TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Henri Nurminen

Estimation Algorithms for Non-Gaussian State-Space Models with Application to Positioning

Julkaisu 1499 • Publication 1499

Tampere 2017

Tampereen teknillinen yliopisto. Julkaisu 1499
Tampere University of Technology. Publication 1499

Henri Nurminen

Estimation Algorithms for Non-Gaussian State-Space Models with Application to Positioning

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Festia Building, Auditorium Pieni Sali 1, at Tampere University of Technology, on the 24th of November 2017, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2017

Doctoral candidate: Henri Nurminen
Laboratory of Automation and Hydraulics
Faculty of Engineering Sciences
Tampere University of Technology
Finland

Supervisor and Instructor: Robert Piché, Professor
Laboratory of Automation and Hydraulics
Faculty of Engineering Sciences
Tampere University of Technology
Finland

Instructor: Simo Ali-Löytty, University Lecturer
Laboratory of Mathematics
Faculty of Natural Sciences
Tampere University of Technology
Finland

Pre-examiners: Lyudmila Mihaylova, Professor
Department of Automatic Control and Systems Engineering
University of Sheffield
United Kingdom

Ondrej Straka, Associate Professor
Department of Cybernetics
University of West Bohemia
Czech Republic

Opponent: Simon Maskell, Professor
Department of Electrical Engineering and Electronics
University of Liverpool
United Kingdom

ISBN 978-952-15-4010-3 (printed)
ISBN 978-952-15-4029-5 (PDF)
ISSN 1459-2045

Abstract

State-space models (SSMs) are used to model systems with hidden time-varying state and observable measurement output. In statistical SSMs, the state dynamics is assumed known up to a random term referred to as the process noise, and the measurements contain random measurement noise. Kalman filter (KF) and Rauch–Tung–Striebel smoother (RTSS) are widely-applied closed-form algorithms that provide the parameters of the exact Bayesian filtering and smoothing distributions for discrete-time linear statistical SSMs where the process and measurement noises follow Gaussian distributions. However, when the SSM involves nonlinear functions and/or non-Gaussian noises, the Bayesian filtering and smoothing distributions cannot in general be solved using closed-form algorithms. This thesis addresses approximate Bayesian time-series inference for two positioning-related problems where the assumption of Gaussian noises cannot capture all useful knowledge of the considered system's statistical properties: map-assisted indoor positioning and positioning using time-delay measurements.

The motion constraints imposed by the indoor map are typically incorporated in the position estimate using the particle filter (PF) algorithm. The PF is a Monte Carlo algorithm especially suited for statistical SSMs where the Bayesian posterior distributions are too complicated to be adequately approximated using a well-known distribution family with a low-dimensional parameter space. In map-assisted indoor positioning, the trajectories that cross walls or floor levels get a low probability in the model. In this thesis, improvements to three different PF algorithms for map-assisted indoor positioning are proposed and compared. In the wall-collision PF, weighted random samples, also known as particles, are moved based on inertial

sensor measurements, and the particles that collide with the walls are downweighted. When the inertial sensor measurements are very noisy, map information is used to guide the particles such that fewer particles collide with the walls, which implies that more particles contribute to the estimation. When no inertial sensor information is used, the particles are moved along the links of a graph that is dense enough to approximate the set of expected user paths.

Time-delay based ranging measurements of e.g. ultra-wideband (UWB) and Global Navigation Satellite Systems (GNSSs) contain occasional positive measurement errors that are large relative to the majority of the errors due to multipath effects and denied line of sight. In this thesis, computationally efficient approximate Bayesian filters and smoothers are proposed for statistical SSMs where the measurement noise follows a skew t-distribution, and the algorithms are applied to positioning using time-delay based ranging measurements. The skew t-distribution is an extension of the Gaussian distribution, which has two additional parameters that affect the heavy-tailedness and skewness of the distribution. When the measurement noise model is heavy-tailed, the optimal Bayesian algorithm is robust to occasional large measurement errors, and when the model is positively (or negatively) skewed, the algorithms account for the fact that most large errors are known to be positive (or negative). Therefore, the skew t-distribution is more flexible than the Gaussian distribution and captures more statistical features of the error distributions of UWB and GNSS measurements. Furthermore, the skew t-distribution admits a conditionally Gaussian hierarchical form that enables approximating the filtering and smoothing posteriors with Gaussian distributions using variational Bayes (VB) algorithms. The proposed algorithms can thus be computationally efficient compared to Monte Carlo algorithms especially when the state is high-dimensional. It is shown in this thesis that the skew-t filter improves the accuracy of UWB based indoor positioning and GNSS based outdoor positioning in urban areas compared to the extended KF. The skew-t filter's computational burden is higher than that of the extended KF but of the same magnitude.

Tiivistelmä

Tila-avaruusmalleilla mallinnetaan järjestelmiä, joilla on tuntematon ajassa muuttuva tila sekä mitatattava ulostulo. Tilastollisissa tila-avaruusmalleissa järjestelmän tilan muutos tunnetaan lukuunottamatta prosessikohinaksi kutsuttua satunnaista termiä, ja mittaukset sisältävät satunnaista mittauskohinaa. Kalmanin suodatin sekä Rauchin Tungin ja Striebelin siloitin ovat yleisesti käytettyjä suljetun muodon estimointialgoritmeja, jotka tuottavat tarkat bayesiläiset suodatus- ja siloitusjakaumat diskreettiaikaisille lineaarisille tilastollisille tila-avaruusmalleille, joissa prosessi- ja mittauskohinat noudattavat gaussisia jakaumia. Jos käsiteltyyn tila-avaruusmalliin kuitenkin liittyy epälineaarisia funktioita tai epägaussisia kohinoita, bayesiläisiä suodatus- ja siloitusjakaumia ei yleensä voida ratkaista suljetun muodon algoritmeilla. Tässä väitöskirjassa tutkitaan approksimatiivista bayesiläistä aikasarjapäättelyä ja sen soveltamista kahteen paikannusongelmaan, joissa gaussinen jakauma ei mallinna riittävän hyvin kaikkea hyödyllistä tietoa tutkitun järjestelmän tilastollisista ominaisuuksista: kartta-avusteinen sisätilapaikannus sekä signaalin kulkuaikamittauksiin perustuva paikannus.

Sisätilakartan tuottamat liikerajoitteet voidaan liittää paikkaestimaattiin käyttäen partikkelisuodattimeksi kutsuttua algoritmia. Partikkelisuodatin on Monte Carlo -algoritmi, joka soveltuu erityisesti tilastollisille tila-avaruusmalleille, joissa bayesiläisen posteriorijakauman tiheysfunktio on niin monimutkainen, että sen approksimointi tunnetuilla matalan parametridimension jakaumilla ei ole mielekästä. Kartta-avusteisessa sisätilapaikannuksessa reitit, jotka leikkaavat seiniä tai kerrostasoja, saavat muita pienemmät todennäköisyydet. Tässä väitöskirjassa esitetään parannuksia kolmeen eri partikkelisuodatusalgoritmiin, joita sovelletaan kartta-avusteiseen sisätilapaikan-

nukseen. Seinätörmayssuodattimessa painolliset satunnaisnäytteet eli partikkelit liikkuvat inertiasensorimittausten mukaisesti, ja seinään törmäävät partikkelit saavat pienet painot. Kun inertiasensorimittauksissa on paljon kohinaa, partikkeleita voidaan ohjata siten, että seinätörmäysten määrä vähenee, jolloin suurempi osa partikkeleista vaikuttaa estimaattiin. Kun inertiasensorimittauksia ei käytetä lainkaan, sisätilakartta voidaan esittää graafina, jonka kaarilla partikkelit liikkuvat ja joka on riittävän tiheä approksimoimaan odotettavissa olevien reittien joukkoa.

Esimerkiksi laajan taajuuskaistan radioista (UWB, *ultra-wideband*) tai paikannussatelliiteista saatavat radiosignaalin kulkuaikaan perustuvat etäisyysmittaukset taas voivat sisältää monipolkuheijastusten ja suoran reitin estymisen aiheuttamia positiivismerkkisiä virheitä, jotka ovat huomattavan suuria useimpiin mittausvirheisiin verrattuna. Tässä väitöskirjassa esitetään laskennallisesti tehokkaita bayesiläisen suodattimen ja siloittimen approksimaatioita tilastollisille tila-avaruusmalleille, joissa mittauskohina noudattaa vinoa t-jakaumaa. Vino t-jakauma on gaussisen jakauman laajennos, ja sillä on kaksi lisäparametria, jotka vaikuttavat jakauman paksuhäntäisyyteen ja vinouteen. Kun mittauskohinaa mallintava jakauma oletetaan paksuhäntäiseksi, optimaalinen bayesiläinen algoritmi ei ole herkkä yksittäisille suurille mittausvirheille, ja kun jakauma oletetaan positiivisesti (tai negatiivisesti) vinoksi, algoritmit hyödyntävät tietoa, että suurin osa suurista virheistä on positiivisia (tai negatiivisia). Vino t-jakauma on siis gaussista jakaumaa joustavampi, ja sillä voidaan mallintaa kulkuaikaan perustuvien mittausten virhejakaumaa tarkemmin kuin gaussisella jakaumalla. Vinolla t-jakaumalla on myös ehdollisesti gaussinen esitys, joka soveltuu suodatus- ja siloitusposteriorien approksimointiin variaatio-Bayes-algoritmilla. Näin ollen esitetyt algoritmit voivat olla laskennallisesti tehokkaampia kuin Monte Carlo -algoritmit erityisesti tilan ollessa korkeaulotteinen. Tässä väitöskirjassa näytetään, että vino-t-virhejakauman käyttö parantaa UWB-radioon perustuvan sisätilapaikannuksen tarkkuutta sekä satelliittipohjaisen ulkopaikannuksen tarkkuutta kaupunkiympäristössä verrattuna laajennettuun Kalmanin suodattimeen. Vino-t-suodatuksen laskennallinen vaativuus on suurempi mutta samaa kertaluokkaa kuin laajennetun Kalmanin suodattimen.

Preface

The research that I present in this thesis was carried out at the Positioning Algorithms group at the Department of Automation Science and Engineering and at the Laboratory of Automation and Hydraulics of Tampere University of Technology between 2013 and 2017. My research was funded by Tampere University of Technology Graduate School and Nokia Corporation. I gratefully acknowledge the additional financial support from the Finnish Society of Automation, Tekniikan edistämissäätiö, the Foundation of Nokia Corporation, and Emil Aaltonen Foundation.

I thank my supervisor and instructor Prof. Robert Piché for providing me with inspiring research topics, useful practical advice, and excellent teaching. I also acknowledge his dedication to promoting high-quality research reporting. I thank my instructor Dr. Simo Ali-Löytty for numerous long and insightful discussions and especially for excellence in mathematics. I am deeply grateful to Dr. Tohid Ardeshiri from the University of Cambridge for support and determination in our long-term co-operation. I also thank my colleagues at the Positioning Algorithms group, especially Dr. Matti Raitoharju, Dr. Philipp Müller, Juha Ala-Luhtala, Dr. Helena Leppäkoski, Zhang Xiaolong, Dr. Pavel Davidson, Mike Koivisto, and Anssi Ristimäki for joint work, peer support, and both educational and entertaining lunch breaks.

Furthermore, I am sincerely thankful to Prof. Fredrik Gustafsson for hosting my research visit to Linköping University in the autumn 2014 and to Prof. Simon Godsill for hosting my research visit to the University of Cambridge in the autumn 2016. I also thank Dr. Jukka Talvitie, Assoc. Prof. Elena-Simona Lohan, Dr. Marzieh Dashti, and

Dr. Rafael Rui for productive co-operations. I am grateful to Dr. Lauri Wirola and Dr. Jari Syrjärinne from HERE Corporation for bringing industry insights to my research. I dedicate special thanks to Prof. Lyudmila Mihaylova and Assoc. Prof. Ondrej Straka for pre-examining this thesis, as well as to Prof. Simon Maskell for acting as my opponent in the public examination of this thesis.

I thank my family for being there for me throughout my life. I thank all my friends who have expanded my worldview during my undergraduate and graduate studies. Finally, I thank my wife Sini for her companionship and acceptance of my devotion to the thesis work.

Kiitän perhettäni ja sukuani läsnäolosta koko elämäni ajan. Kiitän kaikkia ystäviäni, jotka ovat avartaneet maailmankuvaani opintojeni ja väitöskirjatyöni aikana. Erityiskiitokset kuuluvat Tomi Kurkolle väitöskirjani lukemisesta ja hyödyllisistä huomautuksista. Vaimoni Sini on se syy, joka saa minut lähtemään liikkeelle ja palaamaan kotiin joka päivä. Kiitän sinua Sini rakkaudesta, ystävyydestä ja siitä että hyväksyit omistautumiseni väitöskirjatyölle.

Tampere, August 2017

Henri Nurminen

Contents

List of publications

This thesis consists of the introduction and the following six publications and one unpublished manuscript.

PUBLICATIONS

P1. Henri Nurminen, Anssi Ristimäki, Simo Ali-Löytty, and Robert Piché, Particle filter and smoother for indoor localization, *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, October 2013.

P2. Henri Nurminen, Mike Koivisto, Simo Ali-Löytty, and Robert Piché, Motion model for positioning with graph-based indoor map, *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, October 2014.

P3. Mike Koivisto, Henri Nurminen, Simo Ali-Löytty, and Robert Piché, Graph-based map matching for indoor positioning, *10th International Conference on Information, Communications and Signal Processing (ICICS)*, December 2015.

P4. Henri Nurminen, Matti Raitoharju, and Robert Piché, An efficient indoor positioning particle filter using a floor-plan based proposal distribution, *19th International Conference on Information Fusion (FUSION)*, July 2016.

P5. Henri Nurminen, Tohid Ardeshiri, Robert Piché, and Fredrik Gustafsson, Robust inference for state-space models with skewed measurement noise, *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1898–1902, November 2015.

P6. Henri Nurminen, Tohid Ardeshiri, Robert Piché, and Fredrik Gustafsson, A NLOS-robust TOA positioning filter based on a skew-t measurement noise model, *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, October 2015.

UNPUBLISHED MANUSCRIPT

M7. Henri Nurminen, Tohid Ardeshiri, Robert Piché, and Fredrik Gustafsson, Skew-t filter and smoother with improved covariance matrix approximation.

The main contributions of the publications and the unpublished manuscript:

P1. The publication presents a literature review on different aspects of the wall-collision particle filter (PF) and the wall-collision particle smoother (PS) for map-assisted indoor positioning where floor plan restrictions are used to improve the accuracy of the wireless network and hand-held inertial sensor based positioning. In the publication an integrity monitoring method for the PF and a method for re-initialising the PF using a computationally light fallback filter are proposed, and a floor change detection test is presented. Furthermore, the proposed indoor positioning PS is based on a more advanced PS algorithm than the state of the art.

P2. The publication proposes an improvement to the graph-based indoor positioning PF, where the indoor map is represented as a graph. The novel feature is a probabilistic link transition model, i.e. a rule for distributing the user position's probability from one link to the others. In the proposed model, the probability to choose a link is proportional to the size of the subgraph that is accessible through each link option.

P3. The publication proposes an extension to the link transition model of Publication [P2]. The model is extended to indoor map representations where only narrow spaces are represented as graphs and large open spaces are represented as polygons where two-dimensional motion of the PF's particles is allowed.

P4. The publication proposes including some map information in the wall-collision PF's proposal distribution. In the proposed algorithm, the inertial sensor based motion model distribution is distorted by giving more probability to the directions where the distances to the closest walls are larger. This way, fewer particles collide with the walls, and the PF can achieve a better accuracy with comparable computational load because the most probable areas are modelled by more particles.

P5. The publication proposes approximative Bayesian filter and smoother for state-space models (SSMs) where the measurement noise components follows a skew t-distribution. The proposed algorithm retains the computational efficiency and scalability with respect to the state dimension of the Kalman fil-

ter (KF) because the proposed algorithms are based on an analytical approximation called the variational Bayes (VB) method.

P6. The publication proposes application of the skew-t variational Bayes filter proposed in Publication [P5] to indoor positioning using ultra-wideband (UWB) radios. UWB's time of arrival (TOA) measurements' error distribution shows positive skewness and heavy-tailedness because of non-line-of-sight (NLOS) and multipath effects.

M7. The manuscript proposes improvements to the skew-t filter and smoother proposed in Publication [P5]. Firstly, a new VB factorisation of the approximate posterior distribution is proposed, where the mean and covariance matrix of a truncated multivariate normal distribution (TMND) are approximated using an existing expectation propagation (EP) based algorithm. Secondly, a greedy algorithm for ordering the constraints of the TMND within the EP is proposed with an optimality proof. Thirdly, Cramér–Rao lower bounds for the filtering and smoothing distributions of the used SSM as well as an expression of the variational lower bound for the proposed VB factorisation are derived. The new VB factorisation improves the estimation performance of the skew-t filter and smoother by reducing the covariance matrix underestimation common to most VB inference algorithms. The proposed filter is applied to real Global Navigation Satellite System (GNSS) pseudorange data and is shown to outperform some state-of-the-art computationally light filters in accuracy.

My roles in the shared publications and in the unpublished manuscript:

P1. I contributed to developing and implementing the novel features of the proposed PF, ran the PF tests, and had the main role in writing the manuscript except for the PS section IV.

P2. I developed the algorithm except for the point estimator algorithm and contributed to developing the point estimator algorithm together with Mike Koivisto. I implemented the algorithm, ran the tests, and wrote the manuscript.

P3. I developed and implemented the proposed algorithm and wrote the manuscript together with Mike Koivisto.

P4. I proposed the basic idea of the algorithm during conversations with Matti Raitoharju. I developed and implemented the algorithm, ran the tests, and wrote the manuscript.

P5. Robert Piché proposed the basic idea of the algorithms. I developed the details of the algorithms together with Tohid Ardeshiri. I implemented the algorithms, ran the tests, and had the main role in writing Sections I, V, and VI. I had a minor role in writing the other parts of the manuscript, for which Tohid Ardeshiri had the main role.

P6. I collected the UWB data, implemented the algorithms based on the ideas in Publication [P5], and ran the tests. I had the main role in writing the manuscript, for which Tohid Ardeshiri had a minor role.

M7. I proposed the main ideas, developed and implemented the algorithms, ran the tests, and had the main role in writing the manuscript, for which Tohid Ardeshiri had a minor role.

