2 + y'^2 + z'^2 \\
(x'-d)^2 &= (x'+d)^2 \\
x'^2 - 2dx' + d^2 &= x'^2 + 2dx' + d^2 \\
-2dx' &= +2dx' && / \, d \neq 0 \\
x' &= 0
\end{aligned}$$

□

Example 2.12. We have positioning system. $\rho(R_0, R_1) = 2\sqrt{2}$ km. Consider 1 TDOA measurement on 1^{st} receiver. We obtained its value: $\hat{\tau}_1 = 0$ s. By that, we can evaluate surface of possible target positions T as plane $x' = 0$. See on Figure 2.4



(a) in three-dimensional $x'y'z'$ space **(b)** projection into $x'y'$ plane

Figure 2.4: The surface of possible target position T from 1 zero TDOA measurement.

Remark. By previous Proposition 2.11 we solved all situation for TDOA (zero and non-zero time difference of arrival).

As it was mentioned above, the idea is to find the target position T by finding the intersection of 3 surfaces from 3 TDOA measurements. Precise description of the algorithm to find intersection is out of the scope of the text.

Example 2.13. We have passive positioning system with known positions of 4 receivers in cartesian affine frame:

$$R_0 = [-4980, 3154, 318] \text{ m}, \qquad R_1 = [-2143, -2986, 232] \text{ m},$$
$$R_2 = [1997, -3154, 421] \text{ m}, \qquad R_3 = [5001, 3078, 157] \text{ m}.$$

On that receivers obtain 3 TDOA measurements:

$$\hat{\tau}_1 = 5,4891 \cdot 10^{-6} \text{ s}, \qquad \hat{\tau}_2 = 4,4188 \cdot 10^{-6} \text{ s}, \qquad \hat{\tau}_3 = -2,8730 \cdot 10^{-6} \text{ s}.$$

Let us generate surfaces of possible target of individual TDOAs. We will skip the computations, for sake of brevity only discuss the principle of them.

The idea is construct surface (1 sheet of two-sheets hyperboloid) from 1 TDOA in local affine frame $\langle S, \mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3 \rangle$. Then do that for all 3 TDOAs in their own local affine frames. Use inverse rigid transformation to map these 3 surfaces to global cartesian affine frame $\langle O, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \rangle$. When we have 3 surfaces in global cartesian affine frame, we choose some algorithm to find intersection of them. See the situation in Figures 2.5 and 2.6.

It is worth to mention, that due to rotational symmetry about x' axis of the surface of possible target position T, to find matrix $\mathbf{R}(\phi, \theta, \psi)$ is enough to find only angles ϕ and θ. Because in that case is sufficient to consider not overall basis $\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3$ but only the rotation of vector $\mathbf{e}'_1$.

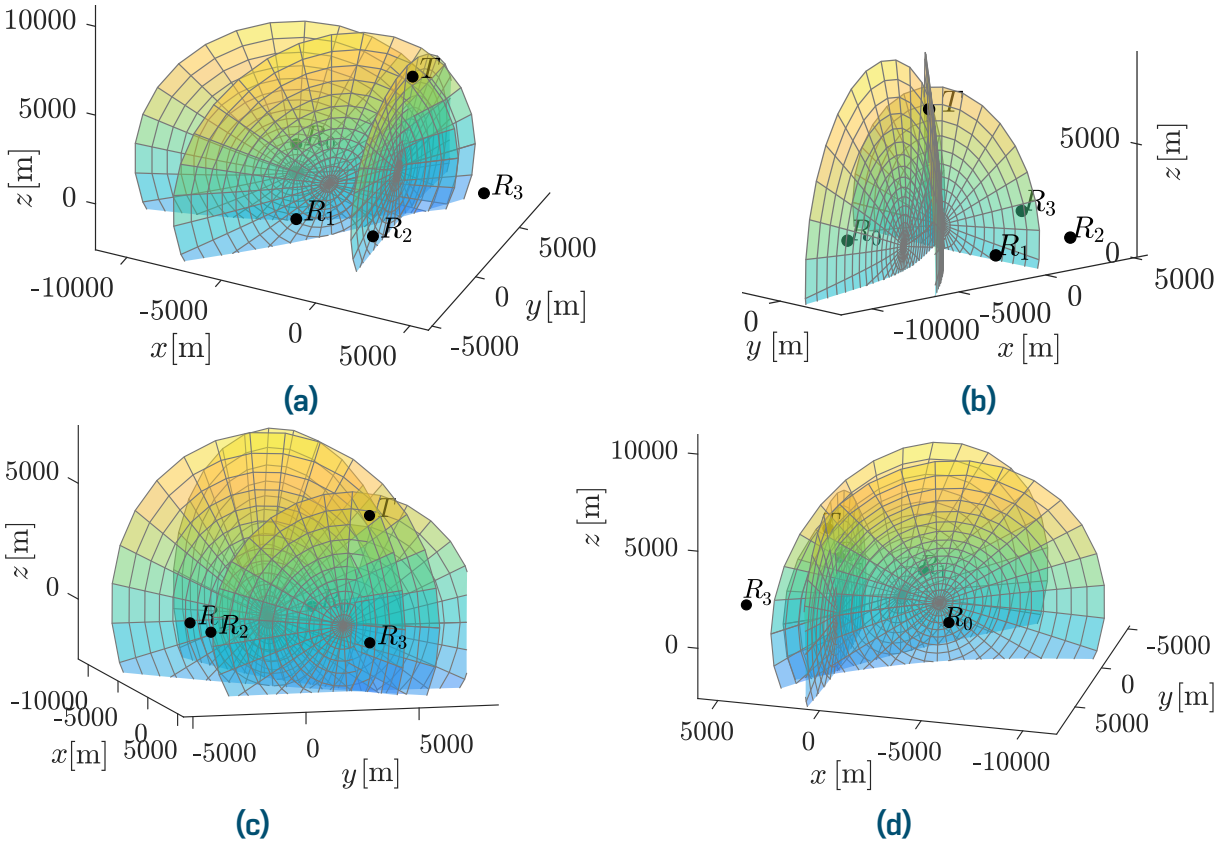
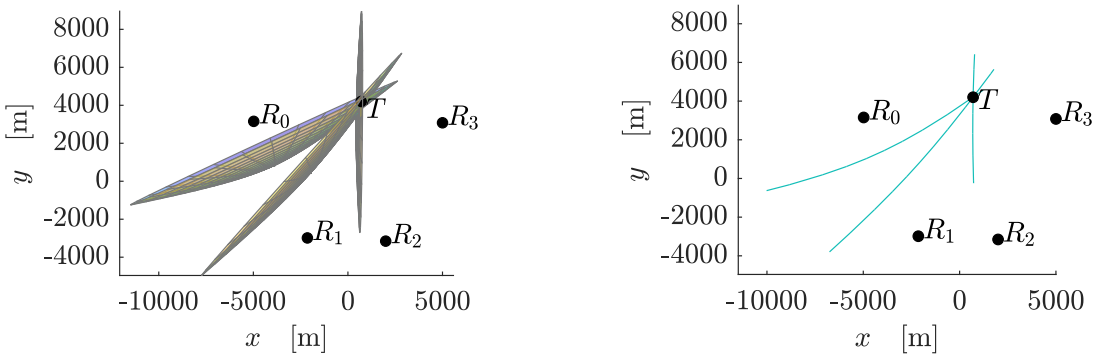


Figure 2.5: Surfaces of possible target position in cartesian affine frame.



(a) view from above **(b)** situation in height of the target

Figure 2.6: Surfaces of possible target position in cartesian affine frame in xy plane.

2.2 | Accuracy of the TDOA Measurements

In the previous section, we dealt with TDOA as precise measurements. In technical practice, it is not a proper approach. TDOA always have some error. From estimation theory including physical properties of radio signal, we can determine Cramér-Rao Lower Bound [14] of TDOA measurements. Let be $\sigma^2_{\hat{\tau},j}$ variance of TDOA on the j^{th} receiver, then following inequality is valid by [35]:

$$\sigma^2_{\hat{\tau},j} \geq \left(2\frac{\mathscr{E}}{N_0}\mathscr{B}^2\right)^{-1}. \tag{2.6}$$

where $\mathscr{E}$ is the signal energy [J], N_0 is the noise spectral power density [W/Hz], and $\mathscr{B}$ [Hz] is a width measure of the signal bandwidth. The errors are dependent on the signal and its properties: noise levels and the signal's bandwidth.

Definition 2.14 (Inexact TDOA on the j^{th} receiver)**.** Consider defined time difference of arrival (TDOA) on the j^{th} receiver denoted as $\hat{\tau}_j$. We call *inexact TDOA on j^{th} receiver* following ordered couple

$$(\hat{\tau}_j, \sigma^2_{\hat{\tau},j}),$$

where $\hat{\tau}_j$ and $\sigma^2_{\hat{\tau},j}$ are considered to be estimates of parameters of normal random variable Υ_j.

In applications (including real data in this text), σ_j^2 is taken as its Cramér-Rao Lower Bound (2.6). 3 TDOAs are considered as normal random vector $\boldsymbol{\Upsilon} \sim \mathsf{N}(\boldsymbol{\mu}_{\hat{\tau}}, \boldsymbol{\Sigma}_{\hat{\tau}})$ with dimension of 3 with estimate of its parameters:

$$\hat{\boldsymbol{\mu}}_{\hat{\tau}} = \begin{bmatrix} \hat{\tau}_1 \\ \hat{\tau}_2 \\ \hat{\tau}_3 \end{bmatrix}, \qquad \hat{\boldsymbol{\Sigma}}_{\hat{\tau}} = \begin{bmatrix} \sigma^2_{\hat{\tau},1} & 0 & 0 \\ 0 & \sigma^2_{\hat{\tau},2} & 0 \\ 0 & 0 & \sigma^2_{\hat{\tau},3} \end{bmatrix}.$$

We are able to reformulate problem of finding intersection of 3 sheets of two-sheets hyperboloid to nonlinear transform from space of TDOA measurements to cartesian space. In previous section, we have denoted $T = [x, y, z]$ as target position in E^3. Let us denote $\boldsymbol{Z}$ as random vector of target position in $\mathbb{R}^3$. Consider in general nonlinear function of random vector g such that

$$\boldsymbol{Z} = g(\boldsymbol{\Upsilon}).$$

In applications (including in this text), it is also assumed that $\boldsymbol{Z}$ has normal distribution, $\boldsymbol{Z} \sim \mathsf{N}(\bar{\mathbf{z}}, \mathbf{R})$.

Remark. It is worth to mention that $\boldsymbol{Z}$ does not have normal distribution, because of nonlinear mapping g. It does not preserve normality. But in practice, it is assumed that distribution of $\boldsymbol{Z}$ is enough close to normal distribution. So, we are able to approximate it by normal distribution. Variance matrix $\mathbf{R}$ is estimated on maximum-likelihood principle. For sake of simplicity, denote the estimation also with $\mathbf{R}$. Mean $\mathbf{z}$ is estimated as $\hat{\bar{\mathbf{z}}} = \boldsymbol{z} = g([\hat{\tau}_1, \hat{\tau}_2, \hat{\tau}_3]^\top)$

Definition 2.15 (Plot)**.** We call *plot* a random variable $\boldsymbol{Z} \sim \mathsf{N}(\boldsymbol{z}, \mathbf{R})$ described above.

Remark. Plots will be considered as raw measurements through this text. In the next chapter, we will start with smoothing them with the aim of getting a better estimate of target position T and the ability of its velocity estimation.

3 | Kalman Filtering

The core of this chapter is Kalman filter. It is an algorithm invented and presented in 1960 in article [13] by Hungarian-American electrical engineer and mathematician Rudolph Emil Kálmán. The first public known usage of this algorithm was in the Apollo program estimating the space module trajectory.

Kalman filter forms a basic concept of recursive Bayesian estimation theory studied in this text. In this chapter, we introduce a scheme of the linear state-space system, which is the hidden Markov model and then derive the Kalman filter. At the end of this chapter, we show that the Kalman filter is optimal Bayesian estimator.

We mainly use Bayesian estimation theory [7].

3.1 | Linear Dynamic System

Definition 3.1 (State vector)**.** Let $\boldsymbol{X} \sim \mathsf{N}(\bar{\mathbf{x}}, \mathbf{P}) \in \mathbb{R}^n$ be random vector of true state with normal distribution and dimension of n. We call it *state vector*.

Definition 3.2 (Control-input vector)**.** Let $\mathbf{u} \in \mathbb{R}^p$ be deterministic vector with dimension of p. In case its elements represent physics quantity we call it *control input vector*.

Definition 3.3 (Measurement vector)**.** Let $\boldsymbol{Z} \sim \mathsf{N}(\bar{\mathbf{z}}, \mathbf{R}) \in \mathbb{R}^m$ be random vector of normal distribution with dimension of m representing noisy measurements. We call it *measurement vector*.

Remark. Latter in the text we will identify general measurement vector in Stochastic Control Theory with plot defined in Chapter 2: Multilateration in order to deal with target tracking problems. Estimation of variance matrix $\mathbf{R}$ in plot will be considered as true value.

Remark. We add to the quantities defined above index symbolizes time index in discrete time series. E.g. $\boldsymbol{X}_k, \mathbf{u_k}, \boldsymbol{Z}_k; k \in \mathbb{N}$ symbolizes defined quantities at time epoch k.

Definition 3.4 (Linear state-space system)**.** Let $\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_k, \ldots \in \mathbb{R}^n$ be sequence of state vectors, $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_k, \ldots \in \mathbb{R}^m$ sequence of measurement vectors and $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_k, \ldots \in \mathbb{R}^p$ be sequence of control-input vectors. We call *linear state-space system* a following system of stochastic difference equations:

$$\boldsymbol{X}_k = \mathbf{F}_k \boldsymbol{X}_{k-1} + \mathbf{G}_k \mathbf{u}_k + \boldsymbol{\Gamma}_k \boldsymbol{V}_k, \tag{3.1}$$

$$\boldsymbol{Z}_k = \mathbf{H}_k \boldsymbol{X}_k + \boldsymbol{W}_k; \tag{3.2}$$

where $\mathbf{F}_k \in \mathcal{M}_n(\mathbb{R})$ is state transition model, $\mathbf{H}_k \in \mathcal{M}_{m,n}(\mathbb{R})$ is observation model and $\mathbf{G}_{k-1} \in \mathcal{M}_{n,p}(\mathbb{R})$ is the control-input model.$\boldsymbol{\Gamma}_k \in \mathcal{M}_{n,q}$ is process noise transform model into state space, $\boldsymbol{V}_k \sim \mathsf{N}(\mathbf{o}, \mathbf{Q}_k)$ is the process noise where $\mathbf{Q}_k \in PD_q(\mathbb{R})$ and $\boldsymbol{W}_k \sim \mathsf{N}(\mathbf{o}, \mathbf{R}_k)$ is the measurement noise. Initial state $\boldsymbol{X}_0 \sim \mathsf{N}(\bar{\mathbf{x}}_0, \mathbf{P}_0)$ is given.

Proposition 3.5. *State vector $\boldsymbol{X}_k$ has normal distribution.*

Proof. By Equation (3.1). For $k = 1$: $\boldsymbol{X}_0 \sim \mathsf{N}(\bar{\mathbf{x}}_0, \mathbf{P}_0)$ is given and $\boldsymbol{V}_k \sim \mathsf{N}(\mathbf{o}, \mathbf{Q}_k)$. $\mathbf{u}_k$ is not a random variable (it is known constant). Then $\boldsymbol{X}_1$ have also normal distribution, because it is linear transformation of normal random variables and constant. Then $\boldsymbol{X}_k$ has normal distribution inductively for $k > 1$. □

Remark. For parameters of normal distribution of $\boldsymbol{X}_k$ we introduce notation

$$\boldsymbol{X}_k \sim \mathsf{N}(\bar{\mathbf{x}}_k, \mathbf{P}_k).$$

To make clear previous terms in Definition 3.4, see the following example of thrown ball with positioning sensor.

Example 3.6. Consider thrown ball with positioning sensor. We track its height. So consider the state transition model and the state vector at time epoch k such that

$$\mathbf{F}_k = \begin{bmatrix} 1 & \Delta t_k \\ 0 & 1 \end{bmatrix}, \qquad \boldsymbol{X}_k = \begin{bmatrix} H \\ \dot{H} \end{bmatrix}.$$

where H is true unknown height, $\dot{H}$ true unknown velocity in vertical direction to the ground, Δt_k is time difference between epoch k and $k-1$; $\Delta t_k := t_k - t_{k-1}$. Control input in that case is known gravitational acceleration $g = 9,81$ m/s.

$$\mathbf{G}_k = \begin{bmatrix} \Delta t_k^2/2 \\ \Delta t_k \end{bmatrix}, \qquad \mathbf{u}_k = \begin{bmatrix} -g \end{bmatrix}, \qquad \mathbf{Q}_k = \begin{bmatrix} \Delta t_k^4/4 & 0 \\ 0 & \Delta t_k^2/2 \end{bmatrix} \sigma_a^2.$$

σ_a^2 [m^2/s^4] is given (expected) variance of ball's total acceleration in vertical direction. Positioning sensor can measure height with known error variance such that

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad \mathbf{R}_k = \begin{bmatrix} \sigma_w^2 \end{bmatrix},$$

where σ_w^2 [m^2] is known. For completeness dimensions in this example are

$$n = 2, \qquad p = 1, \qquad m = 1.$$

3.2 | Recursive Bayesian Estimate of the States

The linear state-space system presented in previous section is one of the simplest dynamic Bayesian networks. The true states $\boldsymbol{X}_k$ are unobserved Markov process and the measurements $\boldsymbol{Z}_k$ are observed states of a hidden Markov model. See on Figure 3.1.

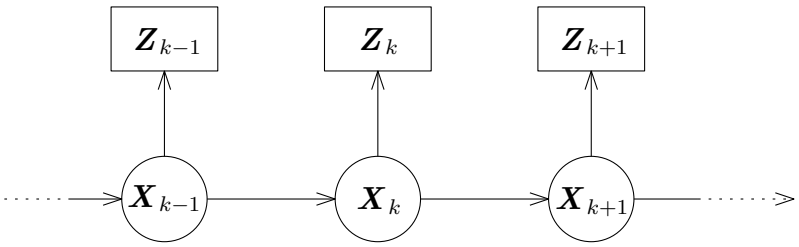


Figure 3.1: Scheme of discussed Hidden Markov Model

Let us denote $\boldsymbol{x}_k$ as a realization of $\boldsymbol{X}_k$. Then, by Markov assumption is valid this equation

$$f(\boldsymbol{x}_k|\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{k-1}) = f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}).$$

Similarly, let us denote $\boldsymbol{z}_k$ as a realization of measurement vector $\boldsymbol{Z}_k$. $\boldsymbol{Z}_k$ is depended only on $\boldsymbol{X}_k$ and is independent from all state vector $\boldsymbol{X}_0,\ldots,\boldsymbol{X}_{k-1}$:

$$f(\boldsymbol{z}_k|\boldsymbol{x}_0,\ldots,\boldsymbol{x}_k) = f(\boldsymbol{z}_k|\boldsymbol{x}_k)$$

Remark. It is worth to recall that by Definition 3.4 initial probability density function $f(\boldsymbol{x}_0)$ is known (respectively given and assumed to be true).

Before continue with study of recursive Bayesian estimation, recall very useful proposition from [1] (Section 3.4, Page 57).

Proposition 3.7. *For conditional density function is equation*

$$f(\boldsymbol{y},\boldsymbol{z}) = f(\boldsymbol{z}|\boldsymbol{y})f(\boldsymbol{y})$$

valid almost everywhere.

Proposition 3.8. *Consider Hidden Markov model described above, probability density function of all states can be written as*

$$f(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_k,\boldsymbol{z}_1,\ldots,\boldsymbol{z}_k) = f(\boldsymbol{x}_0)\prod_{i=1}^{k} f(\boldsymbol{z}_k|\boldsymbol{x}_i)f(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}).$$

Proof. To make it clear, we will do this proof inductively:

- $k = 0$:
$$f(\boldsymbol{x}_0) = f(\boldsymbol{x}_0).$$
- $k = 1$:
$$f(\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{z}_1) = f(\boldsymbol{x}_0,\boldsymbol{x}_1)f(\boldsymbol{z}_1|\boldsymbol{x}_0,\boldsymbol{x}_1) = f(\boldsymbol{x}_0)f(\boldsymbol{x}_1|\boldsymbol{x}_0)f(\boldsymbol{z}_1|\boldsymbol{x}_0,\boldsymbol{x}_1) = \\ = f(\boldsymbol{x}_0)f(\boldsymbol{x}_1|\boldsymbol{x}_0)f(\boldsymbol{z}_1|\boldsymbol{x}_1)$$
- $k = 2$:
$$f(\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{z}_1,\boldsymbol{z}_2) = f(\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{x}_2)f(\boldsymbol{z}_1,\boldsymbol{z}_2|\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{x}_2) = \\ = f(\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{x}_2)f(\boldsymbol{z}_1|\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{x}_2)f(\boldsymbol{z}_2|\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{x}_2) = f(\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{x}_2)f(\boldsymbol{z}_1|\boldsymbol{x}_1)f(\boldsymbol{z}_2|\boldsymbol{x}_2) = \\ = f(\boldsymbol{x}_2|\boldsymbol{x}_0,\boldsymbol{x}_1)f(\boldsymbol{x}_0,\boldsymbol{x}_1)f(\boldsymbol{z}_1|\boldsymbol{x}_1)f(\boldsymbol{z}_2|\boldsymbol{x}_2) = \\ = f(\boldsymbol{x}_2|\boldsymbol{x}_0,\boldsymbol{x}_1)f(\boldsymbol{x}_1|\boldsymbol{x}_0)f(\boldsymbol{x}_0)f(\boldsymbol{z}_1|\boldsymbol{x}_1)f(\boldsymbol{z}_2|\boldsymbol{x}_2) = \\ = f(\boldsymbol{x}_2|\boldsymbol{x}_1)f(\boldsymbol{x}_1|\boldsymbol{x}_0)f(\boldsymbol{x}_0)f(\boldsymbol{z}_1|\boldsymbol{x}_1)f(\boldsymbol{z}_2|\boldsymbol{x}_2)$$
- general k:
$$f(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_k,\boldsymbol{z}_1,\ldots,\boldsymbol{z}_k) = f(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_k)f(\boldsymbol{z}_1,\ldots,\boldsymbol{z}_k|\boldsymbol{x}_0,\ldots,\boldsymbol{x}_k) = \\ = f(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_k)\prod_{i=1}^{k} f(\boldsymbol{z}_i|\boldsymbol{x}_0,\ldots,\boldsymbol{x}_k) = f(\boldsymbol{x}_0,\ldots,\boldsymbol{x}_k)\prod_{i=1}^{k} f(\boldsymbol{z}_i|\boldsymbol{x}_i) = \\ = f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\ldots,\boldsymbol{x}_0)f(\boldsymbol{x}_{k-1},\ldots,\boldsymbol{x}_0)\prod_{i=1}^{k} f(\boldsymbol{z}_i|\boldsymbol{x}_i) = \\ = f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})f(\boldsymbol{x}_{k-1}|\boldsymbol{x}_{k-2})f(\boldsymbol{x}_{k-2},\ldots,\boldsymbol{x}_0)\prod_{i=1}^{k} f(\boldsymbol{z}_i|\boldsymbol{x}_i) = \ldots = \\ = f(\boldsymbol{x}_0)\prod_{i=1}^{k} f(\boldsymbol{z}_k|\boldsymbol{x}_i)f(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}).$$

□

Remark. This proposition shows sequential property of overall density. In next time epoch k we can multiply density at $k-1$ by $f(\boldsymbol{z}_k|\boldsymbol{x}_k)f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ in order to obtain probability density function of all states in the Hidden Markov Model.

3.3 | Useful Matrix Identities

In this section, we mention and prove useful lemmas in order to derive Kalman filter. First one is Woodbury matrix identity [41]. It is also known as *matrix inversion lemma*. It will be used several times during derivation of the Kalman filter.

Lemma 3.9 (Woodbury[1] matrix identity)**.** *Let be* $\mathbf{A}, \mathbf{U}, \mathbf{C}$ *and* $\mathbf{V}$ *be conformable matrices:* $\mathbf{A} \in \mathcal{M}_s(\mathbb{R}), \mathbf{U} \in \mathcal{M}_{s,t}(\mathbb{R}), \mathbf{C} \in \mathcal{M}_t(\mathbb{R}), \mathbf{V} \in \mathcal{M}_{t,s}(\mathbb{R})$*, then*

$$(\mathbf{A}+\mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}. \tag{3.3}$$

Proof. We do direct proof. It also possible to do proof in other ways like algebraic way, derivation via blockwise elimination and derivation from LDU decomposition. We prove following identity given from (3.3):

$$\begin{aligned}
\mathbf{I}_s &= (\mathbf{A}+\mathbf{UCV})\left[\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}\right] \\
&= \left[\mathbf{I}_s - \mathbf{U}(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}\right] + \\
&\quad + \left[\mathbf{UCVA}^{-1} - \mathbf{UCVA}^{-1}\mathbf{U}(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}\right] \\
&= \left[\mathbf{I}_s - \mathbf{UCVA}^{-1}\right] - \\
&\quad - \left[\mathbf{U}(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1} + \mathbf{UCVA}^{-1}\mathbf{U}(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}\right] \\
&= \mathbf{I}_s - \mathbf{UCVA}^{-1} - (\mathbf{U}+\mathbf{UCVA}^{-1}\mathbf{U})(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1} \\
&= \mathbf{I}_s - \mathbf{UCVA}^{-1} - \mathbf{UC}(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1} \\
&= \mathbf{I}_s - \mathbf{UCVA}^{-1} + \mathbf{UCVA}^{-1} \\
&= \mathbf{I}_s.
\end{aligned}$$

□

Following *Weinstein–Aronszajn identity* [34] is useful matrix identity which will be used in proof of a another identity called *fundamental gaussian identity* which will be discussed below.

Lemma 3.10 (Weinstein[2]–Aronszajn[3] identity)**.** *Let be* $\mathbf{U} \in \mathcal{M}_{s,t}(\mathbb{R})$ *and* $\mathbf{V} \in \mathcal{M}_{t,s}(\mathbb{R})$ *matrices. Then*

$$\det(\mathbf{I}_s + \mathbf{UV}) = \det(\mathbf{I}_t + \mathbf{VU}). \tag{3.4}$$

Proof. Let $\mathbf{M} \in \mathcal{M}_{s+t}(\mathbb{R})$ be block matrix comprising $\mathbf{I}_s, -\mathbf{U}, \mathbf{V}, \mathbf{I}_t$:

$$\mathbf{M} = \begin{pmatrix} \mathbf{I}_s & -\mathbf{U} \\ \mathbf{V} & \mathbf{I}_t \end{pmatrix}.$$

[1] Max Atkin Woodbury (*1917), American mathematician.
[2] Alexander Weinstein (1897 - 1979), Russian American mathematician.
[3] Nachman Aronszajn (1907 -1980), Polish American mathematician.

$\mathbf{I}_s$ is invertible, so by the formula for the determinant of a block matrix

$$\det\begin{pmatrix}\mathbf{I}_s & -\mathbf{U}\\ \mathbf{V} & \mathbf{I}_t\end{pmatrix} = \det(\mathbf{I}_s)\det(\mathbf{I}_t - \mathbf{V}\mathbf{I}_s^{-1}(-\mathbf{U})) = \det(\mathbf{I}_t + \mathbf{V}\mathbf{U}).$$

Similarly, $\mathbf{I}_t$ is invertible, so by the formula for the determinant of a block matrix

$$\det\begin{pmatrix}\mathbf{I}_s & -\mathbf{U}\\ \mathbf{V} & \mathbf{I}_t\end{pmatrix} = \det(\mathbf{I}_t)\det(\mathbf{I}_s - (-\mathbf{U})\mathbf{I}_t^{-1}\mathbf{V}) = \det(\mathbf{I}_s + \mathbf{U}\mathbf{V}).$$

Comparing equations above it is clear that

$$\det(\mathbf{I}_s + \mathbf{U}\mathbf{V}) = \det(\mathbf{I}_t + \mathbf{V}\mathbf{U}).$$

□

The last useful identity is *fundamental Gaussian identity* [26]. It plays key role during derivation of a prior and posterior densities in the Kalman filter. It express relation between probability density functions of the normal distribution. The relation between their arguments in exponential function is also called *combination of quadratic forms* [37].

Lemma 3.11 (Fundamental Gaussian identity)**.** *Let be* $\mathbf{\Psi} \in \mathcal{M}_s(\mathbb{R})$ *and* $\mathbf{\Sigma} \in \mathcal{M}_t(\mathbb{R})$ *symmetric positive-definite matrices such that* $s \leq t$*. Let be* $\mathbf{B} \in \mathcal{M}_{t,s}(\mathbb{R})$ *matrix. Then*

$$\mathsf{N}(\mathbf{r}; \mathbf{B}\mathbf{q} + \mathbf{j}, \mathbf{\Sigma}) \cdot \mathsf{N}(\mathbf{q}; \mathbf{p}, \mathbf{\Psi}) = \mathsf{N}(\mathbf{r}; \mathbf{B}\mathbf{p} + \mathbf{j}, \mathbf{\Sigma} + \mathbf{B}\mathbf{\Psi}\mathbf{B}^\top) \cdot \mathsf{N}(\mathbf{q}; \mathbf{e}, \mathbf{E}) \tag{3.5}$$

$\forall \mathbf{r} \in \mathbb{R}^t, \forall \mathbf{q}, \mathbf{p}, \mathbf{j} \in \mathbb{R}^s$ *and* $\mathbf{E}, \mathbf{e}$ *are defined by*

$$\begin{aligned}\mathbf{E}^{-1} &:= \mathbf{B}^\top\mathbf{\Sigma}^{-1}\mathbf{B} + \mathbf{\Psi}^{-1}\\ \mathbf{e} &:= \mathbf{p} + \mathbf{E}\mathbf{B}^\top\mathbf{\Sigma}^{-1}(\mathbf{r} - \mathbf{B}\mathbf{p} - \mathbf{j}).\end{aligned}$$

Proof. Rewrite left side of Equation (3.5):

$$\frac{1}{\sqrt{(2\pi)^t\det(\mathbf{\Sigma})}}\exp\left(-\frac{1}{2}(\mathbf{r} - \mathbf{B}\mathbf{q} - \mathbf{j})^\top\mathbf{\Sigma}^{-1}(\mathbf{r} - \mathbf{B}\mathbf{q} - \mathbf{j})\right)\cdot$$
$$\cdot\frac{1}{\sqrt{(2\pi)^s\det(\mathbf{\Psi})}}\exp\left(-\frac{1}{2}(\mathbf{q} - \mathbf{p})^\top\mathbf{\Psi}^{-1}(\mathbf{q} - \mathbf{p})\right) =$$

$$= \frac{1}{\sqrt{(2\pi)^t(2\pi)^s\det(\mathbf{\Sigma})\det(\mathbf{\Psi})}}\exp\left[-\frac{1}{2}\left((\mathbf{r} - \mathbf{B}\mathbf{q} - \mathbf{j})^\top\mathbf{\Sigma}^{-1}(\mathbf{r} - \mathbf{B}\mathbf{q} - \mathbf{j}) + \right.\right.$$
$$\left.\left. + (\mathbf{q} - \mathbf{p})^\top\mathbf{\Psi}^{-1}(\mathbf{q} - \mathbf{p})\right)\right]$$

Similarly rewrite the right side of Equation (3.5):

$$\frac{1}{\sqrt{(2\pi)^t\det(\mathbf{\Sigma} + \mathbf{B}\mathbf{\Psi}\mathbf{B}^\top)}}\exp\left(-\frac{1}{2}(\mathbf{r} - \mathbf{B}\mathbf{p} - \mathbf{j})^\top(\mathbf{\Sigma} + \mathbf{B}\mathbf{\Psi}\mathbf{B}^\top)^{-1}(\mathbf{r} - \mathbf{B}\mathbf{p} - \mathbf{j})\right)\cdot$$
$$\cdot\frac{1}{\sqrt{(2\pi)^s\det(\mathbf{E})}}\exp\left(-\frac{1}{2}(\mathbf{q} - \mathbf{e})^\top\mathbf{E}^{-1}(\mathbf{q} - \mathbf{e})\right) =$$

$$= \frac{1}{\sqrt{(2\pi)^t(2\pi)^s \det(\boldsymbol{\Sigma}+\mathbf{B}\boldsymbol{\Psi}\mathbf{B}^\top)\det(\mathbf{E})}} \exp\Big[-\frac{1}{2}\Big((\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top(\boldsymbol{\Sigma}+\mathbf{B}\boldsymbol{\Psi}\mathbf{B}^\top)^{-1}\cdot \cdot(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})+(\mathbf{q}-\mathbf{e})^\top\mathbf{E}^{-1}(\mathbf{q}-\mathbf{e})\Big)\Big]$$

Comparing rewritten left and right side of Equation (3.5) it is clear that is enough to prove following two equations:

$$(\mathbf{r}-\mathbf{B}\mathbf{q}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\cdot(\mathbf{r}-\mathbf{B}\mathbf{q}-\mathbf{j})+(\mathbf{q}-\mathbf{p})^\top\boldsymbol{\Psi}^{-1}(\mathbf{q}-\mathbf{p}) = \\ = (\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top(\boldsymbol{\Sigma}+\mathbf{B}\boldsymbol{\Psi}\mathbf{B}^\top)^{-1}\cdot(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})+(\mathbf{q}-\mathbf{e})^\top\mathbf{E}^{-1}(\mathbf{q}-\mathbf{e}) \quad (3.6)$$

and

$$\det(\boldsymbol{\Sigma})\det(\boldsymbol{\Psi}) = \det(\boldsymbol{\Sigma}+\mathbf{B}\boldsymbol{\Psi}\mathbf{B}^\top)\det(\mathbf{E}). \quad (3.7)$$

Firstly prove Equation (3.6). It is proved using matrix completion of the square

$$\begin{aligned}
&(\mathbf{r}-\mathbf{B}\mathbf{q}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{q}-\mathbf{j})+(\mathbf{q}-\mathbf{p})^\top\boldsymbol{\Psi}^{-1}(\mathbf{q}-\mathbf{p}) = \\
&= \mathbf{r}^\top\boldsymbol{\Sigma}^{-1}\mathbf{r}-\mathbf{r}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{q}-\mathbf{q}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{r}+\mathbf{q}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{q}+\mathbf{j}^\top\boldsymbol{\Sigma}^{-1}\mathbf{j}-\mathbf{r}^\top\boldsymbol{\Sigma}^{-1}\mathbf{j}-\mathbf{j}^\top\boldsymbol{\Sigma}^{-1}\mathbf{r}+ \\
&\quad +\mathbf{q}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{j}+\mathbf{j}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{q}+\mathbf{q}^\top\boldsymbol{\Psi}^{-1}\mathbf{q}-\mathbf{q}^\top\boldsymbol{\Psi}^{-1}\mathbf{p}-\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{q}+\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} = \\
&= (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})+\mathbf{q}^\top(\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}+\boldsymbol{\Psi}^{-1})\mathbf{q}-\mathbf{q}^\top(\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{r}-\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{j}+\boldsymbol{\Psi}^{-1}\mathbf{p})- \\
&\quad -(\mathbf{r}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}-\mathbf{j}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}+\mathbf{p}^\top\boldsymbol{\Sigma}^{-1})\mathbf{q}+\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} = \\
&= (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})+\mathbf{q}^\top\mathbf{E}^{-1}\mathbf{q}- \\
&\quad -\mathbf{q}^\top(\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})-\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{p}+\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{p}+\boldsymbol{\Psi}^{-1}\mathbf{p})- \\
&\quad -((\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}-\mathbf{p}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}+\mathbf{p}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}+\mathbf{p}^\top\boldsymbol{\Sigma}^{-1})\mathbf{q}+\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} = \\
&= (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})+\mathbf{q}^\top\mathbf{E}^{-1}\mathbf{q}-\mathbf{q}^\top[\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})+(\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}+\boldsymbol{\Psi}^{-1})\mathbf{p}]- \\
&\quad -[(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}+\mathbf{p}^\top(\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}+\boldsymbol{\Psi}^{-1})]\mathbf{q}+\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} = \\
&= (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})+\mathbf{q}^\top\mathbf{E}^{-1}\mathbf{q}-\mathbf{q}^\top[\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})+\mathbf{E}^{-1}\mathbf{p}]- \\
&\quad -[(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}+\mathbf{p}^\top\mathbf{E}^{-1}]\mathbf{q}+\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} = \\
&= (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})+\mathbf{q}^\top\mathbf{E}^{-1}\mathbf{q}-\mathbf{q}^\top\mathbf{E}^{-1}\mathbf{E}[\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})+\mathbf{E}^{-1}\mathbf{p}]- \\
&\quad -[(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}+\mathbf{p}^\top\mathbf{E}^{-1}]\mathbf{E}\mathbf{E}^{-1}\mathbf{q}+\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} = \\
&= (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})+\mathbf{q}^\top\mathbf{E}^{-1}\mathbf{q}-\mathbf{q}^\top\mathbf{E}^{-1}[\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})+\mathbf{p}]- \\
&\quad -[(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{E}+\mathbf{p}^\top]\mathbf{E}^{-1}\mathbf{q}+\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} = \\
&= (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})+\mathbf{q}^\top\mathbf{E}^{-1}\mathbf{q}-\mathbf{q}^\top\mathbf{E}^{-1}\mathbf{e}-\mathbf{e}^\top\mathbf{E}^{-1}\mathbf{q}+\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} = \\
&= \mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p}+(\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})+\mathbf{q}^\top\mathbf{E}^{-1}\mathbf{q}- \\
&\quad -\mathbf{q}^\top\mathbf{E}^{-1}\mathbf{e}-\mathbf{e}^\top\mathbf{E}^{-1}\mathbf{q}+\mathbf{e}^\top\mathbf{E}^{-1}\mathbf{e}-\mathbf{e}^\top\mathbf{E}^{-1}\mathbf{e} = \\
&= \mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p}+(\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})-\mathbf{e}^\top\mathbf{E}^{-1}\mathbf{e}+(\mathbf{q}-\mathbf{e})^\top\mathbf{E}^{-1}(\mathbf{q}-\mathbf{e})
\end{aligned}$$

In order to prove (3.6) it remains to show that

$$\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p}+(\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j})-\mathbf{e}^\top\mathbf{E}^{-1}\mathbf{e} = (\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top(\boldsymbol{\Sigma}+\mathbf{B}\boldsymbol{\Psi}\mathbf{B}^\top)^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}). \quad (3.8)$$

Substitute into left hand side of Equation (3.8) and then use properties of matrix operations:

$$
\begin{aligned}
&\mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} + (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j}) - \mathbf{e}^\top\mathbf{E}^{-1}\mathbf{e} = \\
&= \mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} + (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j}) - \\
&\quad - (\mathbf{p}+\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}))^\top\mathbf{E}^{-1}(\mathbf{p}+\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})) = \\
&= \mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} + (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j}) - \mathbf{p}^\top\mathbf{E}^{-1}\mathbf{p} - \\
&\quad - (\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}))^\top\mathbf{E}^{-1}(\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})) - \\
&\quad - \mathbf{p}^\top\mathbf{E}^{-1}\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}) - (\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{E}\mathbf{E}^{-1}\mathbf{p} = \\
&= \mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} + (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j}) - \mathbf{p}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{p} - \mathbf{p}^\top\boldsymbol{\Psi}^{-1}\mathbf{p} - \\
&\quad - (\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}))^\top\mathbf{E}^{-1}(\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})) - \\
&\quad - \mathbf{p}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j}) + \mathbf{p}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{p} - (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{p} + \mathbf{p}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{p} = \\
&= (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j}) - (\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{E}\mathbf{E}^{-1}\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}) - \\
&\quad - \mathbf{p}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{j}) - (\mathbf{r}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{p} + \mathbf{p}^\top\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{p} = \\
&= (\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}) - (\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}) = \\
&= (\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top\left[\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{E}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\right](\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}) \\
&= (\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top\left[\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-1}\mathbf{B}(\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B} + \boldsymbol{\Psi}^{-1})\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\right](\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}) = \\
&= |\text{Use Lemma 3.9}| = (\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j})^\top(\boldsymbol{\Sigma}+\mathbf{B}\boldsymbol{\Psi}\mathbf{B}^\top)^{-1}\cdot(\mathbf{r}-\mathbf{B}\mathbf{p}-\mathbf{j}).
\end{aligned}
$$

By that we have proved Equation (3.6). Now, we prove (3.7). At first, we rearrange the equation. It can be rewritten into form

$$
\begin{aligned}
\det(\boldsymbol{\Sigma})\det(\boldsymbol{\Psi}) &= \det(\boldsymbol{\Sigma}+\mathbf{B}\boldsymbol{\Psi}\mathbf{B}^\top)\det(\mathbf{E}) \\
\det(\boldsymbol{\Sigma})\det(\boldsymbol{\Psi}) &= \det(\boldsymbol{\Sigma}+\mathbf{B}\boldsymbol{\Psi}\mathbf{B}^\top)\det(\boldsymbol{\Psi}^{-1}+\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B})^{-1} \\
\det(\boldsymbol{\Psi})\cdot\det(\boldsymbol{\Psi}^{-1}+\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}) &= \det(\boldsymbol{\Sigma}+\mathbf{B}\boldsymbol{\Psi}\mathbf{B}^\top)\cdot\det(\boldsymbol{\Sigma})^{-1} \\
\det(\mathbf{I}_s+\boldsymbol{\Psi}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}\mathbf{B}) &= \det(\mathbf{I}_t+\mathbf{B}\boldsymbol{\Psi}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}).
\end{aligned}
$$

Let us denote

$$\mathbf{J} := \boldsymbol{\Psi}\mathbf{B}^\top\boldsymbol{\Sigma}.$$

Thus

$$\det(\mathbf{I}_s+\mathbf{J}\mathbf{B}) = \det(\mathbf{I}_t+\mathbf{B}\mathbf{J}),$$

which is clear from Weinstein-Aronszajn Lemma 3.10. By that, we have all done and proved the lemma.

□

3.4 | Kalman Filter in Probabilistic View

Definition 3.12 (Transitive density to time epoch k)**.** We call probability density function $f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ as *transitive density to time epoch* k.

Proposition 3.13. *In state space system, transitive density to time epoch* k *is equal to*

$$f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \mathsf{N}(\boldsymbol{x}_k; \mathbf{F}_k\boldsymbol{x}_{k-1}+\mathbf{G}_k\mathbf{u}_k, \boldsymbol{\Gamma}_k\mathbf{Q}_k\boldsymbol{\Gamma}_k^\top).$$

Proof. First of all, we have to check whether $f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ is normal probability density function. Similarly as in Proposition 3.5. By equation (3.1): $\boldsymbol{X}_{k-1} = \boldsymbol{x}_{k-1}$ is known realization of $\boldsymbol{X}_{k-1}$. $\boldsymbol{V}_k \sim \mathsf{N}(\mathbf{o}, \mathbf{Q}_k)$. $\mathbf{u}_k$ is not a random variable (it is known constant). Then $\boldsymbol{X}_k|\boldsymbol{X}_{k-1} = \boldsymbol{x}_{k-1}$ have normal distribution, because it is linear transformation of normal random variable and constants. Consider properties of expected value and variance matrix:

$$\begin{aligned}
\mathsf{E}[\boldsymbol{X}_k|\boldsymbol{X}_{k-1} = \boldsymbol{x}_{k-1}] &= \mathsf{E}[\mathbf{F}_k\boldsymbol{x}_{k-1}] + \mathsf{E}[\mathbf{G}_k\mathbf{u}_k] + \mathsf{E}[\boldsymbol{\Gamma}_k\boldsymbol{V}_k] = \\
&= \mathbf{F}_k\mathsf{E}[\boldsymbol{x}_{k-1}] + \mathbf{G}_k\mathsf{E}[\mathbf{u}_k] + \boldsymbol{\Gamma}_k\mathsf{E}[\boldsymbol{V}_k] = \\
&= \mathbf{F}_k\boldsymbol{x}_{k-1} + \mathbf{G}_k\mathbf{u}_k + \boldsymbol{\Gamma}_k\mathbf{o}_q = \mathbf{F}_k\boldsymbol{x}_{k-1} + \mathbf{G}_k\mathbf{u}_k \\
\mathsf{var}[\boldsymbol{X}_k|\boldsymbol{X}_{k-1} = \boldsymbol{x}_{k-1}] &= \mathsf{var}[\mathbf{F}_k\boldsymbol{x}_{k-1}] + \mathsf{var}[\mathbf{G}_k\mathbf{u}_k] + \mathsf{var}[\boldsymbol{\Gamma}_k\boldsymbol{V}_k\boldsymbol{\Gamma}_k^\top] = \\
&= \mathbf{F}_k\mathsf{var}[\boldsymbol{x}_{k-1}]\mathbf{F}_k^\top + \mathbf{G}_k\mathsf{var}[\mathbf{u}_k]\mathbf{G}_k^\top + \boldsymbol{\Gamma}_k\mathsf{var}[\boldsymbol{V}_k]\boldsymbol{\Gamma}_k^\top = \\
&= \mathbf{F}_k\mathbf{o}_n\mathbf{F}_k^\top + \mathbf{G}_k\mathbf{o}_n\mathbf{G}_k^\top + \mathbf{Q}_k = \boldsymbol{\Gamma}_k\mathbf{Q}_k\boldsymbol{\Gamma}_k^\top
\end{aligned}$$

Parameters of multivariate normal distribution are equal to expected value and variance matrix of random vector, so the proof of the proposition is done. □

Definition 3.14 (Likelihood of the measurement at time epoch k)**.** Probability density function $f(\boldsymbol{z}_k|\boldsymbol{x}_k)$ we call *likelihood of the measurement at time epoch* k.

Proposition 3.15. *In state space system, likelihood of the measurement at time epoch* k *is equal to*

$$f(\boldsymbol{z}_k|\boldsymbol{x}_k) = \mathsf{N}(\boldsymbol{z}_k; \mathbf{H}_k\boldsymbol{x}_k, \mathbf{R}_k).$$

Proof. We have to check, if $\boldsymbol{Z}_k|\boldsymbol{X}_k = \boldsymbol{x}_k$ has normal distribution. By Equation (3.2) it is linear transformation of constant, i.e. $\boldsymbol{x}_k$ as a realization of $\boldsymbol{X}_k$ and normal random vector $\boldsymbol{W}_k$. So $\boldsymbol{Z}_k|\boldsymbol{X}_k = \boldsymbol{x}_k$ has normal distribution. Consider properties of expected value and covariance matrix:

$$\begin{aligned}
\mathsf{E}[\boldsymbol{Z}_k|\boldsymbol{X}_k = \boldsymbol{x}_k] &= \mathsf{E}[\mathbf{H}_k\boldsymbol{x}_k] + \mathsf{E}[\boldsymbol{W}_k] = \\
&= \mathbf{H}_k\mathsf{E}[\boldsymbol{x}_k] + \mathsf{E}[\boldsymbol{W}_k] = \\
&= \mathbf{H}_k\boldsymbol{x}_k + \mathbf{o}_m = \mathbf{H}_k\boldsymbol{x}_k \\
\mathsf{var}[\boldsymbol{Z}_k|\boldsymbol{X}_k = \boldsymbol{x}_k] &= \mathsf{var}[\mathbf{H}_k\boldsymbol{x}_k] + \mathsf{var}[\boldsymbol{W}_k] = \\
&= \mathbf{H}_k\mathsf{var}[\boldsymbol{x}_k]\mathbf{H}_k^\top + \mathsf{var}[\boldsymbol{W}_k] = \\
&= \mathbf{H}_k\mathbf{o}_m\mathbf{H}_k^\top + \mathbf{R}_k = \mathbf{R}_k
\end{aligned}$$

Parameters of multivariate normal distribution are equal to expected value and variance matrix of random vector, so the proof of the proposition is done. □

We recall useful proposition [32] (Page 531) in Hidden Markov Chains.

Proposition 3.16 (Chapman[4]-Kolmogorov[5] equation)**.** *Let state vectors* $\boldsymbol{X}_0 \dots \boldsymbol{X}_k \dots$ *be Markov sequence and* $n > r > s$ *be an arbitrary non-negative integers. Then*

$$f(\boldsymbol{x}_n|\boldsymbol{x}_s) = \int_{-\infty}^{\infty} f(\boldsymbol{x}_n|\boldsymbol{x}_r) f(\boldsymbol{x}_r|\boldsymbol{x}_s)\, \mathrm{d}\boldsymbol{x}_r.$$

[4]Sydney Chapman (1888 - 1970), British mathematician.
[5]Andrey Nikolaevich Kolmogorov (1903 - 1987), Soviet mathematician.

Remark. We denote set of realized measurements $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ up to k^{th} time epoch as $\mathcal{Z}_k$:

$$\mathcal{Z}_k := \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k\}.$$

In Kalman filter at time epoch k, our purpose is to find density function of $\boldsymbol{x}_k$ conditioned on all measurements up to k: $f(\boldsymbol{x}_k|\mathcal{Z}_k)$. In according to Bayesian estimation terminology we call it a *posterior density function*. Use Bayes formula:

$$f(\boldsymbol{x}_k|\mathcal{Z}_k) = f(\boldsymbol{x}_k|\boldsymbol{z}_k, \mathcal{Z}_{k-1}) = \frac{f(\boldsymbol{z}_k|\boldsymbol{x}_k, \mathcal{Z}_{k-1}) f(\boldsymbol{x}_k|\mathcal{Z}_{k-1})}{f(\boldsymbol{z}_k|\mathcal{Z}_{k-1})} = \frac{f(\boldsymbol{z}_k|\boldsymbol{x}_k) f(\boldsymbol{x}_k|\mathcal{Z}_{k-1})}{f(\boldsymbol{z}_k|\mathcal{Z}_{k-1})} \quad (3.9)$$

Quantities in (3.9) have following meaning:

- $f(\boldsymbol{x}_k|\mathcal{Z}_{k-1})$ - a *prior density function*: We use Chapman-Kolmogorov equation in Proposition 3.16:

  $$f(\boldsymbol{x}_k|\mathcal{Z}_{k-1}) = \int_{-\infty}^{\infty} f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) f(\boldsymbol{x}_{k-1}|\mathcal{Z}_{k-1}) \, \mathrm{d}\boldsymbol{x}_{k-1}. \quad (3.10)$$

  where $f(\boldsymbol{x}_{k-1}|\mathcal{Z}_{k-1})$ is a *posterior density function* at previous state $k-1$, which is recursively known. In case $k = 1$ is $f(\boldsymbol{x}_0|\mathcal{Z}_0) = f(\boldsymbol{x}_0)$ given, because we have no measurement at time epoch 0 (i.e. $\mathcal{Z}_0 = \emptyset$) and $f(\boldsymbol{x}_0)$ is given by Definition 3.4. $f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ is known *transitive density to time epoch k* defined by the state-space system (see Proposition 3.13).

- $f(\boldsymbol{z}_k|\boldsymbol{x}_k)$ - *likelihood of the measurement at time epoch k* defined by the state-space system (see Proposition 3.15).

- $f(\boldsymbol{z}_k|\mathcal{Z}_{k-1})$ - *normalization constant*. Use Chapman-Kolmogorov equation:

  $$f(\boldsymbol{z}_k|\mathcal{Z}_{k-1}) = \int_{-\infty}^{\infty} f(\boldsymbol{z}_k|\boldsymbol{x}_k) f(\boldsymbol{x}_k|\mathcal{Z}_{k-1}) \, \mathrm{d}\boldsymbol{x}_k. \quad (3.11)$$

  It is clear that due to integral it is not function of $\boldsymbol{x}_k$ and it is integral over probability space of $\boldsymbol{x}_k$. Recall that both densities in integral are at time step k known.

Lemma 3.17. *Prior density function at time epoch* 1 $f(\boldsymbol{x}_1|\mathcal{Z}_0)$ *is probability density function of normal distribution such that*

$$f(\boldsymbol{x}_1|\mathcal{Z}_0) = \mathsf{N}(\boldsymbol{x}_1; \mathbf{F}_1\bar{\mathbf{x}}_0 + \mathbf{G}_1\mathbf{u}_1, \mathbf{F}_1\mathbf{P}_0\mathbf{F}_1^\top + \mathbf{\Gamma}_1\mathbf{Q}_1\mathbf{\Gamma}_1^\top).$$

Proof. Recall that by Definition 3.4 $f(\boldsymbol{x}_0|\mathcal{Z}_0) = f(\boldsymbol{x}_0) = \mathsf{N}(\boldsymbol{x}_0; \bar{\mathbf{x}}_0, \mathbf{P}_0)$ is given, respectively known. Using Proposition 3.13 substitute into Equation (3.10) and

$$\begin{aligned} f(\boldsymbol{x}_1|\mathcal{Z}_0) &= \int_{-\infty}^{\infty} f(\boldsymbol{x}_1|\boldsymbol{x}_0) f(\boldsymbol{x}_0|\mathcal{Z}_0) \, \mathrm{d}\boldsymbol{x}_0 = \int_{-\infty}^{\infty} f(\boldsymbol{x}_1|\boldsymbol{x}_0) f(\boldsymbol{x}_0) \, \mathrm{d}\boldsymbol{x}_0 = \\ &= \int_{-\infty}^{\infty} \mathsf{N}(\boldsymbol{x}_1; \mathbf{F}_1\boldsymbol{x}_0 + \mathbf{G}_1\mathbf{u}_1, \mathbf{\Gamma}_1\mathbf{Q}_1\mathbf{\Gamma}_1^\top) \cdot \mathsf{N}(\boldsymbol{x}_0; \bar{\mathbf{x}}_0, \mathbf{P}_0) \, \mathrm{d}\boldsymbol{x}_0 = \end{aligned}$$

Use fundamental Gaussian identity (Lemma 3.11).

$$\begin{aligned} \ldots &= \int_{-\infty}^{\infty} \mathsf{N}(\boldsymbol{x}_1; \mathbf{F}_1\bar{\mathbf{x}}_0 + \mathbf{G}_1\mathbf{u}_1, \mathbf{F}_1\mathbf{P}_0\mathbf{F}_1^\top + \mathbf{\Gamma}_1\mathbf{Q}_1\mathbf{\Gamma}_1^\top) \cdot \mathsf{N}(\boldsymbol{x}_0, \hat{\mathbf{e}}, \hat{\mathbf{E}}) \, \mathrm{d}\boldsymbol{x}_0 = \\ &= \mathsf{N}(\boldsymbol{x}_1; \mathbf{F}_1\bar{\mathbf{x}}_0 + \mathbf{G}_1\mathbf{u}_1, \mathbf{F}_1\mathbf{P}_0\mathbf{F}_1^\top + \mathbf{\Gamma}_1\mathbf{Q}_1\mathbf{\Gamma}_1^\top) \cdot \int_{-\infty}^{\infty} \mathsf{N}(\boldsymbol{x}_0, \hat{\mathbf{e}}, \hat{\mathbf{E}}) \, \mathrm{d}\boldsymbol{x}_0 = \\ &= \mathsf{N}(\boldsymbol{x}_1; \mathbf{F}_1\bar{\mathbf{x}}_0 + \mathbf{G}_1\mathbf{u}_1, \mathbf{F}_1\mathbf{P}_0\mathbf{F}_1^\top + \mathbf{\Gamma}_1\mathbf{Q}_1\mathbf{\Gamma}_1^\top); \end{aligned}$$

where

$$\hat{\mathbf{E}} = (\mathbf{F}_1^\top(\mathbf{\Gamma}_1\mathbf{Q}_1\mathbf{\Gamma}_1^\top)^{-1}\mathbf{F}_1 + \mathbf{P}_0^{-1})^{-1},$$
$$\hat{\mathbf{e}} = \bar{\mathbf{x}}_0 + \hat{\mathbf{E}}\mathbf{F}_1^\top(\mathbf{\Gamma}_1\mathbf{Q}_1\mathbf{\Gamma}_1^\top)^{-1}(\boldsymbol{x}_1 - \mathbf{F}_1\bar{\mathbf{x}}_0).$$

□

Introduce notation of parameters of the prior probability density function $f(\boldsymbol{x}_1|\mathcal{Z}_0)$:

$$\bar{\mathbf{x}}_{1|0} := \mathbf{F}_1\bar{\mathbf{x}}_0 + \mathbf{G}_1\mathbf{u}_1;$$
$$\mathbf{P}_{1|0} := \mathbf{F}_1\mathbf{P}_0\mathbf{F}_1^\top + \mathbf{\Gamma}_1\mathbf{Q}_1\mathbf{\Gamma}_1^\top.$$

Lemma 3.18. *The posterior probability density function at time epoch* 1 $f(\boldsymbol{x}_1|\mathcal{Z}_1)$ *is probability density of normal distribution such that*

$$f(\boldsymbol{x}_1|\mathcal{Z}_1) = \mathsf{N}(\boldsymbol{x}_1; \bar{\mathbf{x}}_{1|1}, \mathbf{P}_{1|1}),$$

where

$$\bar{\mathbf{x}}_{1|1} = \bar{\mathbf{x}}_{1|0} + \mathbf{K}_1(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}),$$
$$\mathbf{P}_{1|1} = (\mathbf{I} - \mathbf{K}_1\mathbf{H}_1)\mathbf{P}_{1|0}$$
$$\mathbf{K}_1 = \mathbf{P}_{1|0}\mathbf{H}_1^\top(\mathbf{R}_1 + \mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top)^{-1}$$

Proof. By equations (3.9) and (3.11):

$$f(\boldsymbol{x}_1|\mathcal{Z}_1) = \frac{f(\boldsymbol{z}_1|\boldsymbol{x}_1)f(\boldsymbol{x}_1|\mathcal{Z}_0)}{\int_{-\infty}^{\infty} f(\boldsymbol{z}_1|\boldsymbol{x}_1)f(\boldsymbol{x}_1|\mathcal{Z}_0)\,\mathrm{d}\boldsymbol{x}_1} \tag{3.12}$$

First compute the nominator of the fraction above using Proposition 3.15 and Lemma 3.17:

$$f(\boldsymbol{z}_1|\boldsymbol{x}_1)f(\boldsymbol{x}_1|\mathcal{Z}_0) = \mathsf{N}(\boldsymbol{z}_1; \mathbf{H}_1\boldsymbol{x}_1, \mathbf{R}_1) \cdot \mathsf{N}(\boldsymbol{x}_1; \bar{\mathbf{x}}_{1|0}, \mathbf{P}_{1|0}).$$

Now, use fundamental Gaussian identity (Lemma 3.11)

$$\mathsf{N}(\boldsymbol{z}_1; \mathbf{H}_1\boldsymbol{x}_1, \mathbf{R}_1) \cdot \mathsf{N}(\boldsymbol{x}_1; \bar{\mathbf{x}}_{1|0}, \mathbf{P}_{1|0}) = \mathsf{N}(\boldsymbol{z}_1; \mathbf{H}_1\bar{\mathbf{x}}_{1|0}, \mathbf{R}_1 + \mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top) \cdot \mathsf{N}(\boldsymbol{x}_1; \tilde{\mathbf{e}}, \tilde{\mathbf{E}}),$$

where

$$\tilde{\mathbf{E}}^{-1} = \mathbf{H}_1^\top\mathbf{R}_1^{-1}\mathbf{H}_1 + \mathbf{P}_{1|0}^{-1},$$
$$\tilde{\mathbf{e}} = \bar{\mathbf{x}}_{1|0} + \tilde{\mathbf{E}}\mathbf{H}_1^\top\mathbf{\Sigma}^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}).$$

Similarly compute the denominator of the Equation 3.12 using Proposition 3.15 and Lemma 3.17 and then fundamental Gaussian identity (Lemma 3.11):

$$\begin{aligned}
&\int_{-\infty}^{\infty} f(\boldsymbol{z}_1|\boldsymbol{x}_1)f(\boldsymbol{x}_1|\mathcal{Z}_0)\,\mathrm{d}\boldsymbol{x}_1 = \\
&= \int_{-\infty}^{\infty} \mathsf{N}(\boldsymbol{z}_1; \mathbf{H}_1\boldsymbol{x}_1, \mathbf{R}_1) \cdot \mathsf{N}(\boldsymbol{x}_1; \bar{\mathbf{x}}_{1|0}, \mathbf{P}_{1|0})\,\mathrm{d}\boldsymbol{x}_1 = \\
&= \int_{-\infty}^{\infty} \mathsf{N}(\boldsymbol{z}_1; \mathbf{H}_1\bar{\mathbf{x}}_{1|0}, \mathbf{R}_1 + \mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top) \cdot \mathsf{N}(\boldsymbol{x}_1; \tilde{\mathbf{e}}, \tilde{\mathbf{E}})\,\mathrm{d}\boldsymbol{x}_1 = \\
&= \mathsf{N}(\boldsymbol{z}_1; \mathbf{H}_1\bar{\mathbf{x}}_{1|0}, \mathbf{R}_1 + \mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top) \cdot \int_{-\infty}^{\infty} \mathsf{N}(\boldsymbol{x}_1; \tilde{\mathbf{e}}, \tilde{\mathbf{E}})\,\mathrm{d}\boldsymbol{x}_1 = \\
&= \mathsf{N}(\boldsymbol{z}_1; \mathbf{H}_1\bar{\mathbf{x}}_{1|0}, \mathbf{R}_1 + \mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top).
\end{aligned}$$

Put partial results into the original fraction:

$$\begin{aligned} f(\boldsymbol{x}_1|\mathcal{Z}_1) &= \frac{f(\boldsymbol{z}_1|\boldsymbol{x}_1)f(\boldsymbol{x}_1|\mathcal{Z}_0)}{\int_{-\infty}^{\infty} f(\boldsymbol{z}_1|\boldsymbol{x}_1)f(\boldsymbol{x}_1|\mathcal{Z}_0)\,\mathrm{d}\boldsymbol{x}_1} \\ &= \frac{\mathsf{N}(\boldsymbol{z}_1; \mathbf{H}_1\bar{\mathbf{x}}_{1|0}, \mathbf{R}_1 + \mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top) \cdot \mathsf{N}(\boldsymbol{x}_1; \tilde{\mathbf{e}}, \tilde{\mathbf{E}})}{\mathsf{N}(\boldsymbol{z}_1; \mathbf{H}_1\bar{\mathbf{x}}_{1|0}, \mathbf{R}_1 + \mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top)} = \\ &= \mathsf{N}(\boldsymbol{x}_1; \tilde{\mathbf{e}}, \tilde{\mathbf{E}}). \end{aligned}$$

Introduce notation for posterior variance matrix such that

$$\mathbf{P}_{1|1} := \tilde{\mathbf{E}} = (\mathbf{P}_{1|0}^{-1} + \mathbf{H}_1^\top\mathbf{R}_1^{-1}\mathbf{H}_1)^{-1}.$$

Use Woodbury matrix identity:

$$\begin{aligned} \mathbf{P}_{1|1} &= (\mathbf{P}_{1|0}^{-1} + \mathbf{H}_1^\top\mathbf{R}_1^{-1}\mathbf{H}_1)^{-1} = \\ &= \mathbf{P}_{1|0} - \mathbf{P}_{1|0}\mathbf{H}_1^\top(\mathbf{R}_1 + \mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top)^{-1}\mathbf{H}_1\mathbf{P}_{1|0} = \\ &= \left(\mathbf{I} - \mathbf{P}_{1|0}\mathbf{H}_1^\top(\mathbf{R}_1 + \mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top)^{-1}\mathbf{H}_1\right)\mathbf{P}_{1|0} = \\ &= (\mathbf{I} - \mathbf{K}_1\mathbf{H}_1)\,\mathbf{P}_{1|0}, \end{aligned}$$

where

$$\mathbf{K}_1 := \mathbf{P}_{1|0}\mathbf{H}_1^\top(\mathbf{R}_1 + \mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top)^{-1}$$

is called Kalman gain at time epoch 1. Let us denote mean of posterior distribution as $\bar{\mathbf{x}}_{1|1}$. Use previously derived $\mathbf{P}_{1|1}$.

$$\begin{aligned} \bar{\mathbf{x}}_{1|1} := \tilde{\mathbf{e}} &= \bar{\mathbf{x}}_{1|0} + \tilde{\mathbf{E}}\mathbf{H}_1^\top\mathbf{R}_1^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) \\ &= \bar{\mathbf{x}}_{1|0} + \mathbf{P}_{1|1}\mathbf{H}_1^\top\mathbf{R}_1^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) \\ &= \bar{\mathbf{x}}_{1|0} + (\mathbf{I} - \mathbf{K}_1\mathbf{H}_1)\,\mathbf{P}_{1|0}\mathbf{H}_1^\top\mathbf{R}_1^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) \\ &= \bar{\mathbf{x}}_{1|0} + \mathbf{P}_{1|0}\mathbf{H}_1^\top\mathbf{R}_1^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) - \mathbf{K}_1\mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top\mathbf{R}_1^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) \\ &= \bar{\mathbf{x}}_{1|0} + \mathbf{P}_{1|0}\mathbf{H}_1^\top(\mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top + \mathbf{R}_1)^{-1}(\mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top + \mathbf{R}_1)\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) - \\ &\quad - \mathbf{K}_1\mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top\mathbf{R}_1^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) \\ &= \bar{\mathbf{x}}_{1|0} + \mathbf{K}_1(\mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top + \mathbf{R}_1)\mathbf{R}_1^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) - \\ &\quad - \mathbf{K}_1\mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top\mathbf{R}_1^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) \\ &= \bar{\mathbf{x}}_{1|0} + (\mathbf{K}_1\mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top\mathbf{R}_1^{-1} + \mathbf{K}_1)(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) - \\ &\quad - \mathbf{K}_1\mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top\mathbf{R}_1^{-1}(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) \\ &= \bar{\mathbf{x}}_{1|0} + (\mathbf{K}_1\mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top\mathbf{R}_1^{-1} + \mathbf{K}_1 - \mathbf{K}_1\mathbf{H}_1\mathbf{P}_{1|0}\mathbf{H}_1^\top\mathbf{R}_1^{-1})(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) \\ &= \bar{\mathbf{x}}_{1|0} + \mathbf{K}_1(\boldsymbol{z}_1 - \mathbf{H}_1\bar{\mathbf{x}}_{1|0}) \end{aligned}$$

□

Remark. By Lemmas 3.18 and 3.17 we have in fact derived Kalman filter at time step 1. To derive Kalman filter at general time step k we miss information if previous probability density function $f(\boldsymbol{x}_{k-1}|\mathcal{Z}_{k-1})$ is normal. But in case $f(\boldsymbol{x}_0)$ is normal, which is by Definition 3.4, then we showed by Lemma 3.18 that $f(\boldsymbol{x}_1|\mathcal{Z}_1)$ is normal and then we fulfilled assumption at time step 2 and then all densities $f(\boldsymbol{x}_k|\mathcal{Z}_{k-1})$ and $f(\boldsymbol{x}_k|\mathcal{Z}_k)$ have normal distribution for all $k > 0$.

This conditional estimate we use to estimate hidden state $\boldsymbol{x}_k$. The idea is that target provides the movement with an unknown position (i.e., unknown realization of $\boldsymbol{X}_k$ at time steps k). We have measurements with additional measurement noise $\boldsymbol{W}_k$ (e.g., TDOA measurements, Chapter 2). We aim to estimate the true target position $\boldsymbol{x}_k$ from the measurements. I.e., find the distribution of the $\boldsymbol{X}_k|\mathcal{Z}_k$.

Kalman filter provides the way to compute parameters of $\boldsymbol{X}_k|\mathcal{Z}_k$. Moreover, if we take estimate of the hidden state $\hat{\boldsymbol{x}}_k = \mathsf{E}[\boldsymbol{X}_k|\mathcal{Z}_k]$ we provide best unbiased estimate of the hidden state $\boldsymbol{x}_k$. Called Solution of the Wiener problem.

Theorem 3.19 (Kalman[6] filter)**.** *Following equations are the solution of the Wiener problem:*
Prediction:

$$\begin{aligned}\bar{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k\bar{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_k\mathbf{u}_k,\\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^\top + \boldsymbol{\Gamma}_k\mathbf{Q}_k\boldsymbol{\Gamma}_k^\top.\end{aligned}$$

Innovation:

$$\begin{aligned}\tilde{\boldsymbol{y}}_k &= \boldsymbol{z}_k - \mathbf{H}_k\bar{\mathbf{x}}_{k|k-1},\\ \mathbf{S}_k &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k.\end{aligned}$$

Optimal Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{S}_k^{-1}.$$

Update:

$$\begin{aligned}\bar{\mathbf{x}}_{k|k} &= \bar{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\boldsymbol{y}}_k,\\ \mathbf{P}_{k|k} &= (\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}.\end{aligned}$$

Proof. Firstly, discuss the derivation of the equations. Equations of the prediction can be given by Lemma 3.17 substituting period 0 and 1 with k and $k-1$. As we discussed before the Kalman filter theorem, we know that posterior density has normal distribution by Lemma 3.18.

Equations of innovation, optimal Kalman gain, and finally update comes from Lemma 3.18, but for general k. Similarly, now we can replace indexes 0 and 1 with k and $k-1$.

It remains to discuss that the Kalman filter is the best linear unbiased estimator (solution of the Wiener problem). Introduce notation of an error of the prediction (prior) estimate of the state

$$\boldsymbol{E}_{k|k-1} := \boldsymbol{X}_k - \boldsymbol{X}_k|\mathcal{Z}_{k-1},$$

and update (posterior) estimate of the

$$\boldsymbol{E}_{k|k} := \boldsymbol{X}_k - \boldsymbol{X}_k|\mathcal{Z}_k.$$

In order to prove that the Kalman filter is unbiased, we have to show that $\mathsf{E}[\boldsymbol{E}_{k|k}] = \mathbf{o}_n$.

Assume that given distribution of $\boldsymbol{X}_0 \sim \mathsf{N}(\bar{\mathbf{x}}_0, \mathbf{P}_0)$ accurately reflect the real distribution (is precisely true). Also assume that all parameters in Definition 3.4 like $\mathbf{F}_k, \mathbf{G}_k\mathbf{u}_k, \mathbf{Q}_k, \mathbf{H}_k, \mathbf{R}_k$ precisely fit the reality. We show $\mathsf{E}[\boldsymbol{E}_{k|k}] = \mathbf{o}_n$ inductively.

[6]Rudolf Emil Kálmán (1930 – 2016), Hungarian-American electrical engineer, mathematician, and inventor.

- Consider $k = 1$:

$$\mathcal{Z}_0 = \emptyset \Rightarrow \boldsymbol{X}_1|\mathcal{Z}_0 = \boldsymbol{X}_1 \Rightarrow \mathsf{E}[\boldsymbol{E}_{1|0}] = \mathbf{o}_n$$

Innovation vector $\tilde{\boldsymbol{y}}_1$ is a realization of the random vector $\tilde{\boldsymbol{Y}}_1$.

$$\mathsf{E}[\tilde{\boldsymbol{Y}}_1] = \mathsf{E}[\boldsymbol{Z}_1] - \mathsf{E}[\mathbf{H}_1\boldsymbol{X}_1|\mathcal{Z}_0] = \mathbf{H}_1\mathsf{E}[\boldsymbol{X}_1] - \mathbf{H}_1\mathsf{E}[\boldsymbol{X}_1|\mathcal{Z}_0] = \mathbf{H}_1\mathsf{E}[\boldsymbol{E}_{1|0}] = \mathbf{o}_m.$$

In fact $\tilde{\boldsymbol{Y}}_1 \sim \mathsf{N}(\mathbf{o}_m, \mathbf{S}_1)$. Then consider posterior update

$$\boldsymbol{X}_1|\mathcal{Z}_1 = \boldsymbol{X}_1|\mathcal{Z}_0 + \mathbf{K}_1\tilde{\boldsymbol{Y}}_1.$$

$$\begin{aligned}\mathsf{E}[\boldsymbol{E}_{1|1}] = \mathsf{E}[\boldsymbol{X}_1] - \mathsf{E}[\boldsymbol{X}_1|\mathcal{Z}_1] = \mathsf{E}[\boldsymbol{X}_1] - \mathsf{E}[\boldsymbol{X}_1|\mathcal{Z}_0] - \mathsf{E}[\mathbf{K}_1\tilde{\boldsymbol{Y}}_1] = \\ = \mathsf{E}[\boldsymbol{E}_{1|0}] - \mathsf{E}[\mathbf{K}_1\tilde{\boldsymbol{Y}}_1] = -\mathbf{K}_1\mathsf{E}[\tilde{\boldsymbol{Y}}_1] = \mathbf{o}_n\end{aligned}$$

- Consider $k = 2$:

$$\begin{aligned}\mathsf{E}[\boldsymbol{X}_2] &= \mathbf{F}_2\mathsf{E}[\boldsymbol{X}_1] + \mathbf{G}_2\mathbf{u}_2 && \text{(by Definition 3.4)}\\ \mathsf{E}[\boldsymbol{X}_2|\mathcal{Z}_1] &= \bar{\mathbf{x}}_{2|1} = \mathbf{F}_2\mathsf{E}[\boldsymbol{X}_1|\mathcal{Z}_1] + \mathbf{G}_2\mathbf{u}_2 && \text{(by this Theorem 3.19)}\end{aligned}$$

$$\begin{aligned}\Rightarrow \mathsf{E}[\boldsymbol{E}_{2|1}] &= \mathsf{E}[\boldsymbol{X}_2] - \mathsf{E}[\boldsymbol{X}_2|\mathcal{Z}_1] = \\ &= \mathbf{F}_2\mathsf{E}[\boldsymbol{X}_1] + \mathbf{G}_2\mathbf{u}_2 - \mathbf{F}_2\mathsf{E}[\boldsymbol{X}_1|\mathcal{Z}_1] - \mathbf{G}_2\mathbf{u}_2 = \\ &= \mathbf{F}_2\left(\mathsf{E}[\boldsymbol{X}_1] - \mathsf{E}[\boldsymbol{X}_1|\mathcal{Z}_1]]\right) = \\ &= \mathbf{F}_2\mathsf{E}[\boldsymbol{E}_{1|1}] \\ &= \mathbf{o}_n\end{aligned}$$

Other quantities can be computed in the same way, like in the time epoch $k = 1$ and then in all other time epochs $k > 1$.

So we have shown that the Kalman filter does the unbiased estimation of the hidden state $\boldsymbol{X}_k$. Now we show that it provides the best-unbiased estimation (i.e., unbiased estimation, which minimizes the variance of the error).

$$\begin{aligned}\mathbf{P}_{k|k} &= \mathsf{var}[\boldsymbol{E}_{k|k}] = \\ &= \mathsf{var}[\boldsymbol{X}_k - (\boldsymbol{X}_k|\mathcal{Z}_k)] = \\ &= \mathsf{var}[\boldsymbol{X}_k - ((\boldsymbol{X}_k|\mathcal{Z}_{k-1}) + \mathbf{K}_k\tilde{\boldsymbol{Y}}_k)] = \\ &= \mathsf{var}[\boldsymbol{X}_k - \{(\boldsymbol{X}_k|\mathcal{Z}_{k-1}) + \mathbf{K}_k[\boldsymbol{Z}_k - \mathbf{H}_k(\boldsymbol{X}_k|\mathcal{Z}_{k-1})]\}] \\ &= \mathsf{var}[\boldsymbol{X}_k - \{(\boldsymbol{X}_k|\mathcal{Z}_{k-1}) + \mathbf{K}_k[\mathbf{H}_k\boldsymbol{X}_k + \boldsymbol{W}_k - \mathbf{H}_k(\boldsymbol{X}_k|\mathcal{Z}_{k-1})]\}] \\ &= \mathsf{var}[\boldsymbol{X}_k - (\boldsymbol{X}_k|\mathcal{Z}_{k-1}) - \mathbf{K}_k\mathbf{H}_k\boldsymbol{X}_k - \mathbf{K}_k\boldsymbol{W}_k + \mathbf{K}_k\mathbf{H}_k(\boldsymbol{X}_k|\mathcal{Z}_{k-1})] \\ &= \mathsf{var}[(\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)[\boldsymbol{X}_k - (\boldsymbol{X}_k|\mathcal{Z}_{k-1})] - \mathbf{K}_k\boldsymbol{W}_k] \\ &= (\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)\mathsf{var}[\boldsymbol{X}_k - (\boldsymbol{X}_k|\mathcal{Z}_{k-1})](\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)^\top + \mathsf{var}[\mathbf{K}_k\boldsymbol{W}_k] \\ &= (\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)^\top + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^\top \\ &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{K}_k^\top - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{K}_k^\top + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^\top \\ &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{K}_k^\top - \mathbf{K}_k(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k)\mathbf{K}_k^\top\end{aligned}$$

$$\Rightarrow \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{K}_k^\top - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top \tag{3.13}$$

We aim to minimize $\mathbf{P}_{k|k-1}$ (i.e. $\mathsf{E}[\boldsymbol{E}_{k|k}^\top \boldsymbol{E}_{k|k}]$). In order to do it, we have to minimize its trace. The trace is minimized when its matrix derivative with respect to gain $\mathbf{K}_k$ is equal to zero (because all other quantities in (3.13) are given from Definition 3.4 and from the previous state). Use gradient matrix rules and the fact that $\mathbf{P}_{k|k-1}$ is the symmetric matrix:

$$\frac{\partial \operatorname{tr}(\mathbf{P}_{k|k-1})}{\partial \mathbf{K}_k} = -2\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + 2\mathbf{K}_k\mathbf{S}_k = 0.$$

Solve the equation for the $\mathbf{K}_k$:

$$\mathbf{K}_k\mathbf{S}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top \Rightarrow \mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{S}_k^{-1} \tag{3.14}$$

Use Equation (3.14) to simplify Equation 3.13. At first, multiply (3.14) by $\mathbf{S}_k\mathbf{K}_k^\top$ from the right side:

$$\mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{K}_k^\top,$$

then substitute it into Equation (3.13):

$$\begin{aligned}\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{K}_k^\top - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top = \\ &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} = \\ &= (\mathbf{I}_n - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}.\end{aligned}$$

We have obtained the update formula for posterior covariance in the Kalman filter, minimizing the variance of the error. This is the reason why $\mathbf{K}_k$ is called optimal Kalman gain. □

Remark. In practice, we often work with a system that is not described by the linear state-space system (Definition 3.4). In this thesis, we will face this problem with transition term and control input: instead of $\mathbf{F}_k\boldsymbol{X}_{k-1} + \mathbf{G}_k\mathbf{u}_k$, we will work with some nonlinear mapping $\mathscr{F}_k(\boldsymbol{X}_{k-1}, \mathbf{u}_k)$. In that case, it is not possible to use the Kalman filter. But there other algorithms that can be used. In the text, we will work with the simplest one concept: *Extended Kalman filter* [19]. This algorithm has changed the prediction part:

$$\begin{aligned}\bar{\mathbf{x}}_{k|k-1} &= \mathscr{F}_k(\bar{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k),\\ \mathbf{P}_{k|k-1} &= \hat{\mathbf{F}}_k\mathbf{P}_{k-1|k-1}\hat{\mathbf{F}}_k^\top + \mathbf{Q}_k,\end{aligned}$$

where

$$\hat{\mathbf{F}}_k = \frac{\partial \mathscr{F}_k(\bar{\mathbf{x}}_{k-1|k-1})}{\partial \bar{\mathbf{x}}_{k-1|k-1}}.$$

This can be obviously considered in a more general concept (e.g., nonlinear measurement model), but this is out of the scope of the thesis. Also, the problem of non-linearity can be treat with other algorithms, e.g. *Unscented Kalman filter* [11] or *Particle filters* [2]. Practical application of particle filters on target tracking can be seen in [3]. None of these algorithms are optimal in the sense of minimizing error variance. We call them suboptimal.

Example 3.20. Consider measured plots (Definition 2.15). In this example case, we omit z coordinate, respectively vertical height and only consider xy-plane. State vector consist from position and velocity in x and y axis: $\boldsymbol{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^\top$. Define quantities which must be given (known) by Definition 3.4:

$$\mathbf{F}_k = \begin{bmatrix}1 & \Delta t_k & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & \Delta t_k\\ 0 & 0 & 0 & 1\end{bmatrix},\ \boldsymbol{\Gamma}_k = \begin{bmatrix}\Delta t_k^2/2\\ \Delta t_k\\ \Delta t_k^2/2\\ \Delta t_k\end{bmatrix}\ \mathbf{Q}_k = \sigma_a^2,\ \mathbf{H}_k = \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\\ 0 & 0\end{bmatrix},$$

where $\sigma_a^2 = 0,1\ \mathrm{m}^2/\mathrm{s}^4$ and $\Delta t_k := t_k - t_{k-1}$ is time difference between actual and last time step. There is no control input ($\mathbf{G}_k\mathbf{u}_k = \mathbf{o}_4$) into the state space. Variance of measurements $\mathbf{R}_k$ is known from plots:

$$\mathbf{R}_k = \begin{bmatrix} 100\,000 & 0 \\ 0 & 100\,000 \end{bmatrix};\ \forall k \in \{-1, \dots 682\}.$$

It remains to define initial estimate. In order to improve this initial estimate (because in reality in contrast with theoretical definition it is not precisely known) we do it from 2 measurements instead of 1. We denote them by indices -1 and 0. Let us denote

$$\boldsymbol{z}_0 = \begin{bmatrix} z_0^x \\ z_0^y \end{bmatrix},\ \mathbf{R}_0 = \begin{bmatrix} r_0^{xx} & r_0^{xy} \\ r_0^{yz} & r_0^{yy} \end{bmatrix},\qquad \boldsymbol{z}_{-1} = \begin{bmatrix} z_{-1}^x \\ z_{-1}^y \end{bmatrix},\ \mathbf{R}_{-1} = \begin{bmatrix} r_{-1}^{xx} & r_{-1}^{xy} \\ r_{-1}^{yz} & r_{-1}^{yy} \end{bmatrix}.$$

Define initial distribution such that

$$\bar{\mathbf{x}}_0 := \begin{bmatrix} z_0^x \\ (z_0^x - z_{-1}^x)/\Delta t_0 \\ z_0^y \\ (z_0^y - z_{-1}^y)/\Delta t_0 \end{bmatrix},$$

$$\mathbf{P}_0 := \begin{bmatrix} r_0^{xx} & r_0^{xx}/\Delta t_0 & 0 & 0 \\ r_0^{xx}/\Delta t_0 & (r_0^{xx} + r_{-1}^{xx})/\Delta t_0^2 + \Delta t_0^2\sigma_a^2/4 & 0 & 0 \\ 0 & 0 & r_0^{yy} & r_0^{yy}/\Delta t_0 \\ 0 & 0 & r_0^{yy}/\Delta t_0 & (r_0^{yy} + r_{-1}^{yy})/\Delta t_0^2 + \Delta t_0^2\sigma_a^2/4 \end{bmatrix}.$$

See the results (for positions x_k, y_k) in Figure 3.2. On Figure 3.2 can be seen that despite

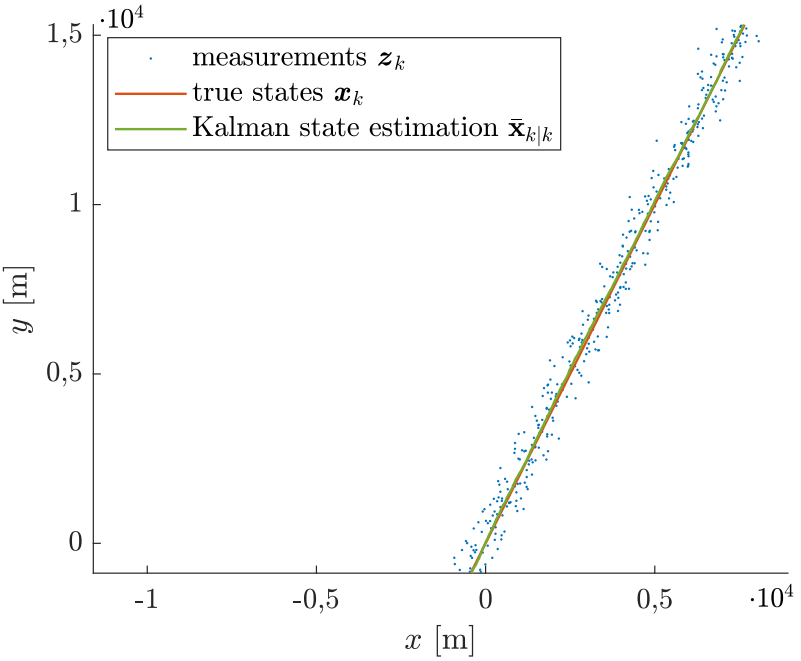


Figure 3.2: Results of estimating of the states x_k, y_k using Kalman filter predicting line movement and low process noise.

noisy measurements we have obtained good results and shown that if we properly set up the parameters of the Kalman filter ($\mathbf{F}_k, \mathbf{G}_k\mathbf{u}_k, \mathbf{Q}_k, \mathbf{H}_k, \bar{\mathbf{x}}_0, \mathbf{P}_0$), then we obtain good results. Especially defined $\mathbf{F}_k$ describe line movement and in Figure 3.2 is shown that

airplane makes a linear movement. In the following examples, we will show what can happen if we do not provide a proper setup (especially transitive matrix $\mathbf{F}_k$ not properly describe a true movement).