Abbreviations

BF bootstrap filter
BLE Bluetooth low energy
CDF cumulative distribution function
CKF cubature Kalman filter
CRLB Cramér–Rao lower bound
EKF extended Kalman filter
EM expectation–maximisation
EP expectation propagation
GM Gaussian mixture
GMF Gaussian mixture filter
GNSS Global Navigation Satellite System
GRUF generalised recursive update filter
IEKF iterated extended Kalman filter
INLA integrated nested Laplace approximation
IoT Internet of Things
KF Kalman filter
KLD Kullback–Leibler divergence
LOS line-of-sight
LTE Long-Term Evolution
NLOS non-line-of-sight
PDF probability density function
PDR pedestrian dead reckoning
PF particle filter
PS particle smoother
RFID radio-frequency identification
RMSE root-mean-square error
RSS received signal strength
RTSS Rauch–Tung–Striebel smoother
spd symmetric positive definite
SSM state-space model
TLL total link length
TMND truncated multivariate normal distribution
TOA time of arrival
UWB ultra-wideband
VB variational Bayes
WLAN wireless local area network

Symbols

$\|\cdot\|$	Euclidean norm
$[\![\cdot]\!]$	Iverson bracket
$\propto$	proportional to
$\mathbf{1}$	column vector of ones
A	state transition matrix
$\alpha^{(i)}$	weight of ith Gaussian mixture component
α_j	jth value of the direction discretisation
C	measurement model matrix
cat	categorical distribution
$\overline{\Delta}_k$	heading change measurement at kth time instant
$\boldsymbol{\epsilon}$	measurement noise expectation value
$\boldsymbol{e}_k$	measurement noise vector at kth time instant
$\mathbb{E}_{p(\boldsymbol{x})}[\boldsymbol{x}]$	expectation value of random vector $\boldsymbol{x}$ with probability density function (PDF) $p(\boldsymbol{x})$
$\operatorname{erf}(\cdot)$	error function
$\phi(\cdot)$	PDF of the standard normal distribution
$\Phi(\cdot)$	cumulative distribution function (CDF) of the standard normal distribution
$\boldsymbol{f}(\cdot)$	state transition function
F^{-1}	inverse CDF
$F_D(\cdot)$	CDF of distribution D
G	gamma distribution
G^{-1}	inverse-gamma distribution
$\Gamma(\cdot)$	gamma function
GIG	generalised inverse-gamma distribution
GM	Gaussian mixture
$\boldsymbol{h}(\cdot)$	measurement model function
$\boldsymbol{\eta}_k$	either process noise or measurement noise at kth time instant
I_n	$n \times n$ identity matrix
K	the index of the last time instant of the estimation time period
$\mathrm{KL}\big(q\big\|p\big)$	Kullback–Leibler divergence (KLD) from distribution q to distribution p
$\operatorname{kurtosis}[x]$	excess kurtosis of random variable x
ℓ_k	footstep length at kth time instant
$\overline{\ell}_k$	footstep length measurement at kth time instant
$\boldsymbol{\mu}^{(i)}$	mean of ith Gaussian mixture component

MVST	multivariate skew t-distribution
N_α	number of direction discretisation values
N_{GM}	number of Gaussian mixture components
$\mathrm{N}(\boldsymbol{\mu}, \Sigma)$	(multivariate) Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ
$\mathrm{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma)$	PDF of $\mathrm{N}(\boldsymbol{\mu}, \Sigma)$ evaluated at $\boldsymbol{x}$
N_+	(multivariate) Gaussian distribution truncated into positive orthant
N_p	number of particles
n_x	dimensionality of vector $\boldsymbol{x}$
$\boldsymbol{\omega}$	process noise expectation value
Ω	set of constraints in variational optimisation
ψ	heading angle of a pedestrian
$\mathrm{P}_{k\vert\ell}$	state covariance matrix of kth time instant given ℓ measurements
$p_e(\cdot)$	measurement noise PDF
$p_{\boldsymbol{w}}(\cdot)$	process noise PDF
$\mathbb{P}(X)$	probability of event X
$p(\boldsymbol{x})$	PDF of random vector $\boldsymbol{x}$, prior distribution of $\boldsymbol{x}$
$p(\boldsymbol{x}\vert\boldsymbol{y})$	conditional PDF of $\boldsymbol{x}$ given $\boldsymbol{y}$, posterior distribution of $\boldsymbol{x}$
$p(\boldsymbol{y}\vert\boldsymbol{x})$	conditional PDF of $\boldsymbol{y}$ given $\boldsymbol{x}$, likelihood function of $\boldsymbol{x}$
Q	process noise covariance matrix
$\hat{q}$	PDF of an optimal VB approximation
$q(\cdot\vert\boldsymbol{y}_{1:k}, \boldsymbol{x}^{(i)}_{1:k-1})$	proposal distribution of ith particle
$q_{\mathrm{BF}}(\cdot\vert\boldsymbol{y}_{1:k}, \boldsymbol{x}^{(i)}_{1:k-1})$	bootstrap filter's proposal distribution
$q_{\mathrm{opt}}(\cdot\vert\boldsymbol{y}_{1:k}, \boldsymbol{x}^{(i)}_{1:k-1})$	optimal proposal distribution
$q^{\overline{\Delta}}$	heading change measurement variance
R	measurement noise covariance matrix
$\mathbb{R}$	set of real numbers
$\mathbb{R}^n$	set of n-dimensional real numbers
$\mathbb{R}_+$	set of positive real numbers
$\boldsymbol{r}_k$	2-dimensional user position at kth time instant
$\boldsymbol{\Sigma}^{(i)}$	covariance matrix of ith Gaussian mixture component
skewness$[x]$	skewness of random variable x
ST	univariate skew t-distribution
τ	resampling parameter in PF
$\boldsymbol{\theta}_{k,i}$	ith component of latent random vector at kth time instant

$\mathrm{t}(\mu, \sigma^2, \nu)$	t-distribution with location μ, scale σ and ν degrees of freedom
$\mathrm{t}(\boldsymbol{x}; \mu, \sigma^2, \nu)$	PDF of $\mathrm{t}(\mu, \sigma^2, \nu)$ evaluated at $\boldsymbol{x}$
$\mathrm{tr}\{\cdot\}$	matrix trace
v_k	speed at kth time instant
$\mathrm{var}[\boldsymbol{x}]$	covariance matrix of random vector $\boldsymbol{x}$
VST	either univariate or multivariate skew t-distribution
$\boldsymbol{w}_k$	process noise vector at kth time instant
$W_k^{(i)}$	weight of the ith particle at kth time instant
$\widetilde{W}_k^{(i)}$	unnormalised weight of the ith particle at kth time instant
$\boldsymbol{x}$	state vector
$\boldsymbol{x}_k$	state vector of kth time instant
$\boldsymbol{x}_{k\|\ell}$	state estimate of kth time instant given ℓ measurements
$\boldsymbol{x}_k^{(i)}$	value of the ith particle at kth time instant
$\boldsymbol{y}$	measurement vector
$\boldsymbol{y}_{1:k}$	measurement vectors of the time instants from first to kth
$\boldsymbol{y}_k$	measurement vector of kth time instant
$\boldsymbol{z}_k$	absolute position measurement at kth time instant

INTRODUCTION

This thesis consists of an introduction chapter, six articles published in scientific conferences and journals, and one unpublished manuscript. This introduction chapter summarises the contribution of the thesis. Section 1 of this introduction chapter defines the positioning problem, the basics of Gaussian and non-Gaussian modelling, and the main research objectives of this thesis. Section 2 explains the theoretical framework on which the proposed algorithms are based. Section 3 introduces the algorithms and the positioning-related application problems considered in Publications [P1], [P2], [P3], [P4], [P5], [P6] and in Manuscript [M7].

1 Background

1.1 Positioning

Positioning means determining the position of a target device with respect to a coordinate system. The commercial and social significance of positioning information and navigation methods has been growing rapidly due to the upsurge in the processing capabilities of personal mobile devices and in the number of applications that are based on location awareness. Positioning is a key component in way finding, rescue services, proximity marketing, mobile games, track-

ing people and equipment in hospitals and industrial environments, among others. The current Internet of Things (IoT) boom emphasises the need of reliable and inexpensive positioning technologies and algorithms [1], [2].

Many positioning applications are based on Global Navigation Satellite Systems (GNSSs) such as GPS, GLONASS, Galileo, and BeiDou. However, there are important use cases where GNSS is unavailable or has inadequate performance, and thus low-cost positioning methods that do not use satellite-based information are necessary [3]–[7]. Often the GNSS precision is lowest where the requirement for the precision is highest: densely built urban areas ("urban canyons") and especially indoor and underground spaces tend to be completely or partially shadowed from the GNSS signals. Even when a GNSS is usable, sophisticated statistical modelling of the navigation signal helps to mitigate the adverse effect of non-line-of-sight (NLOS) signals and multipath effects. Currently no single technology can provide sufficient accuracy in all purposes; different technologies are required for different applications, and there is also a need for hybrid positioning methods, where different technologies complement each other [7].

One way to position a radio receiver without GNSS is to use the radio signals of wireless networks. In wireless network based positioning the measurements are anchored to the coordinate system by either knowledge of the network structure such as the positions of the network's base stations or other knowledge of the received signal's structure in different receiver positions [4], [5]. Positioning can be based on the communication infrastructure such as cellular networks (2G, 3G, Long-Term Evolution (LTE), in future 5G), on wireless local area networks (WLANs) [8] or on positioning-specific wireless transmitters, such as Bluetooth low energy (BLE) [9] and ultra-wideband (UWB) [10]. Commonly used positioning measurements include received signal strength (RSS) and time of arrival (TOA) [4].

RSS positioning can be based on assuming that the closer the positioned target is to a network's base station, the higher the expected RSS level. The RSS measurements are readily available in almost any wireless communication system because it is needed to monitor the quality of the connection to the base station. However, the distance

resolution of the RSS measurements is typically low compared to the noise level, especially at locations far from the base station and in highly obstructed environments such as indoors [4]. Thus, statistical modelling of the RSS measurement is required, and RSS-based positioning is typically assisted by other types of measurements. These measurements include inertial sensors and floor plan information that are especially useful for complementing the wireless network based positioning methods [7], [11]. A central topic in this thesis is how to use floor plan constraints as measurements in indoor position estimation using advanced statistical estimation methods.

TOA positioning uses range estimates obtained by measuring the travelling time of the radio signal between a transmitter at a known location and the receiver whose position is being estimated. TOA measurements are commonly used e.g. with UWB radios whose short-duration pulses enable high time resolution [10]. GNSS positioning is also based on signal propagation time measurements [12]. TOA measurements typically exhibit better accuracy than RSS, tens of centimetres for UWB in line-of-sight (LOS) conditions, but they are susceptible to NLOS and multipath phenomena; when the direct path between the transmitter and receiver is blocked, receptions of reflected signals may occasionally cause positive errors that are large compared to the LOS accuracy, several metres for UWB, for example [10]. A notable feature in the TOA measurements' error distribution as well as in many other time based phenomena is asymmetry: large positive errors are much more frequent than large negative errors. In this thesis, real-time and non-real-time positioning algorithms for TOA time-series data are proposed. The real-time algorithms base the position estimation on the measurements up to and including the estimation time instant, while the non-real-time methods can also use measurements received after the estimation time instant to make fixed-lag or fixed-interval estimation.

There are numerous other positioning technologies that are left out of the scope of this thesis. Other utilisable wireless communication signals include radio-frequency identification (RFID) and ZigBee [6], [13]. Magnetic field anomalies can be used for indoor positioning by matching magnetometer measurements with a pre-collected magnetic field map [5], [7], [14]. The whole 3-dimensional magnetic

field vector can be used if the positioned device's orientation can be estimated using other measurements, and otherwise only the field strength can be used [14]. In vision based positioning, video camera output is used as a positioning measurement [15], [16]. One way to do vision based positioning is estimating the movement of the positioned device including heading change and translation using features of a video camera output [16]. Other signals that can be used for positioning in various ways include infrared radiation, ultrasound, and digital television signals [5].

A key component in all the proposed algorithms is modelling of the measurement errors. Statistical modelling of random errors is discussed in the next subsection.

1.2 Gaussian and non-Gaussian noise models

In general, mathematical models of real-world systems cannot predict actual observations made of the system exactly, but the model predictions contain errors that are referred to as noise. Usually the noise is mainly due to model simplifications that are made because of lacking information or to keep research and/or computational effort at an acceptable level. An example of noise is measurement errors that are seemingly random, and modelling of the conditions that lead to each realised measurement error is impractical. Another example is modelling of pedestrian motion: some general knowledge on how a pedestrian moves can be included in the model, such as average or maximum speed, but modelling of every single movement decision is not feasible without further measurement information.

In this thesis noises are modelled as realisations of random variables described by probability distributions. The Gaussian distribution, also known as the (multivariate) normal distribution, is one of the most commonly used probabilistic noise models. Some physical systems follow the Gaussian distribution by the theory of physics, but in most cases Gaussianity is not an exact provable feature but an assumption that is made for several reasons:

1. The Gaussian distribution admits convenient mathematical properties: marginal and conditional distributions of a multivariate Gaussian distribution are again multivariate Gaussian

distributions. Thus, the maximum likelihood and maximum a posteriori problems for data with Gaussian noise and linear measurement model can be formulated as the well-known and often easily solvable linear least-squares problem.
2. Many time-varying linear systems can be modelled by the linear–Gaussian state-space models (SSMs), which justify the convenient analytical solutions given by the Kalman filter (KF) and Rauch–Tung–Striebel smoother (RTSS) algorithms [17]. Approximative algorithms for nonlinear–Gaussian SSMs have also been studied extensively [18].
3. Some probability distributions can be expressed as conditionally Gaussian distributions, i.e. Gaussian given latent random parameters.
4. Central limit theorems state that with certain conditions the sum of any independent random variables approaches a Gaussian-distributed random variable when the number of the random variables goes to infinity.
5. The Gaussian distribution is the maximum entropy distribution given the first and second moments [19, Ch. 45.2]. That is, if only the mean and variance of a distribution are known, the Gaussian distribution is the approximation that requires the fewest further assumptions of the true distribution in a certain sense.

However, as shown in this thesis, Gaussian models are sometimes inadequate for including all the information available at real-world problems, and using the conventional Gaussian models can result in deteriorated estimation accuracy. Two types of non-Gaussian features are considered in this thesis. Firstly, this thesis considers motion constraints. Motion constraint information is used in map-assisted indoor positioning, where the floor plan information involves highly nonlinear and non-Gaussian features. These models can include deciding which way to continue in a junction of corridors or excluding the paths that cross walls according to the floor plan.

Secondly, this thesis considers skewness and heavy-tailedness. Skewness means asymmetry in the probability distribution. Intuitively speaking, heavy-tailedness means that the distribution generates realisations that are far from the general trend significantly more frequently than the Gaussian distribution. These exceptional reali-

sations are often called outliers. Asymmetry and presence of outlier measurements are deviations from the Gaussianity assumptions of the least-squares methods and the KF, and can cause large estimation errors when Gaussian distribution based algorithms are used [20]. In this thesis it is shown that errors of time delay based measurements of UWB and GNSS in mixed LOS and NLOS conditions admit skewed and heavy-tailed distributions. Therefore, more flexible models and algorithms with computational efficiency comparable to that of the KF are proposed.

A typical feature for positioning problems is observing the data in the form of a time-series. If a model for the movement of the target is available, it is possible to fuse information from multiple time instants' measurements. Doing this fusion efficiently poses some challenges: One needs to be able to model the target state's dynamics with some accuracy. Furthermore, the importance of modelling the measurement errors' distribution is emphasised because the measurement information is fused not only with the other measurements but also with the dynamical model of the state. The class of models used in this thesis for modelling time-series data is the discrete-time statistical SSMs [18]. The SSMs are broadly applicable and are the starting point in a wide literature of closed-form and approximative estimation algorithms.

1.3 Research questions & contributions

This Subsection states the three main research questions considered in this thesis and lists the main contributions of the thesis.

1) What are the best models and algorithms for incorporating floor plan constraints in an indoor positioning algorithm?

Floor plan constraints are typically incorporated in the indoor position estimate using the particle filter (PF) algorithm because of the algorithm's flexibility in what models it can work with. The PF is based on generating weighted (pseudo-)random samples of the tracked person's trajectory. In Publications [P1], [P2], [P3], [P4] different algorithms are proposed for sampling the trajectories such that fewer samples are needed and the algorithm becomes computationally more efficient. As analysed in Section 3.1, the wall collision PF

algorithm studied in [P1] suits best when a relatively precise pedestrian dead reckoning (PDR) is available. In this algorithm the floor plan constraints are used as measurement information. When the PDR measurements are more noisy, the map information can be incorporated in the motion model and the PF's proposal distribution, which can improve the filter's accuracy and reduce the required number of particles [P4]. When no PDR is used, the graph based indoor positioning PF with our proposed link transition rule [P2], [P3] is the most recommendable filter.

2) How can a Bayesian filter and a Bayesian smoother be designed that take account of the skewness and kurtosis of the measurement noise distribution while maintaining an acceptable level of computational complexity and scalability with respect to the problem dimensionality?

Publication [P5] and Manuscript [M7] propose approximative Bayesian filtering and smoothing algorithms that are robust against outlier measurements and take into account asymmetry in the measurement noise distribution. Outliers and asymmetry are modelled using the skew t-distribution, which has four parameters to control the mean (the first moment), variance (the second central moment), skewness (related to the third central moment of the distribution), and kurtosis (heavy-tailedness, related to the fourth central moment) of the distribution. The skew t-distribution is used because it admits a conditionally Gaussian structure that makes the model compatible with the mean-field variational Bayes (VB) algorithm [21], which is a well-known approximation method in statistics and machine learning [22, Ch. 10]. This VB method results in an iterative algorithm that, under suitable conjugacy properties of the model, gives a closed-form approximation of the posterior distribution. Publication [P5] proposes VB based filter and smoother that can be seen as extensions of the KF and RTSS where the mean and covariance matrix of the measurement model are estimated iteratively at each time instant. Manuscript [M7] proposes a different VB based filter and smoother, which involve some further approximations but provide better estimation accuracy and faster convergence due to less restrictive assumptions on the VB approximation.

3) How can NLOS measurements be handled in a computationally efficient way in time delay measurement based positioning algorithms?

In Publication [P6] a proposed skew-t VB filter is applied to indoor positioning using TOA measurements from a UWB network, and in Manuscript [M7] a proposed skew-t filter is applied to GNSS positioning in a dense urban environment. The proposed algorithms provide a statistically principled way to accommodate outlier measurements and to account for the skewness. Because the proposed algorithms are not based on random sampling, they are often computationally lighter than Monte Carlo methods and their performance does not degrade as dramatically when the state dimensionality increases.

2 Estimation theory

2.1 Bayesian inference

The modelling methodology used in this thesis follows the Bayesian paradigm of statistics, where all unknown quantities are treated as random variables, and all the knowledge of the random variable's value is expressed in the probability density function (PDF). The random vector consisting of the unknown state variables is often denoted with $\boldsymbol{x} \in \mathbb{R}^{n_x}$ and the measurement random vector with $\boldsymbol{y} \in \mathbb{R}^{n_y}$. A Bayesian statistical model specifies the prior distribution PDF $p_{\boldsymbol{x}}(\boldsymbol{x})$ and the measurement model $p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})$, which is a PDF when considered as a function of the measurements $\boldsymbol{y}$, and a function that is referred to as the likelihood when considered as a function of $\boldsymbol{x}$. Bayesian inference means finding the conditional probability distribution of the state given the measurements $p_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y})$, which then contains all the knowledge of the state given by the prior information and the measurements. In this thesis, the subscripts in the PDF notations are omitted when not necessary for readability, and the random variable that the PDF is related to is only indicated by the argument, for example $p(\boldsymbol{x})$, $p(\boldsymbol{y}|\boldsymbol{x})$, $p(\boldsymbol{x}|\boldsymbol{y})$. This is done for brevity and simplicity of the notation and to follow a common convention. Random vectors and real vectors are not distinguished in notation.

The Bayesian inference is based on Bayes' rule

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})\, p(\boldsymbol{x})}{\int p(\boldsymbol{y}|\boldsymbol{x})\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}}, \tag{1}$$

which is a direct consequence of the formula of a conditional PDF. Because the denominator in (1) is independent of $\boldsymbol{x}$, this is often written as

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})\, p(\boldsymbol{x}), \tag{2}$$

where the symbol $\propto$ means "proportional to". That is, a key operation in the Bayesian inference is computing the posterior PDF as the product function of the prior PDF and the likelihood function. When necessary, point estimates and different uncertainty statistics of the state $\boldsymbol{x}$ are then computed using the posterior distribution.

A strength of the Bayesian estimation paradigm is that the posterior can be used as a prior when a new measurement is received given that the new measurement is statistically conditionally independent of the previous measurement given $\boldsymbol{x}$. This leads to a recursive Bayesian measurement update procedure, where the previous measurements need not be stored because their information is contained in the posterior. Furthermore, Bayesian statistics intrinsically enables quality monitoring and fusion of different types of measurements because the posterior expresses not only the estimated value of the state but also information on the estimate's uncertainty.