Remark. The Kalman filter setup in the last example corresponds to the discrete white noise model listed in Appendix A used on two coordinates xy.

Example 3.21. Consider measured plots (Definition 2.15) of another airplane with a different trajectory (part of a spiral). Also, in this example case, we omit z coordinate, respectively vertical height, and only consider xy-plane. Take the same setup of the parameters of the Kalman filter set up the parameters of Kalman filter $\mathbf{F}_k, \mathbf{G}_k\mathbf{u}_k, \mathbf{Q}_k, \mathbf{H}_k, \bar{\mathbf{x}}_0, \mathbf{P}_0$ like in Example 3.20. See the results in Figure 3.3.

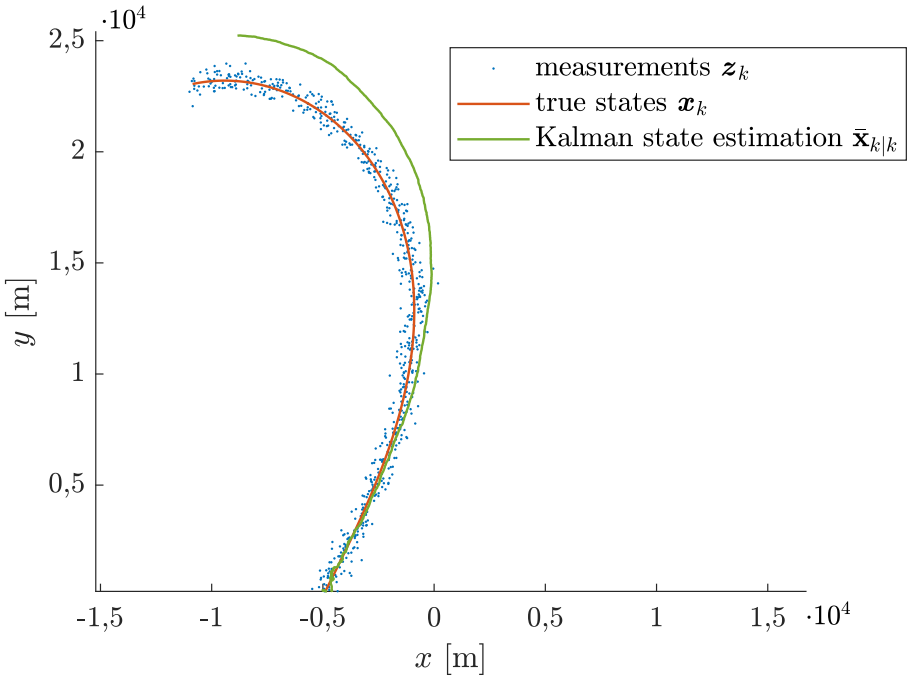


Figure 3.3: Results of estimating of the states x_k, y_k using Kalman filter predicting line movement and low process noise.

From Figure 3.3 can be seen that Kalman set up does not suit the reality, Kalman filter cannot provide unbiased estimation (in proof of the Kalman filter Theorem 3.19 we assumed that Kalman filter parameters are set up properly). $\mathbf{F}_k$ describe line movement, and it is clear that airplane does not provide line movement, and $\mathbf{Q}_k$ is too low. This causes the filter is too confident with prediction. This result is unfeasible. It is necessary to improve that. There are two options for how to do it: change geometry set up - $\mathbf{F}_k, \mathbf{G}_k, \mathbf{u}_k$ or increase process noise - $\mathbf{Q}_k$. In the following example, we do the second option.

Example 3.22. In this example, we try to improve results in previous Example 3.21 by increasing the process noise. Consider set up of parameters of the Kalman filter $(\mathbf{F}_k, \mathbf{G}_k\mathbf{u}_k, \mathbf{Q}_k, \mathbf{H}_k, \bar{\mathbf{x}}_0, \mathbf{P}_0)$ like in Example 3.20 except process noise. The process noise in this example is defined as $\sigma_a = 4$. See results on Figure 3.4.

In Figure 3.4 is shown that by increasing process noise, we have obtained much better results. In following example we provide a first option - change geometry: $\mathbf{F}_k, \mathbf{G}_k, \mathbf{u}_k$.

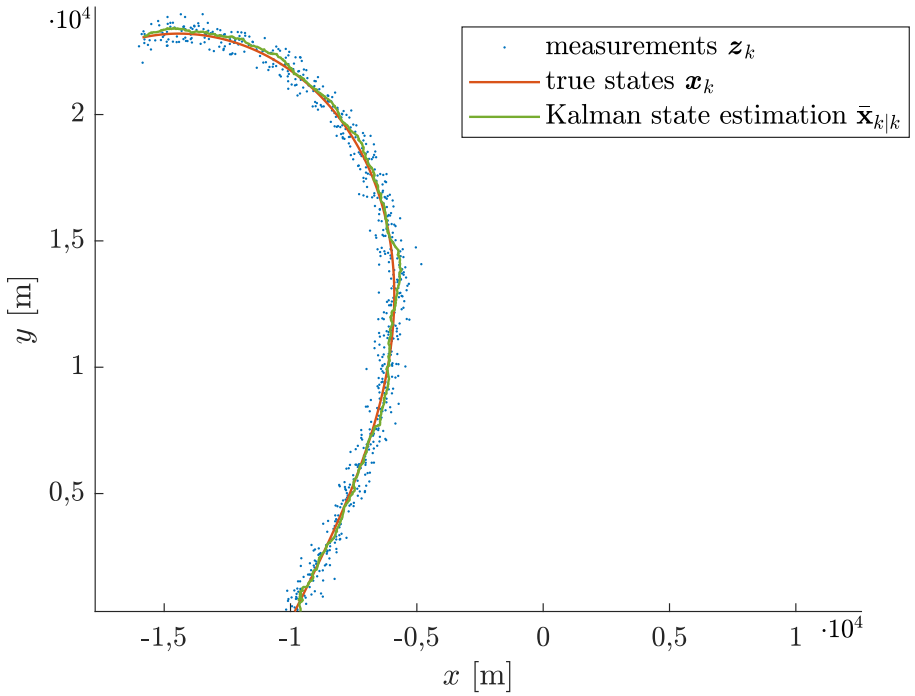


Figure 3.4: Results of estimating of the states x_k, y_k using Kalman filter predicting line movement and higher process noise.

Example 3.23. Consider the same measured data as in Examples 3.21 and Figure 3.4. We add to state vector angular velocity about center of target such that $\boldsymbol{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k, \omega_k]^\top$. Take Kalman filter with following set up parameters (Definition 3.4), but with nonlinear state model.

$$\mathscr{F}_k(\boldsymbol{X}_k, \mathbf{u}_k) = \mathscr{F}_k(\boldsymbol{X}_k) = \begin{bmatrix} X_{k-1} + \frac{\sin(\Omega_{k-1}\Delta t_k)}{\Omega_{k-1}}\dot{X}_{k-1} - \frac{1-\cos(\Omega_{k-1}\Delta t_k)}{\Omega_{k-1}}\dot{Y}_{k-1} \\ \frac{1-\cos(\Omega_{k-1}\Delta t_k)}{\Omega_{k-1}}\dot{X}_{k-1} + Y_{k-1} + \frac{\sin(\Omega_{k-1}\Delta t_k)}{\Omega_{k-1}}\dot{Y}_{k-1} \\ \cos(\Omega_k \Delta t_k)X_{k-1} - \sin(\Omega_k \Delta t_k)Y_{k-1} \\ \sin(\Omega_k \Delta t_k)X_{k-1} + \cos(\Omega_k \Delta t_k)Y_{k-1} \\ \Omega_{k-1} \end{bmatrix}.$$

The complete setup of the Kalman filter can be found in Appendix A: Nearly Constant Turn Rate model. This is evidently more sophisticated geometry. See results on Figure 3.5.

By Example 3.23 it is clear that we are able to improve the results of the Kalman filter by change of process noise (expressing uncertainty of model geometry) or change of the model geometry itself. But there still remains a problem, how to know which process noise or geometry choose. This will be the topic of the following chapters. In the next chapter, we discuss the multiple model approach and then the interactive multiple model algorithm (IMM), which can estimate model geometry with known/given process noise. Then, later we discuss the estimation of the process noise.

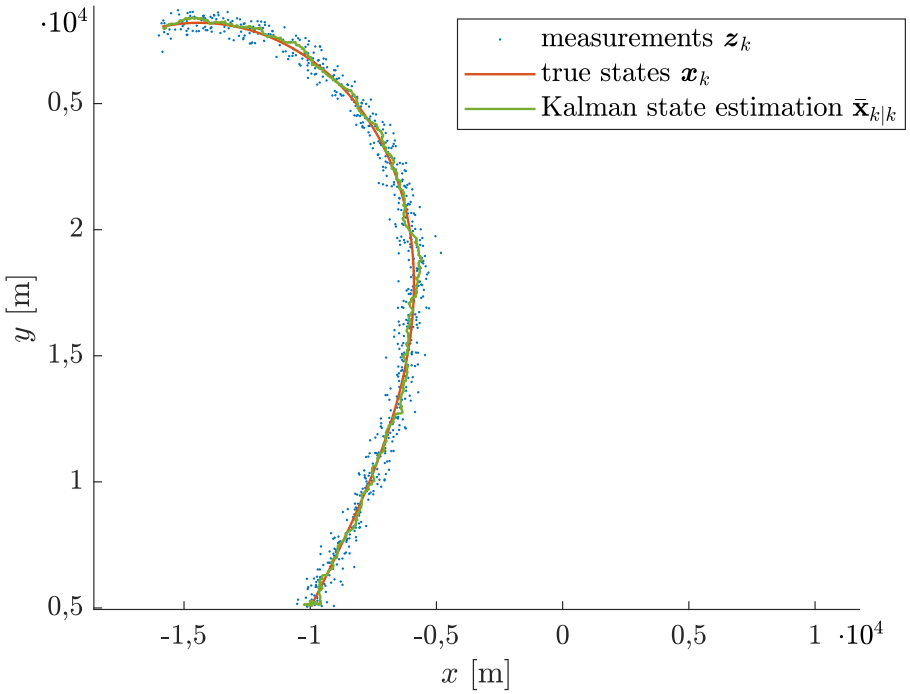


Figure 3.5: Results of estimating of the states x_k, y_k using Kalman filter predicting spiral movement and low process noise.

4 | Multiple Model Approach

In this chapter, we treat the problem with the inadmissible estimation of the state by the improperly given linear stochastic model (Examples 3.20-3.23) by changing the geometry of the model. We exclude Bayesian estimation of the state (Kalman filter) to Bayesian estimation of the state and motion model. We handle with set up of the Kalman filter in mentioned Examples as with random variable. The outcome of this chapter is a derivation of IMM (Interacting Multiple Model algorithm). The most commonly used algorithm for such a purpose.

4.1 | Jump Linear State-Space System

In this section, we describe the jump linear state-space system. It can be view as a generalization of the state space system (Definition 3.4). Before that, we introduce the term of motion model. *Motion model* (or only *model*) represents the set up of the Kalman filter discussed in Chapter 3 in Examples 3.20 – 3.23.

Definition 4.1 (Motion model, model)**.** We call the *motion model* or *model* M_k at time epoch k - in effect during the sampling period ending at k - following random variable defined by ordered eight-tuple:

$$M_k = \langle \mathbf{F}_k, \mathbf{G}_k, \mathbf{u}_k, \mathbf{\Gamma}_k, \mathbf{Q}_k, \mathbf{H}_k; \bar{\mathbf{x}}_0, \mathbf{P}_0 \rangle$$

Quantities in eight-tuple have the same meaning as in Definition 3.4. Examples of such motion models can be found in Appendix A. In principle, they are the set up of the Kalman filter to estimate hidden state.

Proposition 4.2. *Sample space Ω_{M_k} of motion model is uncountable.*

Proof. All elements of defined eight-tuple are member of the finite dimensional vector space over real numbers or set of regular matrices over real numbers. By that is uncountability clear. □

Remark. It is clear that M_k fully define the Kalman filter set up in order to provide Bayesian estimation of the state. But it is not possible to work with infinity (not even uncountable) number of Kalman filter setups, so it is more comfortable to work with some predefined discrete subset of Ω_{M_k}. By that, we make an assumption that realization of M_k belongs to $\mathscr{M} \subset \Omega_{M_k}$, where $\mathscr{M} = \{m^j\}_{j=1}^r$ is predefined discrete subset of Ω_{M_k} and work with M_k as with discrete random variable.

Now, it is opened the question of the coverage of Ω_M by $\mathscr{M}$. We assume in this chapter that the coverage is time-invariant and good enough without any deeper cogitation. In the other way, it is shown by the linear algebra approach in [20] that theoretical optimal selection of $\mathscr{M}_k$ (model set $\mathscr{M}$ at time step k) always leads to a better result than a wider

or smaller selection of $\mathscr{M}_k$. This may happen when the target changes the motion model in time, and we consider time-invariant $\mathscr{M}$. The way to select $\mathscr{M}_k$ is discussed in the series of articles [21], [22], [23], [24], [25]. But these considerations are unfortunately out of the scope of the text.

Definition 4.3 (Jump linear state-space system)**.** Let $\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_k, \ldots \in \mathbb{R}^n$ be sequence of state vectors, $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_k, \ldots \in \mathbb{R}^m$ sequence of measurement vectors. We call *jump linear state-space system* a following system of stochastic difference equations:

$$\boldsymbol{X}_k = \boldsymbol{F}_k(M_k)\boldsymbol{X}_{k-1} + \boldsymbol{G}_k(M_k)\boldsymbol{U}_k(M_k) + \boldsymbol{\Gamma}_k(M_k)\boldsymbol{V}_k(M_k), \tag{4.1}$$

$$\boldsymbol{Z}_k = \boldsymbol{H}_k(M_k)\boldsymbol{X}_k + \boldsymbol{W}_k; \tag{4.2}$$

where $\mathbf{F}_k(M_k) \in \mathcal{M}_n(\mathbb{R})$ is state transition model of M_k, $\mathbf{H}_k(M_k) \in \mathcal{M}_{m,n}(\mathbb{R})$ is observation model of M_k, $\mathbf{G}_k(M_k) \in \mathcal{M}_{n,p}(\mathbb{R})$ is the control-input model and $\mathbf{U}_k(M_k) \in \mathbb{R}^p$ is the control-input of M_k. $\boldsymbol{V}_k(M_k) \sim \mathsf{N}(\mathbf{o}, \boldsymbol{Q}_k(M_k))$ is the process noise of M_k and $\boldsymbol{W}_k \sim \mathsf{N}(\mathbf{o}, \mathbf{R}_k)$ is the measurement noise. Initial state $\boldsymbol{X}_0 \sim \mathsf{N}(\bar{\mathbf{x}}_0, \mathbf{P}_0)$ is given for all M_k. For M_k we define model jump process – Markov chain – with known model transition probabilities

$$\boldsymbol{\Pi} = \{\pi_{ij}\}_{i,j=1}^{r};\ \pi_{ij} = \mathsf{P}\{M_k = m^j, M_{k-1} = m^i\},$$

where $\boldsymbol{\Pi} \in \mathcal{M}_r(\mathbb{R})$ is called model transition probability matrix. The initial probability function $p(M_0)$ is given.

Remark. It is worth emphasizing that in Definition 4.3, we have written matrices $\boldsymbol{F}_k, \ldots, \boldsymbol{H}_k$ in eight-tuple with italics to express that we view these matrices as a random variable in this context. Their realization is given by model.

For sake of clarity, let us provide deeper view into the model jump process. Assume that the set of possible models $\mathscr{M}$ (sample space) is time-invariant; $\mathscr{M} = \{m^j\}_{j=1}^{r}$. For further considerations, denote as $\mathcal{M}_k^\ell$ the ℓ^{th} model history at time epoch k. $\ell \in \{1, \ldots r^k\}$ denote an index of the combination of all possible realizations of models at all time steps up to k. For example, let $r = 2$ and consider time step $k = 2$. Then there are $l = 4$ possible model histories:

$$\begin{aligned}
&\mathcal{M}_2^1 = \{m^1, m^1\},\\
&\mathcal{M}_2^2 = \{m^1, m^2\},\\
&\mathcal{M}_2^3 = \{m^2, m^1\},\\
&\mathcal{M}_2^4 = \{m^2, m^2\}; \qquad m^1, m^2 \in \mathscr{M}.
\end{aligned}$$

In case we are in time step k, consider model m^j which obeys a target and let be this model the last of the sequence $\mathcal{M}_k^\ell$. Introduce notation for index s such that $\mathcal{M}_k^\ell = \{\mathcal{M}_{k-1}^s, m^j\}$. And let be m^i the last model of the sequence $\mathcal{M}_{k-1}^s$. See this in Figure 4.1. This notation will be adhered in further sections. Recall that $p(M_0)$ is given so $\mathsf{P}\{M_0 = m^1\}, \ldots, \mathsf{P}\{M_0 = m^r\}$ are known and also $\boldsymbol{\Pi} = \{\pi_{ij}\}_{i,j=1}^{r}$ is known by Definition 4.3.

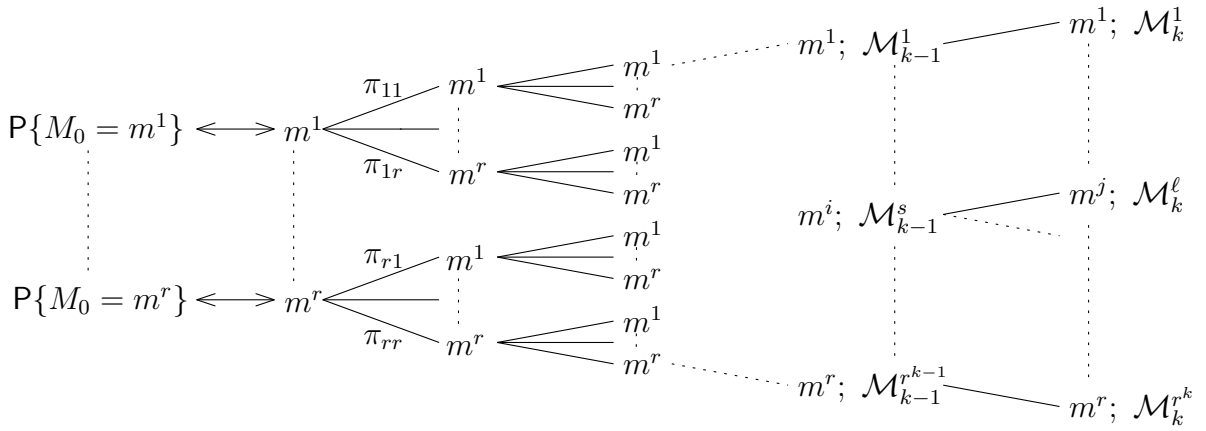


Figure 4.1: Scheme of discussed model jump Markov process

4.2 | Optimal Solution of the Jump Linear State-Space System

We describe and derive the way how to provide an optimal estimation of the hidden state $\boldsymbol{x}_k$. It is optimal under the estimation that the system (Definition 4.3) precisely fits the reality. Unfortunately, this optimal approach is impractical because, as we will show, it has exponential computational complexity.

Proposition 4.4. *The posterior probability density can be written as*

$$f(\boldsymbol{x}_k|\mathcal{Z}_k) = \sum_{l=1}^{r^k} f(\boldsymbol{x}_k|\mathcal{M}_k^\ell, \mathcal{Z}_k) \cdot p(\mathcal{M}_k^\ell|\mathcal{Z}_k).$$

Proof. Directly by *total probability theorem* . □

Remark. Density $f(\boldsymbol{x}_k|\mathcal{M}_k^\ell, \mathcal{Z}_k)$ can be straightforwardly computed by Kalman filter. So the problem is how to compute probability function $p(\mathcal{M}_k^\ell|\mathcal{Z}_k)$. Using the Bayes formula, we see that we are able to derive by applying the Bayes formula recursive relation for this probability function. Let us denote $\nu_k^\ell = p(\mathcal{M}_k^\ell|\mathcal{Z}_k)$.

Proposition 4.5. *ν_k^ℓ can be expressed in following recursive way*

$$\nu_k^\ell = \frac{1}{c} \cdot p(\boldsymbol{z}_k|\mathcal{M}_k^\ell) \cdot \pi_{i,j} \cdot \nu_{k-1}^s,$$

where

$$c = \sum_{l=1}^{r^k} \cdot p(\boldsymbol{z}_k|\mathcal{M}_k^\ell) \cdot \pi_{i,j} \cdot \nu_{k-1}^s$$

is normalizing constant and s is the index of parent sequence such that

$$\mathcal{M}_k^\ell = \{\mathcal{M}_{k-1}^s, m^j\}$$

and i is the index of last model m^i of the parent sequence $\mathcal{M}_{k-1}^s$ (Figure 4.1). $p(\boldsymbol{z}_k|\mathcal{M}_k^\ell)$ is the likelihood of the model sequence $\mathcal{M}_k^\ell$.

Proof. Applying Bayes' formula:

$$\begin{aligned}
\nu_k^\ell &= p(\mathcal{M}_k^\ell|\mathcal{Z}_k) = \\
&= p(\mathcal{M}_k^\ell|\boldsymbol{z}_k, \mathcal{Z}_{k-1}) = |\text{Bayes' formula}| = \\
&= \frac{p(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1}) \cdot p(\mathcal{M}_k^\ell|\mathcal{Z}_{k-1})}{p(\boldsymbol{z}_k|\mathcal{Z}_{k-1})} = |c := p(\boldsymbol{z}_k|\mathcal{Z}_{k-1})| = \\
&= \frac{1}{c} \cdot p(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1}) \cdot p(\mathcal{M}_k^\ell|\mathcal{Z}_{k-1}) = \\
&= \frac{1}{c} \cdot p(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1}) \cdot p(\mathcal{M}_{k-1}^s, m^j|\mathcal{Z}_{k-1}) = |\text{Proposition 3.7}| = \\
&= \frac{1}{c} \cdot p(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1}) \cdot p(m^j|\mathcal{M}_{k-1}^s, \mathcal{Z}_{k-1}) \cdot p(\mathcal{M}_{k-1}^s|\mathcal{Z}_{k-1}) = \\
&\quad = |\mu_{k-1}^s := p(\mathcal{M}_{k-1}^s|\mathcal{Z}_{k-1})| = \\
&= \frac{1}{c} \cdot p(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1}) \cdot p(m^j|\mathcal{M}_{k-1}^s, \mathcal{Z}_{k-1}) \cdot \mu_{k-1}^s = |\text{Markov chain property}| = \\
&= \frac{1}{c} \cdot p(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1}) \cdot p(m^j|\mathcal{M}_{k-1}^s) \cdot \mu_{k-1}^s = |\text{Markov chain property}| = \\
&= \frac{1}{c} \cdot p(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1}) \cdot p(m^j|m^i) \cdot \mu_{k-1}^s \\
&= \frac{1}{c} \cdot p(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1}) \cdot \pi_{i,j} \cdot \mu_{k-1}^s
\end{aligned}$$

Consider normalizing factor $c = p(\boldsymbol{z}_k|\mathcal{Z}_{k-1})$ and use discrete version of Chapman-Kolmogorov equation (Proposition 3.16):

$$c = \sum_{l=1}^{r^k} f(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1}) \cdot p(\mathcal{M}_k^\ell|\mathcal{Z}_{k-1}).$$

By using the same rearrangements as above we obtain

$$c = \sum_{l=1}^{r^k} f(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1}) \cdot \pi_{i,j} \cdot \mu_{k-1}^s.$$

There remains to derive how to compute $f(\boldsymbol{z}_k|\mathcal{M}_k^\ell, \mathcal{Z}_{k-1})$. But for the sake of brevity, we will not do it because we will show by the formula above that this solution is impractical. It is worth mentioning that even we assumed the Markov process, we have not managed to omit $\mathcal{M}_k^\ell$ from the equation, which provides the infeasible exponential computational complexity discussed below. □

In Propositions 4.4 ans 4.5 we derived recursive estimation of the state $\boldsymbol{x}_k$ based on measurements $\mathcal{Z}_k$ with respect to jump linear state-space system (Definition 4.3). It is worth to mention that this solution is optimal (in sense it proved the best unbiased estimation) with respect to given system because we have shown that Kalman filter (used to compute $f(\boldsymbol{x}_k|\mathcal{M}_k^\ell, \mathcal{Z}_k)$) is optimal and other quantities used to compute $p(\mathcal{M}_k^\ell|\mathcal{Z}_k)$ are given by jump linear state-space system.

To judge the computational complexity of the algorithm, introduce the following pair

$$(N_e, N_f),$$

where N_e is *number of estimates* at start of the time step k. For example in optimal solution it is number of values μ_{k-1}^s throw index s. N_f is *number of filters* meaning how

many times we have to use filter (e.g. Kalman filter, Extended Kalman filter) in order to compute posterior update, or equivalently how many quantities we sum in order to compute $f(\boldsymbol{x}_k|\mathcal{Z}_k)$. For example the sum in optimal solution (Proposition 4.4). Thus, computational complexity of the optimal solution of the jump linear state-space system at time step k is

$$(N_e, N_f) = (r^{k-1}, r^k).$$

Unfortunately, it cannot be used in practice because it has exponential computational complexity. Then the optimal solution is impractical (or even uncomputable) because we aim to use that in the long run (e.g., $k = 1, \ldots, 10^4$).

We have to derive approximate algorithms that provide a suboptimal estimation of the state $\boldsymbol{x}_k$. A generalized pseudo-Bayesian estimator of first-order (GPB1) [43] provides the only estimation based on each possible current model (i.e., only single models $m^1, \ldots, m^r$). It has a computational complexity

$$(N_e, N_f) = (1, r).$$

Similarly, a Generalized pseudo-Bayesian estimator of second-order (GPB2) [18] computes estimation under each possible current and previous model (a set of cardinality 2 replaces $\mathcal{M}_k^\ell$). It has a computational complexity.

$$(N_e, N_f) = (r, r^2).$$

In the next section, we describe the interacting multiple model algorithm (IMM) [29]. It provides estimation under each possible current model, but each filter uses a different combination of the previous model – conditioned estimates – called *mixed initial condition*. It has a computational complexity

$$(N_e, N_f) = (r, r)$$

Furthermore, in [36] has been shown that IMM provides similar performance as GPB2, but has the same number of filters as GPB1. This means it has a golden ratio between computational complexity and performance.

4.3 | The Interacting Multiple Model Estimator

In section 4.2 we have described the optimal solution for the jump linear state-space system. However, we stated that it has computational complexity, which is unsuitable for our employment. Then we mentioned the most known suboptimal solution [5] such as Generalized pseudo-Bayesian estimators of first and second orders and the Interacting multiple model estimator. The scope of this section is to introduce and derive the IMM algorithm, which is the most used algorithm for such a purpose in practice in recent years.

Before describing the algorithm itself, it is worth mentioning valuable proposition [4].

Proposition 4.6. *Consider random vector $\boldsymbol{X}$ with Gaussian mixture distribution such that*

$$f(\boldsymbol{x}) = \sum_{i=1}^{s} \alpha_i \cdot \mathsf{N}(\boldsymbol{x}; \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i); \qquad \sum_{i=1}^{s} \alpha_i = 1,$$

then its normal approximation using moment matching method is

$$f(\boldsymbol{x}) \approx \mathsf{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where

$$\boldsymbol{\mu} = \sum_{i=1}^{s} \alpha_i \boldsymbol{\mu}^i, \tag{4.3}$$

$$\boldsymbol{\Sigma} = \sum_{i=1}^{s} \alpha_i \left[\boldsymbol{\Sigma}^i + (\boldsymbol{\mu}^i - \boldsymbol{\mu}) \cdot (\boldsymbol{\mu}^i - \boldsymbol{\mu})^\top \right]. \tag{4.4}$$

Proof. Equation (4.3) is clearly seen by expected value properties. Consider proof of (4.4). Denote event A_i such that

$$A_i := \boldsymbol{X} \sim \mathsf{N}(\boldsymbol{\mu}^i, \Sigma^i).$$

Then

$$\begin{aligned}
\mathsf{E}[(\boldsymbol{x} - \boldsymbol{\mu}) \cdot (\boldsymbol{x} - \boldsymbol{\mu})^\top] &= \sum_{i=1}^{s} \mathsf{E}[(\boldsymbol{x} - \boldsymbol{\mu}) \cdot (\boldsymbol{x} - \boldsymbol{\mu})^\top | A_i] \alpha_i = \\
&= \sum_{i=1}^{s} \mathsf{E}[(\boldsymbol{x} - \boldsymbol{\mu}^i + \boldsymbol{\mu}^i - \boldsymbol{\mu}) \cdot (\boldsymbol{x} - \boldsymbol{\mu}^i + \boldsymbol{\mu}^i - \boldsymbol{\mu})^\top | A_i] \alpha_i = \\
&= \sum_{i=1}^{s} \mathsf{E}[(\boldsymbol{x} - \boldsymbol{\mu}^i) \cdot (\boldsymbol{x} - \boldsymbol{\mu}^i)^\top | A_i] \cdot \alpha_i + \sum_{i=1}^{s} (\boldsymbol{\mu}^i - \boldsymbol{\mu}) \cdot (\boldsymbol{\mu}^i - \boldsymbol{\mu})^\top \cdot \alpha_i = \\
&= \sum_{i=1}^{s} \alpha_i \left[\boldsymbol{\Sigma}^i + (\boldsymbol{\mu}^i - \boldsymbol{\mu}) \cdot (\boldsymbol{\mu}^i - \boldsymbol{\mu})^\top \right].
\end{aligned}$$

□

Proposition 4.6 shows in [6] that approximation is also valid in maximum likelihood sense and is minimizing Kullback[1]–Leibler[2] divergence [16] between normal approximation and Gaussian mixture.

Remark. We are able to consider Kullback-Leibner divergence $D_{\mathrm{KL}}(p\|q)$ as a measure of information loss if we approximate density p by q. So in our case, we measure information loss if we use the normal distribution to approximate the Gaussian mixture.

Now, continue with derivating the Interacting Multiple Model estimator. The key idea of the algorithm is in an expression of the posterior density function. We assume the following approximation.

$$f(\boldsymbol{x}_k | \mathcal{Z}_k) \approx \sum_{j=1}^{r} f(\boldsymbol{x}_k | m_k^j, \mathcal{Z}_k) \cdot p(m_k^j | \mathcal{Z}_k), \tag{4.5}$$

where

$$\begin{aligned}
\mu_k^j &:= p(m_k^j | \mathcal{Z}_k), \\
\mathsf{N}(\boldsymbol{x}_k, \bar{\mathbf{x}}_{k|k}^j, \mathbf{P}_{k|k}^j) &:= f(\boldsymbol{x}_k | m_k^j, \mathcal{Z}_k).
\end{aligned} \tag{4.6}$$

so it can be also written in form

$$f(\boldsymbol{x}_k | \mathcal{Z}_k) \approx \sum_{j=1}^{r} \mathsf{N}(\boldsymbol{x}_k; \bar{\mathbf{x}}_{k|k}^j, \mathbf{P}_{k|k}^j) \cdot \mu_k^j.$$

$\bar{\mathbf{x}}_{k|k}^j$ and $\mathbf{P}_{k|k}^j$ are output of the Kalman filter based on model m^j and prior parameters $\bar{\mathbf{x}}_{k-1|k-1}^j$ and $\mathbf{P}_{k-1|k-1}^j$. We describe how to obtain $\mu_k^j, \bar{\mathbf{x}}_{k|k}^j, \mathbf{P}_{k|k}^j$ in the more concrete way

[1] Solomon Kullback (1907 – 1994), American cryptanalyst and mathematician.
[2] Richard Arthur Leibler (1914 – 2003), American mathematician and cryptanalyst.

using recursion approach. Consider posterior density based on model m^j $f(\boldsymbol{x}_k|m_k^j, \mathcal{Z}_k)$ and use Bayes' formula, then Chapman-Kolmogorov equation (Proposition 3.16):

$$\begin{aligned} f(\boldsymbol{x}_k|m_k^j, \mathcal{Z}_k) &= f(\boldsymbol{x}_k|m_k^j, \boldsymbol{z}_k, \mathcal{Z}_{k-1}) = \\ &= \frac{f(\boldsymbol{z}_k|m_k^j, \boldsymbol{x}_k, \mathcal{Z}_{k-1}) \cdot f(\boldsymbol{x}_k|m_k^j, \mathcal{Z}_{k-1})}{f(\boldsymbol{z}_k|m_k^j, \mathcal{Z}_{k-1})} = \\ &= \frac{f(\boldsymbol{z}_k|m_k^j, \boldsymbol{x}_k) \cdot f(\boldsymbol{x}_k|m_k^j, \mathcal{Z}_{k-1})}{f(\boldsymbol{z}_k|m_k^j, \mathcal{Z}_{k-1})} = \\ &= \frac{f(\boldsymbol{z}_k|m_k^j, \boldsymbol{x}_k)}{f(\boldsymbol{z}_k|m_k^j, \mathcal{Z}_{k-1})} \cdot \int_{-\infty}^{\infty} f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, m_k^j) \cdot f(\boldsymbol{x}_{k-1}|m_k^j, \mathcal{Z}_{k-1}) \, \mathrm{d}\boldsymbol{x}_k. \end{aligned} \tag{4.7}$$

Use total probability theorem on $f(\boldsymbol{x}_{k-1}|m_k^j, \mathcal{Z}_{k-1})$:

$$f(\boldsymbol{x}_{k-1}|m_k^j, \mathcal{Z}_{k-1}) = \sum_{i=1}^{r} f(\boldsymbol{x}_{k-1}|m_{k-1}^i, \mathcal{Z}_{k-1}) \cdot p(m_{k-1}^i|m_k^j, \mathcal{Z}_{k-1})$$

and define

$$\mu_{k-1|k-1}^{ij} := p(m_{k-1}^i|m_k^j, \mathcal{Z}_{k-1}). \tag{4.8}$$

Then

$$f(\boldsymbol{x}_{k-1}|m_k^j, \mathcal{Z}_{k-1}) = \sum_{i=1}^{r} f(\boldsymbol{x}_{k-1}|m_{k-1}^i, \mathcal{Z}_{k-1}) \cdot \mu_{k-1|k-1}^{ij}. \tag{4.9}$$

We know from Equation (4.6) that

$$f(\boldsymbol{x}_{k-1}|m_{k-1}^i, \mathcal{Z}_{k-1}) = \mathsf{N}(\boldsymbol{x}_{k-1}; \bar{\mathbf{x}}_{k-1|k-1}^i, \mathbf{P}_{k-1|k-1}^i).$$

Then (4.9) yields to Gaussian mixture distribution. We consider approximation of it by single Normal distribution $\mathsf{N}(\bar{\mathbf{x}}_{k-1|k-1}^{0j}, \mathbf{P}_{k-1|k-1}^{0j})$ (see Proposition 4.6):

$$f(\boldsymbol{x}_{k-1}|m_k^j, \mathcal{Z}_{k-1}) \approx \mathsf{N}(\boldsymbol{x}_{k-1}; \bar{\mathbf{x}}_{k-1|k-1}^{0j}, \mathbf{P}_{k-1|k-1}^{0j})$$

such that

$$\begin{aligned} \bar{\mathbf{x}}_{k-1|k-1}^{0j} &= \sum_{i=1}^{r} \mu_{k-1|k-1}^{ij} \; \bar{\mathbf{x}}_{k-1|k-1}^{i}, \\ \mathbf{P}_{k-1|k-1}^{0j} &= \sum_{i=1}^{r} \mu_{k-1|k-1}^{ij} [\mathbf{P}_{k-1|k-1}^{i} + (\bar{\mathbf{x}}_{k-1|k-1}^{i} - \bar{\mathbf{x}}_{k-1|k-1}^{0j}) \cdot (\bar{\mathbf{x}}_{k-1|k-1}^{i} - \bar{\mathbf{x}}_{k-1|k-1}^{0j})^{\top}]. \end{aligned} \tag{4.10}$$

To provide this approximation we have to also derive $\mu_{k-1|k-1}^{ij}$:

$$\begin{aligned} \mu_{k-1|k-1}^{ij} &= p(m_{k-1}^i|m_k^j, \mathcal{Z}_{k-1}) = |\text{Bayes' equation}| = \\ &= \frac{p(m_k^j|m_{k-1}^i, \mathcal{Z}_{k-1}) \cdot p(m_{k-1}^i|\mathcal{Z}_{k-1})}{p(m_k^j|\mathcal{Z}_{k-1})} \propto \\ &\propto p(m_k^j|m_{k-1}^i, \mathcal{Z}_{k-1}) \cdot \underbrace{p(m_{k-1}^i|\mathcal{Z}_{k-1})}_{\mu_{k-1}^i} = |\text{Markov chain property}| = \\ &= \underbrace{p(m_k^j|m_{k-1}^i)}_{\pi_{i,j}} \cdot \mu_{k-1}^i = \\ &= \pi_{i,j} \cdot \mu_{k-1}^i \end{aligned}$$

$$\Rightarrow \mu^{ij}_{k-1|k-1} = \frac{\pi_{i,j} \cdot \mu^{i}_{k-1}}{\sum_{h=1}^{r} \pi_{h,j} \cdot \mu^{h}_{k-1}}.$$

It is worth mentioning that we recursively know μ^{i}_{k-1} from the previous time step. The way to obtain it in the current time step we will derive at the end of the IMM estimation when we are computing an estimate of overall posterior distribution $f(\boldsymbol{x}_k|\mathcal{Z}_k)$. Now substitute these computations back into (4.7):

$$f(\boldsymbol{x}_k|m^j_k, \mathcal{Z}_k) \approx \frac{f(\boldsymbol{z}_k|m^j_k, \boldsymbol{x}_k)}{f(\boldsymbol{z}_k|m^j_k, \mathcal{Z}_{k-1})} \cdot \int_{-\infty}^{\infty} f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, m^j_k) \cdot \mathsf{N}(\boldsymbol{x}_{k-1}; \bar{\mathbf{x}}^{0j}_{k-1|k-1}, \mathbf{P}^{0j}_{k-1|k-1})\, \mathrm{d}\boldsymbol{x}_k$$

See the integral in the equation above. The transition density $f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, m^j_k)$ is by Proposition 3.13 equal to

$$f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, m^j_k) = \mathsf{N}(\boldsymbol{x}_k; \mathbf{F}^j_k \boldsymbol{x}_{k-1} + \mathbf{G}^j_k \mathbf{u}^j_k, \mathbf{\Gamma}^j_k \mathbf{Q}^j_k \mathbf{\Gamma}^{j\top}_k),$$

where quantities with upper index j are assigned to model m^j. Use Lemma 3.11 and similar steps as in Lemma 3.17 to derive following equality

$$\begin{aligned}\int_{-\infty}^{\infty} f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, m^j_k) \cdot \mathsf{N}(\boldsymbol{x}_{k-1}; \bar{\mathbf{x}}^{0j}_{k-1|k-1}, \mathbf{P}^{0j}_{k-1|k-1})\, \mathrm{d}\boldsymbol{x}_k = \\ = \mathsf{N}(\boldsymbol{x}_k; \mathbf{F}^j_k \bar{\mathbf{x}}^{0j}_{k-1|k-1} + \mathbf{G}^j_k \mathbf{u}^j_k, \mathbf{F}^j_k \mathbf{P}^{0j}_{k-1|k-1} \mathbf{F}^{j\top}_k + \mathbf{\Gamma}^j_k \mathbf{Q}^j_k \mathbf{\Gamma}^{j\top}_k) \approx \\ \approx f(\boldsymbol{x}_k|m^j_k, \mathcal{Z}_{k-1})\end{aligned}$$

Realize that we have derived approximation of the prior distribution based on j^{th} model. Introduce following notation

$$\begin{aligned}\bar{\mathbf{x}}^j_{k|k-1} &:= \mathbf{F}^j_k \bar{\mathbf{x}}^{0j}_{k-1|k-1} + \mathbf{G}^j_k \mathbf{u}^j_k \\ \mathbf{P}^j_{k|k-1} &:= \mathbf{F}^j_k \mathbf{P}^{0j}_{k-1|k-1} \mathbf{F}^{j\top}_k + \mathbf{\Gamma}^j_k \mathbf{Q}^j_k \mathbf{\Gamma}^{j\top}_k\end{aligned}$$

and recall that

$$f(\boldsymbol{x}_k|m^j_k, \mathcal{Z}_k) \approx \mathsf{N}(\boldsymbol{x}_k; \bar{\mathbf{x}}^j_{k|k-1}, \mathbf{P}^j_{k|k-1}).$$