2.2 Non-Gaussian state-space models

State-space models (SSMs) are used to model an unknown time-varying state vector $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ and noisy observations $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$. In this thesis only discrete-time SSMs are considered because even when the considered process is actually continuous, a discretisation of some type is usually required to do computing with digital computers. The subscript k means that the variable is related to the kth time instant t_k in the given time discretisation. The nonlinear statistical SSM with additive noises is

$$\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0), \tag{3a}$$
$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_{k-1}, \tag{3b}$$
$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{e}_k, \tag{3c}$$

where $p(\boldsymbol{x}_0)$ is the initial prior distribution, and the process noise $\boldsymbol{w}_k$ and the measurement noise $\boldsymbol{e}_k$ are typically assumed to be white stochastic processes that are mutually independent and independent

of $\boldsymbol{x}_0$. Equation (3b) is known as the process model, state transition model, or motion model of the state. Note that the state transition function $\boldsymbol{f}$, the measurement model function $\boldsymbol{h}$, as well as the probability distributions of $\boldsymbol{w}_k$ and $\boldsymbol{e}_k$ can be time-varying, but the subscripts are omitted here for brevity. In a more general case, the $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ can depend nonlinearly also on $\boldsymbol{w}_k$ and $\boldsymbol{e}_k$, but these models can, at least in principle, be handled similarly to the additive models by including the noise terms in the state [18].

An important special case of the SSM (3) is the linear–Gaussian SSM, where the model functions are linear, and the noise processes as well as the initial prior follow a Gaussian distribution as

$$\boldsymbol{x}_0 \sim \mathrm{N}(\boldsymbol{x}_{0|0}, \mathrm{P}_{0|0}), \tag{4a}$$

$$\boldsymbol{x}_k = \mathrm{A}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \quad \boldsymbol{w}_{k-1} \sim \mathrm{N}(\boldsymbol{\omega}, \mathrm{Q}), \tag{4b}$$

$$\boldsymbol{y}_k = \mathrm{C}\boldsymbol{x}_k + \boldsymbol{e}_k, \quad \boldsymbol{e}_k \sim \mathrm{N}(\boldsymbol{\epsilon}, \mathrm{R}), \tag{4c}$$

where A is the state transition matrix, $\boldsymbol{\omega}$ and Q are the process noise mean and covariance matrix, C is the measurement model matrix, and $\boldsymbol{\epsilon}$ and R are the measurement noise mean and covariance matrix.

The Bayesian filter is the following recursive formula that is based on Bayes' rule (1) and gives the filtering posterior $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$:

$$p(\boldsymbol{x}_0|\boldsymbol{y}_{1:0}) = p(\boldsymbol{x}_0), \tag{5a}$$

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) = \int p_{\boldsymbol{w}}(\boldsymbol{x}_k - \boldsymbol{f}(\boldsymbol{x}_{k-1}))\, p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})\, \mathrm{d}\boldsymbol{x}_{k-1}, \tag{5b}$$

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \frac{p_e(\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{x}_k))\, p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}{\int p_e(\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{x}_k))\, p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})\, \mathrm{d}\boldsymbol{x}_k}, \tag{5c}$$

where (5a) is the initialisation, (5b) is the prediction step, and (5c) is the update step. The functions $p_{\boldsymbol{w}}$ and p_e are the PDFs or probability mass functions of the noise terms in (3). The Bayesian smoother for the same SSM gives the smoothing posterior $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:K})$, where K is the index of the last time instant. Smoothing means using also measurements received after the estimation time instant, i.e. $K \geq k$. The Bayesian smoother is the backward recursion

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:K}) = p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \int \frac{p_{\boldsymbol{w}}(\boldsymbol{x}_{k+1} - \boldsymbol{f}(\boldsymbol{x}_k))\, p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{1:K})}{p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{1:k})}\, \mathrm{d}\boldsymbol{x}_{k+1}. \tag{6}$$

That is, the filter incorporates the state transition model to combine all measurement information up to the current time instant t_k, and the smoother uses the state transition model to include also the information of the future measurements in the posterior.

Unfortunately, when the model equations are nonlinear and/or the noises are non-Gaussian, the Bayesian filter and smoother generally become analytically intractable in the sense that posterior statistics such as the moments do not have closed-form expressions and that the number of parameters required to describe the posterior increases over time. That is, the resulting probability density cannot be defined using a limited and small number of parameters. For practical real-time or almost-real-time computational systems this is not acceptable, and instead there has to be a fixed set of parameters that the filter keeps updating over time in a recursive manner. Therefore, the key problem is often how to approximate the Bayesian filtering and/or smoothing formulas in a way that provides an acceptable compromise between accuracy and computational efficiency.

A basic approach is to compute the integrals numerically over a regular grid. However, the computational requirements of the grid method become prohibitive for most practical problems because the number of required grid points increases exponentially with the state dimension ("curse of dimensionality"). The following subsections review some commonly-used approximate Bayesian time-series estimation approaches for non-Gaussian SSMs.

2.3 Kalman-type methods

The Kalman filter (KF) algorithm [17] given in Algorithm 1 is the minimum-mean-square-error filter for linear SSMs (4) within the class of linear filters [23, Ch. 3.2]. The requirements for this are that the initial prior, the process noise and measurement noise have finite and known means $\boldsymbol{x}_{0|0}$, $\boldsymbol{\omega}$ and $\boldsymbol{\epsilon}$ and covariance matrices $\mathrm{P}_{0|0}$, Q and R, and that the matrix $\mathrm{C}\mathrm{P}_{k|k-1}\mathrm{C}^{\mathrm{T}} + \mathrm{R}$ is invertible at each time instant. If the process noise, measurement noise and the initial prior $p(\boldsymbol{x}_0)$ are Gaussian distributions, the filtering and smoothing posteriors are analytically tractable, the filtering posterior being the Gaussian distribution with the KF's output $(\boldsymbol{x}_{k|k}, \mathrm{P}_{k|k})$ as the mean and covariance matrix [23, Ch. 3.1]. When the SSM is non-Gaussian, a nonlinear

filter can provide a smaller root-mean-square error (RMSE) than the KF.

Input: model parameters $\{\boldsymbol{x}_{0|0}, \mathrm{P}_{0|0}, \mathrm{A}, \boldsymbol{\omega}, \mathrm{Q}, \mathrm{C}, \boldsymbol{\epsilon}, \mathrm{R}, K\}$,
measurements $\boldsymbol{y}_k$
Output: state estimate $\boldsymbol{x}_{k|k}$, $\mathrm{P}_{k|k}$ for $k = 0, \ldots, K$
for $k = 1 : K$ **do**
Prediction step
$\boldsymbol{x}_{k|k-1} \leftarrow \mathrm{A}\boldsymbol{x}_{k-1|k-1} + \boldsymbol{\omega}$
$\mathrm{P}_{k|k-1} \leftarrow \mathrm{A}\mathrm{P}_{k-1|k-1}\mathrm{A}^{\mathrm{T}} + \mathrm{Q}$
Update step
$\mathrm{K}_k \leftarrow \mathrm{P}_{k|k-1}\mathrm{C}^{\mathrm{T}}(\mathrm{C}\mathrm{P}_{k|k-1}\mathrm{C}^{\mathrm{T}} + \mathrm{R})^{-1}$
$\boldsymbol{x}_{k|k} \leftarrow \boldsymbol{x}_{k|k-1} + \mathrm{K}_k(\boldsymbol{y}_k - \boldsymbol{\epsilon} - \mathrm{C}\boldsymbol{x}_{k|k-1})$
$\mathrm{P}_{k|k} \leftarrow (\mathrm{I}_{n_x} - \mathrm{K}_k\mathrm{C})\mathrm{P}_{k|k-1}(\mathrm{I}_{n_x} - \mathrm{K}_k\mathrm{C})^{\mathrm{T}} + \mathrm{K}_k\mathrm{R}\mathrm{K}_k^{\mathrm{T}}$
end

Algorithm 1: Kalman filter

The smoothing posteriors of the linear–Gaussian SSM are also Gaussian, and their means and covariance matrices are given by the Rauch–Tung–Striebel smoother (RTSS) [24]. In RTSS, the backward recursion of Algorithm 2 is applied to the corresponding filter output. However, the RTSS is typically also used as an approximation when the filtering posterior is not exactly Gaussian but is approximated by a Gaussian density [18, Ch. 9.1].

Input: model parameters $\{\mathrm{A}, \boldsymbol{\omega}, \mathrm{Q}, K\}$; KF output $\{\boldsymbol{x}_{k|k}, \mathrm{P}_{k|k}\}$ for
$k = 0, \ldots, K$
Output: state estimate $\boldsymbol{x}_{k|K}$, $\mathrm{P}_{k|K}$ for $k = 0, \ldots, K$
for $k = K - 1 : -1 : 0$ **do**
$\mathrm{G}_k \leftarrow \mathrm{P}_{k|k}\mathrm{A}^{\mathrm{T}}(\mathrm{A}\mathrm{P}_{k|k}\mathrm{A}^{\mathrm{T}} + \mathrm{Q})^{-1}$
$\boldsymbol{x}_{k|K} \leftarrow \boldsymbol{x}_{k|k} + \mathrm{G}_k(\boldsymbol{x}_{k+1|K} - \mathrm{A}\boldsymbol{x}_{k|k} - \boldsymbol{\omega})$
$\mathrm{P}_{k|K} \leftarrow \mathrm{P}_{k|k} + \mathrm{G}_k(\mathrm{P}_{k+1|K} - \mathrm{A}\mathrm{P}_{k|k}\mathrm{A}^{\mathrm{T}} - \mathrm{Q})\mathrm{G}_k^{\mathrm{T}}$
end

Algorithm 2: Rauch–Tung–Striebel smoother's backward recursion

A major restriction of the KF and RTSS algorithms is that they are only applicable to models with linear (or affine) state transition and measurement model functions. For approximate filtering and smoothing with nonlinear SSMs, KF and RTSS are often extended by applying

approximate Gaussian moment-matching to the joint distribution of the Gaussian state distribution and its nonlinear image. This type of KF extensions include linearisation based algorithms such as extended Kalman filter (EKF) and iterated extended Kalman filter (IEKF) [25, Ch. 8.3], and numerical integration based algorithms such as cubature Kalman filter (CKF) [26]. These KF extensions can be computationally light compared to Monte Carlo algorithms and do not suffer from the curse of dimensionality, but their accuracy depends on the properties of the considered model.

2.4 Monte Carlo methods

For SSMs with non-Gaussian noises, a nonlinear filter can provide a lower mean-square-error than the KF. The particle filter (PF) [27] is a commonly used nonlinear approximation of the Bayesian filter whose estimate, under mild conditions [28], converges to the minimum-mean-square-error solution when the computational complexity approaches infinity.

The PF, also known as the sequential importance resampling, is an importance sampling based Monte Carlo algorithm, so it uses weighted random samples to approximate the filtering distributions. The random samples $\boldsymbol{x}_k^{(i)}$, $i = 1, \ldots, N_\mathrm{p}$ of the filter state, conventionally referred to as particles, are propagated in time by generating random numbers from a chosen probability distribution $q(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}, \boldsymbol{x}_{1:k-1}^{(i)})$ that is called the proposal distribution. The SSM (3) and the proposal distribution are then used to weight the particles. The unnormalised weight of the ith particle is given by the formula

$$\widetilde{W}_k^{(i)} = \frac{p_e(\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{x}_k^{(i)}))\, p_w(\boldsymbol{x}_k^{(i)} - \boldsymbol{f}(\boldsymbol{x}_{k-1}^{(i)}))}{q(\boldsymbol{x}_k^{(i)}|\boldsymbol{y}_{1:k}, \boldsymbol{x}_{1:k-1}^{(i)})} \cdot W_{k-1}^{(i)}, \tag{7}$$

where $W_{k-1}^{(i)}$ is the ith particle's weight after the previous update. The weighting "corrects" the distribution represented by the particle set to be the filtering posterior in the sense that the weighted average of the particles approximates the expectation value of any function $\boldsymbol{g}$ applied to the filtering posterior distributed random variable

$$\mathop{\mathbb{E}}_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[\boldsymbol{g}(\boldsymbol{x}_k)] \approx \sum_{i=1}^{N_\mathrm{p}} W_k^{(i)} \boldsymbol{g}(\boldsymbol{x}_k^{(i)}), \tag{8}$$

where $\mathbb{E}_p$ denotes the expectation value with respect to the probability distribution p, and $W_k^{(i)} = \widetilde{W}_k^{(i)} / \sum_{j=1}^{N_\mathrm{p}} \widetilde{W}_k^{(j)}$ denotes the particle weight normalised to sum to one. More precisely, the weighted average converges in L^4 sense to the true posterior expectation when $N_\mathrm{p} \to \infty$ provided that the weight update (7) is a bounded function of $\boldsymbol{x}_k^{(i)}$ [28].

An integral part of a PF is resampling [29]. In resampling the particle set is replaced by another uniformly-weighted particle set that approximates the same probability distribution. The resampling step reduces superfluous computation and mitigates the particle degeneracy, i.e. the weight concentrating to only a few particles, by probabilistically removing the particles with the lowest weights and replicating the particles with the highest weights. A commonly used resampling method is multinomial resampling, where the new particle set includes the particles with indices generated independently from the categorical distribution with the particle weights as probabilities. However, there are other methods that have varying properties with respect to how simple they are and how much additional Monte Carlo variance they produce [30]. Because resampling increases the Monte Carlo variance, the particles are typically resampled only when the particle degeneracy is high according to some criteria. A commonly used criterion is that the resampling is done when

$$\frac{1}{\sum_{i=1}^{N_\mathrm{p}} \left(W_k^{(i)} \right)^2} < \tau \cdot N_\mathrm{p}, \tag{9}$$

where $\tau \in (0,1)$ is a parameter, typically $\tau = 0.1$ or $\tau = 0.2$.

The details of the PF algorithm including the multinomial resampling are given in Algorithm 3.

The smoothing posterior distributions can in principle be approximated by storing the particle histories and considering a particle weight as a weight of the whole particle history. However, this particle smoother (PS) solution exhibits high particle degeneracy in the beginning of a long time interval, because the number of effective particles reduces due to resampling [31]. A solution is to keep all filtering particles for each time instant and reweight them in the backwards recursion step. A review of different particle smoothing algorithms is given in [18, Ch. 11].

Input: model parameters $\{p(\boldsymbol{x}_0),\, p_{\boldsymbol{w}},\, p_{\boldsymbol{e}},\, K\}$; filter parameters $\{N_\mathrm{p},$ $q(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}, \boldsymbol{x}_{1:k-1}^{(i)}),\, \tau \in (0,1)\}$
Output: state estimate $\boldsymbol{x}_{k|k}$ for $k = 0, 1, \ldots, K$
Initialisation
$\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0),\ W_0^{(i)} \leftarrow \frac{1}{N_\mathrm{p}},\ i = 1, \ldots, N_\mathrm{p},$
$\boldsymbol{x}_{0|0} \leftarrow \sum_{i=1}^{N_\mathrm{p}} W_0^{(i)} \boldsymbol{x}_0^{(i)}$
for $k = 1 : K$ **do**
  Particle propagation
  $\boldsymbol{x}_k^{(i)} \sim q(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}, \boldsymbol{x}_{1:k-1}^{(i)}),\ i = 1, \ldots, N_\mathrm{p}$
  Weighting
  $\widetilde{W}_k^{(i)} \leftarrow \frac{p_{\boldsymbol{e}}(\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{x}_k^{(i)}))\, p_{\boldsymbol{w}}(\boldsymbol{x}_k^{(i)} - \boldsymbol{f}(\boldsymbol{x}_{k-1}^{(i)}))}{q(\boldsymbol{x}_k^{(i)}|\boldsymbol{y}_{1:k}, \boldsymbol{x}_{1:k}^{(i)})} \cdot W_{k-1}^{(i)},\ i = 1, \ldots, N_\mathrm{p}$
  $W_k^{(i)} \leftarrow \frac{\widetilde{W}_k^{(i)}}{\sum_{j=1}^{N_\mathrm{p}} \widetilde{W}_k^{(j)}},\ i = 1, \ldots, N_\mathrm{p}$
  $\boldsymbol{x}_{k|k} \leftarrow \sum_{i=1}^{N_\mathrm{p}} W_k^{(i)} \cdot \boldsymbol{x}_k^{(i)}$
  Resampling when necessary
  if $1/\sum_{i=1}^{N_p} \left(W_k^{(i)}\right)^2 < \tau \cdot N_p$ **then**
    for $i = 1 : N_p$ **do**
      $j_i \sim \mathrm{cat}(W_k^{(1)}, \ldots, W_k^{(N_\mathrm{p})})$
      $\widetilde{\boldsymbol{x}}_k^{(i)} \leftarrow \boldsymbol{x}_k^{(j_i)}$
    end
    $\boldsymbol{x}_k^{(i)} \leftarrow \widetilde{\boldsymbol{x}}_k^{(i)},\ W_k^{(i)} \leftarrow \frac{1}{N_\mathrm{p}},\ i = 1, \ldots, N_\mathrm{p}$
  end
end

Algorithm 3: Particle filter with multinomial resampling

In addition to the number of particles, the tuning parameters of a PF are (a) the choice of the proposal distribution and (b) the choice of the resampling method [32]. In this thesis, the focus is on (a). In all importance sampling based algorithms the proposal distribution $q(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}, \boldsymbol{x}_{1:k-1}^{(i)})$ plays a vital role [29],[33, Ch. 3.4]. In a SSM and particle filtering context, a common choice is the dynamical model of the state

$$q_{\mathrm{BF}}(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}, \boldsymbol{x}_{1:k-1}^{(i)}) = p_{\boldsymbol{w}}(\boldsymbol{x}_k - \boldsymbol{f}(\boldsymbol{x}_{k-1}^{(i)})), \tag{10}$$

in which case the PF is called a bootstrap filter (BF). This method does not use any information of the newest measurement $\boldsymbol{y}_k$ in the particle propagation, so in some cases only a small portion of the

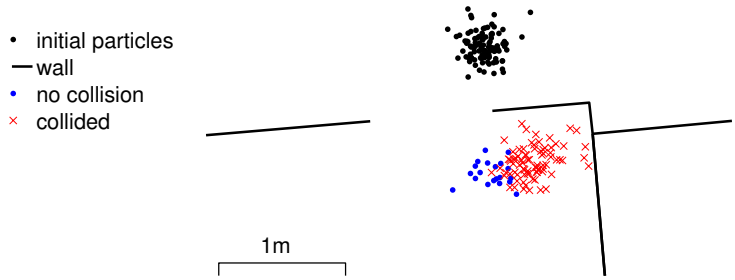


Figure 1: Particle degeneracy of the bootstrap filter in floor plan assisted indoor positioning. When the proposal's uncertainty about the direction is large, most particles get down-weighted by the floor plan likelihood because they crossed a wall, and so do not contribute to the estimate. The heading distribution in this example is $\mathrm{N}(290°, (10°)^2)$ and the footstep length is 1 m.

particles obtain significant weight after a measurement update. This means that computational effort can be wasted on particles that represent very improbable trajectories [29], or some measurements can be "lost" because there are no or few particles in the area where the measurement likelihood would increase the probability density. The latter problem can cause particle degeneracy. Figure 1 illustrates this problem in the floor plan assisted indoor positioning application: most particles take impossible trajectories, i.e. cross walls, and thus the estimation is eventually based on only a few particles. Serious particle degeneracy is typically a result of the likelihood function being very "narrow" compared to the extent of the particle set, e.g. when the measurement noise level is very low compared to the process noise level.