By Proposition 3.15 is

$$f(\boldsymbol{z}_k|m^j_k, \boldsymbol{x}_k) = \mathsf{N}(\boldsymbol{z}_k; \mathbf{H}^j_k \boldsymbol{x}_k, \mathbf{R}_k),$$

where, again, $\mathbf{H}^j_k$ is observation model assigned to j^{th} model. Now, using similar steps as in Lemma 3.18 we can show that we are able to compute $f(\boldsymbol{x}_k|m^j_k, \mathcal{Z}_k) = \mathsf{N}(\boldsymbol{x}_k; \bar{\mathbf{x}}^j_{k|k}, \mathbf{P}^j_{k|k})$ using Kalman filter (Theorem 3.19) such that:

$$\begin{aligned}\bar{\mathbf{x}}^j_{k|k-1} &= \mathbf{F}^j_k \bar{\mathbf{x}}^{0j}_{k-1|k-1} + \mathbf{G}^j_k \mathbf{u}_k, \\ \mathbf{P}^j_{k|k-1} &= \mathbf{F}_k \mathbf{P}^{0j}_{k-1|k-1} \mathbf{F}^{\top}_k + \mathbf{\Gamma}^j_k \mathbf{Q}^j_k \mathbf{\Gamma}^{j\top}_k.\end{aligned}$$

Innovation:

$$\begin{aligned}\tilde{\boldsymbol{y}}^j_k &= \boldsymbol{z}_k - \mathbf{H}^j_k \bar{\mathbf{x}}^j_{k|k-1}, \\ \mathbf{S}^j_k &= \mathbf{H}^j_k \mathbf{P}^j_{k|k-1} \mathbf{H}^{j\top}_k + \mathbf{R}_k.\end{aligned}$$

Optimal Kalman gain:

$$\mathbf{K}^j_k = \mathbf{P}^j_{k|k-1} \mathbf{H}^{j\top}_k \mathbf{S}^{j-1}_k.$$

Update:

$$\bar{\mathbf{x}}^j_{k|k} = \bar{\mathbf{x}}^j_{k|k-1} + \mathbf{K}^j_k \tilde{\boldsymbol{y}}^j_k,$$
$$\mathbf{P}^j_{k|k} = (\mathbf{I}_n - \mathbf{K}^j_k \mathbf{H}^j_k)\mathbf{P}^j_{k|k-1}.$$

Now, go back into (4.5). The last quantity which we do not have computed is $\mu^j_k = p(m^j_k|\mathcal{Z}_k)$ - the posterior probability function of a model (see (4.5) and (4.6)).

$$\begin{aligned}\mu^j_k &= p(m^j_k|\mathcal{Z}_k) = \\ &= p(m^j_k|\boldsymbol{z}_k, \mathcal{Z}_{k-1}) = |\text{Bayes' formula}| = \\ &= \frac{p(\boldsymbol{z}_k|m^j_k, \mathcal{Z}_{k-1}) \cdot p(m^j_k|\mathcal{Z}_{k-1})}{p(\boldsymbol{z}_k|\mathcal{Z}_{k-1})} \propto \\ &\propto p(\boldsymbol{z}_k|m^j_k, \mathcal{Z}_{k-1}) \cdot p(m^j_k|\mathcal{Z}_{k-1}),\end{aligned}$$

where we denote

$$\Lambda^j_k := p(\boldsymbol{z}_k|m^j_k, \mathcal{Z}_{k-1})$$

the likelihood computed using mixed initial conditions (4.10). The approximate approach using mixed initial conditions is why we have to assume that $\boldsymbol{z}_k$ is not independent of $\mathcal{Z}_{k-1}$. Thus we set

$$\Lambda^j_k = \mathsf{N}(\boldsymbol{z}_k; \mathbf{H}^j_k \bar{\mathbf{x}}^j_{k|k-1}, \mathbf{S}^j_k) = \mathsf{N}(\tilde{\boldsymbol{y}}^j_k; \mathbf{o}_m, \mathbf{S}^j_k).$$

Continue with the derivation of the μ^j_k:

$$\begin{aligned}\mu^j_k &\propto \Lambda^j_k \cdot p(m^j_k|\mathcal{Z}_{k-1}) = |\text{Total probability theorem}| = \\ &= \Lambda^j_k \sum_{i=1}^{r} \underbrace{p(m^j_k|m^i_{k-1})}_{\pi_{i,j}} \cdot \underbrace{p(m^i_{k-1}|\mathcal{Z}_{k-1})}_{\mu^i_{k-1}} \\ &= \Lambda^j_k \sum_{i=1}^{r} \pi_{i,j} \cdot \mu^i_{k-1} \\ &\Rightarrow \mu^j_k = \frac{\Lambda^j_k \sum_{i=1}^{r} \pi_{i,j} \cdot \mu^i_{k-1}}{\sum_{h=1}^{r} \Lambda^h_k \sum_{i=1}^{r} \pi_{i,h} \cdot \mu^i_{k-1}}\end{aligned}$$

By this, we have derived all quantities to compute posterior density $f(\boldsymbol{x}_k|\mathcal{Z}_k)$ (4.5). It is worth mentioning that it is clear that it has Gaussian mixture distribution. However, it is more convenient than using normal approximation using Proposition 4.6. It is only for user output and does not affect algorithm performance.

$$f(\boldsymbol{x}_k|\mathcal{Z}_k) \approx \mathsf{N}(\boldsymbol{x}_k; \hat{\bar{\mathbf{x}}}_{k|k}, \hat{\mathbf{P}}_{k|k})$$

such that

$$\hat{\bar{\mathbf{x}}}_{k|k} := \sum_{j=1}^{r} \mu^j_k \bar{\mathbf{x}}^j_{k|k},$$
$$\hat{\mathbf{P}}_{k|k} := \sum_{j=1}^{r} \mu^j_k \cdot [\mathbf{P}^j_{k|k} + (\bar{\mathbf{x}}^j_{k|k} - \hat{\bar{\mathbf{x}}}_{k|k}) \cdot (\bar{\mathbf{x}}^j_{k|k} - \hat{\bar{\mathbf{x}}}_{k|k})^\top].$$

Summarize derived Algorithm IMM:

Algorithm 4.1: The Interacting multiple model estimator (IMM) at time step k

---

Input from previous time step $k-1$**:** previous statistics: $\{\bar{\mathbf{x}}^i_{k-1|k-1}, \mathbf{P}^i_{k-1|k-1}, \mu^i_{k-1}\}_{i=1}^r$ and model parameters by Definition 4.3

- **Mixing:**
  - calculate mixing probabilities $\mu^{ij}_{k-1|k-1}$ for $i, j = 1, \ldots, r$:

$$\mu^{ij}_{k-1|k-1} = \frac{\pi_{i,j} \cdot \mu^{i}_{k-1}}{\sum_{h=1}^{r} \pi_{h,j} \cdot \mu^{h}_{k-1}}$$

  - calculated mixed estimates $\bar{\mathbf{x}}^{0j}_{k-1|k-1}, \mathbf{P}^{0j}_{k-1|k-1}$ for $j = 1, \ldots, r$:

$$\begin{aligned}
\bar{\mathbf{x}}^{0j}_{k-1|k-1} &= \sum_{i=1}^{r} \mu^{ij}_{k-1|k-1}\ \bar{\mathbf{x}}^{i}_{k-1|k-1}, \\
\mathbf{P}^{0j}_{k-1|k-1} &= \sum_{i=1}^{r} \mu^{ij}_{k-1|k-1} [\mathbf{P}^{i}_{k-1|k-1} + (\bar{\mathbf{x}}^{i}_{k-1|k-1} - \bar{\mathbf{x}}^{0j}_{k-1|k-1}) \cdot \\
&\qquad \cdot (\bar{\mathbf{x}}^{i}_{k-1|k-1} - \bar{\mathbf{x}}^{0j}_{k-1|k-1})^{\top}].
\end{aligned}$$

- **Model Matched Kalman Filtering for $j = 1, \ldots, r$:**
  - Prediction:

$$\begin{aligned}
\bar{\mathbf{x}}^{j}_{k|k-1} &= \mathbf{F}^{j}_{k} \bar{\mathbf{x}}^{0j}_{k-1|k-1} + \mathbf{G}^{j}_{k} \mathbf{u}_{k}, \\
\mathbf{P}^{j}_{k|k-1} &= \mathbf{F}_{k} \mathbf{P}^{0j}_{k-1|k-1} \mathbf{F}^{\top}_{k} + \mathbf{\Gamma}^{j}_{k} \mathbf{Q}^{j}_{k} \mathbf{\Gamma}^{j\top}_{k}.
\end{aligned}$$

  - Innovation:

$$\begin{aligned}
\tilde{\boldsymbol{y}}^{j}_{k} &= \boldsymbol{z}_{k} - \mathbf{H}^{j}_{k} \bar{\mathbf{x}}^{j}_{k|k-1}, \\
\mathbf{S}^{j}_{k} &= \mathbf{H}^{j}_{k} \mathbf{P}^{j}_{k|k-1} \mathbf{H}^{j\top}_{k} + \mathbf{R}_{k}.
\end{aligned}$$

  - Optimal Kalman gain:

$$\mathbf{K}^{j}_{k} = \mathbf{P}^{j}_{k|k-1} \mathbf{H}^{j\top}_{k} \mathbf{S}^{j-1}_{k}.$$

  - Update:

$$\begin{aligned}
\bar{\mathbf{x}}^{j}_{k|k} &= \bar{\mathbf{x}}^{j}_{k|k-1} + \mathbf{K}^{j}_{k} \tilde{\boldsymbol{y}}^{j}_{k}, \\
\mathbf{P}^{j}_{k|k} &= (\mathbf{I}_{n} - \mathbf{K}^{j}_{k} \mathbf{H}^{j}_{k}) \mathbf{P}^{j}_{k|k-1}.
\end{aligned}$$

- **Update Model Probability μ^{j}_{k} for $j = 1, \ldots, r$**

$$\begin{aligned}
\Lambda^{j}_{k} &= \mathsf{N}(\tilde{\boldsymbol{y}}^{j}_{k}; \mathbf{o}_{m}, \mathbf{S}^{j}_{k}), \\
\mu^{j}_{k} &= \frac{\Lambda^{j}_{k} \sum_{i=1}^{r} \pi_{i,j} \cdot \mu^{i}_{k-1}}{\sum_{h=1}^{r} \Lambda^{h}_{k} \sum_{i=1}^{r} \pi_{i,h} \cdot \mu^{i}_{k-1}}
\end{aligned}$$

- **Output Estimate Calculation:**

$$\begin{aligned}
\hat{\bar{\mathbf{x}}}_{k|k} &:= \sum_{j=1}^{r} \mu^{j}_{k} \bar{\mathbf{x}}^{j}_{k|k}, \\
\hat{\mathbf{P}}_{k|k} &:= \sum_{j=1}^{r} \mu^{j}_{k} \cdot [\mathbf{P}^{j}_{k|k} + (\bar{\mathbf{x}}^{j}_{k|k} - \hat{\bar{\mathbf{x}}}_{k|k}) \cdot (\bar{\mathbf{x}}^{j}_{k|k} - \hat{\bar{\mathbf{x}}}_{k|k})^{\top}].
\end{aligned}$$

Output for next time step $k+1$: $\{\bar{\mathbf{x}}^{j}_{k|k}, \mathbf{P}^{j}_{k|k}, \mu^{j}_{k}\}_{j=1}^{r}$
Output for user at this step k: $\hat{\bar{\mathbf{x}}}_{k|k}, \hat{\mathbf{P}}_{k|k}$

For the sake of clarity, see the scheme of derived IMM algorithm on Figure 4.2.

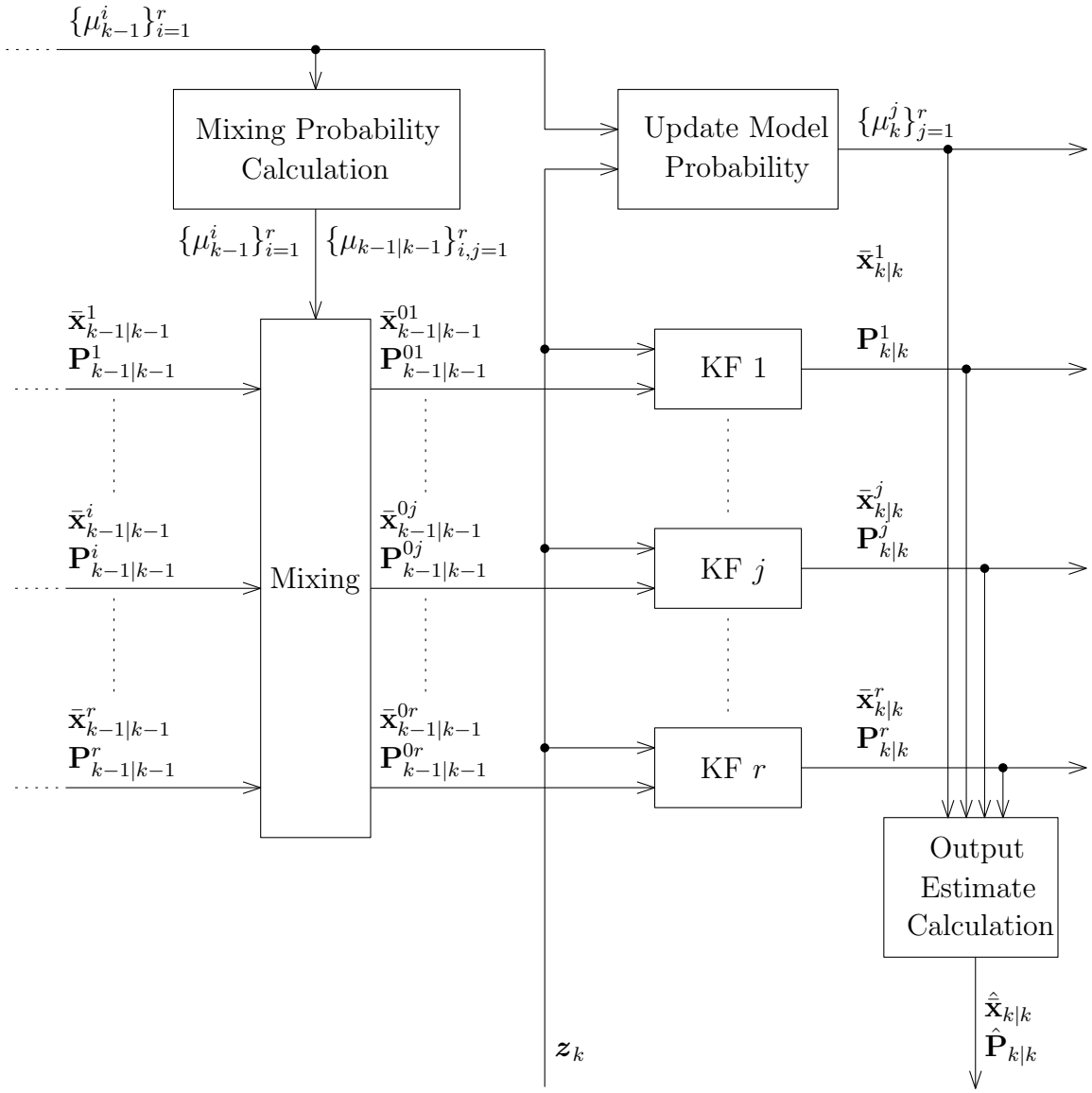


Figure 4.2: Block scheme of the IMM algorithm

Example 4.7. Consider the same data as in Examples 3.21, 3.22, 3.23. Now, we aim to estimate hidden state $\boldsymbol{x}_k$ using derived IMM algorithm.

Consider 4 models listed in Appendix A with parameters:

- DWNA, $\sigma_a^2 = 10^{-10}\ \mathrm{m}^2/\mathrm{s}^4$,
- DWPA, $\sigma_a^2 = 10^{-10}\ \mathrm{m}^2/\mathrm{s}^4$,
- DWPJ, $\sigma_j^2 = 10^{-10}\ \mathrm{m}^2/\mathrm{s}^4$,
- NCTR, $\sigma_a^2 = 0,05\ \mathrm{m}^2/\mathrm{s}^4, \sigma_\alpha^2 = 0,1\ \mathrm{m}^2/\mathrm{s}^4$

and Markov transition matrix

$$\boldsymbol{\Pi} = \begin{bmatrix} 0,55 & 0,15 & 0,15 & 0,15 \\ 0,15 & 0,55 & 0,15 & 0,15 \\ 0,15 & 0,15 & 0,55 & 0,15 \\ 0,15 & 0,15 & 0,15 & 0,55 \end{bmatrix}.$$

Remark. We can see that each chosen model has a different state vector dimension, then we are unable to sum them to provide mixing or output estimate calculation. We must estimate missing vector elements in each state vector. In order to do it, we choose a strategy based on the physical properties of elements. [10] describes more details on this topic.

Remark. In Appendix A we can see that DWNA, DWPA, DWPJ are models only including one coordinate. So we provide estimation for each coordinate (x and y) independently.

See the result on Figure 4.3. Compare with Figures 3.4 and 3.5. Figure 4.4 shows plot-

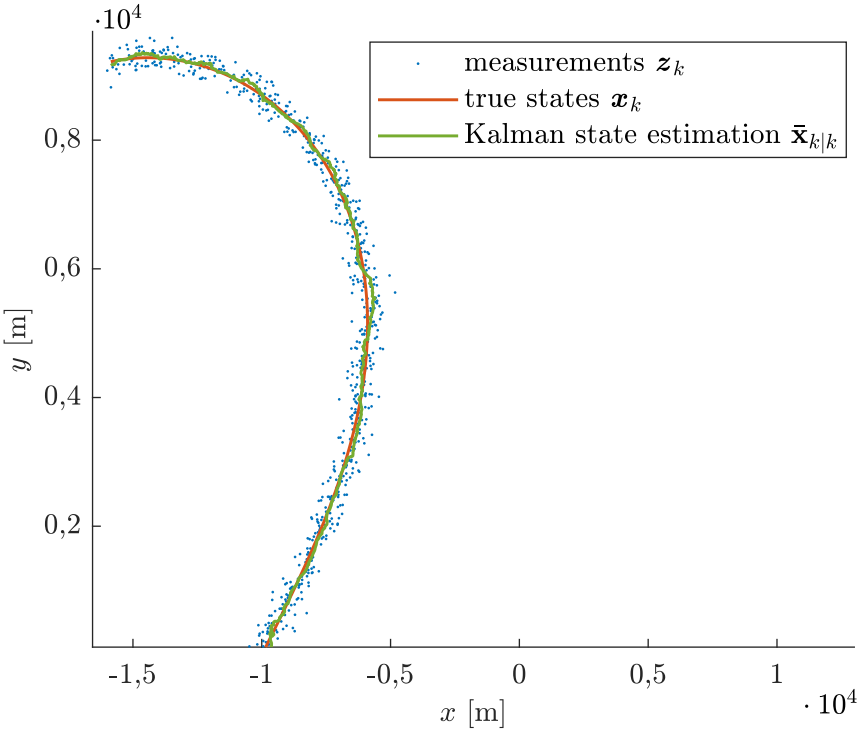


Figure 4.3: Results of estimating states x_k, y_k using IMM estimator.

ted differences between measurements and true (unknown) states compared with plotted differences between estimated states and true states. We can see that with IMM, we can reduce the measurement noise. Recall that time on the x-axis corresponds to real-time such that t_k is the time at time step k. On Figure 4.5 are shown model probabilities

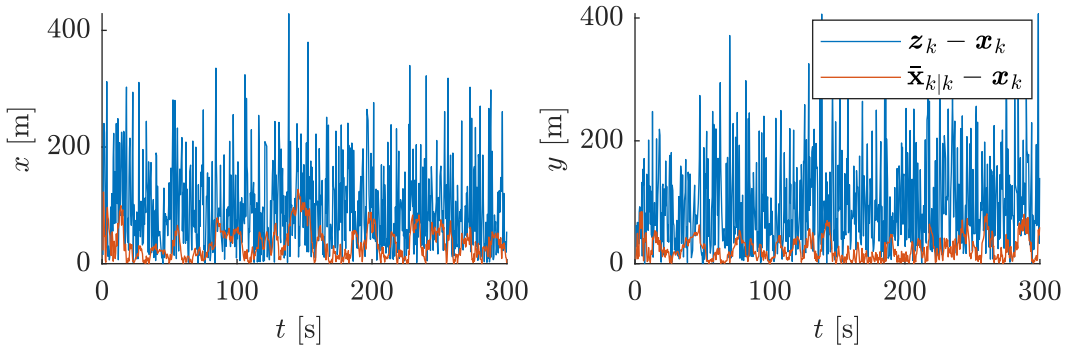


Figure 4.4: Differences between measurements and true (unknown) states compared with plotted differences between estimated states and true states.

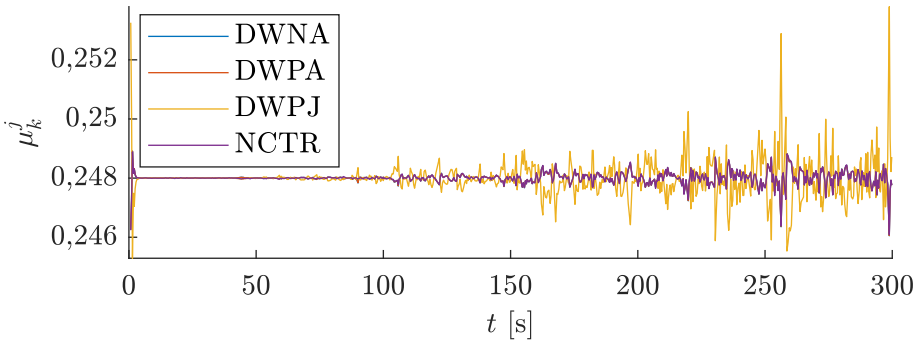


Figure 4.5: Model probabilities $\mu_k^j = p(m_k^j|\mathcal{Z}_k)$ with respect to time.

$\mu_k^j = p(m_k^j|\mathcal{Z}_k)$ with respect to time. Probabilities of model DWPJ hides probabilities of models DWNA and DWPJ on Figure 4.5, because they are almost the same. The fact that they are almost the same comes from the very low process noise of these three models.

This shows the critical problem of IMM algorithm usage. We have to correctly choose the process noise of models, which is sometimes not easy to do, primarily if we use many filters. That is why we continue with our considerations in the next chapter and try to estimate process noise together with the hidden state.

5 | Estimation of the process noise Q

In this chapter, we start by discussing the process noise $\mathbf{Q}$ estimation. In Chapter 2 we described the multiple model approach. In Example 4.7 we seen that if we provide a proper definition of $\mathscr{M}$ (discrete sample space of motion models), we can estimate hidden state $\boldsymbol{x}_k$ with good results (i.e., we solved the problem of unknown geometry). Nevertheless, the reader would notice that we had to set a constant value of process noise $\mathbf{Q}$ for each model. We stated this is a critical step, and if we do not adequately do it, the performance of IMM will degrade. It is worth mentioning that we based the setup of $\mathbf{Q}$ in all models $m^j \in \mathscr{M}$ on the author's previous experience.

So, let us take a deeper (and more mathematician) view into the topic of $\mathbf{Q}$ setup. We do it in a way that we estimate $\mathbf{Q}$ together with a hidden state $\boldsymbol{x}_k$ called *adaptive Kalman filtering*. For the sake of clarity for only one model. At the beginning of the chapter, reformulate Definition 3.4 to define the same linear state-space system but with unknown process noise.

Thus, we temporarily leave the concept of the multiple model approach, but later we see that we can use it to estimate process noise $\mathbf{Q}_k$ in a single motion model.

Remark. In general, we would discuss the same ideas with measurement noise $\mathbf{R}_k$. It does not have to be known. However, in multilateration (see Chapter 2), we assume that we know it precisely and can omit these considerations.

Remark. If we write $\boldsymbol{Q}$ in italics, we want to underline it to view it as a random variable. If we write it only in boldface $\mathbf{Q}$, we want to emphasize to view it as a parameter or realization of $\boldsymbol{Q}$.

Remark. To be precise, we do not estimate process noise but variance matrix $\boldsymbol{Q}$ of the process noise $\boldsymbol{V}_k$. But for the sake of brevity, it is also in literature named as process noise estimation.

Definition 5.1 (Linear state-space model with unknown process noise)**.** Let $\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_k, \ldots \in \mathbb{R}^n$ be sequence of state vectors, $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_k, \ldots \in \mathbb{R}^m$ sequence of measurement vectors, $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_k, \ldots \in \mathbb{R}^p$ be sequence of control-input vectors. We call *linear state-space model with unknown process noise* a following system of stochastic difference equations:

$$\boldsymbol{X}_k = \mathbf{F}_k \boldsymbol{X}_{k-1} + \mathbf{G}_k \mathbf{u}_k + \boldsymbol{V}_k, \tag{5.1}$$

$$\boldsymbol{Z}_k = \mathbf{H}_k \boldsymbol{X}_k + \boldsymbol{W}_k; \tag{5.2}$$

where $\mathbf{F}_k \in \mathcal{M}_n(\mathbb{R})$ is state transition model, $\mathbf{H}_k \in \mathcal{M}_{m,n}(\mathbb{R})$ is observation model and $\mathbf{G}_{k-1} \in \mathcal{M}_{n,p}(\mathbb{R})$ is the control-input model. $\boldsymbol{V}_k \sim \mathsf{N}(\mathbf{o}, \boldsymbol{Q})$ is the process noise and $\boldsymbol{W}_k \sim \mathsf{N}(\mathbf{o}, \mathbf{R}_k)$ is the measurement noise. Initial state $\boldsymbol{X}_0 \sim \mathsf{N}(\bar{\mathbf{x}}_0, \mathbf{P}_0)$ is given.

Remark. We omit time index k in process noise $\boldsymbol{Q}$ since, except the last chapter, we will assume that process noise is constant. We will show in this chapter that we need more

than one measurement to estimate process noise. In the last section, we will consider time-invariant process noise but still assume it is constant over some subset of the set of all measurements.

This chapter aims to estimate the hidden state $\boldsymbol{x}_k$ with unknown process noise $\mathbf{Q}$. This is called *adaptive estimation*. We describe four basic concepts of adaptive estimation/ First two only very bravely since they are not suitable for our purpose. We have tested them, and the results were horrible. Then describe other types of algorithm based on maximum likelihood and Bayesian estimation principles. Compare them in simulation examples to determine which concepts are suitable for our purpose (target tracking using multilateration). In multilateration, we work with known measurement noise $\mathbf{R}_k$ and assumed unknown $\mathbf{Q}$ such that $\operatorname{vol}\mathbf{Q} \ll \operatorname{vol}\mathbf{R}_k$. This property, unfortunately, disqualifies correlation and covariance-matching methods. However, we will show that maximum likelihood and Bayesian concepts work well.

5.1 | Correlation Methods

The main idea of correlation methods comes from the time-series analysis. It is based on the computation of the autocorrelation function of $\boldsymbol{Z}_k$ – output correlation method or innovation $\tilde{\boldsymbol{Y}}_k$ – innovation correlation method. See Kalman filter Theorem 3.19. The second option is stated to be more efficient because innovations $\tilde{\boldsymbol{Y}}_k$ are less correlated than measurements $\boldsymbol{Z}_k$ [30]. Moreover, [12] shows that these two options are equivalent since innovation sequence can be given from observation sequence by Gram[1]-Schmidt[2] orthogonalization process. Thus, we focus on the innovation correlation method. Assume that measurement model $\mathbf{H}_k$ is time-invariant and then omit time index.

We know from Theorem 3.19 that

$$\tilde{\boldsymbol{Y}}_k = \boldsymbol{Z}_k - \mathbf{H}(\boldsymbol{X}_k|\mathcal{Z}_{k-1}) = \mathbf{H}\boldsymbol{X}_k + \boldsymbol{W}_k - \mathbf{H}(\boldsymbol{X}_k|\mathcal{Z}_{k-1}) = \mathbf{H}\boldsymbol{E}_{k|k-1} + \boldsymbol{W}_k \quad \sim \mathsf{N}(\mathbf{o}_m, \mathbf{S}_k); \tag{5.3}$$

Because $\boldsymbol{V}_k, \boldsymbol{W}_k$ are supposed to be white (by Definition 4.3), $\tilde{\boldsymbol{Y}}_k$ is also white, i.e.

$$\mathsf{E}[\tilde{\boldsymbol{Y}}_k \tilde{\boldsymbol{Y}}_{k-i}^\top] = 0; \ i \in \mathbb{N} \setminus \{0\}. \tag{5.4}$$

This is in case Kalman filter is optimal (i.e. process noise in set up fits the true). But in case filter is suboptimal (process noise is untrue) the value can be non-zero. Denote $\boldsymbol{\Phi}_{k,i} = \mathsf{E}[\tilde{\boldsymbol{Y}}_k \tilde{\boldsymbol{Y}}_{k-i}^\top]$ and substitute (5.3) into (5.4) and consider $i > 0$:

$$\begin{aligned}
\boldsymbol{\Phi}_{k,i} &= \mathsf{E}[\tilde{\boldsymbol{Y}}_k \tilde{\boldsymbol{Y}}_{k-i}^\top] = \mathsf{E}[(\mathbf{H}\boldsymbol{E}_{k|k-1} + \boldsymbol{W}_k)(\mathbf{H}\boldsymbol{E}_{k-i|k-i-1} + \boldsymbol{W}_{k-i})^\top] = \\
&= \mathbf{H}\mathsf{E}[\boldsymbol{E}_{k|k-1}\boldsymbol{E}_{k-i|k-i-1}^\top]\mathbf{H}^\top + \mathbf{H}\mathsf{E}[\boldsymbol{E}_{k|k-1}\boldsymbol{W}_{k-i}^\top] + \\
&\quad + \mathsf{E}[\boldsymbol{W}_k \boldsymbol{E}_{k-i|k-i-1}^\top]\mathbf{H}^\top + \mathsf{E}[\boldsymbol{W}_k \boldsymbol{W}_{k-i}^\top] = \\
&= \mathbf{H}\mathsf{E}[\boldsymbol{E}_{k|k-1}\boldsymbol{E}_{k-i|k-i-1}^\top]\mathbf{H}^\top + \mathbf{H}\mathsf{E}[\boldsymbol{E}_{k|k-1}\boldsymbol{W}_{k-i}^\top].
\end{aligned} \tag{5.5}$$

because $\boldsymbol{W}_k$ is assumed to be white then $\mathsf{E}[\boldsymbol{W}_k \boldsymbol{W}_{k-i}^\top] = \mathbf{O}_m$ and prediction/prior error at time $k-i$ is independent from measurement error at time k, then $\mathsf{E}[\boldsymbol{W}_k \boldsymbol{E}_{k-i|k-i-1}^\top]\mathbf{H}^\top = \mathbf{O}_m$. For $i = 0$ it is clear from (5.3) that

$$\boldsymbol{\Phi}_{k,0} = \mathbf{S} = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \mathbf{R}.$$

[1] Jørgen Pedersen Gram (1850 – 1916), Danish actuary, and mathematician.
[2] Erhard Schmidt (1876 – 1959) was a Baltic German mathematician

To evaluate expectation terms in (5.5) it can be derived using Kalman filter equations (Theorem 3.19) following recursive relation for prediction error assuming that filter uses some a priori suboptimal gain $\mathbf{K}^0$:

$$\boldsymbol{E}_{k|k-1} = \mathbf{F}(\mathbf{I}_n - \mathbf{K}^0\mathbf{H})\boldsymbol{E}_{k-1|k-2} - \mathbf{F}\mathbf{K}^0\boldsymbol{W}_{k-1} + \boldsymbol{\Gamma}_{k-1}\boldsymbol{V}_{k-1} \tag{5.6}$$

and carrying (5.6) i steps back:

$$\boldsymbol{E}_{k|k-1} = [\mathbf{F}(\mathbf{I}_n - \mathbf{K}^0\mathbf{H})]^i\boldsymbol{E}_{k-1|k-2} - \sum_{j=1}^{N}[\mathbf{F}(\mathbf{I}_n - \mathbf{K}^0\mathbf{H})]^{-1}\mathbf{F}\mathbf{K}^0\boldsymbol{W}_{k-j} + \\ + \sum_{j=1}^{N}[\mathbf{F}(\mathbf{I}_n - \mathbf{K}^0\mathbf{H})]^{j-1}\boldsymbol{\Gamma}_{k-1}\boldsymbol{V}_{k-j} \tag{5.7}$$

and then can be derived from 5.7 following relation

$$\mathsf{E}[\boldsymbol{E}_{k|k-1}\boldsymbol{E}_{k-i|k-1-i}] = [\mathbf{F}(\mathbf{I}_n - \mathbf{K}^0\mathbf{H})]^i\mathbf{P}_{k|k-1}$$

and similarly

$$\mathsf{E}[\boldsymbol{E}_{k|k-1}\boldsymbol{W}_{k-i}^\top] = -[\mathbf{F}(\mathbf{I}_n - \mathbf{K}^0\mathbf{H})]^{i-1}\mathbf{F}\mathbf{K}^0\mathbf{R}$$

substituting into 5.5 we obtain

$$\boldsymbol{\Phi}_{k,i} = \mathbf{H}[\mathbf{F}(\mathbf{I}_n - \mathbf{K}^0\mathbf{H})]^{i-1}\mathbf{F}[\mathbf{P}_{k|k-1}\mathbf{H}^\top - \mathbf{K}^0\boldsymbol{\Phi}_{k,0}], i = 1, \ldots, N. \tag{5.8}$$

Consider estimation $\hat{\boldsymbol{\Phi}}_{k,i}$ of $\boldsymbol{\Phi}_{k,i}$ such that

$$\hat{\boldsymbol{\Phi}}_{k,i} = \sum_{j=i}^{M}\tilde{\boldsymbol{y}}_j\tilde{\boldsymbol{y}}_{j-i}.$$

To obtain $\boldsymbol{Q}$ (hidden in $\mathbf{P}_{k|k-1}$) and optimal Kalman gains $\mathbf{K}$ (necessary to compute optimal Bayesian estimation of the state), compute N estimates $\hat{\boldsymbol{\Phi}}_{k,i}$, substitute into the system of N equations (5.8) and from that after some matrix rearrangements and use of Moore[3]-Penrose[4] matrix pseudo-inversion [33] we obtain estimates of process noise $\hat{\mathbf{Q}}$. Optimal Kalman gain $\hat{\mathbf{K}}$. It can be shown that estimate of the optimal Kalman gain is asymptotically unbiased and consistent.

Unfortunately, this method does not have good results with multilateration measurements. It is caused, as we stated above, by $\operatorname{vol}\mathbf{Q} \ll \operatorname{vol}\mathbf{R}_k$. The system becomes unstable, and usually, we obtain a result with makes no sense (too large or even a negative estimate of variance).

5.2 | Covariance-Matching Methods

The basic idea is to make residual consistent with their theoretical values.

$$\hat{\mathbf{S}} = \sum_{j=1}^{M}\tilde{\boldsymbol{y}}_j\tilde{\boldsymbol{y}}_j^\top$$

[3]Eliakim Hastings Moore (1862 – 1932), American mathematician.

[4]Sir Roger Penrose(*1931) British mathematical physicist, mathematician, philosopher of science and Nobel Laureate in Physics.

where we choose M to give some statistical smoothing. Use equations in Kalman filter (Theorem 3.19) to obtain the following relations:

$$\begin{aligned}
\mathbf{S} &= \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \mathbf{R} \\
\mathbf{S} &= \mathbf{H}(\mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^\top + \mathbf{\Gamma}\mathbf{Q}\mathbf{\Gamma}^\top)\mathbf{H}^\top + \mathbf{R} \\
\mathbf{S} &= \mathbf{H}(\mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^\top)\mathbf{H}^\top + \mathbf{H}(\mathbf{\Gamma}\mathbf{Q}\mathbf{\Gamma}^\top)\mathbf{H}^\top + \mathbf{R} \\
\\
\Rightarrow \mathbf{H}(\mathbf{\Gamma}\mathbf{Q}\mathbf{\Gamma}^\top)\mathbf{H}^\top &= \mathbf{S} - \mathbf{H}(\mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^\top)\mathbf{H}^\top - \mathbf{R} \\
\mathbf{H}\mathbf{\Gamma}\mathbf{Q}(\mathbf{H}\mathbf{\Gamma})^\top &= \mathbf{S} - \mathbf{H}(\mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^\top)\mathbf{H}^\top - \mathbf{R}
\end{aligned}$$

Assume that $\mathbf{H\Gamma}$ has linearly independent columns, then we can use Moore-Penrose pseudoinverse.

$$\Rightarrow \mathbf{Q} = (\mathbf{H}\mathbf{\Gamma})^+[\mathbf{S} - \mathbf{H}(\mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^\top)\mathbf{H}^\top - \mathbf{R}](\mathbf{H}\mathbf{\Gamma})^{\top+}$$

Now, if we replace $\mathbf{S}$ with an estimate $\hat{\mathbf{S}}$ we obtain an estimate of process noise $\hat{\mathbf{Q}}$ using covariance-matching methods

$$\hat{\mathbf{Q}} = (\mathbf{H}\mathbf{\Gamma})^+ \left[\sum_{j=1}^{M} \tilde{\boldsymbol{y}}_j \tilde{\boldsymbol{y}}_j^\top - \mathbf{H}(\mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^\top)\mathbf{H}^\top - \mathbf{R}\right](\mathbf{H}\mathbf{\Gamma})^{\top+}$$

As correlation methods, this approach is also not suitable for multilateration measurements. A high volume of measurement variance $\mathbf{R}_k$ consumes information about process noise, and we usually obtain a too large or negative estimate of process noise. Which obviously makes no sense.

5.3 | Maximum Likelihood Estimation Methods

Maximum likelihood estimation of the process noise $\mathbf{Q}$ is based on finding extreme of the likelihood function of the parameter (in this case $\mathbf{Q}$). By [30] it can be divided into two basic types of maximum likelihood estimation:

1. *Joint maximum likelihood estimation:* $\mathbf{Q}$ and hidden state $\boldsymbol{x}_k$ are estimated jointly, i.e. joint likelihood $f(\mathcal{Z}_k|\boldsymbol{x}_k, \mathbf{Q})$ is maximized with respect to $\boldsymbol{x}_k, \mathbf{Q}$.
2. *Marginal maximum likelihood estimation:* Marginal likelihood density $f(\mathcal{Z}_k|\mathbf{Q})$ is maximized with respect to $\mathbf{Q}$.

We work with the second one because we use then Kalman filter for hidden state $\boldsymbol{x}_k$ estimation. Consider marginal likelihood function

$$\begin{aligned}
f(\mathcal{Z}_k|\mathbf{Q}) &= f(\boldsymbol{z}_k, \mathcal{Z}_{k-1}|\mathbf{Q}) = \\
&= f(\boldsymbol{z}_k|\mathcal{Z}_{k-1}, \mathbf{Q}_k) \cdot f(\mathcal{Z}_{k-1}|\mathbf{Q}) = \\
&= f(\boldsymbol{z}_k|\mathcal{Z}_{k-1}, \mathbf{Q}) \cdot f(\boldsymbol{z}_{k-1}|\mathcal{Z}_{k-2}, \mathbf{Q}) \cdot \ldots \cdot f(\boldsymbol{z}_1|\mathbf{Q}) = \prod_{i=1}^{k} f(\boldsymbol{z}_i|\mathcal{Z}_{i-1}, \mathbf{Q}).
\end{aligned} \tag{5.9}$$

Lemma 5.2. *For function at time step k*

$$f(\boldsymbol{z}_k|\mathcal{Z}_{k-1}, \mathbf{Q}) = \mathsf{N}(\boldsymbol{z}_k, \mathbf{H}_k\bar{\mathbf{x}}_{k|k-1}(\mathbf{Q}), \mathbf{S}_k(\mathbf{Q})),$$

where $\bar{\mathbf{x}}_{k|k-1}(\mathbf{Q})$ and $\mathbf{S}_k(\mathbf{Q})$ means that their value depends on $\mathbf{Q}$.

Proof. Use Chapman-Kolmogorov equation (Proposition 3.16):

$$f(\boldsymbol{z}_k|\mathcal{Z}_{k-1},\mathbf{Q}) = \int_{-\infty}^{\infty} f(\boldsymbol{z}_k|\boldsymbol{x}_k,\mathbf{Q}) f(\boldsymbol{x}_k|\mathcal{Z}_{k-1},\mathbf{Q})\,\mathrm{d}\boldsymbol{x}_k \tag{5.10}$$

The first term in the integral is independent of $\mathbf{Q}$ (because the single measurement is dependent on the state vector $\boldsymbol{x}_k$ and not the way target comes here and motion model it obeys): $f(\boldsymbol{z}_k|\boldsymbol{x}_k,\mathbf{Q}) = f(\boldsymbol{z}_k|\boldsymbol{x}_k)$. By Proposition 3.15

$$f(\boldsymbol{z}_k|\boldsymbol{x}_k) = \mathsf{N}(\boldsymbol{z}_k, \mathbf{H}_k\boldsymbol{x}_k, \mathbf{R}_k).$$

By Theorem 3.19

$$f(\boldsymbol{x}_k|\mathcal{Z}_{k-1},\mathbf{Q}) = \mathsf{N}(\boldsymbol{x}_k; \bar{\mathbf{x}}_{k|k-1}(\mathbf{Q}), \mathbf{P}_{k|k-1}(\mathbf{Q})).$$

Then rewrite (5.10)

$$f(\boldsymbol{z}_k|\boldsymbol{x}_k,\mathbf{Q}) = \int_{-\infty}^{\infty} \mathsf{N}(\boldsymbol{z}_k, \mathbf{H}_k\boldsymbol{x}_k, \mathbf{R}_k)\mathsf{N}(\boldsymbol{x}_k; \bar{\mathbf{x}}_{k|k-1}(\mathbf{Q}), \mathbf{P}_{k|k-1}(\mathbf{Q}))\,\mathrm{d}\boldsymbol{x}_k$$

and use fundamental Gaussian identity (Lemma 3.11)

$$\begin{aligned} f(\boldsymbol{z}_k|\boldsymbol{x}_k,\mathbf{Q}) &= \int_{-\infty}^{\infty} \mathsf{N}(\boldsymbol{z}_k; \mathbf{H}_k\bar{\mathbf{x}}_{k|k-1}(\mathbf{Q}), \mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k|k-1}(\mathbf{Q})\mathbf{H}_k^\top) \cdot \mathsf{N}(\boldsymbol{x}_k; \tilde{\mathbf{e}}, \tilde{\mathbf{E}})\,\mathrm{d}\boldsymbol{x}_k = \\ &= \mathsf{N}(\boldsymbol{z}_k; \mathbf{H}_k\bar{\mathbf{x}}_{k|k-1}(\mathbf{Q}), \mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k|k-1}(\mathbf{Q})\mathbf{H}_k^\top) \cdot \int_{-\infty}^{\infty} \mathsf{N}(\boldsymbol{x}_k; \tilde{\mathbf{e}}, \tilde{\mathbf{E}})\,\mathrm{d}\boldsymbol{x}_k = \\ &= \mathsf{N}(\boldsymbol{z}_k; \mathbf{H}_k\bar{\mathbf{x}}_{k|k-1}(\mathbf{Q}), \mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k|k-1}(\mathbf{Q})\mathbf{H}_k^\top). \end{aligned}$$

Using notation for innovation variance introduced in Kalman filter (Theorem 3.19):

$$\mathbf{S}_k(\mathbf{Q}) = \mathbf{R}_k + \mathbf{H}_k\mathbf{P}_{k|k-1}(\mathbf{Q})\mathbf{H}_k^\top.$$

□

Substitute result in Lemma 5.2 into (5.9) we obtain the marginal likelihood function

$$f(\mathcal{Z}_k|\mathbf{Q}) = \prod_{i=1}^{k} \mathsf{N}(\boldsymbol{z}_i; \mathbf{H}_i\bar{\mathbf{x}}_{i|i-1}(\mathbf{Q}), \mathbf{S}_i(\mathbf{Q})). \tag{5.11}$$

Then we find $\mathbf{Q}$ which is maximizing (5.11).

Remark. It is worth to mention that for practical reasons of finding maximizer $\mathbf{Q}_k$ and avoiding decimal overflow it can be maximized logarithm of likelihood (because logarithm is monotonous function)

$$\ln f(\mathcal{Z}_k|\mathbf{Q}) = \sum_{i=1}^{k} \ln \mathsf{N}(\boldsymbol{z}_i; \mathbf{H}_i\bar{\mathbf{x}}_{i|i-1}(\mathbf{Q}), \mathbf{S}_i(\mathbf{Q})).$$

Before we do some examples, we should comment on how to compute described maximum likelihood estimation. Respectively, propose the scheme how we suggest computing it. The way we propose in this text is to discretize the sample space of $\mathbf{Q}$. Thus, we assume $\mathbf{Q} \in \mathscr{Q} = \{\mathbf{Q}^j\}_{j=1}^{\#pn}$. $\#pn$ is a large number (e.g., 10^4) to underline the property of the continuous random variable. The lower bound should come from a positive semidefinitness of variance matrix, and the upper bound should be defined by physical

meaning (some very high values of process noise are not physically feasible). See this summarized in Algorithm 5.1 at general time step k:

Algorithm 5.1: ML estimation of the process noise $\mathbf{Q}$

Parameter: discretization of $\mathbf{Q}$'s sample space $\mathscr{Q} = \{\mathbf{Q}^j\}_{j=1}^{\#pn}$
Input from previous time step $k-1$:
prior density:

- $f(\boldsymbol{x}_{k-1}|\mathbf{Q}, \mathcal{Z}_{k-1}) \to \{\bar{\mathbf{x}}_{k-1|k-1}(\mathbf{Q}^j), \mathbf{P}_{k-1|k-1}(\mathbf{Q}^j)\}_{i=1}^{\#pn}$
- $f(\mathcal{Z}_{k-1}|\mathbf{Q}) \to \mathsf{P}\{\mathcal{Z}_{k-1}|\mathbf{Q}^j\};\ j = 1, \ldots, \#pn$

Computation:

- run $\#pn$ Kalman filters for $\mathbf{Q}^j$, $j = 1, \ldots, \#pn$ with output $\bar{\mathbf{x}}_{k|k}(\mathbf{Q}^j), \mathbf{P}_{k|k}(\mathbf{Q}^j)$
- compute approximation of $f(\mathcal{Z}_k|\mathbf{Q})$:

$$\begin{aligned}\mathsf{P}\{\mathcal{Z}_k|\mathbf{Q}^j\} &= \mathsf{P}\{\boldsymbol{z}_k|\mathbf{Q}^j, \mathcal{Z}_k\} \cdot \mathsf{P}\{\mathcal{Z}_{k-1}|\mathbf{Q}^j\} = \\ &= \mathsf{N}(\boldsymbol{z}_k; \mathbf{H}_k\bar{\mathbf{x}}_{k|k}(\mathbf{Q}^j), \mathbf{S}_k(\mathbf{Q}^j)) \cdot \mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_{k-1}\};\ j = 1, \ldots, \#pn\end{aligned}$$

- maximum likelihood estimation:

$$\hat{\mathbf{Q}} = \max_{\mathbf{Q}^j} \mathsf{P}\{\mathcal{Z}_{k-1}|\mathbf{Q}^j\}$$

Output for next time step $k+1$:

- $\bar{\mathbf{x}}_{k|k}(\mathbf{Q}^j), \mathbf{P}_{k|k}(\mathbf{Q}^j);\ j = 1, \ldots, \#pn$
- $\mathsf{P}\{\mathcal{Z}_k|\mathbf{Q}^j\};\ j = 1, \ldots, \#pn$

Output for current state and process noise estimation at time step k:

- $\hat{\mathbf{Q}}, \bar{\mathbf{x}}_{k|k}(\hat{\mathbf{Q}}), \mathbf{P}_{k|k}(\hat{\mathbf{Q}})$

Figure 5.1 shows a proposed maximum likelihood process noise estimation (Algorithm 5.3.1). It is based on the discretization, computing Bayesian estimate of hidden state $\boldsymbol{x}_k$ with Kalman filter at each time step k and then compute a maximum-likelihood estimation of the process noise at time step k. At the end of the time step as output, compute Kalman's estimation of the hidden state with estimated process noise.

For sake of clarity, consider following example for the test application of the proposed scheme.

Example 5.3 (Test example)**.** Consider example to test proposed methods in this chapter. Data (measurements) are generated in following recursive way

$$\begin{aligned}\begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} &= \begin{bmatrix} 1000 \\ 50 \end{bmatrix} \\ \begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix} \cdot V;\ V \sim \mathsf{N}(0, \sigma_a^2), \\ z_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + W; W \sim \mathsf{N}(0, \sigma_w^2); \quad k = 0, \ldots, 1000.\end{aligned}$$

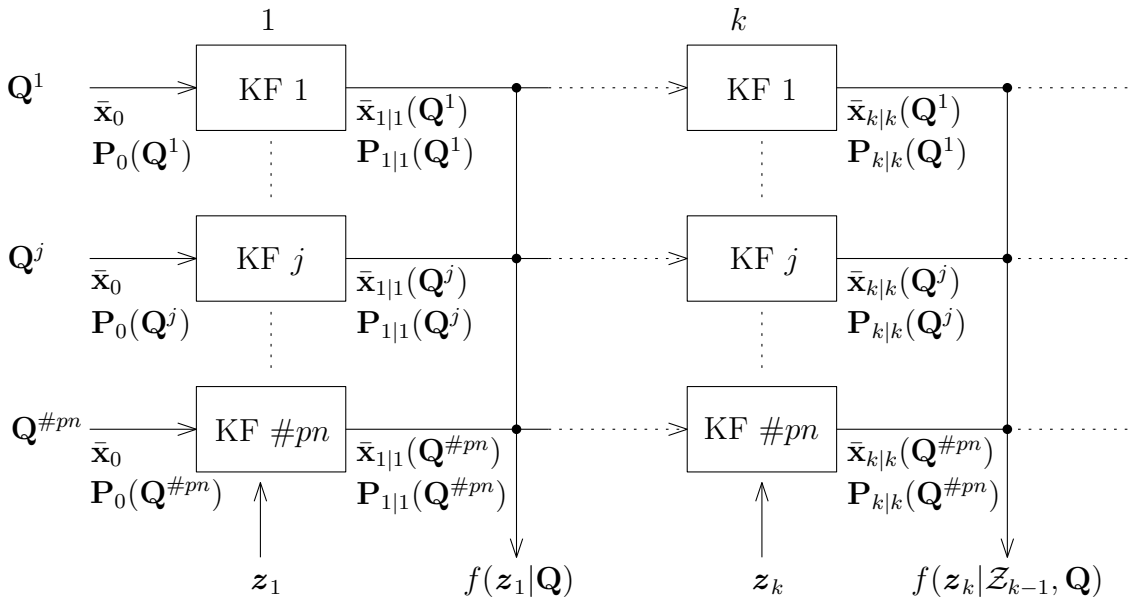


Figure 5.1: Scheme of the proposed way to compute maximum likelihood estimation of $\mathbf{Q}$.

Time step $\Delta t = 1$ s. Where it is worth to mention that measurements noise $\sigma_w^2 = 10^4$ is an order of magnitude larger than process noise $\sigma_a^2 = 10$. The case which we always obtain in multilateration and causes problems to correlation and covariance-matching methods.

The motivation of generating data in such a way is to have states x_k satisfying given process noise σ_a^2, geometry and measurements satisfying measurement noise σ_w^2. Then, it will be possible to assess the performance of adaptive algorithms to estimate process noise, which is unknown for them but known for evaluation.

Firstly, see Figures 5.2 and 5.3 of such a generated trajectory by the proposed way. Resulting estimated states $\bar{\mathbf{x}}_{k|k}(\hat{\sigma_a^2})(\equiv$ Bayesian estimation of the state $\boldsymbol{x}_k$ by Kalman filter using process noise $\sigma_a^2)$ in this figures are given such that $\hat{\sigma}_a^2$ maximizes $f(\mathcal{Z}_k|\sigma_a^2)$ - the key principle of the maximum likelihood estimation. It can be seen the detail of

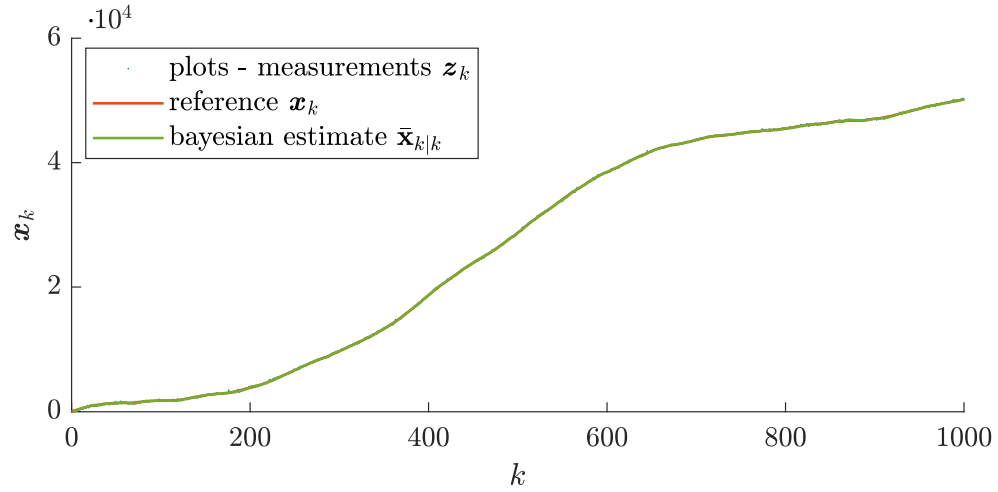


Figure 5.2: Picture of generated trajectory satisfying given value of the process noise $\mathbf{Q}$ with results of filtration.

generated trajectory on Figure 5.3.

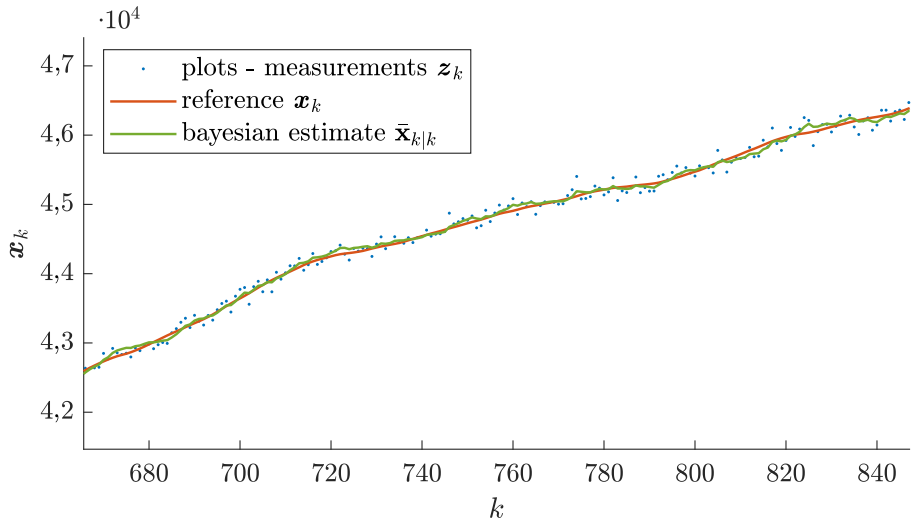


Figure 5.3: Detail of generated trajectory with results of filtration.

Since now, let us discuss the performance of σ_a^2 estimation. In this example, we use following discretization of process noise sample space:

$$\mathscr{Q} = \{0,05; 0,1; \ldots; 50\}$$

for $\#pn = 1000$. On Figure 5.4 can bee result of maximum likelihood estimation throw all time steps. To obtain a better view into properties of this maximum likelihood estimation,

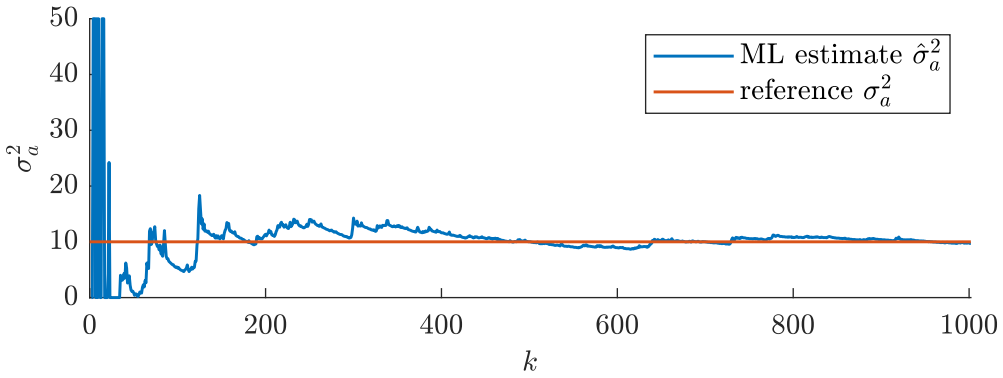


Figure 5.4: Result of estimating process noise $\mathbf{Q}$ using maximum likelihood estimation throw all time steps.

run $\#MC = 100$ Monte Carlo generation of the trajectory and then, for each, provide an estimation of $\mathbf{Q}$. See the result for all Monte Carlo runs on Figure 5.5.

For a set of Monte Carlo runs, we can compute empirical characteristics of the estimation such as sample mean and sample variance for each time step k. Consider sample mean on Figure 5.6. There is also plotted 95% confidence interval for mean at each time step under the assumption that estimates follow the normal distribution. To asses variance consider Fisher information matrix of the process noise at time step k by [28]

$$\mathbf{J}_k(\sigma_a^2) = \frac{k}{2(\sigma_s^2)^2}.$$

Its inverse is Cramér-Rao bound for unbiased estimate. Consider sample variance on Figure 5.7 and detail on Figure 5.8. It is worth mentioning that on Figures 5.7 and

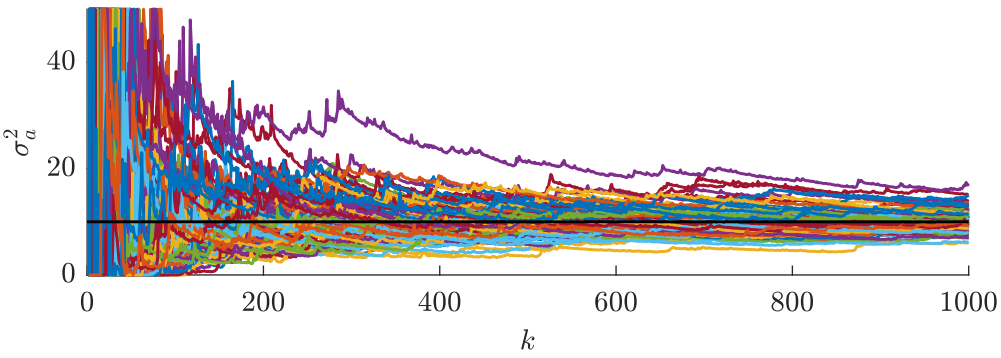


Figure 5.5: Result of maximum likelihood process noise estimation throw all time steps for all Monte Carlo runs.

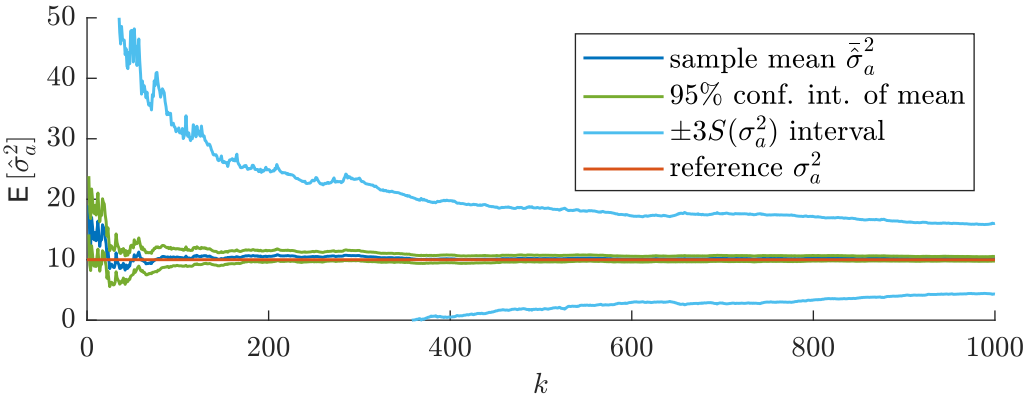


Figure 5.6: Sample mean throw all Monte Carlo runs.

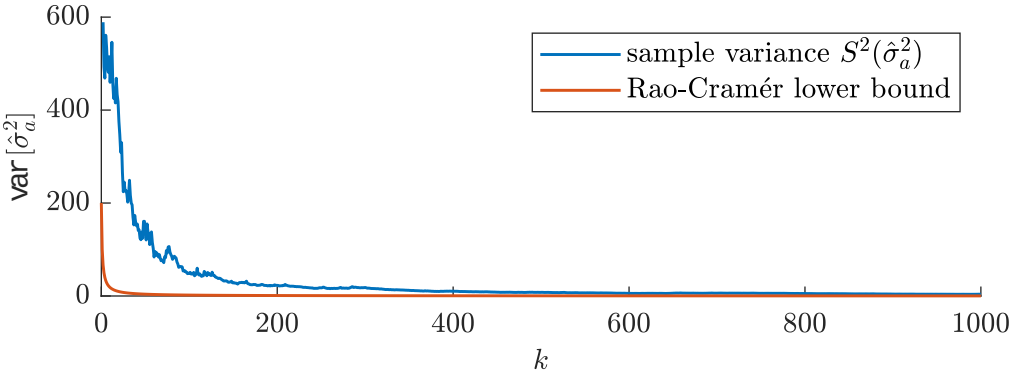


Figure 5.7: Sample variance throw all Monte Carlo runs.

5.8 we have drawn Cramér-Rao lower bound in order to show the minimal theoretical variance of unbiased estimation. Nevertheless, we have not shown that our proposed maximum likelihood estimation is unbiased, so we keep only this as an informal quantity. Although it can be seen from Figure 5.6 that since $k \approx 400$ is the sample mean of our estimation close to the true one. See these results in Table 5.1 at selected time steps $k = 10, 50, 100, 200, 400, 1\,000$. Also on Figure 5.6 we have drawn confidence interval and $\pm 3S(\sigma_a^2)$ are corresponding to normal distribution. It is worth to mention that for higher indexes the hypothesis of normal distribution is not rejected by Anderson-Darling test.

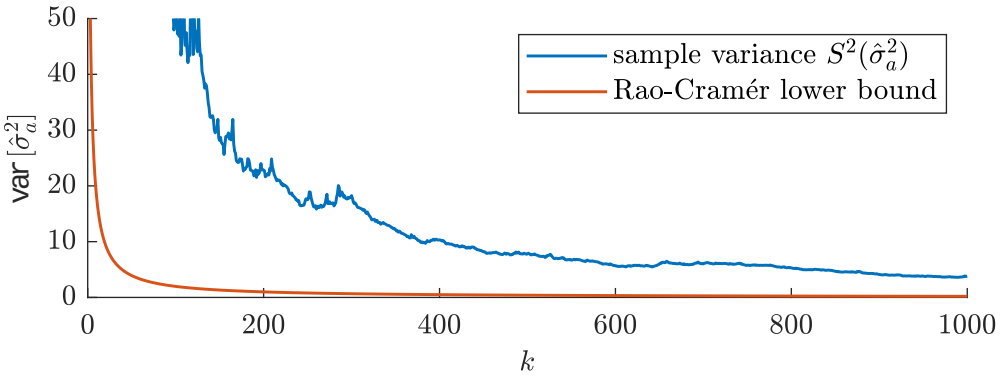


Figure 5.8: Detail of the sample variance throw all Monte Carlo runs.

Table 5.1: Results of the process noise maximum likelihood estimation in the selected time steps.

$k =$	s. mean	95 % conf. int.	s. var.	AD p-value
10	16,39	$\langle 11,75; 21,02\rangle$	545,89	$< 0,005$
50	10,82	$\langle 8,35; 13,29\rangle$	155,07	$< 0,005$
100	10,34	$\langle 8,98; 11,71\rangle$	47,26	$< 0,005$
200	10,55	$\langle 9,63; 11,47\rangle$	21,62	$< 0,005$
400	10,14	$\langle 9,51; 10,78\rangle$	10,26	0,009
1 000	10,16	$\langle 9,78; 10,55\rangle$	3,81	0,106

See Figures 5.9 - 5.14 to obtain a imagination of the shape of the likelihood function $f(\mathcal{Z}_k|\mathbf{Q})$ or its natural logarithm $\ln f(\mathcal{Z}_k|\mathbf{Q})$ at time steps $k = 1, 5, 20, 50, 100, 1\,000$.

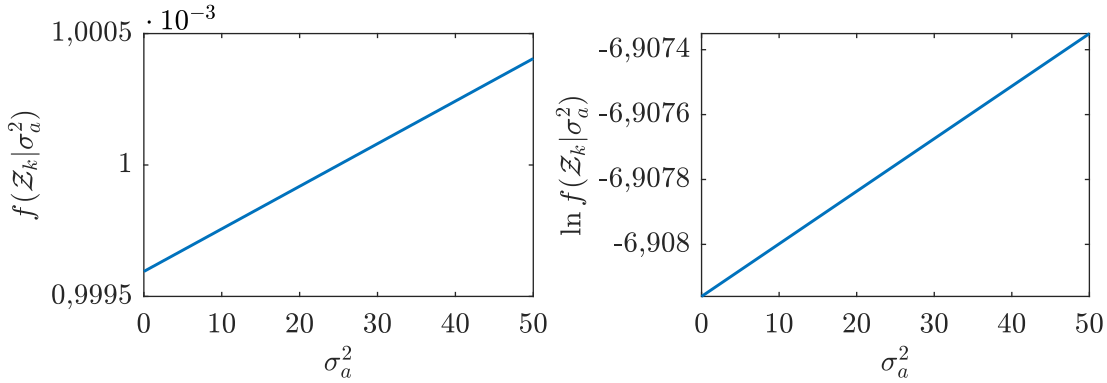


Figure 5.9: Likelihood function of σ_a^2 and its natural logarithm at time step $k = 1$.

We can see on the figures above that it is necessary to have some significant set of measurements (at time steps $k = 1$ and $k = 5$, it was impossible to compute the likelihood function). It is why we have to assume likelihood concerning all measurements up to k instead of only actual measurement $f(\boldsymbol{z}_k|\mathbf{Q}, \mathcal{Z}_{k-1})$. This likelihood does not contain enough information to compute maximizer, as shown in Figures 5.15 and 5.16.

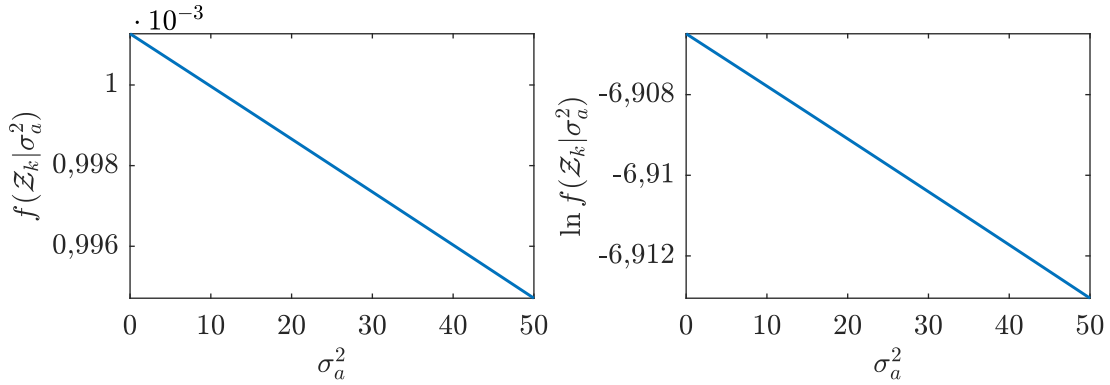


Figure 5.10: Likelihood function of σ_a^2 and its natural logarithm at time step $k = 5$.

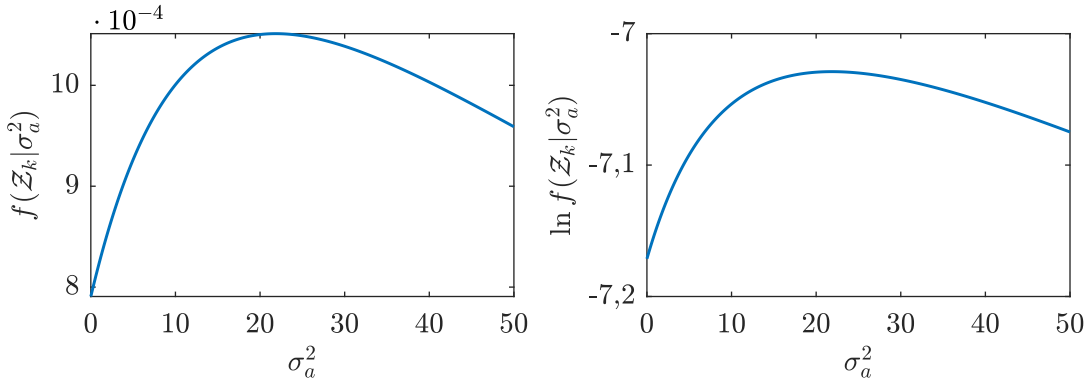


Figure 5.11: Likelihood function of σ_a^2 and its natural logarithm at time step $k = 20$.

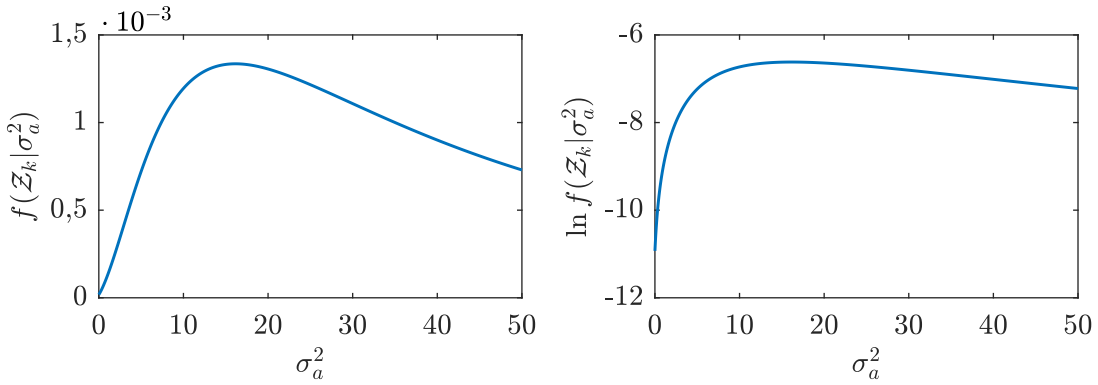


Figure 5.12: Likelihood function of σ_a^2 and its natural logarithm at time step $k = 50$.

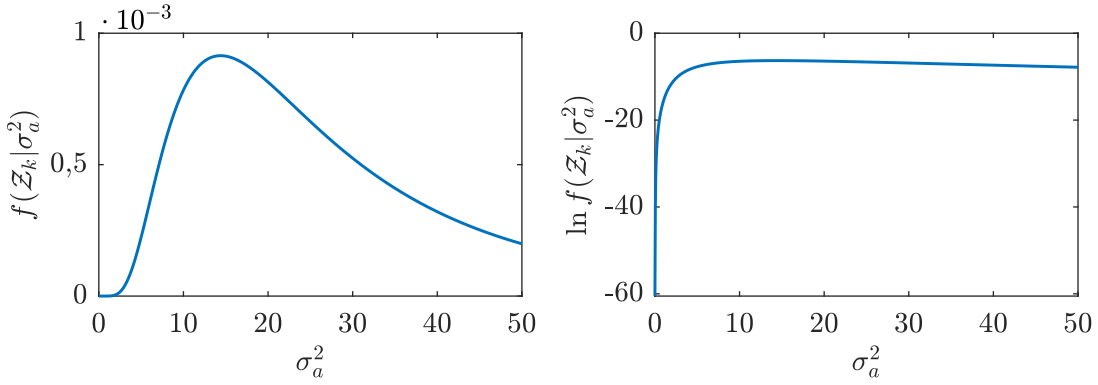


Figure 5.13: Likelihood function of σ_a^2 and its natural logarithm at time step $k = 100$.

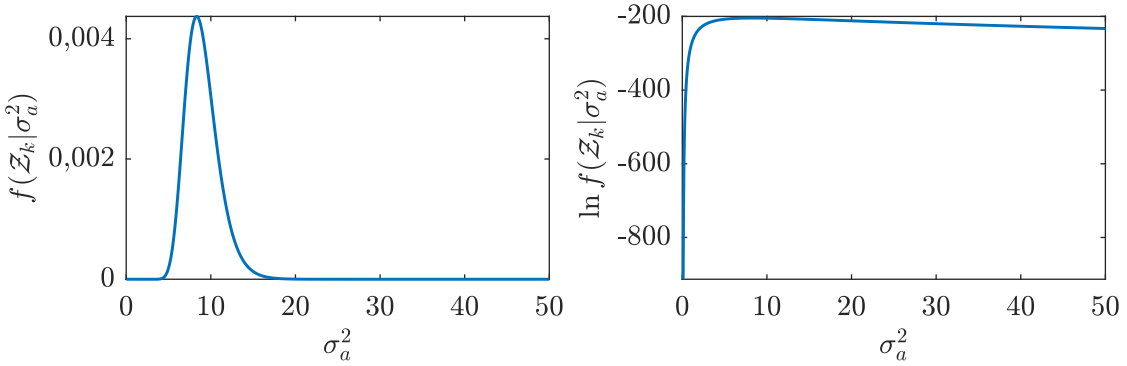


Figure 5.14: Likelihood function of σ_a^2 and its natural logarithm at time step $k = 1\,000$.

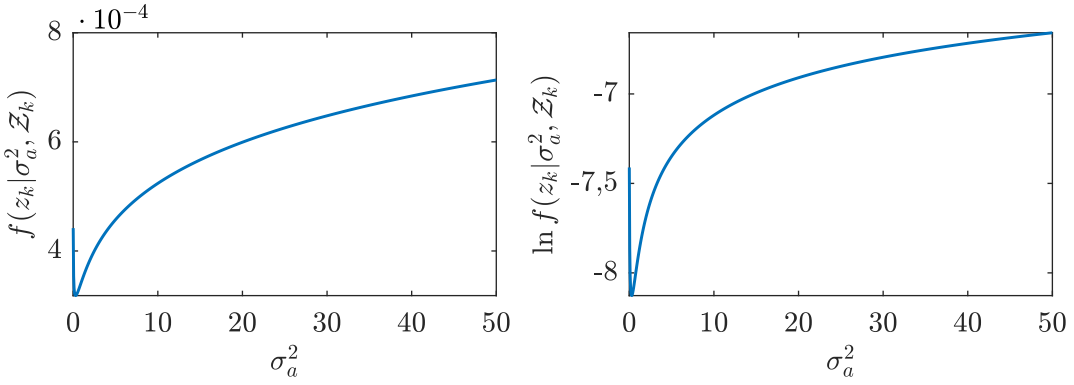


Figure 5.15: Likelihood function $f(z_k|\sigma_a^2, \mathcal{Z}_{k-1})$ and its natural logarithm at time step $k = 70$.

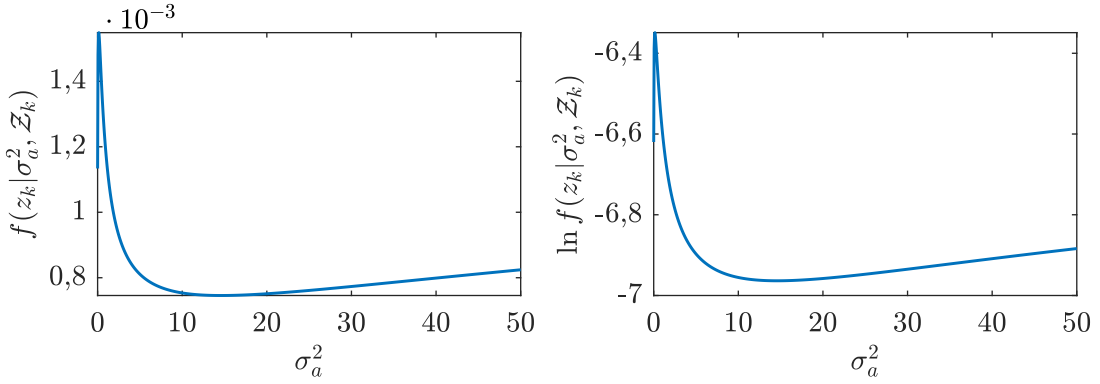


Figure 5.16: Likelihood function $f(z_k|\sigma_a^2, \mathcal{Z}_{k-1})$ and its natural logarithm at time step $k = 100$.

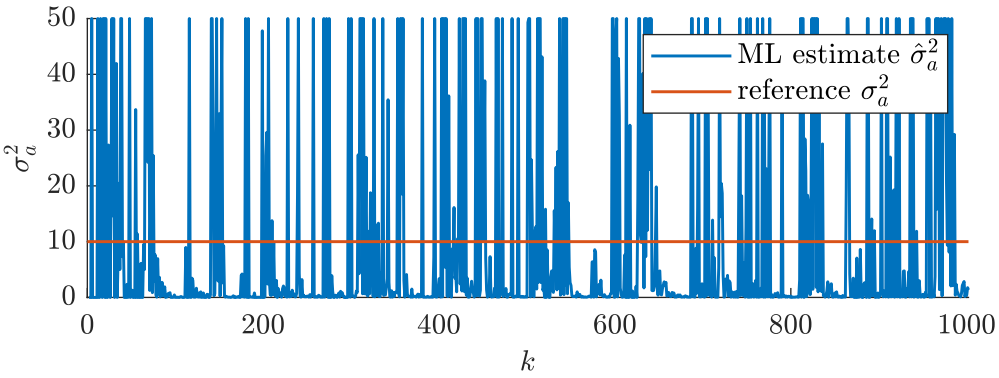


Figure 5.17: Result of maximum likelihood estimation using likelihood function $f(z_k|\sigma_a^2, \mathcal{Z}_{k-1})$ throw all time steps k.

5.4 | Bayesian Estimation Method

In this section, we describe the Bayesian estimation of the process noise $\mathbf{Q}$ at time step k. We aim to compute following posterior density function $f(\boldsymbol{x}_k, \mathbf{Q}|\mathcal{Z}_k)$. By Proposition 3.7 we can split this density into a product of two following densities

$$f(\boldsymbol{x}_k, \mathbf{Q}|\mathcal{Z}_k) = \underbrace{f(\boldsymbol{x}_k|\mathbf{Q}, \mathcal{Z}_k)}_{\text{Kalman filter}} \cdot f(\mathbf{Q}|\mathcal{Z}_k)$$

Where $f(\boldsymbol{x}_k|\mathbf{Q}, \mathcal{Z}_k)$ is computed by Kalman filter with given $\mathbf{Q}$. So, we focus our consideration on the derivation of the posterior density of the process noise $f(\mathbf{Q}|\mathcal{Z}_k)$. Use Bayes equation

$$\begin{aligned} f(\mathbf{Q}|\mathcal{Z}_k) &= f(\mathbf{Q}|\boldsymbol{z}_k, \mathcal{Z}_{k-1}) = \\ &= \frac{f(\boldsymbol{z}_k|\mathbf{Q}, \mathcal{Z}_{k-1}) \cdot f(\mathbf{Q}|\mathcal{Z}_{k-1})}{f(\boldsymbol{z}_k|\mathcal{Z}_k)} = \\ &\propto f(\boldsymbol{z}_k|\mathbf{Q}, \mathcal{Z}_{k-1}) \cdot f(\mathbf{Q}|\mathcal{Z}_{k-1}) \end{aligned}$$

Likelihood

$$f(\boldsymbol{z}_k|\mathbf{Q}, \mathcal{Z}_{k-1}) = \mathsf{N}(\boldsymbol{z}_k, \mathbf{H}_k \bar{\mathbf{x}}_{k|k}(\mathbf{Q}), \mathbf{P}_{k|k}(\mathbf{Q}))$$

by Lemma 5.2 and prior density $f(\mathbf{Q}|\mathcal{Z}_{k-1})$ is known from previous time step and the initial prior $f(\mathbf{Q})$ must be given. Bayesian estimation of the hidden state and process noise such that

$$[\hat{\boldsymbol{x}}_k, \hat{\mathbf{Q}}] = \mathsf{E}[\boldsymbol{X}_k, \boldsymbol{Q}|\mathcal{Z}_k]$$

leads to intractable integral. Thus, we use marginal estimation. Firstly provide Bayesian estimation of the process noise

$$\hat{\mathbf{Q}} = \mathsf{E}[\boldsymbol{Q}|\mathcal{Z}_k]$$

And then use Kalman filter to compute

$$f(\boldsymbol{x}_k|\hat{\mathbf{Q}}, \mathcal{Z}_k) = \mathsf{N}(\boldsymbol{x}_k, \bar{\mathbf{x}}_{k|k}(\mathbf{Q}), \mathbf{P}_{k|k}(\mathbf{Q})).$$

Unfortunately computation of $\mathsf{E}[\boldsymbol{Q}|\mathcal{Z}_k]$ leads to intractable integral too. To compute it, we could use approximation techniques such as *variational Bayes approximation* [42]. In

this text, we use discretization of the process noise sample space, such as maximum likelihood estimation. Consider the way compute the Bayesian estimation in the summarized Algorithm 5.4.2:

Algorithm 5.2: Bayesian estimation of the process noise $\mathbf{Q}$

Parameters:

- discretization of $\mathbf{Q}$'s sample space $\mathscr{Q} = \{\mathbf{Q}^j\}_{j=1}^{\#pn}$
- initial prior density $f(\mathbf{Q}) \to \mathsf{P}\{\mathbf{Q}^j\};\ j = 1, \ldots, \#pn$

Input from previous time step $k-1$:
prior density:

- $f(\boldsymbol{x}_{k-1}|\mathbf{Q}, \mathcal{Z}_{k-1}) \to \{\bar{\mathbf{x}}_{k-1|k-1}(\mathbf{Q}^j), \mathbf{P}_{k-1|k-1}(\mathbf{Q}^j)\}_{i=1}^{\#pn}$
- $f(\mathbf{Q}|\mathcal{Z}_{k-1}) \to \mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_{k-1}\};\ j = 1, \ldots, \#pn$

Computation:

- run $\#pn$ Kalman filters for $\mathbf{Q}^j$, $j = 1, \ldots, \#pn$ with output $\bar{\mathbf{x}}_{k|k}(\mathbf{Q}^j), \mathbf{P}_{k|k}(\mathbf{Q}^j)$
- compute approximation of $f(\mathbf{Q}|\mathcal{Z}_k)$:

$$\begin{aligned}\mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k\} &:= \mathsf{P}\{\boldsymbol{z}_k|\mathbf{Q}^j, \mathcal{Z}_k\} \cdot \mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k\} = \\ &= \mathsf{N}(\boldsymbol{z}_k; \mathbf{H}_k\bar{\mathbf{x}}_{k|k}(\mathbf{Q}^j), \mathbf{S}_k(\mathbf{Q}^j)) \cdot \mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_{k-1}\};\ j = 1, \ldots, \#pn\end{aligned}$$

- normalize computed $\mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k\}$:

$$\mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k\} := \frac{\mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k\}}{\sum_{j=1}^{\#pn} \mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k\}} \tag{5.12}$$

- Bayesian estimation of the process noise :

$$\hat{\mathbf{Q}} = \sum_{j=1}^{\#pn} \mathbf{Q}^j \mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k\} \tag{5.13}$$

- State estimation $\hat{\boldsymbol{x}}_k$

$$\bar{\mathbf{x}}_{k|k}(\hat{\mathbf{Q}}), \mathbf{P}_{k|k}(\hat{\mathbf{Q}})$$

Output for next time step $k+1$:

- $\bar{\mathbf{x}}_{k|k}(\mathbf{Q}^j), \mathbf{P}_{k|k}(\mathbf{Q}^j);\ j = 1, \ldots, \#pn$
- $\mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k\};\ j = 1, \ldots, \#pn$

Output for current state and process noise estimation at time step k:

- $\hat{\mathbf{Q}}, \bar{\mathbf{x}}_{k|k}(\hat{\mathbf{Q}}), \mathbf{P}_{k|k}(\hat{\mathbf{Q}})$

For more sake of clarity consider scheme of this algorithm on Figure 5.18.