The best proposal distribution would be sampling directly from the posterior $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$. With this choice each particle would have an equal contribution to the result. This is typically not straightforward, and it is shown in [29] that the distribution

$$q_{\mathrm{opt}}(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}, \boldsymbol{x}_{1:k-1}^{(i)}) = p(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{x}_{k-1}^{(i)}) \tag{11}$$

is the optimal proposal distribution for the PF in the sense that it minimises the conditional variance of the particle weight given $\boldsymbol{y}_{1:k}$ and $\boldsymbol{x}_{1:k-1}^{(i)}$; in fact this variance is zero because the particle weighting

for the optimal proposal distribution

$$\widetilde{W}_k^{(i)} = p_e(\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{x}_{k-1}^{(i)})) \tag{12}$$

does not depend on the generated particle value $\boldsymbol{x}_k^{(i)}$. In most cases where a PF is applied, the optimal proposal distribution is not usable directly because the analytical form of q_{opt} is too complicated for efficient random number generation. However, the lesson to be learned is that the proposal should use as much information of the current time instant's measurement as possible.

A PF's tendency to suffer from particle degeneracy is highly dependent on the model that the filter is based on. In many practical situations, the used SSM can also be tuned. A means for reducing the degeneracy of a BF is to include some measurements in the process model instead of the measurement model and thus reduce the dissimilarity between the particle locations' distribution and the true posterior. For example, in indoor positioning the pedestrian dead reckoning (PDR) output is typically used in the target person's motion model and in moving the particles instead of weighting them. Furthermore, the floor plan map can be used to constrain the set of possible trajectories the person can take, and these constraints can be very precise measurements, thus increasing particle degeneracy. Section 3.1 and Publications [P2], [P3], [P4] discuss how to modify the motion model and the PF's proposal distribution in map-assisted indoor positioning such that particle degeneracy can be mitigated.

2.5 Analytical approximation based methods

Designing approximate filtering or smoothing algorithms requires compromising between complexity and accuracy. Kalman-type filters can be computationally light, easy to implement and tune, and do not suffer from the curse of dimensionality, but they can suffer from unpredictable errors when the SSM is non-linear and/or non-Gaussian. The PF is very flexible, applicable to almost any dynamical SSM at least in principle, but it can be computationally expensive and challenging to tune. Furthermore, the PF suffers from the curse of dimensionality: the performance can degrade significantly already with dimensions below ten, and the particle methods become practically useless when the state dimension is high. Roughly speaking, the

required number of particles increases exponentially as a function of the state dimensionality [34]. Furthermore, PFs are known to show poor performance in static models and dynamical models with low process noise levels [35].

Variational Bayes filter & smoother

The posterior distributions can also be approximated using an analytical approximation such that the number of parameters in the posterior does not increase over time. This can be a compromise between the computational ease and scalability of the KF and the flexibility of the PF. In Bayesian inference, analytical approximations are closed-form PDFs that approximate the true probability distribution. The methodology studied in this thesis is the variational Bayes (VB) approximation. The VB algorithm has similarities with the expectation–maximisation (EM) algorithm, but a difference is that the VB algorithm gives a whole probability distribution as an approximation instead of just a point estimate. This makes the VB algorithm suitable for approximate Bayesian time-series inference. The VB algorithm does not give the maximum a posteriori estimate like the EM, but the approximation is optimal in a sense explained in this section. This section also explains how VB based approximate Bayesian filters and smoothers can be applied to account for certain non-Gaussian features of the measurement noise distribution but still circumvent some of the PFs' undesired features.

Variational inference uses the calculus of variations to obtain a probability distribution $\hat{q}(\boldsymbol{x})$ that approximates the true distribution $p(\boldsymbol{x})$ optimally in the sense that the approximative distribution minimises a functional within a class of functions Ω [21],[36],[37],[22, Ch. 10.1],[38, Ch. 21.2]. In the VB approach this functional is the Kullback–Leibler divergence (KLD) from the approximative distribution to the exact posterior, as this choice provides convenient analytical formulas in certain situations. The KLD from distribution q to distribution p is defined as

$$\mathrm{KL}\big(q\,\big\|\,p\big) = \int q(\boldsymbol{x}) \log\!\left(\frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}\right) \mathrm{d}\boldsymbol{x}. \tag{13}$$

Within Bayesian inference, the VB method thus approximates the exact posterior $p(\boldsymbol{x}|\boldsymbol{y})$ with the probability distribution $\hat{q}(\boldsymbol{x})$ that is a

solution of the optimisation problem

$$\hat{q} = \arg\min_{q \in \Omega} \mathrm{KL}\big(q(\boldsymbol{x}) \big\| p(\boldsymbol{x}|\boldsymbol{y})\big). \tag{14}$$

where Ω is the set of restrictions that ensure that q will be a practical approximation for the considered application. In the case of recursive filtering, for example, a sensible requirement for the constraint set Ω is that the variational posterior approximation maintains the same functional form as the initial distribution of the filter recursion. For Gaussian filtering, for example, the approximation of the filtering posterior should be a Gaussian distribution.

The choice of the KLD $\mathrm{KL}\big(q \big\| p\big)$ ("the reverse KLD") as the objective functional is motivated by analytical convenience: the minimisation of $\mathrm{KL}\big(q \big\| p\big)$ requires evaluating expectations with respect to q, and the algorithm designer can control the functional form of q and thus enable analytical evaluation of the necessary expectations. Apart from the analytical convenience, the KLD $\mathrm{KL}\big(p \big\| q\big)$ ("the forward KLD") would in fact be a more natural choice as the optimisation objective because there the expectation in the KLD's definition is taken with respect to the exact distribution p. One implication of the reverse KLD is that VB tends to underestimate the uncertainty in the approximated distribution, since VB often simply gives low q-density to the areas where the approximation error is bound to be large due to the constraint set Ω. See further discussion about the KLDs and other divergence measures in [38, Ch. 21.2].

An often convenient choice of the restriction set Ω is to use the (structured) mean-field approximation [22, Ch. 10.1.1],[38, Ch. 21.3]. Assume that the state consists of vector-valued variables $\boldsymbol{x}_{1:M}$ with distribution $p(\boldsymbol{x}_{1:M})$. Then, the approximative distribution $\hat{q}$ is restricted to be of the form

$$\hat{q}(\boldsymbol{x}_{1:M}) = \prod_{i=1}^{M} \hat{q}_i(\boldsymbol{x}_i), \tag{15}$$

where each $\hat{q}_i(\boldsymbol{x}_i)$ is a probability distribution. That is, the approximation is restricted to the class of distributions where the components $\boldsymbol{x}_i$ are statistically mutually independent. This choice is motivated

by the identity

$$\begin{aligned}\arg\min_{q_i} \mathrm{KL}\Big(q_i(\boldsymbol{x}_i)\prod_{j\neq i} q_j(\boldsymbol{x}_j)\Big\|p(\boldsymbol{x}_{1:M})\Big)\\ = \frac{\exp\big(\mathbb{E}_{q_{1:i-1,i+1:M}}\big[\log p(\boldsymbol{x}_{1:M})\big]\big)}{\int \exp\big(\mathbb{E}_{q_{1:i-1,i+1:M}}\big[\log p(\boldsymbol{x}_{1:M})\big]\big)\,\mathrm{d}\boldsymbol{x}_i}\end{aligned} \tag{16}$$

that holds for any form of the distribution $p(\boldsymbol{x}_{1:M})$ for which the required expectations exist [38, Ch. 21.3]. Therefore, for many models the cyclic iteration of

$$q_i(\boldsymbol{x}_i) = \frac{\exp\big(\mathbb{E}_{q_{1:i-1,i+1:M}}\big[\log p(\boldsymbol{x}_{1:M}|\boldsymbol{y})\big]\big)}{\int \exp\big(\mathbb{E}_{q_{1:i-1,i+1:M}}\big[\log p(\boldsymbol{x}_{1:M}|\boldsymbol{y})\big]\big)\,\mathrm{d}\boldsymbol{x}_i} \tag{17}$$

converges to

$$\hat{q} = \arg\min_{q\in\Omega_{\mathrm{VB}}} \mathrm{KL}\big(q(\boldsymbol{x}_{1:M})\big\|p(\boldsymbol{x}_{1:M}|\boldsymbol{y})\big), \tag{18}$$

where

$$\Omega_{\mathrm{VB}} = \big\{q\,\big|\,\exists q_{1:M}: q(\boldsymbol{x}_{1:M}) \equiv \textstyle\prod_{i=1}^{M} q_i(\boldsymbol{x}_i)\big\}. \tag{19}$$

The cyclic iteration of (17) is called the (structured) mean-field VB algorithm [38, Ch. 21.5].

In the context of dynamical systems, VB approximations can be useful especially when the process noise and/or the measurement noise distributions can be expressed as marginals of conditionally Gaussian hierarchical models, i.e. Gaussian given a set of M vector-valued latent random variables $\boldsymbol{\theta}_{k,1:M}$ with prior $p(\boldsymbol{\theta}_{k,1:M})$ and some functions $\boldsymbol{\mu}$ and Σ:

$$\boldsymbol{\eta}_k|\boldsymbol{\theta}_{k,1:M} \sim \mathrm{N}\big(\boldsymbol{\mu}_{\boldsymbol{\eta}}(\boldsymbol{\theta}_{k,1:M}), \Sigma_{\boldsymbol{\eta}}(\boldsymbol{\theta}_{k,1:M})\big), \tag{20}$$

where $\boldsymbol{\eta}_k$ is either process or measurement noise. The linear and conditionally Gaussian SSM is thus

$$\boldsymbol{x}_0 \sim \mathrm{N}(\boldsymbol{x}_{0|0}, \mathrm{P}_{0|0}), \tag{21a}$$

$$\boldsymbol{x}_k = \mathrm{A}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \quad \boldsymbol{w}_{k-1}|\boldsymbol{\theta}_{k,1:M} \sim \mathrm{N}\big(\boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_{k,1:M}), \Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_{k,1:M})\big), \tag{21b}$$

$$\boldsymbol{y}_k = \mathrm{C}\boldsymbol{x}_k + \boldsymbol{e}_k, \qquad \boldsymbol{e}_k|\boldsymbol{\theta}_{k,1:M} \sim \mathrm{N}\big(\boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_{k,1:M}), \Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_{k,1:M})\big). \tag{21c}$$

In a filtering context, the VB iteration can then be applied to the set of $M+1$ vector-valued variables $\{\left[\begin{smallmatrix}\boldsymbol{x}_{k-1}\\ \boldsymbol{x}_k\end{smallmatrix}\right], \boldsymbol{\theta}_{k,1}, \ldots, \boldsymbol{\theta}_{k,M}\}$ using the factorisation

$$p(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, \boldsymbol{\theta}_{k,1:M} | \boldsymbol{y}_k) \approx q(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k) \prod_{i=1}^{M} q_i(\boldsymbol{\theta}_{k,i}). \tag{22}$$

The functions $\boldsymbol{\mu}_w$, Σ_w, $\boldsymbol{\mu}_e$, and Σ_e should be chosen such that the model admits suitable conjugacy properties: at each iteration the approximation q is to be a distribution for which the expectation in the updates (17) can be evaluated efficiently enough. Importantly, assuming that the previous filtering posterior $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$ is a Gaussian distribution, this VB approximation of the posterior $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$, i.e. $\int \hat{q}(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k)\, \mathrm{d}\boldsymbol{x}_{k-1}$, will be another Gaussian distribution. This makes the filtering algorithm recursive. The algorithm of the VB filter for linear and conditionally Gaussian SSMs are given in Algorithm 4, and the algorithm of the corresponding VB smoother is given in Algorithm 5. These algorithms are derived in Appendix B. The termination criterion for the VB iteration can be reaching a fixed number of iterations, a small enough change in the state estimate, or a small enough decrease in the reverse KLD, for example. The KLD-decrease criterion is equivalent to requiring a small enough increase in the variational lower bound [22, Ch. 10.1], which can be easier to evaluate.

The conditionally Gaussian distributions of the form (20) can be Gaussian scale mixtures where $\boldsymbol{\mu}_\eta$ is a constant, Gaussian location mixtures where Σ_η is a constant, or Gaussian location–scale mixtures where both $\boldsymbol{\mu}_\eta$ and Σ_η are non-constant functions. Examples of well-known distributions that can be expressed in the form of (20) are the t-distribution, the skew Gaussian distribution, the skew t-distribution, the Laplace distribution, the generalised hyperbolic distribution, the α-stable distribution, and the (discrete) Gaussian mixture. Table 1 lists hierarchical representations of some well-known multivariate conditionally Gaussian distributions. This thesis concentrates on the skew t-distribution, and VB approximations for filtering and smoothing for the SSMs with skew-t-distributed measurement noise are discussed in depth in Section 3.2 of this introduction and in Publications [P5], [P6] and Manuscript [M7].

Input: initial prior parameters $\boldsymbol{x}_{0|0}$, $\mathrm{P}_{0|0}$; model matrices A, C;
$\boldsymbol{\theta}$-dependent functions $\{\boldsymbol{\mu}_{\boldsymbol{w}}, \Sigma_{\boldsymbol{w}}, \boldsymbol{\mu}_{\boldsymbol{e}}, \Sigma_{\boldsymbol{e}}\}$; prior distribution $p_{\boldsymbol{\theta}}$
Output: state estimate $\boldsymbol{x}_{k|k}$, $\mathrm{P}_{k|k}$ for $k = 0, \ldots, K$
for $k = 1 : K$ **do**
  for $i = 1 : M$ **do**
    $q_i(\boldsymbol{\theta}_{k,i}) \leftarrow \tilde{q}_i(\boldsymbol{\theta}_{k,i})$ ▷VB initialisation, e.g
    $\tilde{q}_i(\boldsymbol{\theta}_{k,i}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{\theta}_k) \, \mathrm{d}\boldsymbol{\theta}_{k,-i}$
  end
  while *not converged* **do**
    $\overline{\mathrm{Q}}_{k-1} \leftarrow \mathbb{E}_{q_{1:M}}[\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1}]^{-1}$
    $\overline{\boldsymbol{\omega}}_{k-1} \leftarrow \overline{\mathrm{Q}}_{k-1} \, \mathbb{E}_{q_{1:M}}[\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k)]$
    $\overline{\mathrm{R}}_k \leftarrow \mathbb{E}_{q_{1:M}}[\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1}]^{-1}$
    $\overline{\boldsymbol{\epsilon}}_k \leftarrow \overline{\mathrm{R}}_k \, \mathbb{E}_{q_{1:M}}[\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k)]$
    $\boldsymbol{x}_{k|k-1} \leftarrow \mathrm{A}\boldsymbol{x}_{k-1|k-1} + \overline{\boldsymbol{\omega}}_{k-1}$
    $\mathrm{P}_{k|k-1} \leftarrow \mathrm{A}\mathrm{P}_{k-1|k-1}\mathrm{A}^{\mathrm{T}} + \overline{\mathrm{Q}}_{k-1}$
    $\mathrm{K} \leftarrow \begin{bmatrix} \mathrm{P}_{k|k-1} \\ \mathrm{P}_{k-1|k-1}\mathrm{A}^{\mathrm{T}} \end{bmatrix} \mathrm{C}^{\mathrm{T}} (\mathrm{C}\mathrm{P}_{k|k-1}\mathrm{C}^{\mathrm{T}} + \overline{\mathrm{R}}_k)^{-1}$
    $\overline{\boldsymbol{x}} \leftarrow \begin{bmatrix} \boldsymbol{x}_{k|k-1} \\ \boldsymbol{x}_{k-1|k-1} \end{bmatrix} + \mathrm{K}(\boldsymbol{y}_k - \overline{\boldsymbol{\epsilon}}_k - \mathrm{C}\boldsymbol{x}_{k|k-1})$
    $\overline{\mathrm{P}} \leftarrow (\mathrm{I} - \mathrm{K}[\,\mathrm{C}\ \mathrm{O}\,]) \begin{bmatrix} \mathrm{P}_{k|k-1} & \mathrm{A}\mathrm{P}_{k-1|k-1} \\ \mathrm{P}_{k-1|k-1}\mathrm{A}^{\mathrm{T}} & \mathrm{P}_{k-1|k-1} \end{bmatrix}$
    for $i = 1 : M$ **do**
      $q_i(\boldsymbol{\theta}_{k,i}) \propto \exp\Big(-\frac{1}{2} \mathbb{E}_{q_{-i}} \big[\log|\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)| + \log|\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)|$
      $+ (\boldsymbol{y}_k - \mathrm{C}\overline{\boldsymbol{x}}_{1:n_x} - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} (\boldsymbol{y}_k - \mathrm{C}\overline{\boldsymbol{x}}_{1:n_x} - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))$
      $+ \mathrm{tr}\big\{ \overline{\mathrm{P}}_{1:n_x,1:n_x} \mathrm{C}^{\mathrm{T}} \Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} \mathrm{C} \big\}$
      $+ ([\,\mathrm{I}\ -\mathrm{A}\,]\overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} ([\,\mathrm{I}\ -\mathrm{A}\,]\overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))$
      $+ \mathrm{tr}\big\{ \overline{\mathrm{P}} \begin{bmatrix} \mathrm{I} \\ -\mathrm{A}^{\mathrm{T}} \end{bmatrix} \Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} [\,\mathrm{I}\ -\mathrm{A}\,] \big\} - 2 \log p_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \big] \Big)$
    end
  end
  $\boldsymbol{x}_{k|k} \leftarrow \overline{\boldsymbol{x}}_{1:n_x}$
  $\mathrm{P}_{k|k} \leftarrow \overline{\mathrm{P}}_{1:n_x,1:n_x}$
end

Algorithm 4: VB filter for linear and conditionally Gaussian SSMs. The notation $\boldsymbol{\theta}_k$ is short-hand for $\boldsymbol{\theta}_{k,1:M}$, and the notation q_{-i} is short-hand for $\prod_{j \neq i} q_j(\boldsymbol{\theta}_{k,j})$.

Input: initial prior parameters $\boldsymbol{x}_{0|0}$, $\mathrm{P}_{0|0}$; model matrices A, C;
$\boldsymbol{\theta}$-dependent functions $\{\boldsymbol{\mu}_{\boldsymbol{w}}, \Sigma_{\boldsymbol{w}}, \boldsymbol{\mu}_{\boldsymbol{e}}, \Sigma_{\boldsymbol{e}}\}$; prior distribution $p_{\boldsymbol{\theta}}$
Output: state estimate $\boldsymbol{x}_{k|K}$, $\mathrm{P}_{k|K}$ for $k = 0, \ldots, K$
for $k = 1 : K$ **do**
  for $i = 1 : M$ **do**
    $q_i(\boldsymbol{\theta}_{k,i}) \leftarrow \tilde{q}_i(\boldsymbol{\theta}_{k,i})$ ▷VB initialisation, e.g.
    $\tilde{q}_i(\boldsymbol{\theta}_{k,i}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_{k,-i}$
  end
end
while *not converged* **do**
  for $k = 1 : K$ **do**
    $\overline{\mathrm{Q}}_{k-1} \leftarrow \mathbb{E}_{q_{1:M}}[\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1}]^{-1}$
    $\overline{\boldsymbol{\omega}}_{k-1} \leftarrow \overline{\mathrm{Q}}_{k-1} \mathbb{E}_{q_{1:M}}[\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k)]$
    $\overline{\mathrm{R}}_k \leftarrow \mathbb{E}_{q_{1:M}}[\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1}]^{-1}$
    $\overline{\boldsymbol{\epsilon}}_k \leftarrow \overline{\mathrm{R}}_k \mathbb{E}_{q_{1:M}}[\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k)]$
    $\boldsymbol{x}_{k|k-1} \leftarrow \mathrm{A}\boldsymbol{x}_{k-1|k-1} + \overline{\boldsymbol{\omega}}_{k-1}$
    $\mathrm{P}_{k|k-1} \leftarrow \mathrm{A}\mathrm{P}_{k-1|k-1}\mathrm{A}^{\mathrm{T}} + \overline{\mathrm{Q}}_{k-1}$
    $\mathrm{K} \leftarrow \mathrm{P}_{k|k-1}\mathrm{C}^{\mathrm{T}}(\mathrm{C}\mathrm{P}_{k|k-1}\mathrm{C}^{\mathrm{T}} + \overline{\mathrm{R}}_k)^{-1}$
    $\boldsymbol{x}_{k|k} \leftarrow \boldsymbol{x}_{k|k-1} + \mathrm{K}(\boldsymbol{y}_k - \overline{\boldsymbol{\epsilon}}_k - \mathrm{C}\boldsymbol{x}_{k|k-1})$
    $\mathrm{P}_{k|k} \leftarrow (\mathrm{I} - \mathrm{K}\mathrm{C})\mathrm{P}_{k|k-1}$
  end
  for $k = K : -1 : 1$ **do**
    $\mathrm{G}_{k-1} \leftarrow \mathrm{P}_{k-1|k-1}\mathrm{A}^{\mathrm{T}}\mathrm{P}_{k|k-1}^{-1}$
    $\boldsymbol{x}_{k-1|K} \leftarrow \boldsymbol{x}_{k-1|k-1} + \mathrm{G}_{k-1}(\boldsymbol{x}_{k|K} - \boldsymbol{x}_{k|k-1})$
    $\mathrm{P}_{k-1|K} \leftarrow \mathrm{P}_{k-1|k-1} + \mathrm{G}_{k-1}(\mathrm{P}_{k|K} - \mathrm{P}_{k|k-1})\mathrm{G}_{k-1}^{\mathrm{T}}$
    $\overline{\boldsymbol{x}} \leftarrow \begin{bmatrix} \boldsymbol{x}_{k|K} \\ \boldsymbol{x}_{k-1|K} \end{bmatrix}$
    $\overline{\mathrm{P}} \leftarrow \begin{bmatrix} \mathrm{P}_{k|K} & \mathrm{P}_{k|K}\mathrm{G}_{k-1}^{\mathrm{T}} \\ \mathrm{G}_{k-1}\mathrm{P}_{k|K} & \mathrm{P}_{k-1|K} \end{bmatrix}$
    for $i = 1 : M$ **do**
      $q_i(\boldsymbol{\theta}_{k,i}) \propto \exp\Big(-\frac{1}{2}\mathbb{E}_{q_{-i}}\big[\log|\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)| + \log|\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)|$
      $+ (\boldsymbol{y}_k - \mathrm{C}\overline{\boldsymbol{x}}_{1:n_x} - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))^{\mathrm{T}}\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1}(\boldsymbol{y}_k - \mathrm{C}\overline{\boldsymbol{x}}_{1:n_x} - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))$
      $+ \mathrm{tr}\big\{\overline{\mathrm{P}}_{1:n_x,1:n_x}\mathrm{C}^{\mathrm{T}}\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1}\mathrm{C}\big\}$
      $+ ([\mathrm{I} \; -\mathrm{A}]\overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))^{\mathrm{T}}\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1}([\mathrm{I} \; -\mathrm{A}]\overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))$
      $+ \mathrm{tr}\big\{\overline{\mathrm{P}}\begin{bmatrix} \mathrm{I} \\ -\mathrm{A}^{\mathrm{T}} \end{bmatrix}\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1}[\mathrm{I} \; -\mathrm{A}]\big\} - 2\log p_{\boldsymbol{\theta}}(\boldsymbol{\theta})\big] \Big)$
    end
  end
end

Algorithm 5: VB smoother for linear and conditionally Gaussian SSMs. The notation $\boldsymbol{\theta}_k$ is short-hand for $\boldsymbol{\theta}_{k,1:M}$, and the notation q_{-i} is short-hand for $\prod_{j \neq i} q_j(\boldsymbol{\theta}_{k,j})$.

Table 1: Hierarchical formulations of some conditionally Gaussian multivariate distributions. The notation $[\![\cdot]\!]$ is the Iverson bracket. For further information on the latent variables' distributions, see Appendix A.

Distribution	Parameters	Latents $\boldsymbol{\theta}$	Prior $p(\boldsymbol{\theta})$	$\mu_{\boldsymbol{\eta}}(\boldsymbol{\theta})$	$\Sigma_{\boldsymbol{\eta}}(\boldsymbol{\theta})$
Gaussian	$\boldsymbol{\mu}, \Sigma$			$\boldsymbol{\mu}$	Σ
t-distribution	$\boldsymbol{\mu}, \Sigma, \nu$	λ	$\mathrm{G}(\lambda; \frac{\nu}{2}, \frac{\nu}{2})$	$\boldsymbol{\mu}$	$\lambda^{-1}\Sigma$
skew Gaussian	$\boldsymbol{\mu}, \Sigma, \Delta$	$\boldsymbol{u}$	$\mathrm{N}_+(\boldsymbol{u}; \mathbf{0}, \mathrm{I})$	$\boldsymbol{\mu} + \Delta\boldsymbol{u}$	Σ
skew t-distribution	$\boldsymbol{\mu}, \Sigma, \Delta, \nu$	$\boldsymbol{u}, \lambda$	$\mathrm{N}_+(\boldsymbol{u}; 0, \lambda^{-1}\mathrm{I})\, \mathrm{G}(\lambda; \frac{\nu}{2}, \frac{\nu}{2})$	$\boldsymbol{\mu} + \Delta\boldsymbol{u}$	$\lambda^{-1}\Sigma$
Laplace	$\boldsymbol{\mu}, \Sigma$	λ	$\mathrm{G}^{-1}(\lambda; 1, 1)$	$\boldsymbol{\mu}$	$\lambda^{-1}\Sigma$
generalised hyperbolic	$\boldsymbol{\mu}, \Sigma, \boldsymbol{\delta}, \ell, d, a$	λ	$\mathrm{GIG}(\lambda; \ell, d, \sqrt{a^2 - \boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\delta}})$	$\boldsymbol{\mu} + \lambda\boldsymbol{\delta}$	$\lambda\Sigma$
Gaussian mixture	$\alpha^{(1:N_{\mathrm{GM}})}$, $\boldsymbol{\mu}^{(1:N_{\mathrm{GM}})}, \Sigma^{(1:N_{\mathrm{GM}})}$	ξ	$\mathrm{cat}(\xi; \alpha^{(1:N_{\mathrm{GM}})})$	$\sum_{i=1}^{N_{\mathrm{GM}}} [\![\xi = i]\!] \cdot \boldsymbol{\mu}^{(i)}$	$\sum_{i=1}^{N_{\mathrm{GM}}} [\![\xi = i]\!] \cdot \Sigma^{(i)}$

Other analytical approximations for non-Gaussian state-space models

The Bayesian filter also has an analytical recursive solution when the SSM is linear and the prior distributions of the random variables $\boldsymbol{x}_0$, $\boldsymbol{w}_k$, and $\boldsymbol{e}_k$ in (4) are the Gaussian mixture (GM) distributions

$$\mathrm{GM}(\boldsymbol{x}; \alpha^{(1:N_{\mathrm{GM}})}, \boldsymbol{\mu}^{(1:N_{\mathrm{GM}})}, \Sigma^{(1:N_{\mathrm{GM}})}) = \sum_{i=1}^{N_{\mathrm{GM}}} \alpha^{(i)} \cdot \mathrm{N}(\boldsymbol{x}; \boldsymbol{\mu}^{(i)}, \Sigma^{(i)}), \tag{23}$$

where $\alpha^{(1:N_{\mathrm{GM}})}$ are the mixture weights for which $\sum_{i=1}^{N_{\mathrm{GM}}} \alpha^{(i)} = 1$. The resulting filter is known as the Gaussian mixture filter (GMF) or the Gaussian sum filter [39]. The posterior is computed using the KF formulas for each mixture component, and there is a closed-form formula also for the posterior component weights. GMFs are often also used as approximate Bayesian filters by approximating the noise distributions or non-Gaussian posterior distributions as GMs. A challenge with GMFs is that if each time instant introduces a new GM-distributed noise term, the number of Gaussian posterior components increases exponentially over time. Therefore, mixture reduction algorithms [40] are used, and these algorithms can require heavy computations and introduce hard-to-analyse approximation errors to the posterior density.

The expectation propagation (EP) [41] is an approximate inference algorithm that approximates the true distribution with a chosen exponential family class of distributions. EP is usable for cases where the true distribution can be factorised such that each factor times the chosen exponential family distribution has efficient-to-evaluate expectations of the sufficient statistics of the chosen exponential family distribution. The intuition behind this algorithm is reducing the forward KLD from the true posterior to the approximation given the functional forms of each exponential family factor. Because the forward KLD is often a more natural objective than the reverse KLD, the EP algorithm is expected to outperform VB when applicable.

The integrated nested Laplace approximation (INLA) [42] is an algorithm for approximate Bayesian updating of conditionally Gaussian models. The algorithm is based on approximating the marginal posterior of the latent variables $p(\boldsymbol{\theta}|\boldsymbol{y})$ and numerically integrating the

Laplace approximations of the conditional state posterior $p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$. This can get computationally expensive when the number of latent variables is large.

Recently, several interesting estimation algorithms for non-Gaussian SSMs have been proposed [43]–[46]. In [43], [44] analytic approximation based filter and smoother for SSMs where both process and measurement noise are t-distributed are proposed. The filter maintains a multivariate t posterior approximation, and at each time instant the joint distribution of the posterior and the process noise is approximated as a multivariate t-distribution instead of two independent t-distributions, and the joint distribution of the filter prediction and the measurement noise is again approximated as multivariate t.

The article [45] proposes an approximate Bayesian measurement update for models where the prior distribution is an exponential family distribution. The algorithm is based on linearising the log-likelihood with respect to the sufficient statistics of the exponential family distribution. A challenge with this algorithm is that there are often several options in how to write the likelihood as a function of the sufficient statistics, and the choice should be made such that the linearised likelihood would resemble the exact likelihood as much as possible. No tool for measuring the goodness of the linearisation has been proposed.

In the generalised recursive update filter (GRUF) [46] a univariate non-Gaussian noise component $\tilde{e}$ is expressed as a nonlinear mapping of the standard Gaussian random variable e as $\tilde{e} = F^{-1}(\Phi(e))$, where Φ is the standard Gaussian cumulative distribution function (CDF) and F^{-1} is the true noise distribution's inverse CDF. The posterior of this nonlinear model is then approximated using a KF extension. This method is useful only with KF extensions whose linearisation point or sigma point set is propagated iteratively, as otherwise $\tilde{e}$ is anyway approximated just as Gaussian noise. The algorithm proposed in [46] is based on the recursive update filter, and in this case the iterations lack convergence results, and a divergence example is indeed shown in [46].

3 Estimation algorithms with positioning applications

3.1 Map constraints in indoor localisation

Map-assisted indoor positioning is based on sensor fusion algorithms that combine information from various independent measurement sensors. Typical measurement types include absolute position measurements, pedestrian dead reckoning (PDR), and map information. The absolute position measurements can be based on wireless networks such as wireless local area network (WLAN), Bluetooth low energy (BLE), or ultra-wideband (UWB). Common measurement types include received signal strength (RSS) and time of arrival (TOA), which are often used as range measurements. Global Navigation Satellite System (GNSS) measurements can be used whenever available, but in indoor spaces their accuracy is typically weak due to severe signal strength attenuation caused by the building structures [3]. PDR means measuring and computing the displacement from the previous position, and the PDR can be based on inertial sensors such as accelerometers and gyroscopes and other sensors such as magnetometers and barometers. When the sensors are hand-held, the PDR typically includes footstep detection as well as heading and altitude change measurements. The footsteps can be detected using the acceleration's norm [47]; the user's heading change during a footstep is then estimated from the gyroscope's angular velocity measurements projected to the horizontal plane, which is given by the accelerometer-based direction of gravity [48]. The footstep length can also be estimated using some heuristic criterion such as the step duration [49]. Map information is typically in some form of floor plan. A floor plan is a collection of walls and floor levels which limit the set of allowed transitions in the building: a footstep cannot be taken through a wall, a floor can only be changed through a staircase or elevator, etc.

The different measurement types complement each other. Absolute position measurements tend to be noisy but their accuracy does not degrade over time during the tracking. PDR, in contrast, often has high short-term accuracy and high data rate, but its long-term accuracy is weak due to sensor biases and lack of absolute position

information. Indoor map information provides accuracy improvements that can be crucial for the user experience: incorporation of the map information can increase the probability that the positioning system detects the correct floor level and room, which in many cases is a sufficient positioning accuracy.

Map-assisted outdoor positioning systems have been introduced already in the 1970s [50], [51]. In outdoor vehicle navigation, the map information is relatively straightforward to represent, since the users typically move only along the road network, and the road network has a natural representation as a collection of nodes and links with zero width. This graph can be used as a basis for matching the track estimate with the map using geometric or other methods, such as point-to-curve or curve-to-curve matching, or particle filters (PFs), for example [52, Ch. 3.1].

However, in indoor spaces there are more degrees of freedom. A typical indoor floor plan does not represent the allowed tracks, but is instead a collection of walls and doors as well as floor levels and staircases that actually represent the prohibited translations. A common probabilistic approach is to give zero or very small probability to those trajectories that cross the walls. In this thesis the PF based on this approach is referred to as the wall-collision PF. Another approach is to generate a graph whose links approximate the set of allowed trajectories such as corridors and entrances to rooms, and whose nodes join the links. In this thesis the PF based on the node–link approach is referred to as the graph-based PF.

In Publications [P1],[P2],[P3],[P4] we propose three different types of indoor positioning PF algorithms that use different levels of map and inertial sensor based directional information. In [P1] we propose a wall-collision PF where the particles are moved using the PDR measurements and reweighted using the absolute position measurements and map measurements. In [P4] we propose a wall-collision PF where the particle propagation can use PDR information but is also based on the floor plan, which enables the PF to work efficiently with more noisy heading change measurements because the proposal distribution can be expected to be closer to the posterior. In [P2] and [P3] we propose graph-based PFs to be used when there is no PDR available. These three PF algorithms are described in the

following subsections.

Wall-collision particle filter

In the wall-collision PF for indoor positioning [47], [53]–[65], [P1], the state contains at least the two position coordinates $\boldsymbol{r}_k$ and the heading φ_k. The PDR output is used in the particle propagation and the motion model. The heading change measurement $\overline{\Delta}_k$ along with a white noise term is added to the heading of each particle. The heading is then used for propagation of the position coordinates of the same particle. The length of the displacement can be obtained from a footstep length measurement, or it can be a variable in the PF's state as discussed in the following paragraph. If no other motion model is used, this is a bootstrap PF, since the PDR model is used as both proposal distribution and process model distribution. After the map-matching phase, possible wireless network based absolute position measurements, such as measurements of WLAN access points' RSSs as in [P1], are used to update the particle weights. Finally, the particle weights are normalised and resampling and weight equalisation are performed if necessary. The particles are again used as a starting point for the next time instant.

In Publication [P1], footstep length measurements are not used; instead the footstep length ℓ_k is a component in the PF's state, assuming that the footstep length is a random-walk process. The state transition model is thus

$$\varphi_k = \varphi_{k-1} + \overline{\Delta}_{k-1} + w^{\overline{\Delta}}_{k-1} \tag{24a}$$

$$\ell_k = \ell_{k-1} + w^{\ell}_{k-1} \tag{24b}$$

$$\boldsymbol{r}_k = \boldsymbol{r}_{k-1} + \ell_k \cdot \begin{bmatrix} \cos\varphi_k \\ \sin\varphi_k \end{bmatrix} + \boldsymbol{w}^{r}_{k-1}. \tag{24c}$$

where $w^{\overline{\Delta}}_{k-1}$, w^{ℓ}_{k-1}, and $\boldsymbol{w}^{r}_{k-1}$ are noise terms that can be modelled as Gaussian or t-distributed. The PF-learned footstep length has the advantage that it is better able to recover from erroneous estimation than a system that uses measured step lengths with a small variance [66]. This model also circumvents the requirement to learn a user-specific parameter, a requirement that is characteristic to the heuristics proposed in [49]. However, if the footstep length estimation is correcting large errors in position, the footstep length estimate

will converge to an incorrect value, which might cause overshooting at future time instants [P1]. Furthermore, sudden stops or other changes in the footstep length are not modelled by this approach; this problem can to some extent be mitigated using the multiple model approach such as that of [P2], where a small portion of particles are propagated using the assumption that the user has stopped.

In Publication [P1] the user's altitude is also a state variable in the PF. The altitude model is based on barometer based altitude change measurements and map-based knowledge of where it is possible to change floor. Wireless networks can also provide absolute altitude or floor information [67].

In [47], [53]–[55], [57], [62], [68], [69] a particle's weight is set to zero whenever the particle crosses a wall, whereas in [59], [60], [64], [P1] wall-crossing particles' weights are multiplied by a small number $\epsilon/(1-\epsilon)$ where ϵ, $0 \leq \epsilon \ll 1$, is the probability that in the map there is an erroneous wall that crosses the line segment between two successive positions of the particle. The latter solution increases robustness against map errors, but it is problematic if all the particles penetrate a wall within a few successive steps [P1]. In such a case, the estimate trajectory will also cross the wall and the estimate will lag slightly [P1].

A problem in the wall-collision PF is that recovery from relatively small positioning errors can be slow if there are wall constraints between the true position and the particle cloud. This is demonstrated in Figure 2. In Publication [P1] it is proposed that a computationally light Kalman-type fallback filter is run parallel to the PF as both divergence monitoring and reinitialisation tool, such that when there are no particles close to the fallback estimate, some of the particles are re-located using the fallback estimate. A recommended fallback filter is the PDR-KF of [66] or its robustified version [P1], which fuses PDR and absolute position measurements' information using a linear state-space model (SSM). This fallback filter does not use map information, which is an advantage because the goal is to avoid getting stuck behind the walls in the map. The latest absolute position measurement is used for divergence monitoring in [32], [70], but using the filter estimate as the fallback solution can be more robust against outlier measurements in both divergence monitoring and

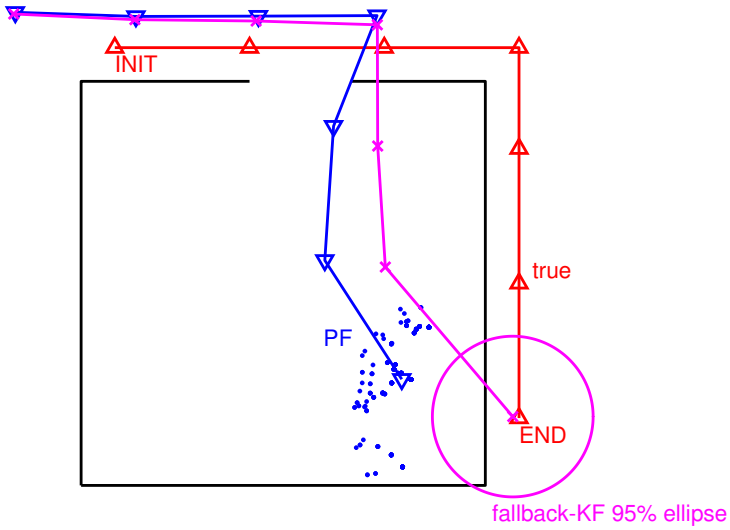


Figure 2: The wall-collision PF can get stuck behind the wall constraints because of small positioning error earlier; the KF-based fallback estimate is used to make the PF recover by reinitialising some particles. In this simulated example the motion model (24) with noisy heading change measurements, wall constraints, and an absolute position measurement at the last time instant are used.

reinitialisation because the filter can to some extent accommodate a single outlier.

A problem in heading change estimation is that the gyroscope noise is not actually white but has a constant bias part. For simplicity, it is often assumed that the white noise part dominates enough so that map-matching fixes the bias in the long run, but the accumulation of the bias can also be estimated by the particle filter [64], [65]. In [57] a KF is used for accurate inertial sensor modelling, and the output of the Kalman filter is given as a step measurement for the particle filter. This approach allows estimation of inertial sensor biases with the KF, which is more efficient than estimation with PF, since the sensor bias model is linear. The authors also point out that the sampling frequency of the KF can be considerably higher than that of the PF, because the Kalman filter is computationally light.