Now consider the same test example as Example 5.3 and provide proposed Bayesian estimation of the process noise.

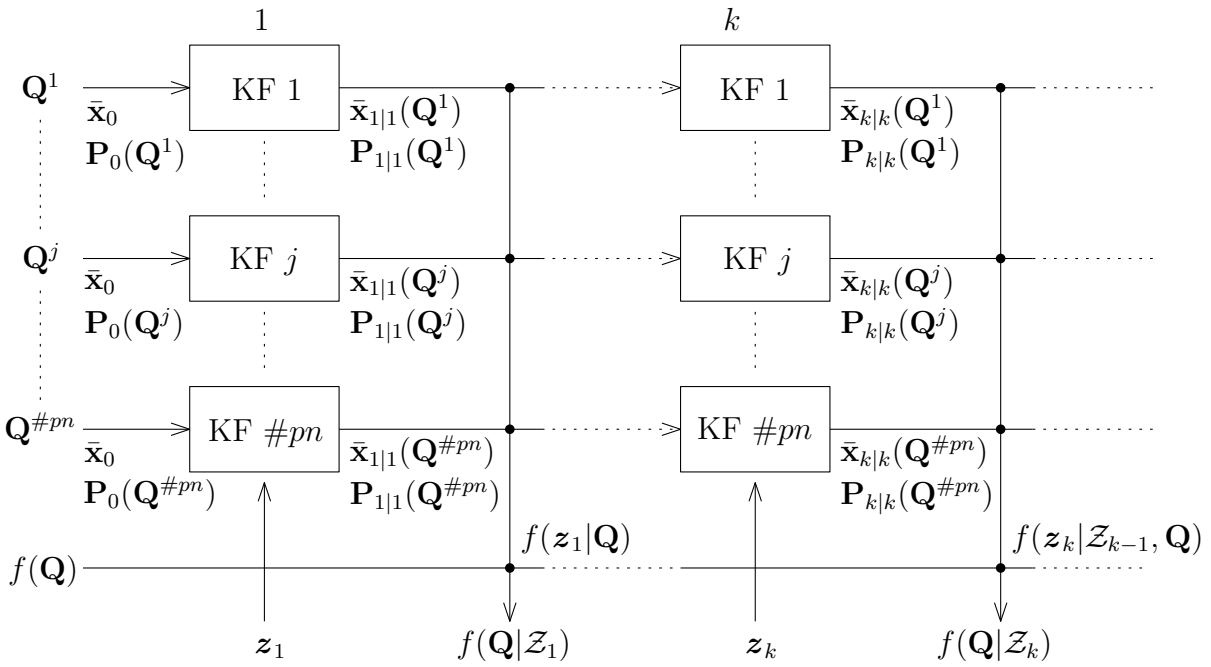


Figure 5.18: Scheme of the proposed way to compute Bayesian estimation of $\mathbf{Q}$.

Example 5.4 (Test example)**.** Consider the same generated data as in Example 5.3. To compute estimation of the process noise σ_a^2 we use Bayesian estimation method. We discretize process noise sample space $\Omega_{\sigma_a^2} \approx \mathscr{Q}$ such that

$$\mathscr{Q} = \{0,05; 0,1; \ldots; 50\} \Rightarrow \#pn = 1\,000.$$

Then consider two initial prior distributions. First one consider uniform distribution on interval $\langle 0; 50\rangle$:

$$f(\sigma_a^2) = \mathsf{U}(\sigma_a^2; 0, 50).$$

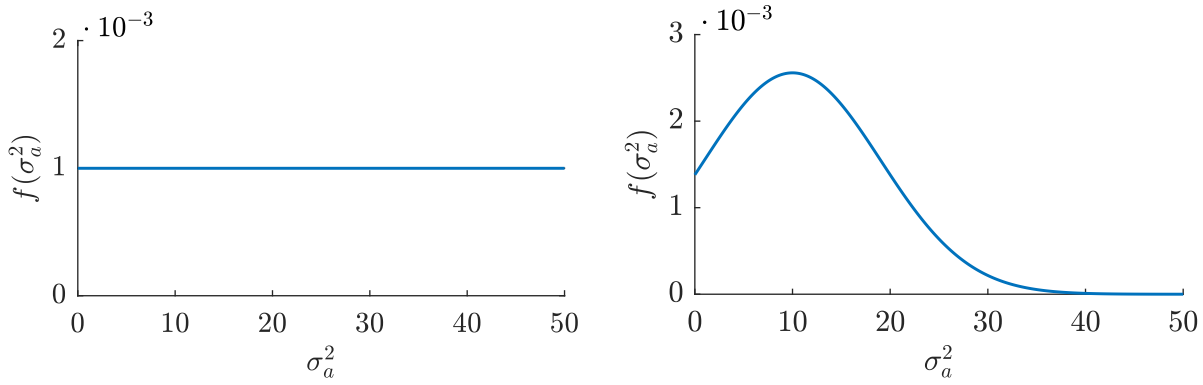
This has meaning that we have no information about the process noise. Second initial prior distribution is following normal distribution

$$f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 9).$$

This distribution can be considered, for instance, from previous experiences or from the fact that it is less probable that the target has high process noise (i.e., > 30). See these initial distributions on Figure 5.19

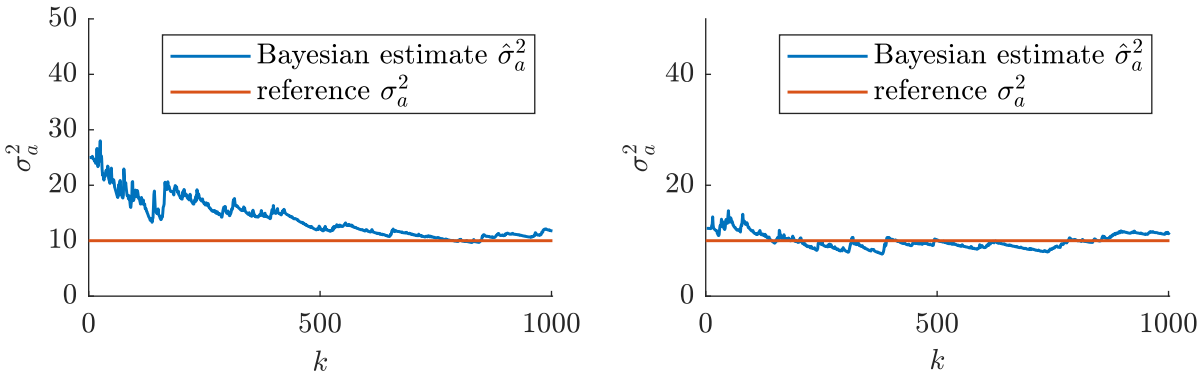
On Figure 5.20 can be seen the result of Bayesian estimation throw all time steps. The first thing we notice from Figure 5.20 is that we have obtained significantly better results at the beginning in comparison with maximum-likelihood estimation. Also, as expected, normal initial prior distribution has even better performance at the beginning because it contains more information about the noise.

Run $\#MC = 100$ Monte Carlo runs to get a more consistent view of properties of proposed Bayesian estimation. See on Figure 5.21 sample mean throw all Monte Carlo runs of the process noise estimation by Bayesian method throw all time steps. Then on Figure 5.22 can be seen sample variance. From the Figures 5.21 and 5.22 it is clear that we have obtained better performance from the one with normal initial prior distribution. It is intuitive since the system have more information about the noise at the beginning, but as time index increases, the difference gradually disappear. But From the Figure 5.22



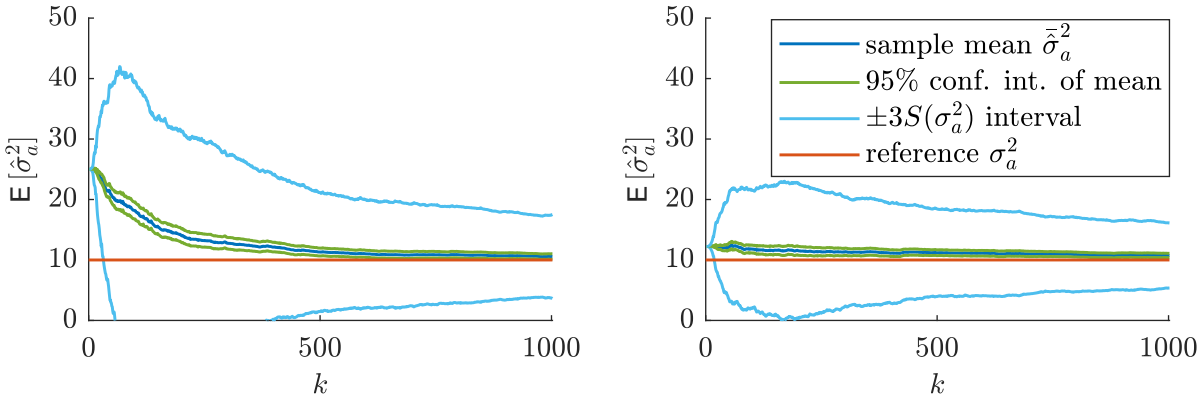
(a) $f(\sigma_a^2) = \mathsf{U}(\sigma_a^2; 0, 50)$ **(b)** $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 9)$

Figure 5.19: Considered initial prior probability densities of the process noise σ_a^2



(a) $f(\sigma_a^2) = \mathsf{U}(\sigma_a^2; 0, 50)$ **(b)** $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 9)$

Figure 5.20: Result of process noise $\mathbf{Q}$ estimation throw the all time steps.



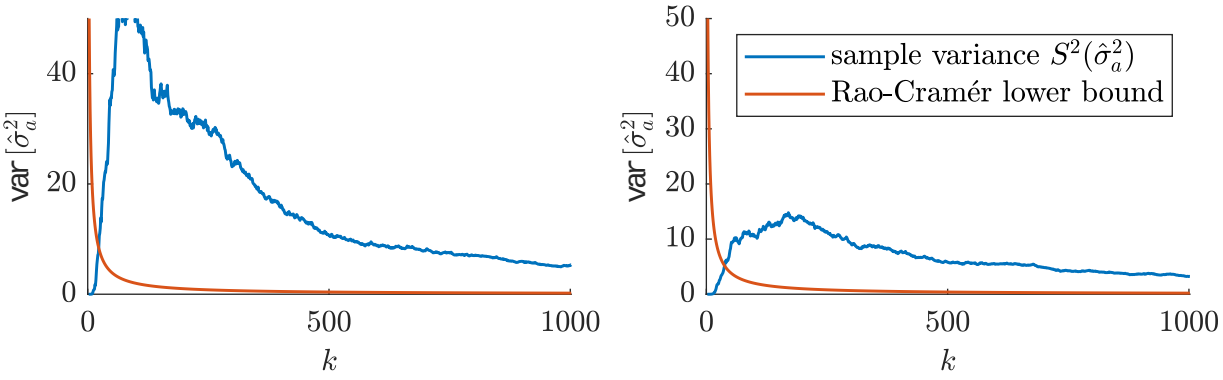
(a) $f(\sigma_a^2) = \mathsf{U}(\sigma_a^2; 0, 50)$ **(b)** $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 9)$

Figure 5.21: Sample mean of process noise $\mathbf{Q}$ estimation throw all Monte Carlo runs.

it can be see that (b) has smaller variance for all time indices. Compare Figures 5.20, 5.21 and 5.22 with the maximum likelihood corresponding Figures 5.2, 5.6, 5.7 and5.8.

Now see the following result together with maximum likelihood estimation at Table 5.2. Summarizing results of estimation at time index $k = 10, 50, 100, 200, 400, 1\,000$.

Comparing Tables 5.1 and 5.2 we can conclude that Bayesian estimation has better performance at the beginning and maximum likelihood estimation has better performance



(a) $f(\sigma_a^2) = \mathsf{U}(\sigma_a^2; 0, 50)$ **(b)** $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 9)$

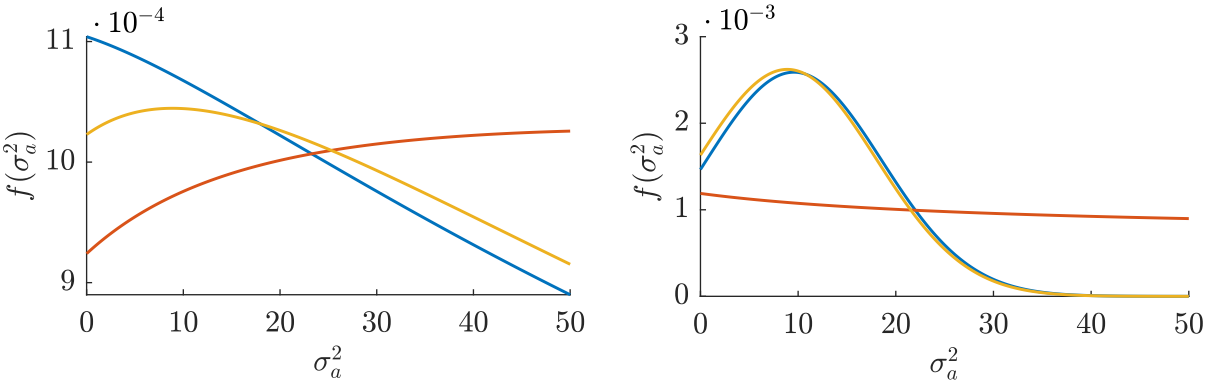
Figure 5.22: Sample variance of process noise $\mathbf{Q}$ estimation throw all Monte Carlo runs.

Table 5.2: Results of the process noise Bayesian estimation in the selected time steps.

	$\mathsf{U}(\sigma_a^2; 0, 50)$			$\mathsf{N}(\sigma_a^2; 10, 9)$		
$k =$	s. mean	95 % conf. int.	s. var.	s. mean	95 % conf. int.	s. var.
10	24,99	$\langle 24,82; 25,16\rangle$	0,74	12,23	$\langle 12,19; 12,27\rangle$	0,04
50	20,38	$\langle 19,14; 21,63\rangle$	39,32	12,28	$\langle 11,67; 12,89\rangle$	9,42
100	17,93	$\langle 16,50; 19,35\rangle$	51,35	11,80	$\langle 11,15; 12,44\rangle$	10,50
200	14,03	$\langle 12,89; 15,17\rangle$	33,11	11,50	$\langle 10,77; 12,23\rangle$	13,52
400	12,12	$\langle 11,33; 12,90\rangle$	15,72	11,14	$\langle 10,58; 11,69\rangle$	7,76
1 000	10,57	$\langle 10,11; 11,03\rangle$	5,33	10,74	$\langle 10,39; 11,10\rangle$	3,24

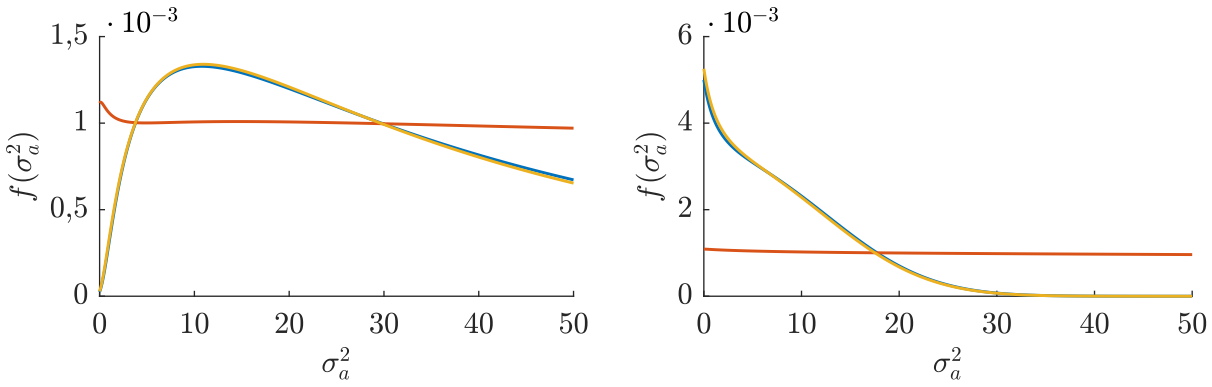
in higher time indices. It comes from that our proposed maximum likelihood method strictly assume that process noise is time-invariant. In Bayesian estimation, we use likelihood $f(z_k|\sigma_a^2, \mathcal{Z}_{k-1})$ reflecting only current measurement and information about previous states and measurements is contained in prior density. It is worth to mention that for higher time index ($k > 100$) for all time steps the hypothesis of normal distribution of the estimation was not rejected by the Anderson-Darling test.

At the end of this section about Bayesian estimation of the process noise see Figures 5.23, 5.24, 5.25 and 5.26 to have imagination about shape of the likelihood, prior and posterior densities.



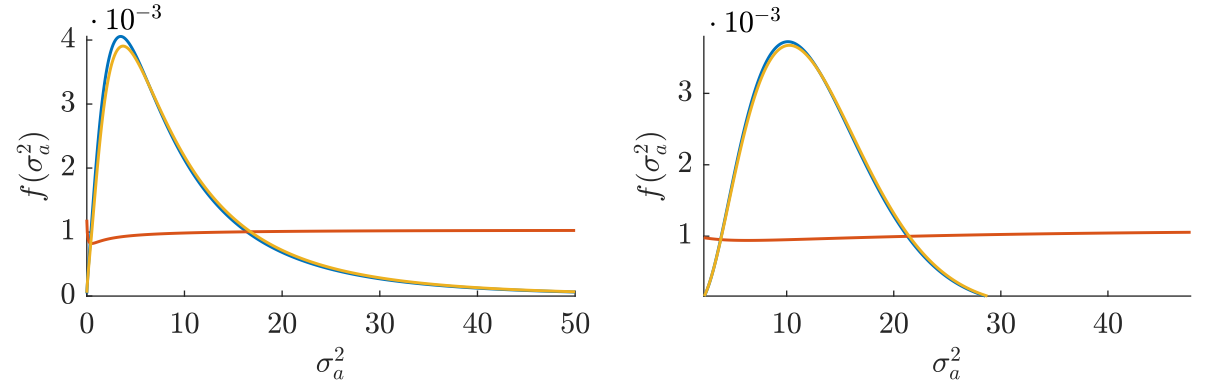
(a) $f(\sigma_a^2) = \mathsf{U}(\sigma_a^2; 0, 50)$ **(b)** $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 9)$

Figure 5.23: Prior, likelihood and posterior density at time step $k = 20$.



(a) $f(\sigma_a^2) = \mathsf{U}(\sigma_a^2; 0, 50)$ **(b)** $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 9)$

Figure 5.24: Prior, likelihood and posterior density at time step $k = 50$.



(a) $f(\sigma_a^2) = \mathsf{U}(\sigma_a^2; 0, 50)$ **(b)** $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 9)$

Figure 5.25: Prior, likelihood and posterior density at time step $k = 100$.

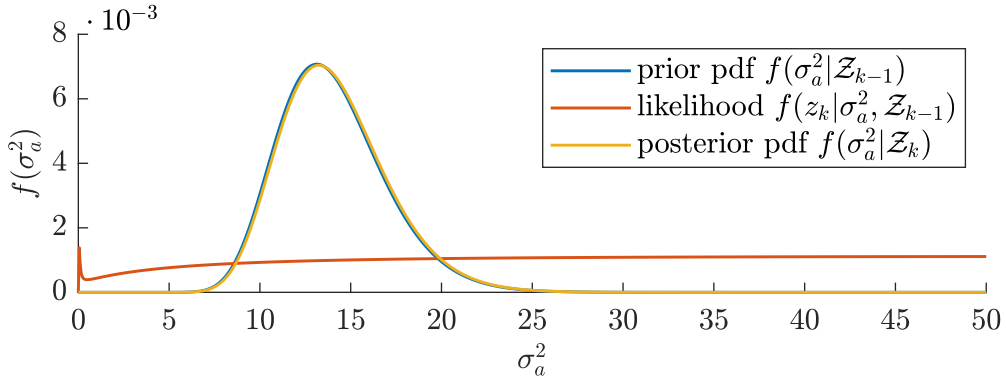


Figure 5.26: Prior, likelihood and posterior density at time step $k = 50$ with initial prior density $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 9)$.

5.5 | Relation Between Maximum Likelihood and Bayesian Estimation Methods

In this section, we briefly discuss the relation between proposed maximum likelihood method and Bayesian method to estimate process noise $\mathbf{Q}$. Consider the following proposition.

Proposition 5.5. *Posterior density function $f(\mathbf{Q}|\mathcal{Z}_k)$ is equal to*

$$f(\mathbf{Q}|\mathcal{Z}_k) \propto f(\mathcal{Z}_k|\mathbf{Q}) \cdot f(\mathbf{Q}) = \left(\prod_{i=1}^{k} f(\boldsymbol{z}_i|\mathcal{Z}_{i-1}, \mathbf{Q}) \right) \cdot f(\mathbf{Q})$$

where $f(\mathcal{Z}_k|\mathbf{Q})$ is likelihood function which is maximized in maximum likelihood estimation and $f(\mathbf{Q})$ is initial prior probability density function of the process noise (i.e. $f(\mathbf{Q}) = f(\mathbf{Q}|\mathcal{Z}_0)$).

Proof. The proposition is clearly valid by applying Bayes's formula:

$$f(\mathbf{Q}|\mathcal{Z}_k) = \frac{f(\mathcal{Z}_k|\mathbf{Q}) \cdot f(\mathbf{Q})}{f(\mathcal{Z}_k)}.$$

Then relation

$$f(\mathcal{Z}_k|\mathbf{Q}) = \prod_{i=1}^{k} f(\boldsymbol{z}_i|\mathcal{Z}_{i-1}, \mathbf{Q})$$

is derived at the beginning of Section 5.3. □

Recall that we consider discretized sample space of the process noise $\mathscr{Q}$ in our proposed maximum likelihood and Bayesian estimation algorithms (see Sections 5.3 and 5.3). This approach provides some interesting property, which we mention in the following proposition.

Proposition 5.6. *Consider discretized sample space of the process noise $\mathscr{Q}$. Then, in case initial prior distribution is non-informal $f(\mathbf{Q}) = \mathsf{U}(\mathbf{Q}; \mathscr{Q})$*

$$p(\mathbf{Q}|\mathcal{Z}_k) = p(\mathcal{Z}_k|\mathbf{Q}).$$

Proof. By Proposition 5.5 (or by Bayes's formula in general) we can write

$$p(\mathbf{Q}|\mathcal{Z}_k) \propto p(\mathcal{Z}_k|\mathbf{Q}) \cdot p(\mathbf{Q}).$$

Since $p(\mathbf{Q})$ gives the same values $\forall \mathbf{Q} \in \mathscr{Q}$, then term $p(\mathbf{Q})$ does not influence the product of the probability functions. □

We just have shown, that if we set non-informal initial prior $f(\mathbf{Q})$ in our proposed discretized Bayesian and maximum likelihood methods, the only difference between them is representative value. In case of Bayesian estimation it is expected value and in case of maximum likelihood method it is modus.

We try to use this property to improve performance of the maximum likelihood method at the beginning of simulation. So introduce improved maximum likelihood estimation of the process noise described in following Algorithm 5.3.

Algorithm 5.3: Improved ML estimation of the process noise $\mathbf{Q}$

Parameter:

- discretization of $\mathbf{Q}$'s sample space $\mathscr{Q} = \{\mathbf{Q}^j\}_{j=1}^{\#pn}$
- initial prior density $f(\mathbf{Q}) \to \mathsf{P}\{\mathbf{Q}^j\}; j = 1, \ldots, \#pn$

Input from previous time step $k-1$:
prior density:

- $f(\boldsymbol{x}_{k-1}|\mathbf{Q}, \mathcal{Z}_{k-1}) \to \{\bar{\mathbf{x}}_{k-1|k-1}(\mathbf{Q}^j), \mathbf{P}_{k-1|k-1}(\mathbf{Q}^j)\}_{i=1}^{\#pn}$
- $f(\mathcal{Z}_{k-1}|\mathbf{Q}) \to \mathsf{P}\{\mathcal{Z}_{k-1}|\mathbf{Q}^j\};\ j = 1, \ldots, \#pn$

Computation:

- run $\#pn$ Kalman filters for $\mathbf{Q}^j$, $j = 1, \ldots, \#pn$ with output $\bar{\mathbf{x}}_{k|k}(\mathbf{Q}^j), \mathbf{P}_{k|k}(\mathbf{Q}^j)$
- compute approximation of $f(\mathbf{Q}|\mathcal{Z}_k)$:

$$\begin{aligned}\mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k\} &= \mathsf{P}\{\boldsymbol{z}_k|\mathbf{Q}^j, \mathcal{Z}_k\} \cdot \mathsf{P}\{\mathcal{Z}_{k-1}|\mathbf{Q}^j\} \cdot \mathsf{P}\{\mathbf{Q}^j\} = \\ &= \mathsf{N}(\boldsymbol{z}_k; \mathbf{H}_k\bar{\mathbf{x}}_{k|k}(\mathbf{Q}^j), \mathbf{S}_k(\mathbf{Q}^j)) \cdot \mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_{k-1}\} \cdot \mathsf{P}\{\mathbf{Q}^j\};\ j = 1, \ldots, \#pn\end{aligned}$$

- improved maximum likelihood estimation:

$$\hat{\mathbf{Q}} = \max_{\mathbf{Q}^j} \mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k\}$$

Output for next time step $k+1$:

- $\bar{\mathbf{x}}_{k|k}(\mathbf{Q}^j), \mathbf{P}_{k|k}(\mathbf{Q}^j);\ j = 1, \ldots, \#pn$
- $\mathsf{P}\{\mathcal{Z}_k|\mathbf{Q}^j\};\ j = 1, \ldots, \#pn$

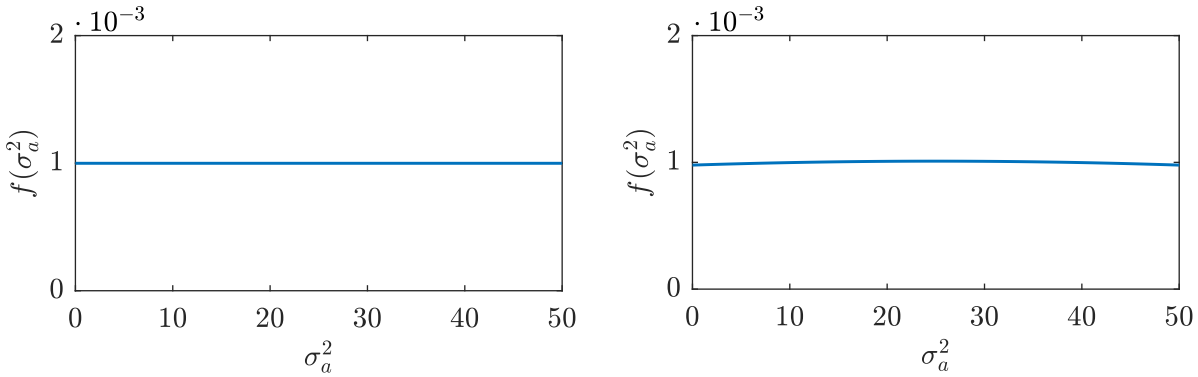
Output for current state and process noise estimation at time step k:

- $\hat{\mathbf{Q}}, \bar{\mathbf{x}}_{k|k}(\hat{\mathbf{Q}}), \mathbf{P}_{k|k}(\hat{\mathbf{Q}})$

Try the algorithm of the improved maximum likelihood estimation on the test example.

Example 5.7 (Test example)**.** Consider the same data as in Example 5.3. Also consider the same parameters of the maximum likelihood estimation as in this example and add initial prior distribution $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 25, 100)$. See this initial distribution and compare with non-informal discrete distribution in Figure 5.27. Consider mentioned distributions as discretization of the continuous one. For simplicity write the continuous one.

Figure 5.27 shows that non-informal and proposed initial distributions are almost same, but the proposed one has maximizer. See the result of the improved maximum likelihood estimator on the Figure 5.28. Compare with Figure 5.4. See that we have improved performance at the beginning. This is obviously caused by the initial prior with maximizer. To obtain more consistent view of proposed improved maximum likelihood



(a) $f(\sigma_a^2) = \mathsf{U}(\sigma_a^2; 0, 50)$ **(b)** $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 25, 100)$

Figure 5.27: Non-informal and proposed initial distributions.

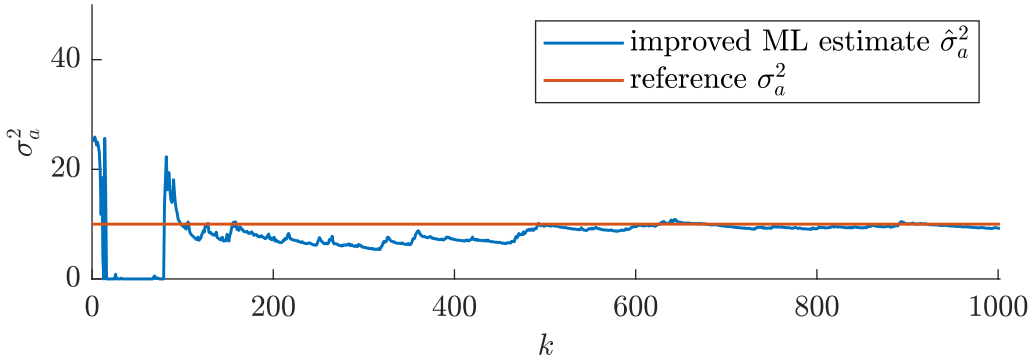


Figure 5.28: Estimation of the process noise by the improved maximum likelihood estimation.

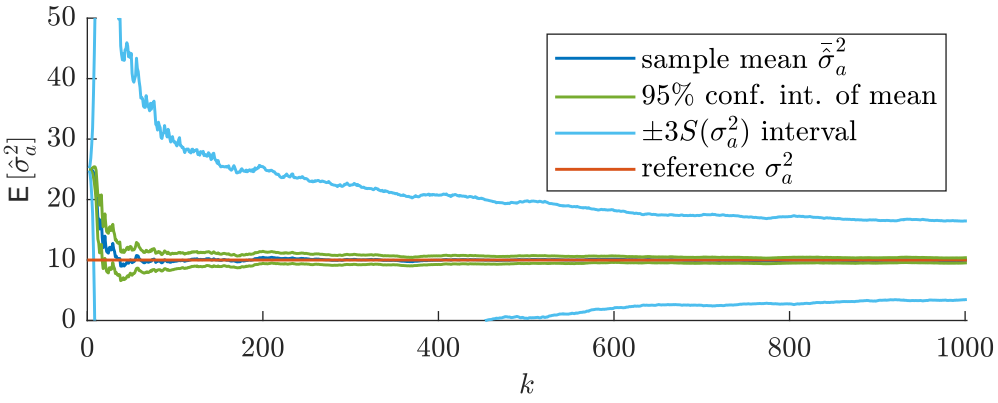


Figure 5.29: Sample mean throw all Monte Carlo runs.

estimation, run $\#MC = 100$ Monte Carlo simulations. See the results – sample mean and variance on Figures 5.29 and 5.30. Compare with Figures 5.6, 5.7 and 5.8. Notice that at the beginning we have significantly decreased the variance. To underline these conclusion from the results on the figures, see Table 5.3 and compare with Table 5.1.

In Table 5.3 comparing with Table 5.1 we can see that at time steps $k = 10, 50, 100$ we have significantly decreased sample variance. However, latter the variance has slightly increased. This is the situation, which can be followed also in comparison with Bayesian estimation. If we set initial prior, which can be wrong, or better say not precise, at

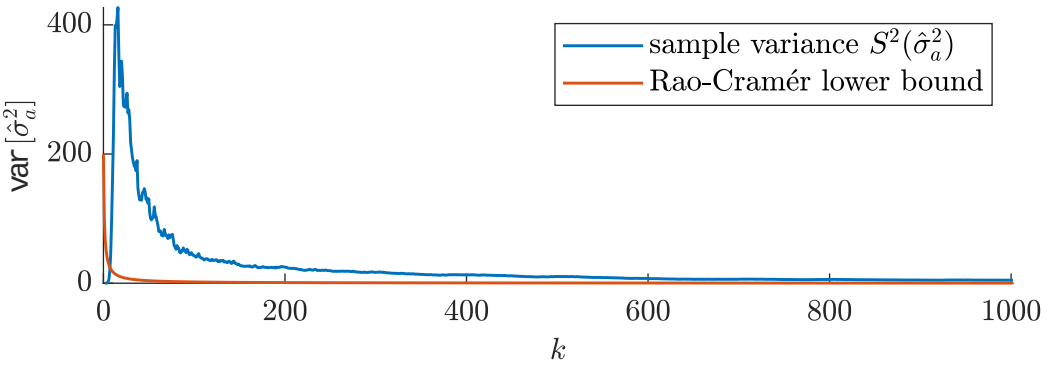


Figure 5.30: Sample variance throw all Monte Carlo runs.

Table 5.3: Results of the process noise improved maximum likelihood estimation in the selected time steps.

$k =$	s. mean	95 % conf. int.	s. var.	AD p-value
10	17,06	$\langle 13,46; 20,67\rangle$	326,22	$< 0,005$
50	9,74	$\langle 7,75; 11,74\rangle$	100,16	$< 0,005$
100	9,83	$\langle 8,57; 11,10\rangle$	40,25	$< 0,005$
200	10,38	$\langle 9,39; 11,37\rangle$	24,55	$< 0,005$
400	9,96	$\langle 9,24; 10,68\rangle$	12,95	$0,246$
1 000	9,97	$\langle 9,53; 10,40\rangle$	4,70	$0,738$

beginning algorithm has at least some information and can reduce uncertainty of the estimation, but in the long run imprecise prior information can slightly increase the variance. Nevertheless, in that case, the variance increase is almost negligible. We state this as success, with omissible modification of the non-informal prior we obtain significant variance reduction at the beginning of simulation.

What about we have more information at the beginning? Consider initial prior $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 6)$. See the result on Figure 5.31. Compare with Figure 5.4. By that, no doubts

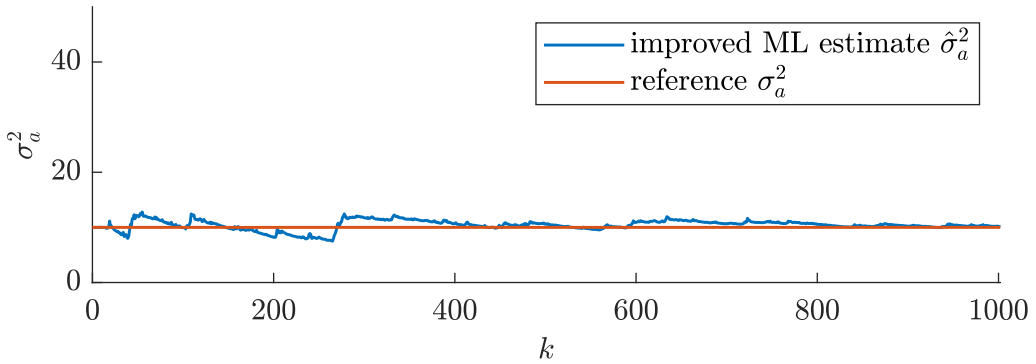


Figure 5.31: Estimation of the process noise by the improved maximum likelihood estimation with initial prior distribution $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 10, 6)$.

we have obtained better result.

5.6 | Use IMM to Estimate Process Noise

We can use IMM algorithm described in Chapter 4 to estimate process noise. We can define set of motion models with same parameters beside process noise $\mathbf{Q}$ and initial covariance $\mathbf{P}_0$, which is directly affected by the process noise value. Write down this idea. Set $\#m = \#pn$ – number of models is same as number of discrete values of possible process noise $\mathbf{Q}$ and define $\mathscr{M} = \{m^j\}_{j=1}^{\#pn}$ such that

$$m^j = \langle \mathbf{F}_k, \mathbf{G}_k, \mathbf{u}_k, \mathbf{\Gamma}_k, \mathbf{Q}^j, \mathbf{H}_k, \bar{\mathbf{x}}_0, \mathbf{P}_0^j \rangle;\ j = 1, \ldots, \#pn. \tag{5.14}$$

Try this idea on the data from the test example (Example 5.3).

Example 5.8. Consider the same generated data as in Example 5.3 (Test example). Define set of possible models such that it is set of DWNA models (see Appendix A) with different parameters σ_a^2 from the set

$$\sigma_a^2 \in \{0,05;\ 0,1;\ \ldots;\ 50\}. \tag{5.15}$$

One could notice that this completely underline the definition in Equation (5.14). Because different process noise (by DWNA definition $\mathbf{Q}_k = \sigma_a^2$) only influence initial variance $\mathbf{P}_0$. Last, it remains to define Markov transition matrix. Consider

$$\mathbf{\Pi} = \{\pi_{ij}\}_{i,j=1}^{\#pn};\ p_{ij} = \begin{cases} 0,703 & i = j, \\ 0,003 & i \neq j. \end{cases} \tag{5.16}$$

See the result on Figure 5.32.

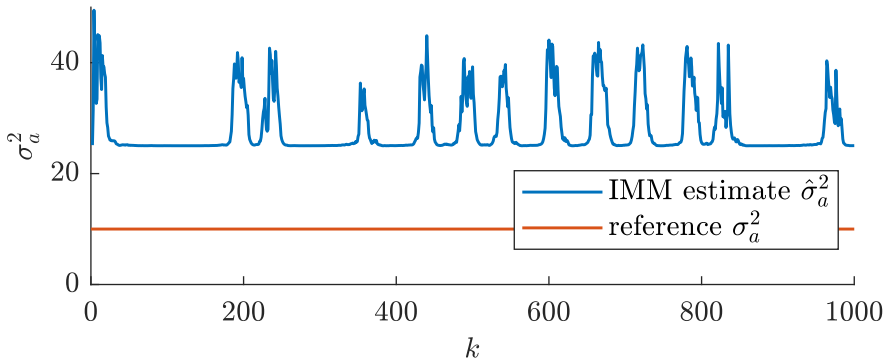


Figure 5.32: Estimating process noise using IMM with $\mathscr{M}$ defined in (5.15) and $\mathbf{\Pi}$ in (5.16).

Then try to change definition of the Markov transition matrix $\mathbf{\Pi}$ such that

$$\mathbf{\Pi} = \{\pi_{ij}\}_{i,j=1}^{\#pn};\ p_{ij} = \begin{cases} 0,9901 & i = j, \\ 10^{-4} & i \neq j. \end{cases} \tag{5.17}$$

See the result on Figure 5.33. On Figures 5.32 and 5.33 we can see that we have obtained bad result in sense of process noise estimation. They can be more reasons of this weak performance, but we conclude that the main one is mixing stage (see Algorithm 4.1) because models (i.e. values of the process noise $\mathbf{Q}^j$) with low probability are included into prediction (prior distribution).

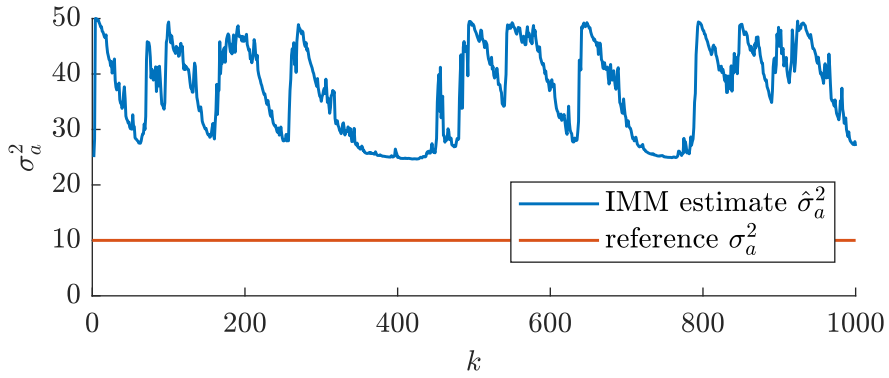


Figure 5.33: Estimating process noise using IMM with $\mathscr{M}$ defined in (5.15) and $\mathbf{\Pi}$ in (5.17).