Article [71] does not use PDR at all, but instead uses a constant ve-

locity model as importance and state transition distribution. This approach assumes the user's acceleration to be a white Gaussian process. This assumption is not justified for pedestrians indoors because sudden changes in the user's velocity are possible: the user can stop or make 90- or 180-degree turns relatively frequently. Other alternatives would be random-walk or Lévy-flight models [72]. Article [53] uses a model in which the heading follows the uniform distribution, but the footstep length is estimated from the accelerometer data. With these models only little information is used in particle propagation. This can result in inefficient particle filters because large numbers of particles collide with the walls and become almost meaningless for the estimation.

Wall-collision particle filter using map in proposal distribution

Maps can also be used for formulating the proposal distribution and for state transition modelling. This can be advantageous especially when the PDR output has high noise levels, because a large portion of particles penetrating a wall can result in sample degeneracy of the PF. This means that only a few particles have almost all weight. One way to avoid sample degeneracy is to use a proposal distribution that is as close to the final posterior or as close to the optimal proposal distribution as possible, as discussed in Section 2.4 of this Introduction chapter. With the map constraint measurements this means that the algorithm should minimise the wall collisions rate. A constraint for this minimisation is, however, that a particle cloud for which the PDR and map measurements contradict should eventually vanish.

Publication [P4] proposes a PF where the proposal distribution combines PDR and map information. In the indoor map, the building is covered by a grid, where each grid point is assigned an angular probability density function (PDF), which is computed off-line. Following [60], the probability of a direction is a non-decreasing function of the distance to the closest wall in the considered direction. Because the wall distances are computed in a discrete direction space, the angular PDF is a piece-wise constant function. An example of such an angular PDF is presented in Figure 3.

The proposal distribution is now defined to be the product of the

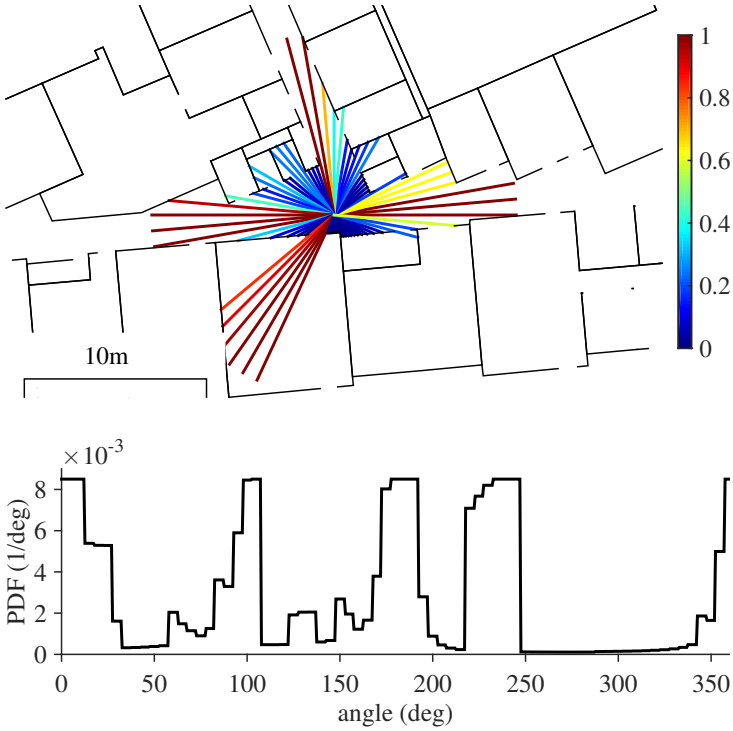


Figure 3: The angular PDF proposed in [P4]. In the upper figure the lengths of the radial line segments represent the distances to the closest wall, while the colours represent the angular PDF with 0.7 m footstep length. The figure is quoted from [P4].

map-based angular PDF and the PDR and particle heading based Gaussian distribution, which becomes a mixture of truncated Gaussian distributions

$$q(\varphi_k) \propto \sum_{j=1}^{N_\alpha} w_j^i \, \mathrm{N}_{[\alpha_j-\pi/N_\alpha, \alpha_j+\pi/N_\alpha]}(\varphi_k; \varphi_{k-1}^i + \overline{\Delta}_{k-1}, \mathrm{var}[v_{k-1}^{\overline{\Delta}}]) \quad (25)$$

where α_j, $j = 1, 2, \ldots, N_\alpha$ is the jth value in the direction discretisation, φ_{k-1}^i is the heading value of the ith particle, and

$$w_j^i = s(\alpha_j, \boldsymbol{r}_k^{(i)}) \cdot \mathbb{P}(\widetilde{\varphi} \in [\alpha_j - \pi/N_\alpha, \alpha_j + \pi/N_\alpha]), \quad (26)$$

where $\widetilde{\varphi} \sim \mathrm{N}(\varphi_{k-1}^i + \overline{\Delta}_{k-1}, \mathrm{var}[v_{k-1}^{\overline{\Delta}}])$, and $s(\alpha_j, \boldsymbol{r}_k^{(i)})$ is the angular PDF value in the direction α_j from the grid point that is closest to the position $\boldsymbol{r}_k^{(i)}$. The computer simulations presented in [P4] show that when

the heading change measurement $\overline{\Delta}_k$ has a high noise level, the proposed algorithm improves the positioning accuracy, or reduces the required number of particles and reduces the resampling rate compared to the conventional wall-collision PF. When the PDR is very precise, the proposed algorithm is slightly less accurate and computationally more expensive than the conventional wall-collision PF.

A further detail studied in [P4] is whether the angular PDF should be part of the state transition model (in addition to being part of the proposal distribution) or not. This choice affects the particle weighting. The tests show that when the track is a typical pedestrian's indoor path, it is advantageous to include the map-based angular PDF both in the state transition model and in the proposal distribution.

Kaiser et al. [60] propose a map-based state transition model that is computed using the laws of gas effusion. The aim is to assign more probability to the directions where the distance to the closest wall is longer. However, in [60] the map-based state transition model does not affect the proposal distribution but is only used in the particle weighting. In articles [56], [58] a particle's propagation is repeated until the wall collision is avoided, and the particle is weighted down only after several trials of getting a non-wall-colliding step. This is a heuristic process and can be computationally heavy. One difference to the approach of [P4] is that the estimation is only affected when some particles are actually colliding, while the approach of [P4] uses the map information already some steps before collisions. The latter approach is based on the assumption that the map-based angular PDF is indeed useful information along with the PDR on how people usually move in indoor space.

Graph-based particle filter

If a graph representation of the floor plan is available and reliable PDR is not used, the graph-based indoor positioning PF [71], [73]–[77] is one way to improve the sample efficiency of the PF, as demonstrated in Publications [P2] and [P3]. In the graph-based floor plan, the most expected user paths are condensed into links (edges) of a graph. The links are undirected line segments that are connected by nodes (vertices) according to their real-world connectivity. In

Figure 4: A graph representation of an indoor map is shown in black, the actual floor plan in grey, and the graph-based PF's particles in red.

a typical graph-based PF, the particles only move on the links as demonstrated in Figure 4. This is the key for the improved sample efficiency compared to the wall-collision PF, as the number of degrees of freedom in the particle's movement is reduced: the model is specified by motion in a one-dimensional line segment and a link transition rule. In the graph-based approach the particles do not cross walls, so the map is not used to downweight particles. The node–link model's resolution should be comparable with or higher than that of the other position measurements, as otherwise the graph modelling will only reduce the positioning accuracy [P2].

At each node there is a probabilistic link transition model for each arriving link, according to which the next link is chosen for each particle arriving at the node via this arriving link. In Publication [P2] a link transition rule is proposed that is based on the quantity that we refer to as the total link length (TLL). The TLL measures the total length of the node–link system that is accessible by choosing the considered link alternative. The TLL can be automatically precomputed off-line for each end of each link. Only the links that are closer to the current node than some maximum distance are considered. This is a more natural motion model for indoor spaces than a uniformly random link transition model, because in any building there are main passageways such as corridors and halls, where the user is more likely to continue on the passageway than to choose a

Figure 5: A graph representation of an indoor map is shown in black, the open space polygons with light blue, the actual floor plan in grey, and the graph-based PF's particles in red.

certain room. In fact, computing the parameters of the TLL-based link transition model can be considered as an automated method for detecting the main passageways of a building without any prior knowledge of the functions of the different spaces. However, if there are data about the frequencies with which people usually choose the links, this information can be added to the link transition model as discussed in [P2]. In Publication [P2], the TLL-based link transition rule is also compared with that of [71], where the links resulting in smallest changes in direction are given the most probability. This model can also be less realistic than the proposed model, as corridors can take 90-degree turns, for example. The presented real-data tests show that the TLL-based link transition model is a useful model especially when the absolute position measurements have a low update frequency, say, with larger than 5-second intervals.

The pure graph representation does not model well large open areas. A solution is to add open space polygons that the particles can enter and exit and where the particles are allowed to move in a two-dimensional space [78], [79]. An example of such a representation is shown in Figure 5. In Publication [P3] the TLL-based link transition rule is extended to maps represented as a combination of graphs and open space polygons.

The graph-based indoor positioning PF is not the most natural solution when PDR measurements are used, because the true heading

most of the time does not exactly correspond to the link direction. In [P2], the PF fuses only wireless network based absolute position measurements with the map. However, in article [77] a rudimentary inertial sensor based turn detection is used in determining the link transition probabilities. Article [77] proposes using regular graphs, where the building is covered by a square grid, and each grid point is a node. Such graphs are straightforward to generate using the floor plan and enable better utilisation of the possible PDR's measurement resolution, but storing the full link map can require a lot of memory, and a dense grid can again reduce the PF's sample efficiency.

Some authors do not use Monte Carlo at all but instead use grid filters where the user position's probability is computed at each node of the graph, which typically is a regular grid [80], [81]. Grid filters have drawbacks: the grid can have a lot of points with very small probabilities that cause computational overhead, and grids that cover large areas can require a lot of memory. Furthermore, the computational load of the grid filters increases exponentially with the state dimensionality, making grid filters mostly unusable in practice for state dimensionalities above two or three.

Comparison of map-constraint particle filters

To compare the performance of the wall-collision PF, the graph-based PF, and the wall-collision PF with map and PDR based proposal distribution, an extension of the simulation scenario of [P4] is presented here. In this test, three test tracks were designed in a campus building of Tampere University of Technology. The tracks are depicted in Fig. 6. Track 1 stays in corridors, track 2 goes through doors and visits rooms, while track 3 includes motion in open spaces and transition from an open space to a corridor. Similarly to [P4], the test tracks' paths were defined by hand, while the footstep lengths ℓ_k were simulated from the model

$$v_0 \sim \mathrm{N}(0, 0.2718^2), \tag{27a}$$

$$\begin{bmatrix} \ell_k \\ v_k \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} 0.7 + 0.9748\, v_{k-1} \\ 0.95\, v_{k-1} \end{bmatrix}, \begin{bmatrix} 0.3208 & 0.4751 \\ 0.4751 & 0.9504 \end{bmatrix} \right), \tag{27b}$$

and the footstep length measurements $\overline{\ell}_k$ and heading change measurements $\overline{\Delta}_k$ were generated by

$$\overline{\ell}_k \sim \mathrm{N}\big(\|\boldsymbol{r}_k - \boldsymbol{r}_{k-1}\|, (0.7\,\mathrm{m} \cdot 2^\circ \tfrac{\pi}{180^\circ})^2\big) \tag{28a}$$

$$\overline{\Delta}_k \sim \mathrm{N}\big(\Delta_k - 0.3^\circ \tfrac{\pi}{180^\circ}, q^{\overline{\Delta}} - (0.3^\circ \tfrac{\pi}{180^\circ})^2\big), \tag{28b}$$

where $\boldsymbol{r}_k$ is the 2-dimensional position, Δ_k is the true heading change in radians, and $q^{\overline{\Delta}}$ is a parameter. The model includes a gyro bias of -0.3° per step. The absolute position measurements $\boldsymbol{z}_k$ were generated by

$$\boldsymbol{z}_k \sim \mathrm{N}(\boldsymbol{r}_k, (4\,\mathrm{m})^2 \cdot \mathrm{I}_2). \tag{29}$$

As stated in [P2], the motion model has greater influence on the accuracy when the position measurements' time interval is quite large. Thus, in this test the interval is relatively large; the measurements are received every 20 steps, which roughly corresponds to measurement intervals between 10 and 20 seconds.

Figure 6 presents the root-mean-square error (RMSE) of each filter averaged over 1000 Monte Carlo replications with different values. The average RMSEs are shown as a function of the heading change measurement variance $q^{\overline{\Delta}}$, and in the case $q^{\overline{\Delta}}$=Inf neither footstep length nor heading change measurements are used. KF is the PDR-KF used in [P4], and each PF uses the KF as the fallback solution. Wall-collision PF is the "PFc" filter in [P4], MPD-PF is the wall-collision PF with map and PDR based motion model and proposal distribution denoted as PF2 in [P4], and graph PF is the total link length based filter proposed in [P3]. Graph PF uses the PDR only in the fallback estimate and to detect steps and stops. In graph PF each particle's footstep length value is propagated using the model (27). When PDR is not used, wall-collision PF and MPD-PF also generate the footstep lengths using the model (27), while the heading is generated from the uniform distribution in wall-collision PF and from the map-based angular PDF in MPD-PF. Each PF uses 400 particles.

Figure 6 shows that when the heading change measurement variance is very small, say $(0.5^\circ)^2$, the map information makes little difference, so the KF provides the lowest RMSE. With somewhat noisier heading change measurements, the wall-collision PF improves the accuracy, especially with Track 1 that stays in corridors. When the heading

change measurements are very noisy, with variance between $(5°)^2$ and $(20°)^2$, using the map information also in the motion model and the proposal distribution makes the PF more accurate at the cost of somewhat increased computational burden per particle. When the heading change measurements' variance is $(45°)^2$ or there is no PDR at all, the graph-based PF provides the lowest RMSE. This can be explained with the improved sample efficiency in the PF.

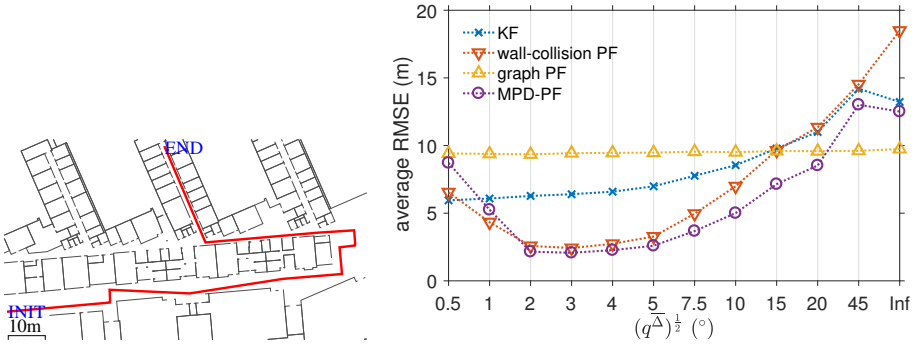
3.2 State-space models with skewed and heavy-tailed measurement noise distribution

Skewness of a probability distribution is a measure of asymmetry of the distribution with respect to its median. Intuitively, positive (negative) skewness means that large positive (negative) deviations from the median are more likely than large negative (positive) deviations. The most commonly used formal definition for skewness for a univariate random variable x with three finite moments is the third standardised central moment

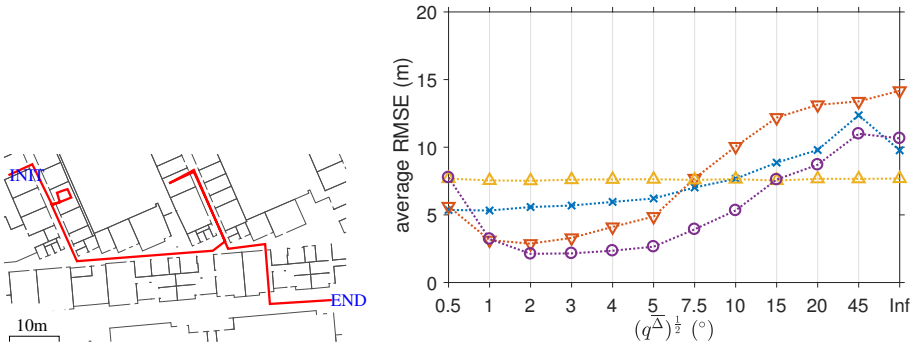
$$\text{skewness}[x] = \frac{\mathbb{E}[(x - \mathbb{E}[x])^3]}{\mathbb{E}[(x - \mathbb{E}[x])^2]^{\frac{3}{2}}}. \tag{30}$$

This formal measure is not defined for distributions whose required moments are not finite, even though the intuitive skewness exists. For multivariate distributions, skewness can be defined for each univariate marginal random variable. Examples of negatively-skewed, symmetric, and positively-skewed probability distributions are shown in Figure 7.

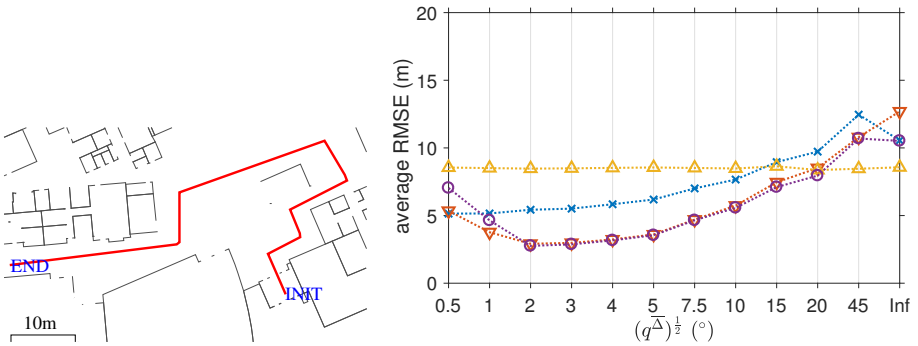
Skewness in real-world phenomena is typically caused by physical or other restrictions, which diminish or remove the probability of one side of the distribution. For example, a time of arrival (TOA) measurement is softly bounded from below by some maximum speed restrictions while disruptions can occasionally cause late arrivals, which makes the TOA's probability distribution positively skewed [P6]. RSS measurement is an example of a physical field that has an empty space value in each point of the space, and disruptive obstacles can usually only reduce the strength of the field; this results in negative skewness [82],[83].



(a) Track 1



(b) Track 2



(c) Track 3

Figure 6: Comparison of map-constraint PFs. MPD-PF is the wall-collision PF that uses the map and PDR based motion model and proposal distribution. Track 1 tests behaviour in corridors, track 2 tests doors and rooms, and track 3 tests open space. The track figures are quoted from [P4].

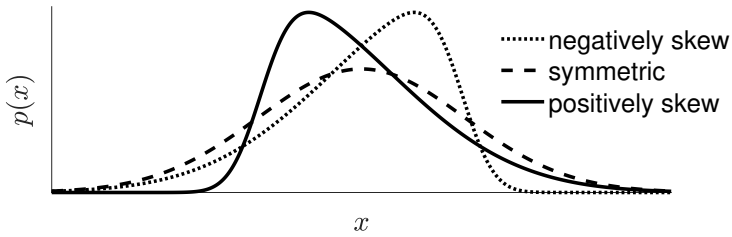


Figure 7: Examples of negatively-skewed, symmetric, and positively-skewed probability distributions

Intuitively speaking, the heavy-tailedness of a probability distribution measures how large a portion of the distribution's variance comes from the tails. Within statistics, heavy-tailedness is commonly referred to as kurtosis. A univariate heavy-tailed distribution has thus a narrower peak than a light-tailed distribution with equal variance, as the increase of variance due to the probability mass far from the peak is compensated in the peak area. One formal definition of (excess) kurtosis of a univariate random variable x is the fourth standardised central moment

$$\text{kurtosis}[x] = \frac{\mathbb{E}[(x - \mathbb{E}[x])^4]}{\mathbb{E}[(x - \mathbb{E}[x])^2]^2} - 3, \tag{31}$$

where the subtraction of 3 sets the excess kurtosis of any univariate Gaussian distribution to zero. The distributions with strictly positive excess kurtosis are called leptokurtic distributions, and these distributions can be defined to have heavier tails than the Gaussian distributions. A comparison of heavy-tailed and light-tailed probability distributions is shown in Figure 8.

Heavy-tailed distributions model quantities that have a definite trend to be close to a center value while there is also a significant probability that the value is arbitrarily far from the center value. For example, almost any real-world measurement system in engineering produces heavy-tailed error distributions due to a possibility of sensor failures and other anomalies [20]. Basing statistical estimation on a heavy-tailed error distribution makes the estimation more robust against outlier measurement, i.e. measurements that are far from the main bulk of measurements [84].

In Publications [P5], [P6] and Manuscript [M7], modelling of skew-

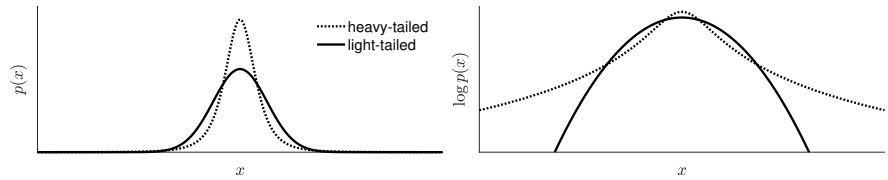


Figure 8: Examples of symmetric heavy-tailed and light-tailed probability distributions with equal variances. The heavy-tailed distribution's peak is narrower but tails heavier. The PDFs in the left and the log-PDFs of the same distributions in the right.

ness and heavy-tailedness is applied to positioning using distance measurements that are based on TOA of an electromagnetic signal. Such ranging measurements are currently used in the most prevalent positioning technology, GNSS, and also by emerging UWB radio based positioning systems.

Publications [P5], [P6] and Manuscript [M7] propose approximative Bayesian filtering and smoothing algorithms for linear SSMs with skew-t-distributed measurement noise

$$\boldsymbol{x}_0 \sim \mathrm{N}(\boldsymbol{x}_{0|0}, \mathrm{P}_{0|0}), \tag{32a}$$

$$\boldsymbol{x}_k = \mathrm{A}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \quad \boldsymbol{w}_{k-1} \sim \mathrm{N}(\boldsymbol{0}, \mathrm{Q}), \tag{32b}$$

$$\boldsymbol{y}_k = \mathrm{C}\boldsymbol{x}_k + \boldsymbol{e}_k, \quad [\boldsymbol{e}_k]_i \sim \mathrm{ST}(\mu_i, r_i, \delta_i, \nu_i), \tag{32c}$$

where $\mathrm{ST}(\mu_i, r_i, \delta_i, \nu_i)$ is the (univariate) skew t-distribution [85]–[87] with location parameter $\mu_i \in \mathbb{R}$, squared-spread parameter $r_i \in \mathbb{R}_+$, skewness parameter $\delta_i \in \mathbb{R}$, and degrees-of-freedom parameter $\nu_i \in \mathbb{R}_+$. The skew t-distribution was chosen as the basis of the algorithms because it has favourable conjugacy properties, because it is flexible, modelling both skewness and heavy-tailedness, because it has only four parameters, and because it is well-known within the statistics community. The favourable conjugacy properties include the fact that the skew t-distribution can be represented as a continuous location–scale mixture of Gaussian distributions. This enables applying the variational Bayes (VB) filter and smoother for linear conditionally Gaussian SSMs in Algorithms 4 and 5. Based on the formulation as a conditionally Gaussian random variable according to Table 1, the univariate random variable $z \sim \mathrm{ST}(\mu, r, \delta, \nu)$ has the

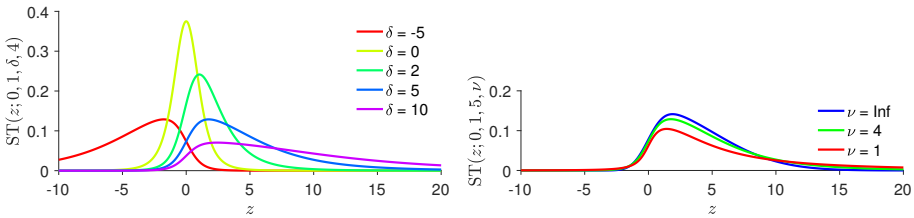


Figure 9: Skew t-distribution with different values of the skewness parameter δ (left) and the degrees-of-freedom parameter ν (right)

hierarchical formulation [88]

$$z|u,\lambda \sim \mathrm{N}(\mu+\delta u, \lambda^{-1} r) \tag{33a}$$

$$u|\lambda \sim \mathrm{N}_+(0, \lambda^{-1}) \tag{33b}$$

$$\lambda \sim \mathrm{G}(\tfrac{\nu}{2}, \tfrac{\nu}{2}), \tag{33c}$$

where $\mathrm{N}_+(\cdot,\cdot)$ is the Gaussian distribution truncated into positive numbers (or positive orthant in multivariate case). This results in the PDF $\mathrm{ST}(z;\mu,r,\delta,\nu) = \int p(z,u,\lambda)\,\mathrm{d}u\,\mathrm{d}\lambda$ whose analytical form is given in Appendix A. Figure 9 illustrates the effect of the skewness parameter δ and the degrees-of-freedom parameter ν to the shape of the skew t-distribution: The larger the magnitude of δ is, the more skewed the distribution, and the sign of δ determines the direction of skewness. The smaller the parameter ν is, the more heavy-tailed the distribution. A skew t-distribution with $\delta=0$ is Student's t-distribution, a skew t-distribution with $\nu\to\infty$ becomes a skew Gaussian distribution, and a skew t-distribution with $\delta=0$ and $\nu\to\infty$ becomes a Gaussian distribution.

In [P5] the VB iteration is derived for the SSM where the measurement is a vector of conditionally independent univariate skew-t-distributed measurements, while the VB based algorithm proposed in [M7] also applies to models with the multivariate skew-t-distributed measurement noise. For the multivariate skew-t-distributed random variable $\boldsymbol{z} \sim \mathrm{MVST}(\boldsymbol{\mu}, \mathrm{R}, \Delta, \nu)$ in $\mathbb{R}^{n_z}$, the hierarchical formulation is [88]

$$\boldsymbol{z}|\boldsymbol{u},\lambda \sim \mathrm{N}(\boldsymbol{\mu}+\Delta\boldsymbol{u}, \lambda^{-1}\mathrm{R}) \tag{34a}$$

$$\boldsymbol{u}|\lambda \sim \mathrm{N}_+(\boldsymbol{0}, \lambda^{-1}\mathrm{I}) \tag{34b}$$

$$\lambda \sim \mathrm{G}(\tfrac{\nu}{2}, \tfrac{\nu}{2}), \tag{34c}$$

where $\boldsymbol{u} \in \mathbb{R}_+^{n_z}$ and $\lambda \in \mathbb{R}_+$ are latent random variables, and $\boldsymbol{\mu} \in \mathbb{R}^{n_z}$, $\mathrm{R} \in \mathbb{R}^{n_z \times n_z}$ (symmetric positive definite (spd)), $\Delta \in \mathbb{R}^{n_z \times n_z}$, and $\nu \in \mathbb{R}_+$ are the parameters of the multivariate skew t-distribution. Its PDF is given in Appendix A. The form of the skewness parameter matrix Δ determines the form of skewness: a form $\Delta = \left[\boldsymbol{v}, \mathrm{O}_{n_z \times (n_z-1)}\right]$, for example, induces skewness in the direction of the vector $\boldsymbol{v}$, and a diagonal Δ-matrix induces skewness in the direction of each coordinate axis having a non-zero diagonal element in Δ. Roughly speaking, the key difference between a multivariate skew t-distribution and a product of independent univariate skew t-distributions is that in the former an outlier measurement vector is an outlier in all dimensions, while in the latter the measurement components are independent. A review of different important special cases of the multivariate skew t-distribution can be found in [89].

The Bayesian Cramér–Rao lower bounds (CRLBs) for filtering and smoothing distributions of the linear and skew-t SSMs (32) are derived in Manuscript [M7]. It is also shown that an increase in the skewness as well as in the heavy-tailedness can decrease the filtering CRLB significantly even though the measurement variance is constant, which suggests that a nonlinear filter can be significantly better than the KF. In the following subsection, two different VB based filters and smoothers for the SSM (32) are explained and compared. One methodology is also applicable to the case where the measurement noise in (32c) is modelled as multivariate skew-t-distributed.

Comparison of two VB factorisations

Because the factor variables in the structured mean field VB factorisation (15) can be vectors, the VB factorisation is an algorithm design choice. VB algorithms seek an optimal approximation given a specific factorisation; that is, the accuracy of the VB approximation is determined by the factorisation. If random variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are highly dependent, separating them into different factors can worsen the approximation. On the other hand, assigning $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ into a joint factor requires that the obtained joint distribution $q(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be treated with sufficient mathematical convenience.

This subsection discusses this choice of the VB factorisation for the case of a Gaussian prior

$$p(\boldsymbol{x}) = \mathrm{N}(\boldsymbol{x}; \hat{\boldsymbol{x}}, \hat{\mathrm{P}}) \tag{35}$$

and skew-t-distributed measurement noise

$$\boldsymbol{y} = \mathrm{C}\boldsymbol{x} + \boldsymbol{e}, \quad \boldsymbol{e} \sim \mathrm{VST}(\boldsymbol{\mu}, \mathrm{R}, \Delta, \nu), \tag{36}$$

where C is the measurement model matrix, and VST means that the measurement noise $\boldsymbol{e}$ is either a collection of independent univariate skew-t-distributed random variables, in which case R and Δ are diagonal matrices, or a multivariate skew-t-distributed random vector. Consider the hierarchical formulation

$$\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{u}, \Lambda \sim \mathrm{N}(\mathrm{C}\boldsymbol{x} + \boldsymbol{\mu} + \Delta\boldsymbol{u}, \Lambda^{-1}\mathrm{R}) \tag{37a}$$

$$\boldsymbol{u}|\Lambda \sim \mathrm{N}_+(\boldsymbol{0}, \Lambda^{-1}) \tag{37b}$$

$$\Lambda \sim p_\Lambda. \tag{37c}$$

In the case of independent univariate skew t-distributions Λ is a diagonal matrix with the prior $p_\Lambda(\Lambda) = \prod_{i=1}^{n_y} \mathrm{G}(\Lambda_{ii}; \frac{\nu_i}{2}, \frac{\nu_i}{2})$, while in the case of a multivariate skew t-distribution Λ is of the form $\lambda \cdot \mathrm{I}_{n_y}$, where $p_\Lambda(\lambda) = \mathrm{G}(\lambda; \frac{\nu}{2}, \frac{\nu}{2})$. The considered factorisations for the VB approximation of the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$ are

$$q^{(-)}(\boldsymbol{x}, \boldsymbol{u}, \Lambda) = q^{(-)}(\boldsymbol{x})\, q^{(-)}(\boldsymbol{u})\, q^{(-)}(\Lambda), \tag{38}$$

which is used in the filter and smoother proposed in Publication [P5], and

$$q^{(+)}(\boldsymbol{x}, \boldsymbol{u}, \Lambda) = q^{(+)}(\boldsymbol{x}, \boldsymbol{u})\, q^{(+)}(\Lambda), \tag{39}$$

which is used in Manuscript [M7]. The VB iteration for the factorisation (−) for Gaussian prior and the independent univariate skew-t measurement noise is derived in the appendix of Publication [P5], and one cycle of the VB iteration is of the form

$$q^{(-)}(\boldsymbol{x}) = \mathrm{N}(\boldsymbol{x}; \cdot, \cdot) \tag{40a}$$

$$q^{(-)}(\boldsymbol{u}) = \mathrm{N}_+(\boldsymbol{u}; \cdot, \cdot) \tag{40b}$$

$$q^{(-)}(\Lambda) = \prod_{i=1}^{n_y} \mathrm{G}(\Lambda_{ii}; \cdot, \cdot), \tag{40c}$$

where the squared-scale matrix of $q^{(-)}(\boldsymbol{u})$ is diagonal. The approximate VB iterations for the factorisation (+) are derived in Manuscript [M7]. The difference is that in (+) the variables $\boldsymbol{x}$ and $\boldsymbol{u}$ are not approximated as independent, but instead the joint VB distribution of $\boldsymbol{x}$ and $\boldsymbol{u}$ is a truncated multivariate normal distribution (TMND)

$$q^{(+)}(\boldsymbol{x},\boldsymbol{u}) \propto \mathrm{N}\left(\left[\begin{smallmatrix}\boldsymbol{x}\\ \boldsymbol{u}\end{smallmatrix}\right];\cdot,\cdot\right)\cdot[\![\boldsymbol{u}\geq\boldsymbol{0}]\!], \tag{41a}$$

$$q^{(-)}(\Lambda) = \prod_{i=1}^{n_{\boldsymbol{y}}} \mathrm{G}(\Lambda_{ii};\cdot,\cdot), \tag{41b}$$

where $[\![\cdot]\!]$ is the Iverson bracket. This has the following consequences:

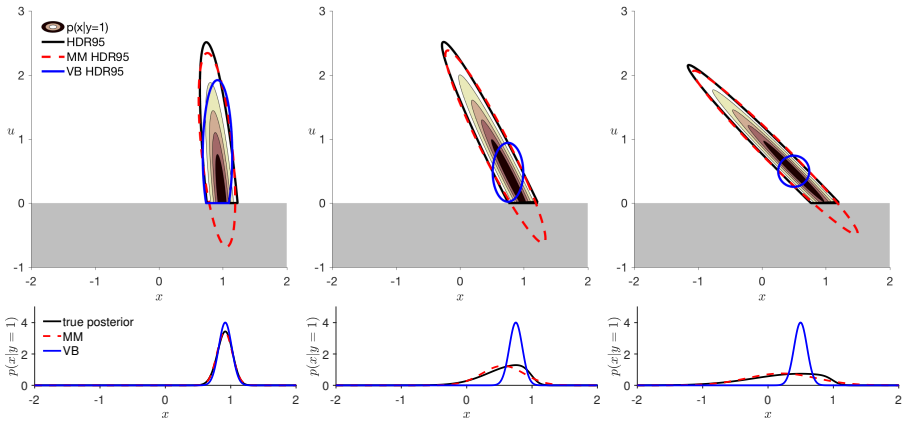
1. If $\boldsymbol{x}$ and $\boldsymbol{u}$ are highly dependent *a posteriori*, the factorisation (+) can be significantly more accurate.
2. A method for computing the mean and covariance matrix of a certain type of TMND is required.
3. The factorisation (+) does not result in a special case of Algorithm 4 because the approximation of $\boldsymbol{x}$ is not Gaussian. To obtain a recursive filtering algorithm, the marginal $\int q^{(+)}(\boldsymbol{x},\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}$ is approximated as Gaussian.

The state $\boldsymbol{x}$ and the latent variable $\boldsymbol{u}$ are indeed prone to high posterior correlations, as illustrated in the following example. Based on the hierarchical model (37) the measurement model (36) essentially implies that given the scaling variable Λ we are observing the sum $\mathrm{C}\boldsymbol{x}+\Delta\boldsymbol{u}$ plus Gaussian noise. Figure 10 illustrates two approximations of the posterior distribution of the model

$$p(x,u) = \mathrm{N}(x;0,1)\cdot\mathrm{N}_+(u;0,1) \tag{42a}$$

$$p(y|x,u) = \mathrm{N}(y;x+\delta u, 0.1^2) \tag{42b}$$

with the measurement value $y=1$ and with δ values 0.1, 0.5, and 1. The figures show that when δ is large, x and u are highly correlated. This makes the VB approximation fail by seriously underestimating the covariance matrix, and moment matching seems to provide a better approximation of the joint posterior and the marginal posterior of x. This reflects the well-known uncertainty underestimation property of the VB approximation discussed in Section 2.5 and in [22, Ch. 10.1], and the fact that this can become a problem with a



(a) $\delta = 0.1$ **(b)** $\delta = 0.5$ **(c)** $\delta = 1$

Figure 10: Comparison of approximation methods for a truncated multivariate normal distribution. The figures show the posteriors $p(x, u|y=1)$ (upper row) and the marginal posteriors $p(x|y=1)$ (lower row) of the model (42), and the 95 % highest density regions (HDR95) for moment matching (MM) and variational Bayes (VB) approximations.

highly skewed skew t-distribution due to high posterior correlation between the variables $\boldsymbol{x}$ and $\boldsymbol{u}$. Moment matching is used in the factorisation (+) when $\boldsymbol{x}$ and $\boldsymbol{u}$ are in a joint VB factor. Manuscript [M7] shows in detail that the factorisation (+) can indeed provide a clear improvement in the accuracy of time-series estimation compared to the factorisation (−) when skewness is high. Furthermore, in Manuscript [M7] it is also demonstrated that the convergence of the VB approximation (−) can be slow in highly-skewed cases; this is a weakness from which the approximation (+) does not suffer.

A downside of the approximation (+) is that the evaluation of the moments of distributions of the form (41a) is not straightforward. Closed-form formulas exist [90], but their computation includes several evaluations of multivariate Gaussian distributions' cumulative distribution function (CDF), which can be very expensive when there are more than two or three measurements per time instant [M7]. Straightforward rejection sampling type Monte Carlo is also inefficient when a large probability is truncated. Therefore, Manuscript

[M7] proposes evaluating the moments using the expectation propagation (EP) based algorithm proposed in [91]. As illustrated by Figure 3 of Manuscript [M7], this approximation means choosing one truncating linear constraint at a time, and approximating the previous phase's output Gaussian truncated by the current constraint as Gaussian using moment-matching. In Manuscript [M7] a greedy approach for ordering the constraints' processing sequence is proposed: at each phase the constraint that truncates most probability is applied next. The used EP algorithm is simple and computationally efficient, requires no tuning, and accepts any form of squared-scale matrix thus enabling skew-t inference with any form of multivariate skew t-distribution. Furthermore, the simulations in [M7] show that the skew-t filter using this algorithm is close in RMSE to the optimal filter. The convergence of the EP iterations is not guaranteed, but [91] states that the proposed algorithm works well when the constraints are rectangular. Provided that the EP gives the exact mean and covariance matrix, the obtained (+)-type VB iteration (41) is exact for the filtering posterior. The exact VB algorithm for the smoothing posterior is impractical because of large dimension of the TMND, but the tests in [M7] indicate that the approximative VB algorithm where the EP is applied already in the forward filtering step of the smoother performs well.

The VB factorisation (+) of the skew-t inference problem is not of the form (22) unlike the factorisation (−). A consequence of this is that the variational approximation of the posterior of $\boldsymbol{x}$, i.e. $\int q(\boldsymbol{x}, \boldsymbol{u})\, \mathrm{d}\boldsymbol{u}$ is not Gaussian but a marginal distribution of the untruncated components of a TMND. In the skew-t filter proposed in Manuscript [M7], the approximate state posterior $\int q(\boldsymbol{x}, \boldsymbol{u})\, \mathrm{d}\boldsymbol{u}$ is approximated as Gaussian with the mean and covariance matrix given by the sequential truncation. In the simulated and real-data tests of Manuscript [M7] this approach still outperforms the algorithm based on the factorisation (−) in accuracy of filtering and smoothing, and in the filtering case it yields RMSEs close to the PF.