Now, try to play with the definition of the $\mathscr{M}$. Take $\#pn = \#m = 3$ and

$$\sigma_a^2 \in \{5, 10, 15\}. \tag{5.18}$$

and define Markov transition matrix

$$\mathbf{\Pi} = \begin{bmatrix} 0,8 & 0,1 & 0,1 \\ 0,1 & 0,8 & 0,1 \\ 0,1 & 0,1 & 0,8 \end{bmatrix} \tag{5.19}$$

See the result on Figure 5.34. We see that we have obtained such an excellent result. We

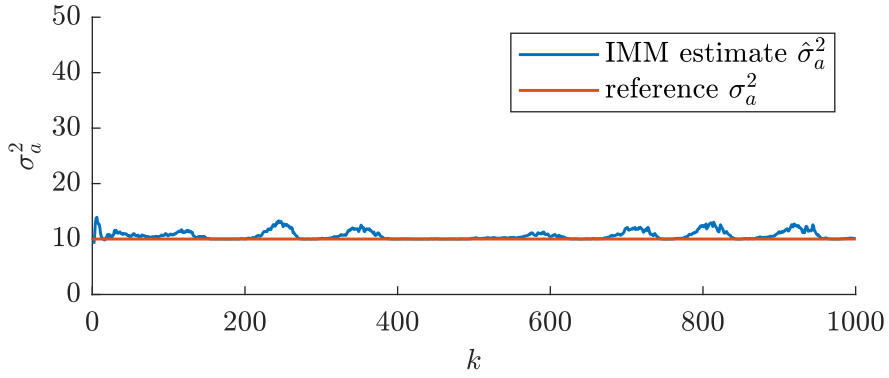


Figure 5.34: Estimating process noise using IMM with $\mathscr{M}$ defined in (5.18) and $\mathbf{\Pi}$ in (5.19).

can state that increasing the $\#m$ does not necessarily mean improving performance in using the IMM algorithm. Now, try to make a slight change in the definition of the $\mathscr{M}$ and $\mathbf{\Pi}$. Define $\mathscr{M}$ such that

$$\sigma_a^2 = \{5, 10, 15, 20, 25, 30\} \tag{5.20}$$

and

$$\mathbf{\Pi} = \begin{bmatrix} 0,75 & 0,05 & 0,05 & 0,05 & 0,05 & 0,05 \\ 0,05 & 0,75 & 0,05 & 0,05 & 0,05 & 0,05 \\ 0,05 & 0,05 & 0,75 & 0,05 & 0,05 & 0,05 \\ 0,05 & 0,05 & 0,05 & 0,75 & 0,05 & 0,05 \\ 0,05 & 0,05 & 0,05 & 0,05 & 0,75 & 0,05 \\ 0,05 & 0,05 & 0,05 & 0,05 & 0,05 & 0,75 \end{bmatrix} \tag{5.21}$$

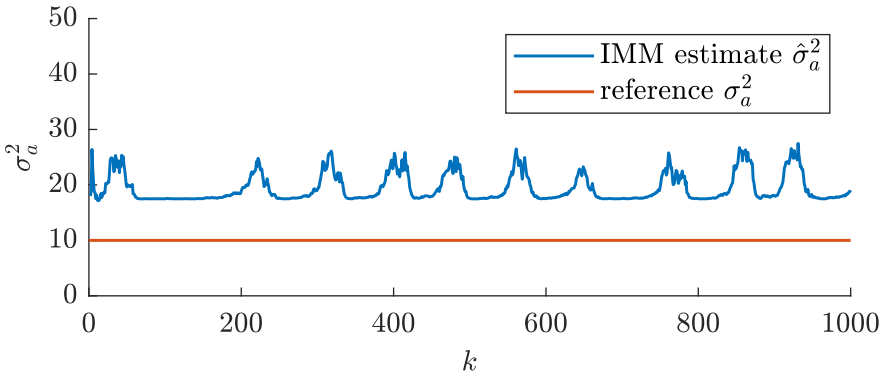


Figure 5.35: Estimating process noise using IMM with $\mathscr{M}$ defined in (5.20) and $\mathbf{\Pi}$ in (5.21).

See the result on Figure 5.35. Figure 5.35 shows that we have obtained poor results. We just have shown the critical problem of the IMM algorithm. It critically depends on the definition of the $\mathscr{M}$, and adding new models into it does not mean improving performance. In general, IMM is more suitable to estimate models with different geometry than process noise estimation. Nevertheless, it is derived to do it.

At the end of this section, we make a conclusion about using the IMM algorithm to estimate process noise. In general, we have not obtained good results. The estimated value of the process noise is often significantly higher than the true one. If we have chosen the "proper" model set $\mathscr{M}$, we have obtained good results. The author would like to notice that during the work with IMM, he observed that IMM estimates process noise correctly if possible values of $\sigma_a^2 \equiv \mathscr{M}$, in this case, are symmetric about the true (reference) value. But this is though since we do not know a priori the true value to construct the symmetric set of possible values of the process noise about that. By that, we consider the IMM algorithm to be not well suited to estimate process noise.

Remark. The author would like to mention that by setting

$$\mathbf{\Pi} = \mathbf{I}_{\#pn}$$

and enough dense set of the possible process noise values, e.g., (5.15) we obtain the same algorithm as the proposed Bayesian estimation algorithm described in Section 5.4 with the discretization of the process noise sample space defined by (5.15). However, setting identity as Markov transition matrix leads to division by zero in the mixing stage after some time steps. But the idea is we can see the proposed Bayesian estimation as some limit case of the IMM algorithm. Identity as Markov transition matrix provides that algorithm omit mixing stage and each state of the filter remains the same in prediction. Considering this fact and comparing Figures 5.32 and 5.20, we conclude that the mixing stage is the main cause of the poor performance of the IMM algorithm. As mentioned above, the models (process noise values) are taken into account, while in Bayesian estimation are ignored.

Discuss the opportunity to set a Markov probability transition matrix ($\mathbf{\Pi}$). Nevertheless, we never know that. We must guess it (use some experience from previous applications). This fact implies the idea of the application of these two algorithms. IMM algorithm at the beginning assumes a discrete set of motion models (usually not large since we do not know hundreds of different types of geometry trajectories and we have shown that increasing number of models does not necessarily mean better performance).

Then we consider the Markov transition probability matrix. We also take this assumption into account in state estimation. We assume that the actual state can be "between" of two model condition state estimation $\bar{\mathbf{x}}^i_{k|k}$ and $\bar{\mathbf{x}}^j_{k|k}$. Then we have to provide a weighted average to compute state estimation $\hat{\boldsymbol{x}}_k$.

On the other side, to provide the Bayesian estimation described in this chapter, we assume continuous sample space of process noise $\mathbf{Q}$. Then the Markov transition matrix is not considered since we cannot define it in continuous space. We provide the discretization only as one of the possible ways to compute intractable integral, and then we assume that the discretization is dense enough. Thus we believe we can find the true value of parameter $\mathbf{Q}$ and then provide state estimation based on this value.

5.7 | Non-constant Process Noise Problem

We have considered constant process noise during the test examples in this chapter. In this section, we test algorithms and then propose modifications to estimate the non-constant process noise of the target. So, reformulate the used test example. For the sake of brevity, consider only Bayesian estimation in this section. We can apply the used idea to the same principle on other algorithms.

Example 5.9. Consider the same data as in Example 5.3 with one exception.

$$\mathbf{Q}_k = \sigma_a^2 = \begin{cases} 15 \text{ m}^2/\text{s}^4 & k < 500 \\ 1 \text{ m}^2/\text{s}^4 & k \geq 500 \end{cases} \tag{5.22}$$

Run $\#MC = 100$ Monte Carlo runs to obtain more consistent view about the performance. Consider initial prior distribution

$$f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 15, 6).$$

See the result on Figure 5.36. Figure 5.36 shows that the Bayesian estimation moves to

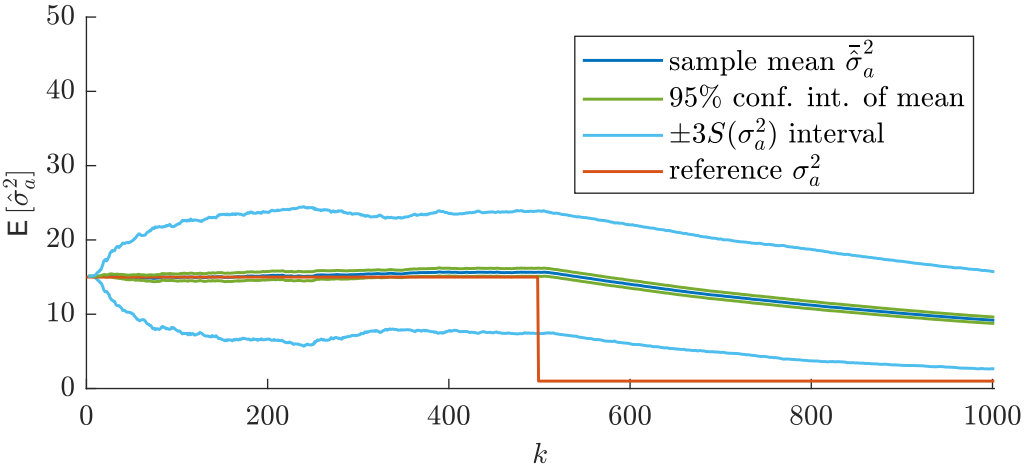


Figure 5.36: Estimation of the process noise by the Bayesian estimation with initial prior distribution $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 15, 6)$ throw $\#MC = 100$ Monte Carlo runs.

the true value after a dramatic change of the process noise very slow.

We propose the improvement to obtain better results. Introduce following notation:

$$\mathcal{Z}_k^h = \{\boldsymbol{z}_{k-h+1}, \ldots, \boldsymbol{z}_k\}.$$

Other words, h means history – number of last considered measurements. The idea is replace the estimation base on all measurements up to k: $\mathcal{Z}_k$ with only last h measurements. Such that provide following Bayesian estimation

$$f(\mathbf{Q}|\mathcal{Z}_k^h) \propto f(\mathcal{Z}_k^h|\mathbf{Q}) \cdot f(\mathbf{Q}) = \left(\prod_{i=k-h+1}^{k} f(\boldsymbol{z}_i|\mathbf{Q}, \mathcal{Z}_{i-1}^{h-i}) \right) \cdot f(\mathbf{Q}). \tag{5.23}$$

For practical reason, we consider following approximation

$$\left(\prod_{i=k-h+1}^{k} f(\boldsymbol{z}_i|\mathbf{Q}, \mathcal{Z}_{i-1}^{h-i}) \right) \cdot f(\mathbf{Q}) \approx \left(\prod_{i=k-h+1}^{k} f(\boldsymbol{z}_i|\mathbf{Q}, \mathcal{Z}_{i-1}^{h}) \right) \cdot f(\mathbf{Q}). \tag{5.24}$$

Equation (5.23) above is analogue to Proposition 5.5. We use approximation (5.24) to be able to store information and significantly reduce computational complexity. See proposed Algorithm 5.4: Bayesian estimation with abbreviated history.

Algorithm 5.2: Bayesian estimation of the process noise $\mathbf{Q}$ with abbreviated history

---

Parameters:

- history h: number of last considered measurements
- discretization of $\mathbf{Q}$'s sample space $\mathscr{Q} = \{\mathbf{Q}^j\}_{j=1}^{\#pn}$
- initial prior density $f(\mathbf{Q}) \rightarrow \mathsf{P}\{\mathbf{Q}^j\};\ j = 1, \ldots, \#pn$

Input from previous time step $k-1$:
prior density:

- $f(\boldsymbol{x}_{k-1}|\mathbf{Q}, \mathcal{Z}_{k-1}^h) \rightarrow \{\bar{\mathbf{x}}_{k-1|k-1}(\mathbf{Q}^j), \mathbf{P}_{k-1|k-1}(\mathbf{Q}^j)\}_{j=1}^{\#pn}$
- $f(\mathcal{Z}_{k-1}^h|\mathbf{Q}) \rightarrow \mathsf{N}(\boldsymbol{z}_i; \mathbf{H}_i\bar{\mathbf{x}}_{i|i}(\mathbf{Q}^j), \mathbf{S}_i(\mathbf{Q}^j));\ i = k-h+1,\ \ldots,\ k-1,\ j = 1, \ldots, \#pn$

Computation:

- run $\#pn$ Kalman filters for $\mathbf{Q}^j$, $j = 1, \ldots, \#pn$ with output $\bar{\mathbf{x}}_{k|k}(\mathbf{Q}^j), \mathbf{P}_{k|k}(\mathbf{Q}^j)$
- compute approximation of $f(\mathbf{Q}|\mathcal{Z}_k)$:

$$\begin{aligned}
\mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k^h\} &:= \mathsf{P}\{\mathcal{Z}_k^h|\mathbf{Q}\} \cdot \mathsf{P}\{\mathbf{Q}^j\} = \\
&\approx \left(\prod_{i=k-h+1}^{k} \mathsf{P}\{\boldsymbol{z}_i|\mathbf{Q}^j, \mathcal{Z}_{i-1}^h\} \right) \cdot \mathsf{P}\{\mathbf{Q}^j\} \\
&= \prod_{i=k-h+1}^{k} \Big(\mathsf{N}(\boldsymbol{z}_i; \mathbf{H}_i\bar{\mathbf{x}}_{i|i}(\mathbf{Q}^j), \mathbf{S}_i(\mathbf{Q}^j)) \Big) \cdot \mathsf{P}\{\mathbf{Q}^j\};\ j = 1, \ldots, \#pn
\end{aligned}$$

- normalize computed $\mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k^h\}$:

$$\mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k^h\} := \frac{\mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k^h\}}{\sum_{j=1}^{\#pn} \mathsf{P}\{\mathbf{Q}^j|\mathcal{Z}_k^h\}}$$

- Bayesian estimation of the process noise :

$$\hat{\mathbf{Q}} = \sum_{j=1}^{\#pn} \mathbf{Q}^j \mathsf{P}\{\mathbf{Q}^j | \mathcal{Z}_k^h\}$$

- State estimation $\hat{\boldsymbol{x}}_k$

$$\bar{\mathbf{x}}_{k|k}(\hat{\mathbf{Q}}), \mathbf{P}_{k|k}(\hat{\mathbf{Q}})$$

Output for next time step $k+1$:

- $\mathsf{N}(\boldsymbol{z}_i; \mathbf{H}_i \bar{\mathbf{x}}_{i|i}(\mathbf{Q}^j), \mathbf{S}_i(\mathbf{Q}^j));\ i = k-h+2,\ \ldots,\ k,\ j = 1, \ldots, \#pn$

Output for current state and process noise estimation at time step k:

- $\hat{\mathbf{Q}}, \bar{\mathbf{x}}_{k|k}(\hat{\mathbf{Q}}), \mathbf{P}_{k|k}(\hat{\mathbf{Q}})$

Remark. It is worth mentioning that this algorithm consumes much more memory and has higher computational complexity than other algorithm described above. We have to store in memory in each time step $h-1$ likelihoods for $\#pn$ discretized values of the process noise $\mathbf{Q}$, and during the computation, we have to compute the pointwise product of them. We can derive the same principle in maximum likelihood estimation algorithm with abbreviated history and improved maximum likelihood estimation with abbreviated history.

Now, try the algorithm on the example.

Example 5.10. Consider the same data as in Example 5.9. Use Bayesian estimation method with abbreviated history. Run $\#MC = 100$ Monte Carlo runs to obtain more consistent view about the performance. Consider initial prior distribution

$$f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 5, 10).$$

To compare history parameter h influence on the result, consider tree values: $h = 50, 100, 250$. See the result on Figures 5.37, 5.38, 5.39 and compare also with Figure 5.36.

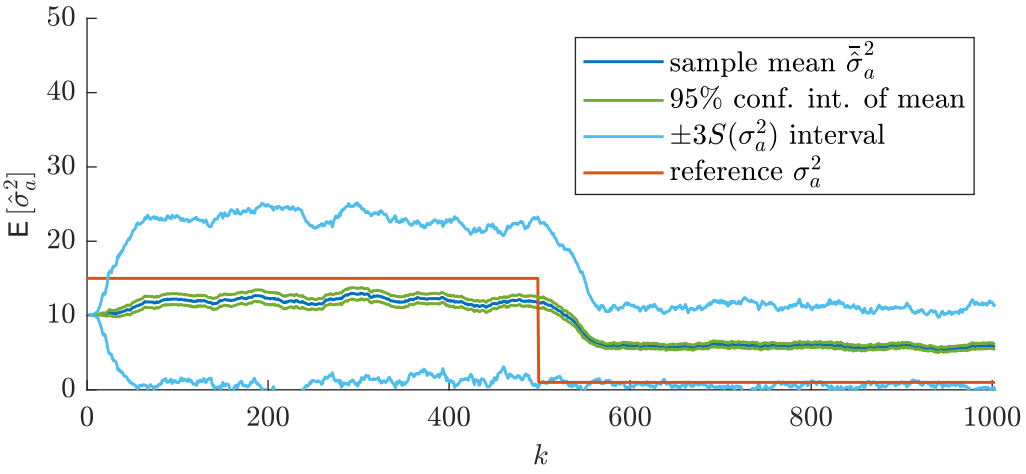


Figure 5.37: Estimation of the process noise by the Bayesian estimation with abbreviated history, $h = 50$ and initial prior distribution $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 5, 10)$ throw $\#MC = 100$ Monte Carlo runs.

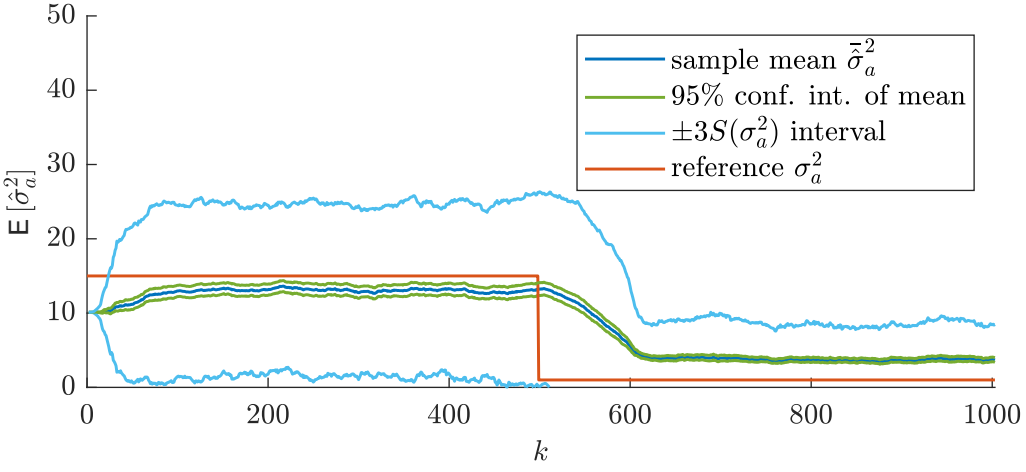


Figure 5.38: Estimation of the process noise by the Bayesian estimation with abbreviated history, $h = 100$ and initial prior distribution $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 5, 10)$ throw $\#MC = 100$ Monte Carlo runs.

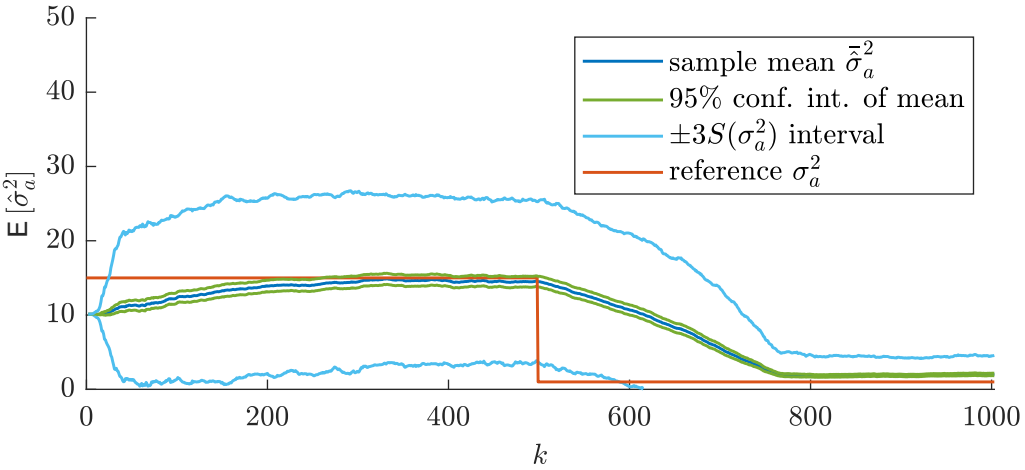


Figure 5.39: Estimation of the process noise by the Bayesian estimation with abbreviated history, $h = 250$ and initial prior distribution $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2; 5, 10)$ throw $\#MC = 100$ Monte Carlo runs.

On the figures above, we can see that higher history is estimation more precise but has slower adaptation on the changes. It makes sense since, with higher history, the estimation is more accurate because the algorithm has ‚more time' to ‚eliminate' potentially imprecise initial prior $f(\mathbf{Q})$. On the other side, more time intervals are included in the estimation; then, the algorithm slowly reacts to changes.

6 | Combination of the Multiple Model Approach and the Process Noise Estimation

In Chapter 4 we have considered improving Kalman filter estimation with the estimation of the model geometry. In Chapter 5 we have considered improving the Kalman filter with known motion model geometry but improving the hidden state estimation with unknown process noise estimation. In this chapter, we try to combine this problem. The idea is to have an estimator which estimates hidden state together with model geometry and process noise. We start with the exclusion of the described and derived IMM algorithm.

6.1 | Combination of IMM and Bayesian Estimation

The idea of this section is to take the IMM algorithm (Section 4.3). And exclude each considered motion model with adaptive estimation of the process noise $\mathbf{Q}$. The adaptive estimation of each model runs independently, and only process noise estimation at each time step is used to provide Kalman filter estimation of the model. Thus, the only difference between this proposed scheme and the original IMM algorithm is that we do not have a given process noise to models but provide its estimation at each time step for each model and use it in IMM instead of the given one. See the result in the following example.

Example 6.1. Consider the same data as in Example 4.7. Define IMM with two models. For each model we provide Bayesian estimation of the process noise, so define, instead of the constant process noise, the initial prior distribution.

- DWNA, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 1, 9)$;
- DWPA, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 1, 9)$,

set initial model probabilities

$$\mu_1^1 = 0,9; \qquad \mu_1^2 = 0,1$$

and Markov transition matrix

$$\mathbf{\Pi} = \begin{bmatrix} 0,99 & 0,01 \\ 0,01 & 0,99 \end{bmatrix}.$$

See the results on Figures 6.1 and 6.2.

Unfortunately, we did not obtain a good result in the sense of model estimation. Now, try to explain the cause. See the process noise estimation σ_a^2 on Figure 6.3.

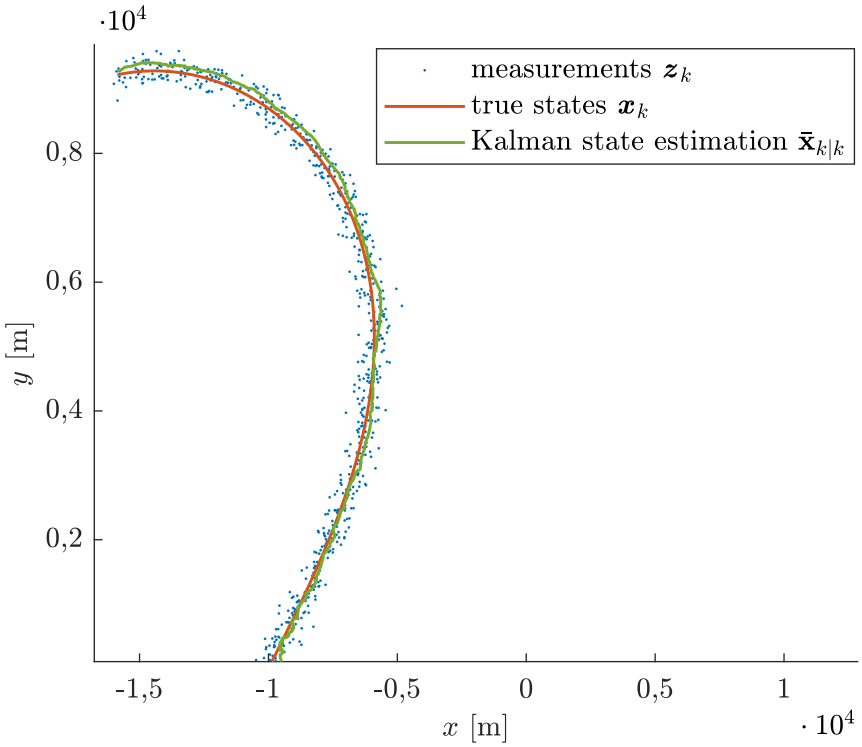


Figure 6.1: Result of IMM state estimation using IMM with process noise estimation in the motion models.

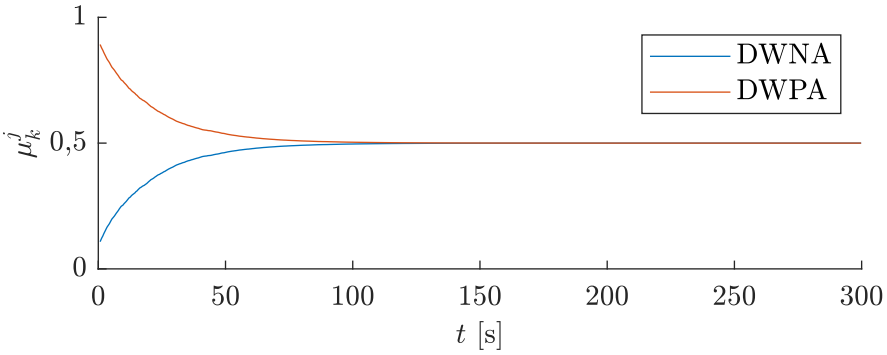
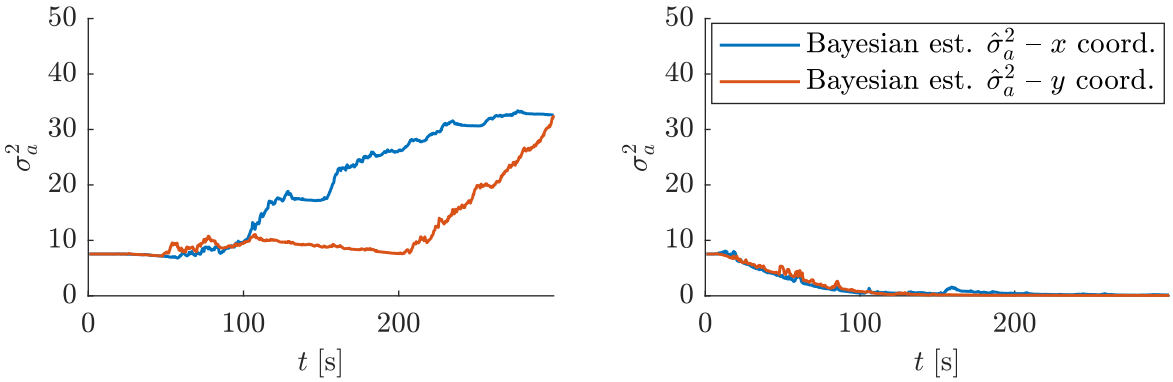


Figure 6.2: Model probabilities $\mu_k^j = p(m_k^j|\mathcal{Z}_k)$ with respect to time.



(a) DWNA model. **(b)** DWPA model.

Figure 6.3: Process noise σ_a^2 estimation.

The critical problem is that model estimation by IMM and process noise estimation are based on the same likelihood function (derived from a considered linear dynamic state-space system). Then both Bayesian analysis in the DWNA and DWPA model estimate process noise to provide hidden state estimation. IMM cannot decide which model is more suitable since they have similar likelihoods after using the estimated process noise.

In other words, Bayesian estimation of the process noise in the DWNA set the noise high enough to provide hidden state estimation with good results. Bayesian analysis in the DWPA decreases the process noise since the model suits trajectory well enough. But in the end, IMM cannot decide which is better.

Thus, unfortunately, we have to conclude that it is unsuitable to use a combination of the IMM algorithm and process noise estimation.

6.2 | Model Decision Based on the Process Noise Estimation

See again Figure 6.3. It shows that the model DWPA is more suitable than the DWNA model. Because it has significantly lower estimated process noise σ_a^2. The DWPA models' geometry better suits the target trajectory. So, use the idea that runs models independently, estimate process noise, and choose the most suitable one by some decision rule (e.g., the one with the lowest process noise).

Denote AKF - Adaptive Kalman filter, the Kalman filter with estimating process noise. Consider, as in IMM, discretized sample set of possible models $\mathscr{M} = \{m^j\}_{j=1}^{\#m}$. Run all $\#m$ models independently with process noise estimation. At each time step, obtain from each AKF state estimation and variance matrix $\bar{\mathbf{x}}_{k|k}^j, \mathbf{P}_{k|k}^j$ and process noise estimate $\hat{\mathbf{Q}}_k^j; j = 1, \ldots, \#m$. Then, by some predefined decision rule, from the set of process noise estimates $\{\hat{\mathbf{Q}}_k^j\}_{j=1}^{\#m}$ choose the most suitable model m^i by the decision rule. Then provide run Kalman filter with the chosen model and estimated process noise $\hat{\mathbf{Q}}_k^i$. We obtain output $\bar{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}$. See the proposed algorithm in the scheme in Figure 6.4.

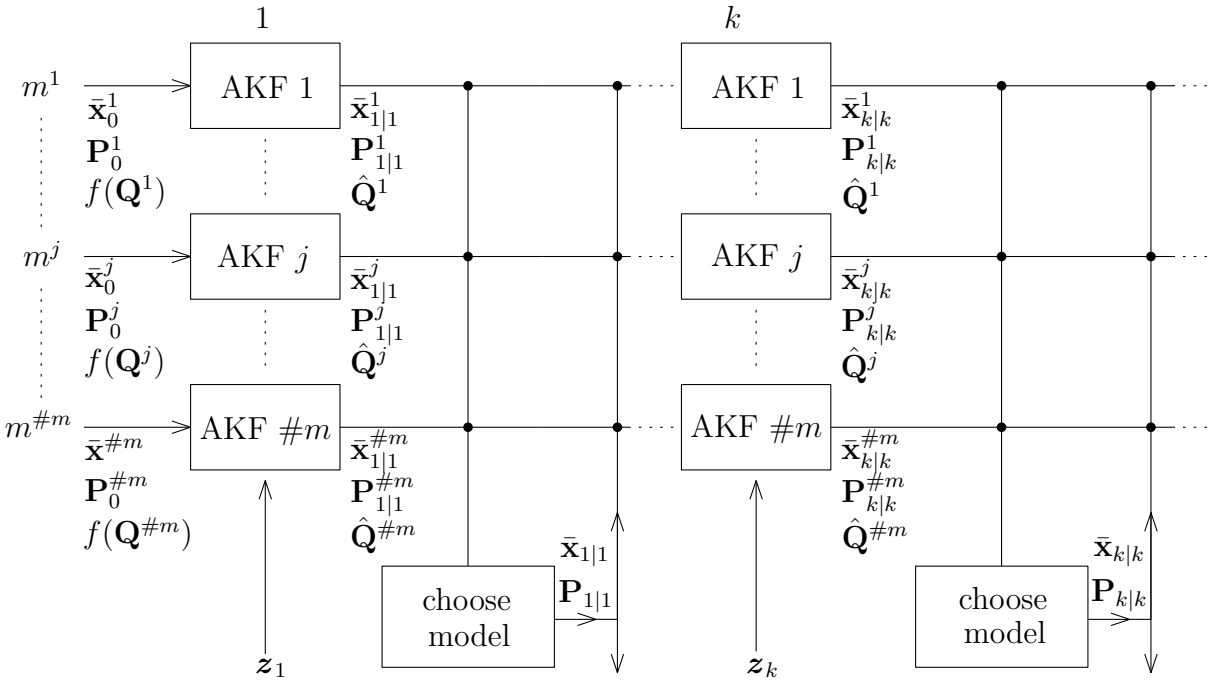


Figure 6.4: Scheme of the proposed algorithm model decision based on the process noise estimation

Example 6.2. Consider the same situation as in Example 6.1. Consider these $\#m = 3$ models and define initial prior for the process noise for each model.

- 1^{st} model: DWNA, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 1, 9)$;
- 2^{nd} model: DWPA, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 1, 9)$;
- 3^{rd} model: DWPJ, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 0, 01, 9)$

and define decision rule such that

- if $\hat{\mathbf{Q}}_k^1 < 1$ choose DWNA;
- else if $\hat{\mathbf{Q}}_k^2 < 1$ choose DWPA;
- else $\hat{\mathbf{Q}}_k^3 < 1$ choose DWPJ.

See the result on Figure 6.5. Estimation of the process noise for DWNA and DWPA models $(\mathbf{Q}^1, \mathbf{Q}^2)$ can be seen on Figure 6.3. The estimation of the process noise of the DWPJ model $(\mathbf{Q}^3)$ can be seen on Figure 6.6.

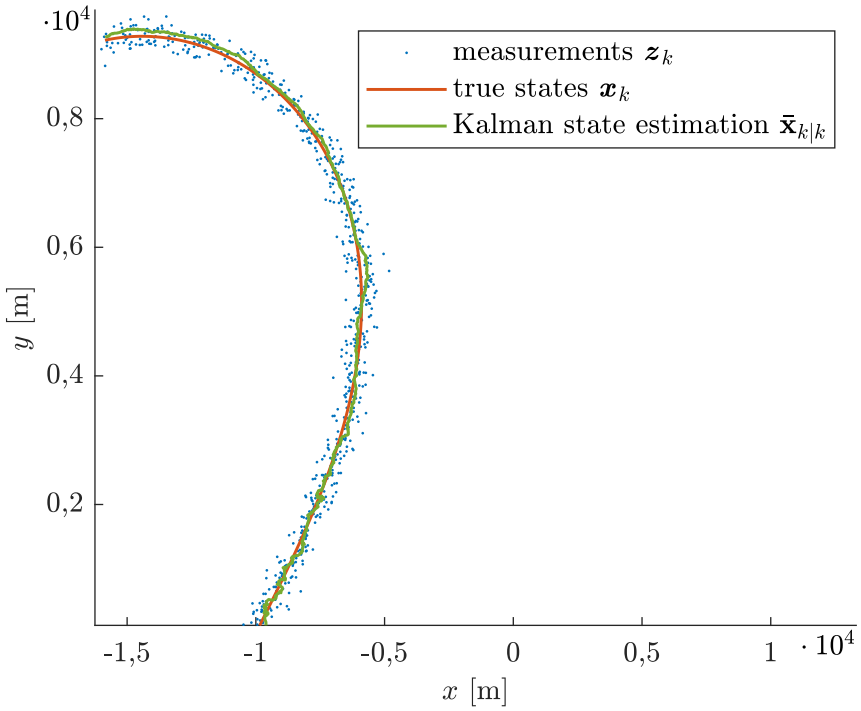


Figure 6.5: Scheme of the proposed algorithm model decision based on the process noise estimation

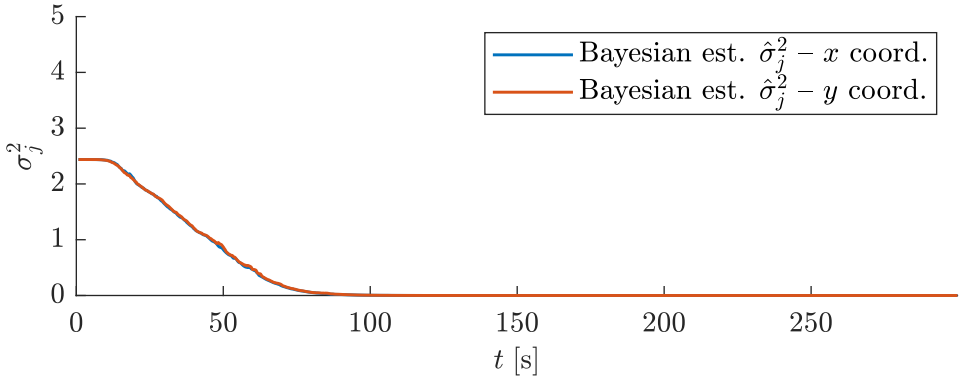


Figure 6.6: Scheme of the proposed algorithm model decision based on the process noise estimation

Example 6.3 (Broken line)**.** We consider this simulation called the 'broken line' to test how adaptive estimators reflect a dramatic change of target trajectory. It is evident that the trajectory is unrealistic, but we aim to test the marginal case of dramatic trajectory change. The measurements correspond to multilateration plots (see Chapter 2) with variance matrix

$$\mathbf{R}_k = \begin{bmatrix} 10^6 & 0 & 0 \\ 0 & 10^6 & 0 \\ 0 & 0 & 10^6 \end{bmatrix}$$

Consider these $\#m = 3$ models and define initial prior for the process noise for each model.

- 1^{st} model: DWNA, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 0, 001, 5)$;
- 2^{nd} model: DWPA, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 0, 001, 5)$;
- 3^{rd} model: DWPJ, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 0, 001, 5)$

and define decision rule such that

- if $\hat{\mathbf{Q}}_k^1 < 2$ choose DWNA;
- else if $\hat{\mathbf{Q}}_k^2 < 1$ choose DWPA;
- else $\hat{\mathbf{Q}}_k^3 < 1$ choose DWPJ.

To estimate process noise of the particular models, use Bayesian estimation of the process noise with abbreviated history. Figures 6.8, 6.9, 6.10 show the process noise estimation in the particular models for x and y coordinates.

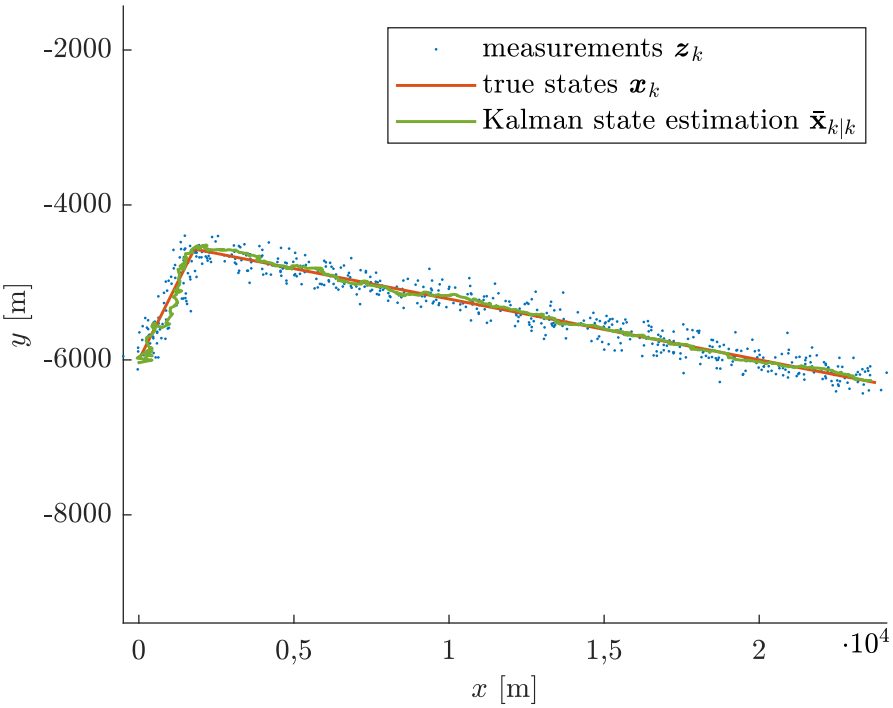


Figure 6.7: Estimating simulated ‚broken line' trajectory with given measurements by multilateration using model decision based on the process noise estimation.

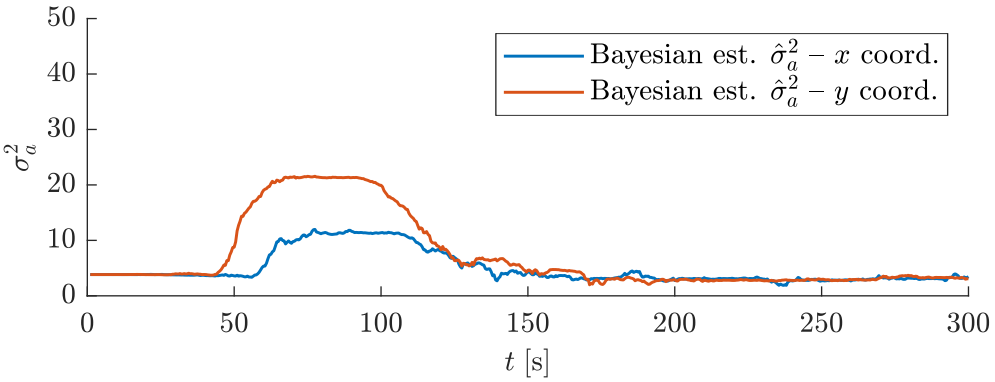


Figure 6.8: Bayesian estimation with abbreviated history $h = 100$ of the process noise in DWNA model.

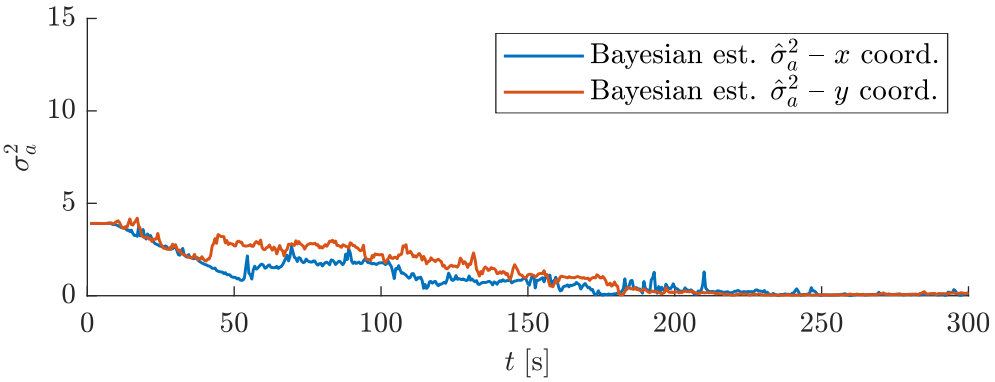


Figure 6.9: Bayesian estimation with abbreviated history $h = 100$ of the process noise in DWPA model.

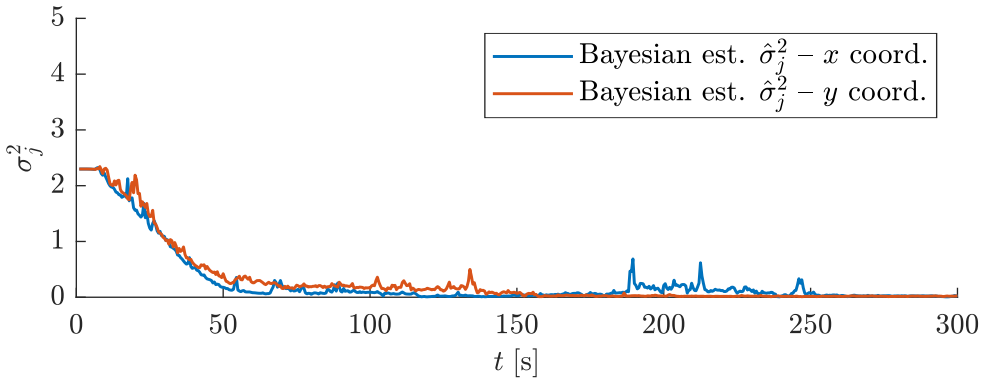


Figure 6.10: Bayesian estimation with abbreviated history $h = 100$ of the process noise in DWPJ model.

Despite the unrealistic trajectory, we can see that we managed to remove measurement uncertainty and provided good hidden state adaptive estimation results.

7 | Application of Adaptive Estimation to Multilateration Measurements

At the end of the thesis, try to use developed algorithm in Section 6.2 on multilateration measurements of aircraft trajectory. Consider simulated data but corresponding to real measurements data. The trajectory is simulated. It is corresponding to highly maneuvering acrobatic airplane. It is maneuvering on the edge of the real possibilities of the airplane. The measurements corresponds to multilateration plots (see Chapter 2) with variance matrix

$$\mathbf{R}_k = \begin{bmatrix} 10^6 & 0 & 0 \\ 0 & 10^6 & 0 \\ 0 & 0 & 10^6 \end{bmatrix}$$

Consider these $\#m = 3$ models and define initial prior for the process noise for each model.

- 1^{st} model: DWNA, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 0, 001, 5)$;
- 2^{nd} model: DWPA, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 0, 001, 5)$;
- 3^{rd} model: DWPJ, $f(\sigma_a^2) = \mathsf{N}(\sigma_a^2, 0, 001, 5)$

and define decision rule such that

- if $\hat{\mathbf{Q}}_k^1 < 2$ choose DWNA;
- else if $\hat{\mathbf{Q}}_k^2 < 1$ choose DWPA;
- else $\hat{\mathbf{Q}}_k^3 < 1$ choose DWPJ.

To estimate the process noise of the particular models, use Bayesian estimation of the process noise with abbreviated history $h = 100$.

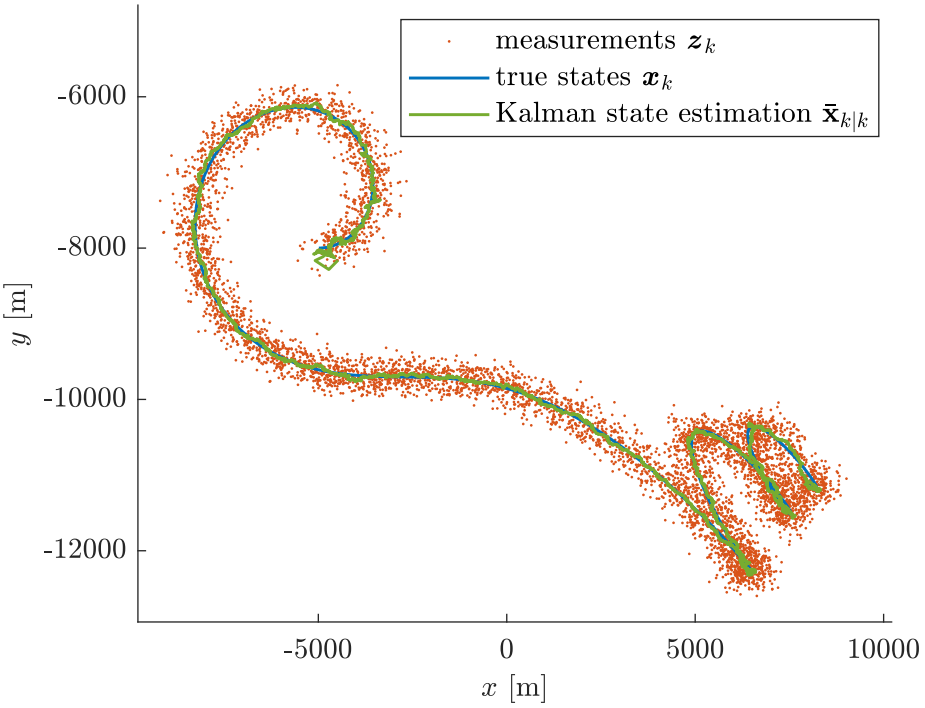


Figure 7.1: Estimating simulated ‚spiral' trajectory with given measurements by multilateration.

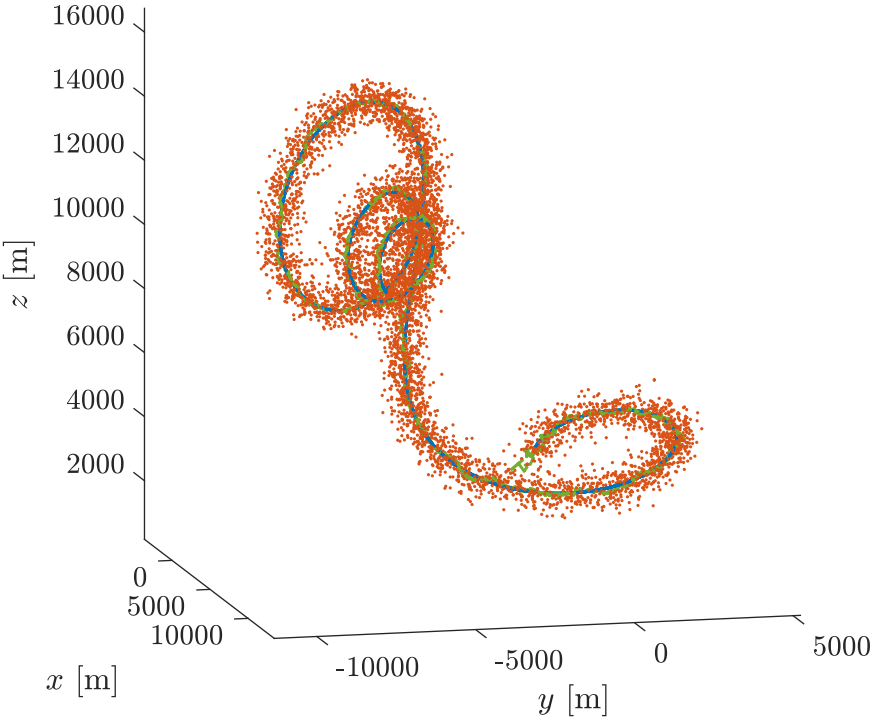


Figure 7.2: Estimating simulated ‚spiral' trajectory with given measurements by multilateration.

Now, see Figures 7.3, 7.4, 7.5 to obtain imagination about process noise estimation in particular models.

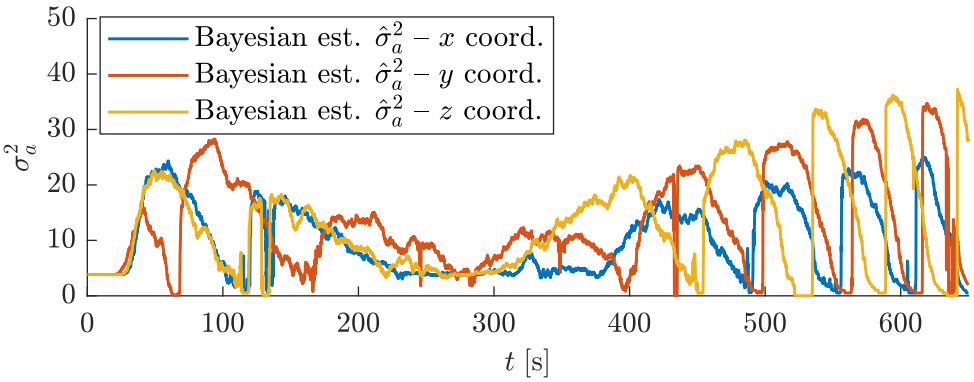


Figure 7.3: Bayesian estimation with abbreviated history $h = 100$ of the process noise in DWNA model.

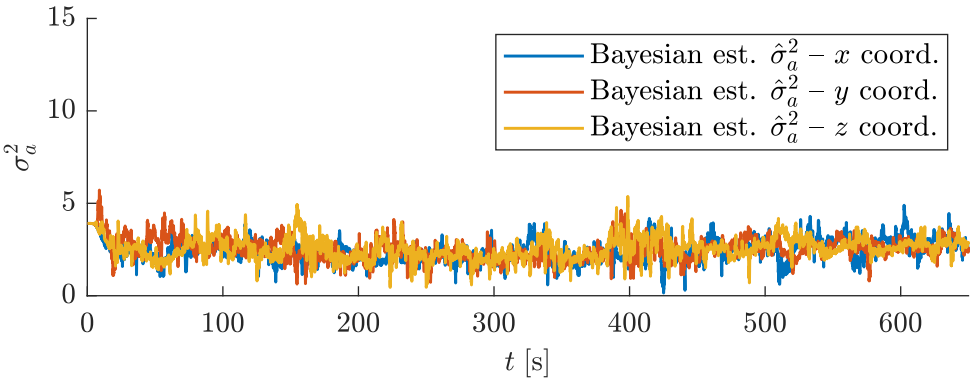


Figure 7.4: Bayesian estimation with abbreviated history $h = 100$ of the process noise in DWPA model.

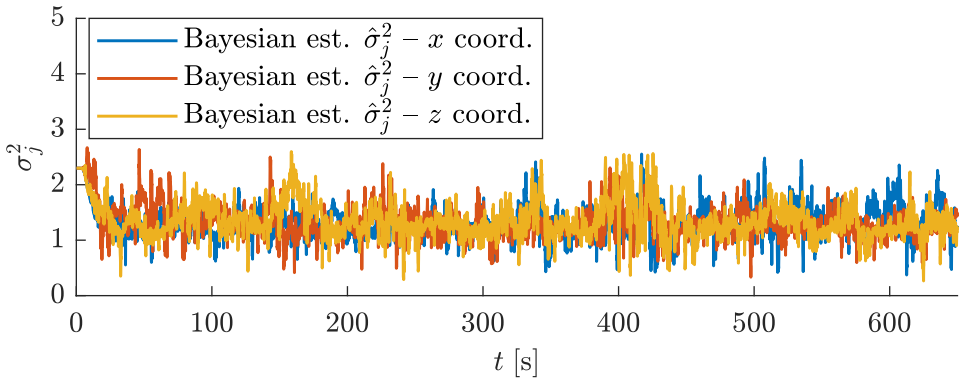


Figure 7.5: Bayesian estimation with abbreviated history $h = 100$ of the process noise in DWPJ model.

We can conclude that we have obtained perfect results. The process noise in the DWNA model has reacted the most from all models.

8 | Conclusion

This text deal with target tracking from multilateration measurements. In the beginning, we have described multilateration and its geometrical interpretation since the author has considered it as interesting from a mathematical point of view. It has been mentioned the motivation throw all text. Multilateration measurements are noisy, i.e., they are inaccurate, and we have aimed to remove the uncertainty from the measurements. We can do this because we have many measurements since the positioning system often does it (e.g., every 0,1 second). However, the crucial problem remains that target moves between the measurements.

In the second chapter, the Kalman filter was derived. It is a very known and used algorithm for such a purpose. We have shown that the Kalman filter works pretty well, but its parameters have to be appropriately chosen. We have divided these parameters into two groups - geometrical parameters defining the geometry of the trajectory and uncertainty parameter - process noise representing how much the geometry parameter would be wrong.

Two ways to improve the Kalman filter have been considered. Estimation of the geometry and estimation of the process noise. We have both described. As estimation of the geometry, we have considered the IMM algorithm. This algorithm is mostly used in real target tracking. Its disadvantage is that we have to choose a proper geometry model (motion model) and uncertainty of each geometry - process noise. This is crucial, and the author would like to mention that it was strange during the implementation. It has taken time to find proper values of process noise to some considered set of multilateration measurements. However, in case we managed to find it we have obtained a good result.

After that it adaptive filtering was studied. We have considered one geometry, but together with state, we have estimated process noise. In literature, this is studied for the general case and are known four basic principles in order to do that: correlation, covariance-matching, maximum-likelihood, and Bayesian methods. We have described all four but states that only the last two are suitable for multilateration measurements. During the implementation, we found that high measurement noise makes the first two unworkable.

Then, we have developed, described, and tested maximum-likelihood and Bayesian estimation methods. We have also introduced improvements of the Bayesian and maximum likelihood method to estimate process noise. We obtained good results also in non-constant process noise estimation.

At the end of the thesis, we have discussed the combination of geometry and process noise estimation. Unfortunately, combination with IMM was not possible. We have introduced another one standing on the principle of comparing estimated process noise of particular models. We applied this to simulated measurement data corresponding to the real one.

A | Used motion models

In this appendix are listed used motion models (other words Kalman filter set up) in the text. More details and more models can be found in [17]. Consider the time difference between actual and last time epoch

$$\Delta t_k = t_k - t_{k-1}.$$

DISCRETE WHITE NOISE ACCELERATION (DWNA)

This model is so-called 1D motion, because it describes motion in one dimension. Thus we define it for one dimension and in practical application it would be used also other dimensions in a similar way.

$$\mathbf{F}_k = \begin{bmatrix} 1 & \Delta t_k \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{\Gamma}_k = \begin{bmatrix} \Delta t_k^2/2 \\ \Delta t_k \end{bmatrix}, \qquad \mathbf{Q}_k = \sigma_a^2, \qquad \mathbf{H}_k = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Initial estimate at $k = 0$ is made from two time epochs which we denote $k = 0$ and $k = -1$.

$$\bar{\mathbf{x}}_0 = \begin{bmatrix} \boldsymbol{z}_0 \\ (\boldsymbol{z}_0 - \boldsymbol{z}_{-1})/\Delta t_0 \end{bmatrix}, \qquad \mathbf{P}_0 = \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_0/\Delta t_0 \\ \mathbf{R}_0/\Delta t_0 & (\mathbf{R}_0 + \mathbf{R}_{-1})/t^2 + \sigma_a^2/4 \cdot t^2 \end{bmatrix}$$

Dimensions of the model are equal to

$$n = 2 \qquad m = 1 \qquad p = 0 \qquad q = 1.$$

Parameter of the model is variance of acceleration σ_a^2.

DISCRETE WIENER PROCESS ACCELERATION (DWPA)

This model is so-called 1D motion, because it describes motion in one dimension. Thus we define it for one dimension and in practical application it would be used also other dimensions in a similar way.

$$\mathbf{F}_k = \begin{bmatrix} 1 & \Delta t_k & \Delta t_k^2/2 \\ 0 & 1 & \Delta t_k \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{\Gamma}_k = \begin{bmatrix} \Delta t_k^2/2 \\ \Delta t_k \\ 1 \end{bmatrix}, \qquad \mathbf{Q}_k = \sigma_a^2, \qquad \mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Initial estimate at $k = 0$ is made from two time epochs which we denote $k = 0$ and $k = -1$.

$$\bar{\mathbf{x}}_0 = \begin{bmatrix} \boldsymbol{z}_0 \\ (\boldsymbol{z}_0 - \boldsymbol{z}_{-1})/\Delta t_0 \\ 0 \end{bmatrix}, \qquad \mathbf{P}_0 = \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_0/\Delta t_0 & 0 \\ \mathbf{R}_0/\Delta t_0 & (\mathbf{R}_0 + \mathbf{R}_{-1})/t^2 + \sigma_a^2/4 \cdot t^2 & \Delta t_0 \sigma_a^2 \\ 0 & \Delta t_0 \sigma_a^2 & \sigma_a^2 \end{bmatrix}$$

Dimensions of the model are equal to

$$n = 3 \qquad m = 1 \qquad p = 0 \qquad q = 1.$$

Parameter of the model is variance of acceleration σ_a^2.

DISCRETE WIENER PROCESS JERK (DWPJ)

This model is so-called 1D motion, because it describes motion in one dimension. Thus we define it for one dimension and in practical application it would be used also other dimensions in a similar way.

$$\mathbf{F}_k = \begin{bmatrix} 1 & \Delta t_k & \Delta t_k^2/2 & \Delta t_k^3/6 \\ 0 & 1 & \Delta t_k & \Delta t_k^2/2 \\ 0 & 0 & 1 & \Delta t_k \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{\Gamma}_k = \begin{bmatrix} \Delta t_k^3/6 \\ \Delta t_k^2/2 \\ \Delta t_k \\ 1 \end{bmatrix}, \qquad \mathbf{Q}_k = \sigma_j^2, \qquad \mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

Initial estimate at $k = 0$ is made from two time epochs which we denote $k = 0$ and $k = -1$.

$$\bar{\mathbf{x}}_0 = \begin{bmatrix} \boldsymbol{z}_0 \\ (\boldsymbol{z}_0 - \boldsymbol{z}_{-1})/\Delta t_0 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{P}_0 = \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_0/\Delta t_0 & 0 & 0 \\ \mathbf{R}_0/\Delta t_0 & (\mathbf{R}_0 + \mathbf{R}_{-1})/\Delta t_0^2 + 3/4 \cdot \Delta t_0^2 \sigma_j^2 & (\Delta t_0/2 + \Delta t_0^3/3 + \Delta t_0^5/3)\sigma_j^2 & (\Delta t_0^4/3 + \Delta t_0^2/3)\sigma_j^2 \\ 0 & (\Delta t_0/2 + \Delta t_0^3/3 + \Delta t_0^5/3)\sigma_j^2 & (1 + 2\Delta t_0^2)\sigma_j^2 & 2\Delta t_0 \sigma_j^2 \\ 0 & (\Delta t_0^4/3 + \Delta t_0^2/3)\sigma_j^2 & 2\Delta t_0 \sigma_j^2 & (1 + \Delta t_0^2)\sigma_j^2 \end{bmatrix}$$

Dimensions of the model are equal to

$$n = 4 \qquad m = 1 \qquad p = 0 \qquad q = 1.$$

Parameter of the model is variance of jerk σ_j^2.

NEARLY CONSTANT TURN RATE (NCTR)

This is so-called 2D motion model, because it described motion in plane. Thus we define it in two dimensions (in xy plane). It can used in other planes (xz and yz), but is impractical. Consider state vector

$$\boldsymbol{X}_k = \begin{bmatrix} X_k & \dot{X}_k & Y_k & \dot{Y}_k & \Omega_k \end{bmatrix}^\top,$$

where Ω is angular velocity about z axis with center in target.

$$\mathscr{F}_k(\boldsymbol{X}_k, \mathbf{u}_k) = \mathscr{F}_k(\boldsymbol{X}_k) = \begin{bmatrix} X_{k-1} + \frac{\sin(\Omega_{k-1}\Delta t_k)}{\Omega_{k-1}}\dot{X}_{k-1} - \frac{1-\cos(\Omega_{k-1}\Delta t_k)}{\Omega_{k-1}}\dot{Y}_{k-1} \\ \frac{1-\cos(\Omega_{k-1}\Delta t_k)}{\Omega_{k-1}}\dot{X}_{k-1} + Y_{k-1} + \frac{\sin(\Omega_{k-1}\Delta t_k)}{\Omega_{k-1}}\dot{Y}_{k-1} \\ \cos(\Omega_k \Delta t_k)X_{k-1} - \sin(\Omega_k \Delta t_k)Y_{k-1} \\ \sin(\Omega_k \Delta t_k)X_{k-1} + \cos(\Omega_k \Delta t_k)Y_{k-1} \\ \Omega_{k-1} \end{bmatrix}$$

$$\mathbf{\Gamma}_k = \begin{bmatrix} \Delta t_k^2/2 & 0 \\ \Delta t_k^2/2 & 0 \\ \Delta t_k & 0 \\ \Delta t_k & 0 \\ 0 & \Delta t_k \end{bmatrix}, \qquad \mathbf{Q}_k \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_\alpha^2 \end{bmatrix}, \qquad \mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Initial estimate at $k = 0$ is made from two time epochs which we denote $k = 0$ and $k = -1$. Consider measurement vector and measurement variance $\mathbf{R}_k$ such that

$$\boldsymbol{z}_k = \begin{bmatrix} z_k^x \\ x_k^y \end{bmatrix}, \qquad \mathbf{R}_k = \begin{bmatrix} r_k^{xx} & r_k^{xy} \\ r_k^{yx} & r_k^{yy} \end{bmatrix}$$

.

$$\bar{\mathbf{x}}_0 = \begin{bmatrix} z_0^x \\ (z_0^x - z_{-1}^x)/\Delta t_0 \\ z_0^y \\ (z_0^y - z_{-1}^y)/\Delta t_0 \\ 0 \end{bmatrix},$$

$$\mathbf{P}_0 = \begin{bmatrix} r_0^{xx} & r_0^{xx}/\Delta t_0 & 0 & 0 & 0 \\ r_0^{xx}/\Delta t_0 & (r_0^{xx} + r_{-1}^{xx})/t^2 + \sigma_a^2/4 \cdot t^2 & 0 & 0 & 0 \\ 0 & 0 & r_0^{yy} & r_0^{yy}/\Delta t_0 & 0 \\ 0 & 0 & r_0^{yy}/\Delta t_0 & (r_0^{yy} + r_{-1}^{yy})/t^2 + \sigma_a^2/4 \cdot t^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\alpha^2 \end{bmatrix}.$$

Recall that due to transition model is nonlinear we have use Extended Kalman filter [15] instead of Kalman filter 3.

Parameter of the model is variance of acceleration σ_a^2 and angular acceleration σ_α^2.

B | Used probability distributions

In this appendix are listed distribution used in the text. See [39] and [9].

DISCRETE UNIFORM DISTRIBUTION

We say discrete random variable X has uniform distribution, denote $X \sim \mathsf{UD}(\mathcal{K})$, where $\mathcal{K}$ is set considered as parameter, iff it has probability function

$$p(x) = \begin{cases} 1/m; & x \in \mathcal{K} \\ 0; & \text{otherwise.} \end{cases} =: \mathsf{UD}(x; \mathcal{K})$$

where $m := \operatorname{card} \mathcal{K}$. Mean and variance are

$$\mathsf{E}[X] = \frac{m+1}{2},$$
$$\mathsf{D}[X] = \frac{m^2-1}{12}.$$

UNIFORM DISTRIBUTION

We say continuous random variable X has uniform distribution, denote $X \sim \mathsf{U}(a,b)$, where parameters are $a, b \in \mathbb{R}; a < b$, iff it has probability density function

$$f(x) = \begin{cases} \frac{1}{b-a}; & x \in \langle a; b \rangle, \\ 0; & \text{otherwise.} \end{cases} =: \mathsf{U}(x; a, b)$$

with mean and variance

$$\mathsf{E}[X] = \frac{a+b}{2},$$
$$\mathsf{D}[X] = \frac{(b-a)^2}{12}.$$

NORMAL DISTRIBUTION

We say continuous random vector $\boldsymbol{X}$ has uniform distribution, denote $\boldsymbol{X} \sim \mathsf{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where parameters are $\boldsymbol{\mu} \in \mathbb{R}^n, \boldsymbol{\Sigma} \in PD_n(\mathbb{R})$, iff it has probability density function

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^k \det(\Sigma)}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) =: \mathsf{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with vector mean and variance matrix

$$\mathsf{E}[\boldsymbol{X}] = \boldsymbol{\mu},$$
$$\mathsf{D}[\boldsymbol{X}] = \boldsymbol{\Sigma}$$

GAUSSIAN MIXTURE DISTRIBUTION

We say continuous random variable $\boldsymbol{X}$ has Gaussian mixture distribution, denote $\boldsymbol{X} \sim \mathsf{GM}(\{\boldsymbol{\mu}^i\}_{i=1}^s, \{\boldsymbol{\Sigma}^i\}_{i=1}^s, \{\alpha_i\}_{i=1}^s)$, where parameters are $\boldsymbol{\mu}^i \in \mathbb{R}^n, \boldsymbol{\Sigma}^i \in PD_n(\mathbb{R}), \alpha_i \in (0,1)$ such that

$$\sum_{i=1}^{s} \alpha_i = 1,$$

iff it has probability density function

$$f(\boldsymbol{x}) = \sum_{i=1}^{s} \alpha_i \cdot \mathsf{N}(\boldsymbol{x}; \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i),$$

with vector mean and variance matrix

$$\mathsf{E}[\boldsymbol{X}] = \sum_{i=1}^{s} \alpha_i \boldsymbol{\mu}^i =: \boldsymbol{\mu}, \tag{B.1}$$

$$\mathsf{D}[\boldsymbol{X}] = \sum_{i=1}^{s} \alpha_i \left[\boldsymbol{\Sigma}^i + (\boldsymbol{\mu}^i - \boldsymbol{\mu}) \cdot (\boldsymbol{\mu}^i - \boldsymbol{\mu})^\top\right]. \tag{B.2}$$

See Proposition 4.6.

STUDENT'S DISTRIBUTION

We say continuous random variable X has student's distribution, denote $X \sim \mathsf{t}(n)$ with parameter degree of freedom $n \in \langle 1; \infty)$, iff it has probability density function

$$f(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)\sqrt{\pi n}} \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}} =: \mathsf{t}(x; n)$$

with mean and variance

$$\mathsf{E}[X] = 0; n > 1,$$
$$\mathsf{D}[X] = \frac{n}{n-2}; n > 2.$$

Bibliography

[1] ANDĚL, Jiří. *Základy matematické statistiky.* 3. opr. vyd. Praha: Matfyzpress, 2011. ISBN 978-80-7378-001-2

[2] ARULAMPALAM M. S., MASKELL S., GORDON N. and CLAPP T., *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*, in IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174-188, Feb. 2002, doi: 10.1109/78.978374.

[3] BENKO, M. *Comparison of filters in target tracking.* Brno: Brno University of Technology, Faculty of Mechanical engineering, 2019. 42 pages. Supervisor doc. RNDr. Libor Žák, Ph.D.

[4] BAR-SHALOM, Yaakov, LI, X.-Rong and KIRUBARAJAN, Thia. *Estimation with Applications to Tracking and Navigation.* [New York]: Wiley 2001. ISBN: 9780471416555

[5] BAR-SHALOM, Y., CHALLA, S. and BLOM H. A. P., *IMM estimator versus optimal estimator for hybrid systems.* IEEE Transactions on Aerospace and Electronic Systems, **41**(3), 86-991, 2005, 10.1109/TAES.2005.1541443.

[6] BISHOP, Christopher M. *Pattern recognition and machine learning.* [New York]: Springer, 2006. Information science and statistics. ISBN 03-873-1073-8.

[7] CONGDON, Peter. *Applied Bayesian modelling.* 2nd ed. Wiley, 2014. ISBN 978-1119951513

[8] DUNÍK, J., STRAKA, O., KOST, O. and Havlík, J., *Noise covariance matrices in state-space models: A survey and comparison of estimation methods—Part I.* International Journal of Adaptive Control and Signal Processing, 31, 1505-1543, 2017.

[9] FRÜHWIRTH-SCHNATTER, Sylvia. *Finite Mixture and Markov Switching Models*, Springer, 2006. ISBN 978-1-4419-2194-9

[10] GRANSTRÖM., K, WILLETT, P. and BAR-SHALOM Y. *Systematic approach to IMM mixing for unequal dimension states.* IEEE Transactions on Aerospace and Electronic Systems, **51**(4), 2975-2986, 2015. 10.1109/TAES.2015.150015.

[11] JULIER, S.J.; UHLMANN, J.K. (2004). *Unscented filtering and nonlinear estimation.* Proceedings of the IEEE. **92** (3): 401–422. doi:10.1109/jproc.2003.823141.

[12] KAILATH, T. *An innovations approach to least-squares estimation–Part I: Linear filtering in additive white noise.* IEEE Transactions on Automatic Control, **13**(6), 646 – 655, December 1968, 10.1109/TAC.1968.1099025.

[13] KALMAN, R. E. *A New Approach to Linear Filtering and Prediction Problems.* Journal of Basic Engineering. 1960, **82**(1), 35-45. ISSN 0021-9223. DOI:10.1115/1.3662552.

[14] KAY, Steven M. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory.* Prentice Hall, 1993. ISBN 0-13-345711-7.

[15] KHAN, Rafiullah, KHAN, Sarmad Ullah, KHAN, Shahid and KHAN, Mohammad Usman. *Localization Performance Evaluation of Extended Kalman Filter in Wireless Sensors Network.* Procedia Computer Science, **32**, 2014, 117-124. 10.1016/j.procs.2014.05.405.

[16] KULLBACK, S., *Letter to the Editor: The Kullback–Leibler distance.* The American Statistician. **41** (4), 340–341.10.1080/00031305.1987.10475510.

[17] LI X. Rong and JILKOV V. P. *Survey of maneuvering target tracking. Part I. Dynamic models.* IEEE Transactions on Aerospace and Electronic Systems, **39**(4), 1333-1364, 2003. 10.1109/TAES.2003.1261132.

[18] LI X., LIU Y., MIHAYLOVA L., YANG L., WEDDELL S. and GUO F., *Enhanced Multiple Model GPB2 Filtering Using Variational Inference*, 2019 22th International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, 2019, 1-9.

[19] LI, Q., LI, R., JI, K. and DAI, W., *Kalman Filter and Its Application*, 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), Tianjin, China, 2015, 74-77, doi: 10.1109/ICINIS.2015.35.

[20] LI, Xiao-Rong and BAR-SHALOM, Y., *Multiple-model estimation with variable structure*, IEEE Transactions on Automatic Control, **41**(4), 478-493, April 1996, doi: 10.1109/9.489270.

[21] LI, Xiao-Rong, *Multiple-model estimation with variable structure. II. Model-set adaptation*, IEEE Transactions on Automatic Control, **45**(11), 2047-2060, Nov. 2000, doi: 10.1109/9.887626.

[22] LI Xiao-Rong, ZHANG, Youmin and ZHI, Xiaorong, *Multiple-model estimation with variable structure. III. Model-group switching algorithm*, Proceedings of the 36th IEEE Conference on Decision and Control, San Diego, CA, USA, 1997, 3114-3119, 4, doi: 10.1109/CDC.1997.652320.

[23] LI Xiao-Rong, ZHANG, Youmin and ZHI, Xiaorong, *Multiple-model estimation with variable structure. IV. Design and evaluation of model-group switching algorithm*, IEEE Transactions on Aerospace and Electronic Systems, **35**(1), pp. 242-254, Jan. 1999, doi: 10.1109/7.745695.

[24] LI, X. and ZHANG, Youmin, *Multiple-model estimation with variable structure. V. Likely-model set algorithm.* Aerospace and Electronic Systems, IEEE Transactions, 36, 448 - 466, 2000, 10.1109/7.845222.

[25] LI, X., JILKOV, Vesselin and RU, Jifeng, *Multiple-model estimation with variable structure - Part VI: Expected-mode augmentation.* Aerospace and Electronic Systems, IEEE Transactions, 41, 853- 867, 2005, 10.1109/TAES.2005.1541435.

[26] MAHLER, Ronald P. S. *Statistical Multisource-Multitarget Information Fusion.* Norwood: Artech house, 2007. ISBN 13: 978-1-59693-092-6.

[27] MANTILLA-GAVIRIA, Ivan A., Mauro LEONARDI, Gaspare GALATI and Juan V. BALBASTRE-TEJEDOR. Localization algorithms for multilateration (MLAT) systems in airport surface surveillance. *Signal, Image and Video Processing.* 2015, **9**, 1549–1558.

[28] MATISKO Peter, HAVLENA Vladimír, Cramér-Rao Bounds for Estimation of Linear System Noise Covariances, Journal of Mechanical Engineering and Automation, **2**(2), 2012, pp. 6-11. 10.5923/j.jmea.20120202.02.

[29] MAZOR, E., AVERBUCH, A., BAR-SHALOM, Yaakov and DAYAN, Joshua. *Interacting multiple model methods in target tracking: A survey.* Aerospace and Electronic Systems, IEEE Transactions on. 34, 103 - 123, 1998, 10.1109/7.640267.

[30] MEHRA, R. K., *Approaches to adaptive filtering.* IEEE Symposium on Adaptive Processes (9th) Decision and Control, Austin, Texas, USA, 1970, 141-141. 10.1109/SAP.1970.269992.

[31] MEHRA, R. K., *On the identification of variances and adaptive Kalman filtering.* IEEE Transactions on Automatic Control, **15**(2), 175-184, 1970. 10.1109/TAC.1970.1099422.

[32] PAPOULIS, A. *Probability, Random Variables, and Stochastic Processes.* 2nd ed. New York: McGraw-Hill, 1984.

[33] PENROSE, Roger. *A generalized inverse for matrices.* Proceedings of the Cambridge Philosophical Society. **51** (3), 406–13,1955. 10.1017/S0305004100030401.

[34] POZRIKIDIS, C. *An Introduction to Grids, Graphs, and Networks.* Oxford University Press, 2014. ISBN 9780199996735.

[35] QUAZI, A. H., *An Overview on the Time Delay Estimation in Active and Passive Systems for Target Localization.*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-29, No. 3 pp. 527–533, June 1981.

[36] RYAN, Pitre, JILKOV, Vesselin and LI, X. *A comparative study of multiple-model algorithms for maneuvering target tracking.* Proceedings of SPIE - The International Society for Optical Engineering, 2005, 5809. 10.1117/12.609681.

[37] SALMOND, D. J. *Tracking in Uncertain Environments.* Farnborough, Hampshire: Royal Aerospace Establishment, Ministry of Defence, 1989, 282.

[38] SCHWEPPE, F. *Evaluation of likelihood functions for Gaussian signals.*, IEEE Transactions on Information Theory,**11**(1), 61-70, January 1965, 10.1109/TIT.1965.1053737.

[39] THOMOPOULOS, Nick T. *Statistical Distributions: Applications and Parameter Estimates.* Springer, 2017. ISBN 978-3-319-65112-5.

[40] WONG, S., R. JASSEMI-ZARGANI, D. BROOKES and B. KIM. *A Geometric Approach to Passive Target Localization.* S&T Organization, North Atlantic Treaty Organization, 1-18.

[41] WOODBURY, Max A. *Inverting modified matrices*. Memorandum Rept. 42, Statistical Research Group, Princeton University, Princeton, New Jersey, 1950.

[42] XU, Hong, DUAN, Keqing, YUAN Huadong, XIE, Wenchong, WANG, Yongliang. *Black box variational inference to adaptive Kalman filter with unknown process noise covariance matrix.* Signal Processing, **169**, 2020, https://doi.org/10.1016/j.sigpro.2019.107413.

[43] ZHAO S.,AHN C. K.,SHI P.,SHMALIY Y. and LIU F., *Bayesian State Estimation for Markovian Jump Systems: Employing Recursive Steps and Pseudocodes*, IEEE Systems, Man, and Cybernetics Magazine,**5**(2),27-36, April 2019, doi: 10.1109/MSMC.2018.2882145.

List of Used Symbols and Abbreviations

Abbreviations

AD	Anderson-Darling
DWNA	discrete white noise acceleration
DWPA	discrete Wiener process acceleration
DWPJ	discrete Wiener process jerk
GPB1	generalized pseudo-Bayesian estimator of the first order
GPB2	generalized pseudo-Bayesian estimator of the first order
IMM	Interacting Multiple Model estimator
NCTR	nearly constant turn rate
MLAT	multilateration
ML	maximum-likelihood
TDOA	time difference of arrival
TOA	time of arrival

Symbols

A, B	general points in Euclidian space E^3
$\mathscr{B}$	width measure of the signal bandwidth
c	speed of light
cov	covariance between two random vectors
D	variance of the random variable
Δt_k	time difference between time at time epoch k and $k-1$: $\Delta t_k := t_k - t_{k-1}$
E^3	Euclidian space with dimension 3

$\mathbf{e}_i$	i^{th} vector of cartesian basis
$\mathbf{e}'_i$	i^{th} vector of local basis
$\mathscr{E}$	signal energy
E	expected value / mean
$\boldsymbol{E}_{k\|k-1}$	error of the prediction (prior) estimate of the hidden state
$\boldsymbol{E}_{k\|k}$	error of the posterior estimate of the hidden state
$f(\cdot)$	density probability function
$F(\cdot)$	cumulative distribution function
$\mathbf{F}$	state transition model
$\boldsymbol{F}(M)$	state transition model viewed as random variable, the realization is given by realization m of the model M
$\mathscr{F}$	non-linear state transition model
$\hat{\mathbf{F}}$	Jacobi matrix of the non-linear state transition model
g	non-linear mapping from TDOA space (positioning system) into E^3
$\mathbf{G}$	control-input model
$\boldsymbol{G}(M)$	control-input model viewed as random variable, the realization is given by realization m of the model M
GM	Gaussian mixture distribution
Γ	process noise transform model
h	history parameter, number of last considered measurements
$\mathbf{H}$	measurement model
$\boldsymbol{H}(M)$	measurement model viewed as random variable, the realization is given by realization m of the model M
$\mathbf{I}_s$	identity matrix of size $s \times s$
k	time index
$\mathbf{K}$	optimal Kalman gain
m	dimension of the measurement vector
m^j	j^{th} model in $\mathscr{M}$
M	motion model / model
$\mathcal{M}_s(\mathbb{R})$	$s \times s$ square matrix over real numbers
$\mathcal{M}_{s,t}(\mathbb{R})$	$s \times t$ matrix over real numbers

$\mathscr{M}$	discretization of the sample space Ω_{M_k} (set of the possible motion models)
$\mathcal{M}_{\|}^{\ell}$	ℓ^{th} history of active models up to time epoch k
$\boldsymbol{\mu}_{\hat{\tau}}$	mean vector of the inexact TDOA measurements in the positioning system
μ_k^j	posterior probability of the j^{th} model at time step k
n	dimension of the state vector
N_e	number of estimations
N_f	number of filters
N	normal distribution
$\mathbb{N}$	set of all natural number ($= \{0, 1, 2, \ldots\}$)
N_0	noise spectral power density
ν_k^{ℓ}	posterior probability of the ℓ^{th} model history at time step k
p	dimension of the control-input vector
$p(\cdot)$	probability function
$\mathbf{P}$	variance matrix of the state random vector $\boldsymbol{X}$
$\mathsf{P}\{\cdot\}$	probability
$PD_s(\mathbb{R})$	set of $s \times s$ positive definite matrices
q	dimension of the process noise vector
O	origin of the global cartesian affine frame
$\mathbf{O}_s$	zero matrix of size $s \times s$
$\mathbf{O}_{s,t}$	zero matrix of size $s \times t$
$\mathbf{o}_s$	zero vector of s-dimensional vector space
Ω_{M_k}	sample space of the motion model
$\Omega_{\mathbf{Q}}$	sample space of the process noise
P	point in Euclidian space E^3
$\mathbf{P}_{k\|k}$	Bayesian estimate of $\mathbf{P}_k$
$\mathbf{P}_{k\|k}^j$	Bayesian estimate of $\mathbf{P}_k$ by the j^{th} model
$\mathbf{P}_{k\|k}^{0j}$	mixed Bayesian estimate of $\mathbf{P}_k$ for the j^{th} model
$\boldsymbol{\Pi}$	Markov transition matrix

π_{ij}	element of Markov transition matrix
$\mathbf{Q}$	process noise variance matrix
$\boldsymbol{Q}(M)$	variance matrix of the process noise viewed as random variable, the realization is given by realization m of the model M
$\mathscr{Q}$	discretization of the $\Omega_{\mathbf{Q}}$
R_i	i^{th} receiver in the positioning system
$\mathbf{R}$	variance matrix of inexact measurements
$\mathbf{R}(\phi, \theta, \psi)$	rotation matrix in $SO(3)$
$\mathbb{R}$	real numbers
$\mathbb{R}^3$	vector space with dimension 3
ρ	Euclidian metric
S	origin of the local affine frame
$\mathbf{S}$	innovation variance matrix
$SO(3)$	Group of rotation matrices with rank 3
$\sigma^2_{\hat{\tau},j}$	variance of TDOA measurements on j^{th} receiver
σ_a^2	variance of the acceleration
$\hat{\mathbf{\Sigma}}_{\hat{\tau}}$	variance matrix of the inexact TDOA measurements in the positioning system
T	target, point in Euclidian space E^3
t_k	time at time epoch k
$\mathbf{t}$	vector in $\mathbb{R}^{\not\models}$
t	student's distribution
$\hat{\tau}_i$	TOA on the i^{th} receiver
τ_i	TDOA on i^{th} receiver (with respect to R_0 receiver)
$\mathbf{u}$	control-input vector
U	uniform distribution
$\boldsymbol{U}(M)$	control-input vector viewed as random vector, the realization is given by realization m of the model M
UD	discrete uniform distribution
$\boldsymbol{V}$	process noise vector

$\boldsymbol{V}(M)$	process noise vector viewed as random vector, the realization is depended on realization m of the model M
var	variance matrix of the random vector
$\boldsymbol{W}$	measurement noise vector
xyz	axis with respect to global cartesian affine frame
$x'y'z'$	axis with respect to local affine frame
$\boldsymbol{x}$	unknown realization of the state vector (unknown/hidden state)
$\bar{\mathbf{x}}$	mean vector of $\boldsymbol{X}$
$\bar{\mathbf{x}}_{k\|k}$	Bayesian estimate of $\bar{\mathbf{x}}_k$
$\bar{\mathbf{x}}^j_{k\|k}$	Bayesian estimate of $\bar{\mathbf{x}}_k$ by the j^{th} model
$\bar{\mathbf{x}}^{0j}_{k\|k}$	mixed Bayesian estimate of $\bar{\mathbf{x}}_k$ for the j^{th} model
$\boldsymbol{X}$	state vector (random vector)
$\tilde{\boldsymbol{y}}$	innovation realization
Υ_j	inexact TDOA measurement on the j^{th} receiver
$\boldsymbol{\Upsilon}$	inexact TDOA measurements in the positioning system
$\boldsymbol{Z}$	inexact measurement in E^3, in latter chapter identified with particular TDOA measurements transformed into E^3
$\bar{\mathbf{z}}$	mean of $\boldsymbol{Z}$
$\boldsymbol{z}$	realization of $\boldsymbol{Z}$
$\mathcal{Z}_k$	set of all measurements up to time epoch k, $\mathcal{Z}_k = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k\}$
$\mathcal{Z}^h_k$	set of last h measurements up to time epoch k, $\mathcal{Z}^h_k = \{\boldsymbol{z}_{k-h+1}, \ldots, \boldsymbol{z}_k\}$
$\|\cdot\|$	Euclidian norm of a vector in $\mathbb{R}^3$
$\|\cdot\|$	absolute value
$(\cdot,\cdot,\cdot)$	vector in Euclidian or vector space
$\langle A, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\rangle$	affine frame