Time-of-arrival positioning in mixed LOS–NLOS conditions

TOA measurements are measurements of the duration of electromagnetic radiation's flight time from a transmitter to the receiver. A TOA

measurement can thus be converted into a distance measurement by scaling with the speed of light. This is a widely used type of positioning measurement, as all GNSSs are based on measuring the TOA [12]. Currently, TOA measurements are also being introduced in indoor positioning, as UWB radio based positioning systems produce high-resolution TOA measurements thanks to the high time resolution provided by the wide frequency band [10]. Practical implementation of TOA positioning requires very precise synchronisation of some form between the transmitters and the receivers. In GNSSs the synchronisation is implemented by precisely synchronising the satellites' clocks and estimating just one receiver clock bias parameter simultaneously with the 3-dimensional position [12]. One way to circumvent the synchronisation problem in UWB positioning is to use two-way ranging, where the receiver mirrors the signal back to the transmitter and the transmitter knows both the reception and transmission times according to its own clock [92].

In line-of-sight (LOS) conditions the TOA measurements can be very precise; typical errors are within a few meters for GNSS [12] and within tens of centimetres for UWB [10]. However, non-line-of-sight (NLOS) conditions can reduce the estimation accuracy dramatically because the direct signal path is blocked and the distance is measured using a reflected signal component or because the starting point of the pulse is thresholded to a wrong time instant. This introduces a measurement bias whose magnitude can be several meters for UWB, which makes the measurement error distribution heavy-tailed. Furthermore, the bias is almost always positive because the signal cannot travel faster than the speed of light, which makes the typical TOA error distribution positively skewed. The NLOS bias can have a significant effect on the positioning accuracy because many indoor environments are packed with obstacles that cause NLOS conditions.

In Publication [P6] the skew-t filter is applied to UWB and inertial sensor based indoor positioning in mixed LOS–NLOS indoor conditions, where the RMSE reduction enabled by the skew-t filter is roughly 45% for the extended Kalman filter (EKF). In Manuscript [M7] the skew-t filter is applied to GNSS pseudorange based outdoor positioning in central London (UK), where tall buildings frequently cause NLOS measurements, and the RMSE reduction is roughly 35% for the

EKF and 25% for the Student's t filter of [93]. In these papers we show that the error distribution of the TOA measurements in real-world conditions is positively skewed and that modeling the skewness in the filtering algorithm can improve real-world positioning accuracy compared to the Gaussian and Student's t based measurement noise models.

4 Conclusions and future work

This thesis studies estimation algorithms for some non-Gaussian state-space models (SSMs) that are involved in positioning. In Section 1.3 the research problem is formulated as three research questions, and the answers obtained in this thesis as well as some future research directions are outlined here.

Research question 1) What are the best models and algorithms for incorporating floor plan constraints in an indoor positioning algorithm? For map-assisted indoor positioning, three different models and particle filters (PFs) are compared: the wall-collision PF is shown to suit best for cases where relatively precise pedestrian dead reckoning (PDR) is available, the wall-collision PF with map and PDR based proposal distribution is recommended for cases with noisy PDR, and the graph-based PF is recommended for cases where PDR is not used. The main contributions are thorough analysis of the wall-collision PF and a method to monitor the PF's integrity using a backup Kalman filter (KF), an algorithm to mitigate the indoor positioning PF's sample degeneracy by adding map information to the PF's proposal distribution and/or motion model, and improving the graph-based PF using the so-called total link length (TLL) as a basis for propagating the particles without PDR or prior information about the functions of the different parts of the building.

Floor estimation is a crucial part of any large-scale indoor positioning system, and map information is also usable in floor estimation, as the map can contain information on staircases, elevators and escalators, etc. The promising performance of barometer-assisted 3-dimensional indoor positioning PF's has been demonstrated in [77], [94], [P1], among others. The TLL-based link transition rule in [P2] is also extended to a multi-floor case. However, map-assisted

floor estimation remains a challenging research problem, where an interesting question is what to do when different measurements contradict: the filter should cope with a scenario where barometer and absolute position measurements indicate a floor change but none of the particles is inside a staircase or an elevator, for example.

Research question 2) How can a Bayesian filter and a Bayesian smoother be designed that take account of the skewness and kurtosis of the measurement noise distribution while maintaining an acceptable level of computational complexity and scalability with respect to the problem dimensionality? In this thesis, two different approximative Bayesian filters and smoothers are proposed for inference of SSMs with skew-t-distributed measurement noise. The skew t-distribution models data that contain outliers and show asymmetry in distribution. The algorithms are based on two different factorisations made for the variational Bayes (VB) approximation. One factorisation provides analytical formulas for the VB smoother and recursive filter. With the other factorisation, additional approximations are used to enable a less coarse VB factorisation. The tests show that the latter approximation converges faster and provides better accuracy due to more accurate approximation of the posterior covariance matrix, especially when the measurement error distributions are highly skewed. The Cramér–Rao lower bounds (CRLBs) are also derived for the filtering and smoothing distributions of the skew-t SSM.

The KF, which is optimal for Gaussian-distributed measurement noises, is widely used because of mathematical and computational convenience. The work presented in this thesis shows that also more fine-grained characteristics of the noise distribution can be modelled and that this can improve the estimation performance. Sampling-free approximation methods such as the VB algorithms can provide close-to-optimal mean-square-errors without dramatically sacrificing the KF's computational complexity and scalability with respect to the state dimensionality. The proposed skew-t inference algorithms are new tools in the kit that can in many applications enable the best compromise between accuracy and complexity. Even when sampling based methods are found to be the best solution, the proposed analytical approximations could be used e.g. for approximate

Rao–Blackwellisation of the sampling procedures or to obtain better proposal distributions.

Research question 3) How can non-line-of-sight (NLOS) measurements be handled in a computationally efficient way in time delay measurement based positioning algorithms? It is shown in this thesis that the skew-t modelling of the time delay measurement noise can improve the positioning filter's accuracy compared to the state-of-the-art Gaussian and Student's t model based filter in obstructed indoor spaces and densely built urban areas. In addition to positioning, skewed distributions have been reported in many fields such as finance, economics, psychiatry, and environmetrics [M7]. Thus, skew-t based time-series inference has a lot of potential applications. The author of this thesis hopes that this thesis encourages practitioners in different fields to consider alternatives to the "extremes" in the simplicity–accuracy axis, i.e. KF extensions and PF. In this thesis the skew-t measurement noise distribution's parameters are assumed to be known. A future research direction that can further widen the applicability of the skew-t modelling is developing skew-t based recursive system identification methods where all parameters are inferred from the data; this work has already been started [95].

References

[1] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey", *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[2] D. Macagnano, G. Destino, and G. Abreu, "Indoor Positioning: a Key Enabling Technology for IoT Applications", in *IEEE World Forum on Internet of Things (WF-IoT)*, 2014.

[3] K. Pahlavan, X. Li, and J.-P. Mäkelä, "Indoor Geolocation Science and Technology", *IEEE Communications Magazine*, vol. 40, no. 2, pp. 112–118, 2002.

[4] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements", *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, 2005.

[5] R. Mautz, “Indoor Positioning Technologies”, Habilitation Thesis, ETH Zurich, 2012. DOI: 10.3929/ethz-a-007313554.

[6] D. Dardari, P. Closas, and P. M. Djurić, “Indoor Tracking: Theory, Methods, and Technologies”, *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, 2015.

[7] P. Davidson and R. Piché, “A Survey of Selected Indoor Positioning Methods for Smartphones”, *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1347–1370, 2017.

[8] P. Bahl and V. N. Padmanabhan, “RADAR: An In-Building RF-based User Location and Tracking System”, *19th International conference on Computer Communications (InfoCom)*, vol. 2, pp. 775–784, 2000.

[9] A. Kotanen, M. Hännikäinen, H. Leppäkoski, and T. D. Hämäläinen, “Experiments on Local Positioning with Bluetooth”, in *International Conference on Information Technology: Computers and Communications (ITCC)*, 2003.

[10] S. Gezici, T. Zhi, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, “Localization via Ultra-Wideband Radios”, *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, 2005.

[11] I. Spassov, “Algorithms for map-aided autonomous indoor pedestrian positioning and navigation”, PhD thesis, École Polytechnique Fédérale de Lausanne, 2007. DOI: 10.5075/epfl-thesis-3961.

[12] J. J. Spilker Jr. and B. W. Parkinson, “Overview of GPS Operation and Design”, in *Global Positioning System: Theory and Applications Volume I*, B. W. Parkinson and J. J. Spilker Jr., Eds., AIAA, 1994, ch. 2.

[13] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, “Locating the Nodes – Cooperative localization in wireless sensor networks”, *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[14] J. Haverinen and A. Kemppainen, “Global indoor self-localization based on the ambient magnetic field”, *Robotics and Autonomous Systems*, no. 57, pp. 1028–1035, 2009.

[15] A. J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera", in *9th IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2003.

[16] L. Ruotsalainen, "Vision-Aided Pedestrian Navigation for Challenging GNSS Environments", PhD thesis, Tampere University of Technology, 2013. [Online]. Available: `http://URN.fi/URN:ISBN:978-951-711-303-8`.

[17] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Transactions of the ASME – Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[18] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, UK: Cambridge University Press, 2013.

[19] S. Kotz, N. Balakrishnan, and N. L. Johnson, *Continuous Multivariate Distributions, Volume 1: Models and Applications*, 2nd edition. John Wiley & Sons, 2000.

[20] R. K. Pearson, "Outliers in process modeling and identification", *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 55–63, 2002.

[21] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An Introduction to Variational Methods for Graphical Models", *Machine Learning*, vol. 37, pp. 183–233, 1999.

[22] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[23] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, ser. Prentice-Hall information and system sciences. Prentice-Hall, 1979.

[24] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum Likelihood Estimates of Linear Dynamic Systems", *Journal of the American Institute of Aeronautics and Astronautics*, vol. 3, no. 8, pp. 1445–1450, 1965.

[25] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, ser. Mathematics in Science and Engineering. Academic Press, 1970, vol. 64.

[26] I. Arasaratnam and S. Haykin, "Cubature Kalman Filters", *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[27] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.

[28] I. S. Mbalawata and S. Särkkä, "On the L^4 convergence of particle filters with general importance distributions", in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2014.

[29] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering", *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[30] J. D. Hol, T. B. Schön, and F. Gustafsson, "On Resampling Algorithms for Particle Filters", in *IEEE Nonlinear Statistical Signal Processing Workshop (NSSPW)*, 2006, pp. 79–82.

[31] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian non-linear state space models", *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[32] F. Gustafsson, "Particle filter theory and practice with positioning applications", *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, 2010.

[33] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter, Particle Filters for Tracking Applications*. Boston, London: Artech House, 2004.

[34] P. Rebeschini, "Nonlinear Filtering in High Dimension", PhD thesis, Princeton University, 2014. [Online]. Available: `http://arks.princeton.edu/ark:/88435/dsp0105741t92j`.

[35] G. Storvik, "Particle Filters for State-Space Models With the Presence of Unknown Static Parameters", *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 281–289, 2002.

[36] M. J. Beal, "Variational Algorithms for Approximate Bayesian Inference", PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003. [Online]. Available: `https://www.cse.buffalo.edu//faculty/mbeal/thesis/`.

[37] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference", *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 131–146, 2008.

[38] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: The MIT Press, 2012.

[39] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums", *Automatica*, vol. 7, no. 4, pp. 465–479, 1971.

[40] J. L. Williams and P. S. Maybeck, "Cost-function-based hypothesis control techniques for multiple hypothesis tracking", *Mathematical and Computer Modelling*, vol. 43, no. 9–10, pp. 976–989, 2006.

[41] T. P. Minka, "Expectation propagation for approximate Bayesian inference", in *17th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001, pp. 362–369.

[42] H. Rue, S. Martino, and N. Chopin, "Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations", *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 71, no. 2, pp. 319–392, 2009.

[43] M. Roth, E. Özkan, and F. Gustafsson, "A Student's t filter for heavy tailed process and measurement noise", in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2013.

[44] M. Roth, T. Ardeshiri, E. Özkan, and F. Gustafsson, *Robust Bayesian Filtering and Smoothing Using Student's t Distribution*, Arxiv, 2017. [Online]. Available: `https://arxiv.org/abs/1703.02428`.

[45] T. Ardeshiri, U. Orguner, and F. Gustafsson, *Bayesian Inference via Approximation of Log-likelihood for Priors in Exponential Family*, Arxiv, 2015. [Online]. Available: `https://arxiv.org/abs/1510.01225`.

[46] M. Raitoharju, R. Piché, and H. Nurminen, "A Systematic Approach for Kalman-type Filtering with non-Gaussian Noises", in *19th International Conference on Information Fusion (FUSION)*, 2016.

[47] H. Leppäkoski, J. Collin, and J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals", *Journal of Signal Processing Systems*, vol. 71, no. 3, pp. 287–296, 2013.

[48] J. Collin, O. Mezentsev, and G. Lachapelle, "Indoor Positioning System Using Accelerometry and High Accuracy Heading Sensors", in *GPS/GNSS Conference*, 2003.

[49] W. Chen, R. Chen, Y. Chen, H. Kuusniemi, and J. Wang, "An effective pedestrian dead reckoning algorithm using a unified heading error model", in *IEEE/ION Position Location and Navigation Symposium (PLANS)*, 2010, pp. 340–347.

[50] R. L. French and G. M. Lang, "Automatic route control system", *IEEE Transactions on Vehicular Technology*, vol. 22, no. 2, pp. 36–41, 1973.

[51] T. W. Lezniak, R. W. Lewis, and R. A. McMillen, "A dead reckoning/map correlation system for automatic vehicle tracking", *IEEE Transactions on Vehicular Technology*, vol. 26, no. 1, pp. 47–60, 1977.

[52] P. Davidson, "Algorithms for Autonomous Personal Navigation Systems", PhD thesis, Tampere University of Technology, 2013. [Online]. Available: `http://URN.fi/URN:ISBN:978-952-15-3215-3`.

[53] H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U. D. Hanebeck, "WLAN-based pedestrian tracking using particle filters and low-cost MEMS sensors", in *4th Workshop on Positioning, Navigation and Communication (WPNC)*, 2007, pp. 1–7.

[54] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments", in *10th International Conference on Ubiquitous computing (UbiComp)*, 2008, pp. 114–123.

[55] Widyawan, M. Klepal, and S. Beauregard, "A Backtracking Particle Filter for Fusing Building Plans with PDR Displacement Estimates", in *5th Workshop on Positioning, Navigation and Communication (WPNC)*, 2008, pp. 207–212.

[56] S. Beauregard, Widyawan, and M. Klepal, "Indoor PDR performance enhancement using minimal map information and particle filters", in *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2008, pp. 141–147.

[57] B. Krach and P. Roberston, "Cascaded estimation architecture for integration of foot-mounted inertial sensors", in *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2008, pp. 112–119.

[58] P. Blanchart, L. He, and F. Le Gland, "Information fusion for indoor localization", in *12th International Conference on Information Fusion (FUSION)*, 2009, pp. 2083–2090.

[59] P. Davidson, J. Collin, and J. Takala, "Application of particle filters for indoor positioning using floor plans", in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, 2010, pp. 1–4.

[60] S. Kaiser, M. Khider, and P. Robertson, "A human motion model based on maps for navigation systems", *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, pp. 1–14, 2011.

[61] N. Kothari, B. Kannan, and M. B. Dias, "Robust Indoor Localization on a Commercial Smart-Phone", Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-11-27, 2011. [Online]. Available: `http://www.ri.cmu.edu/publication_view.html?pub_id=6902`.

[62] J. Pinchin, C. Hide, and T. Moore, "A particle filter approach to indoor navigation using a foot mounted inertial navigation system and heuristic heading information", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012.

[63] M. Khider, S. Kaiser, and P. Robertson, "A Novel Three Dimensional Movement Model for Pedestrian Navigation", *Journal of Navigation*, vol. 65, no. 2, pp. 245–264, 2012.

[64] M. Kirkko-Jaakkola, J. Collin, and J. Takala, "Using Building Plans and Self-Contained Sensors with GNSS Initialization for Indoor Navigation", in *IEEE 77th Vehicular Technology Conference (VTC)*, 2013.

[65] F. Zampella, A. R. Jiménez Ruiz, and F. Seco Granja, "Indoor Positioning Using Efficient Map Matching, RSS Measurements, and an Improved Motion Model", *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1304–1317, 2015.

[66] M. Raitoharju, H. Nurminen, and R. Piché, "Kalman filter with a linear state model for PDR+WLAN positioning and its application to assisting a particle filter", *EURASIP Journal on Advances in Signal Processing*, vol. 1, no. 33, 2015.

[67] E.-S. Lohan, J. Talvitie, P. Figueiredo e Silva, H. Nurminen, S. Ali-Löytty, and R. Piché, "Received Signal Strength models for WLAN and BLE-based indoor positioning in multi-floor buildings", in *International Conference on Localization and GNSS (ICL-GNSS)*, 2015.

[68] P. Kemppi, T. Rautiainen, V. Ranki, F. Belloni, and J. Pajunen, "Hybrid positioning system combining angle-based localization, pedestrian dead reckoning and map filtering", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2010.

[69] M. Kessel and M. Werner, "Automated WLAN calibration with a backtracking particle filter", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012.

[70] B. Turgut and R. P. Martin, "Restarting particle filters: an approach to improve the performance of dynamic indoor localization", in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2009.

[71] F. Evennou, M. François, and E. Novakov, "Map-aided indoor mobile positioning system using particle filter", in *IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 4, 2005, pp. 2490–2494.

[72] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the Levy-Walk Nature of Human Mobility", *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 630–643, 2011.

[73] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, "Particle filters for mobile robot localization", in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., New York: Springer-Verlag, 2001, pp. 470–498.

[74] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz, "Voronoi tracking: location estimation using sparse and noisy sensor data", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003, pp. 723–728.

[75] P.-Y. Gilliéron, I. Spassov, and B. Merminod, "Indoor Navigation Enhanced by Map-Matching", *European Journal of Navigation*, vol. 3, no. 3, 2005.

[76] M. I. Khan and J. Syrjärinne, "Investigating Effective Methods for Integration of Building's Map with Low cost Inertial Sensors and Wifi-based Positioning", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2013, pp. 884–891.

[77] F. Ebner, T. Fetzer, F. Deinzer, L. Köping, and M. Grzegorzek, "Multi Sensor 3D Indoor Localisation", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015.

[78] B. Ferris, D. Hähnel, and D. Fox, "Gaussian Processes for Signal Strength-Based Location Estimation", in *Robotics, Sciences and Systems Conference (RSS)*, 2006.

[79] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, "Graph-based Data Fusion of Pedometer and WiFi Measurements for Mobile Indoor Positioning", in *ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2014, pp. 147–158.

[80] J. Liu, R. Chen, L. Pei, R. Guinness, and H. Kuusniemi, "A Hybrid Smartphone Indoor Positioning Solution for Mobile LBS", *Sensors*, vol. 12, no. 12, pp. 17 208–17 233, 2012.

[81] J. Trogh, D. Plets, L. Martens, and W. Joseph, "Advanced Real-Time Indoor Tracking Based on the Viterbi Algorithm and Semantic Data", *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.

[82] K. Kaemarungsi and P. Krishnamurthy, "Analysis of WLAN's received signal strength indication for indoor location fingerprinting", *Pervasive and Mobile Computing*, vol. 8, no. 2, pp. 292–316, 2012.

[83] B. I. Ahmad, T. Ardeshiri, P. Langdon, S. J. Godsill, and T. Popham, "Modelling Received Signal Strength from On-Vehicle BLE Beacons Using Skewed Distributions: A Preliminary Study", in *20th International Conference on Information Fusion (FUSION)*, 2017, pp. 1027–1033.

[84] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Approximate Inference in State-Space Models With Heavy-Tailed Noise", *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5024–5037, 2012.

[85] M. D. Branco and D. K. Dey, "A General Class of Multivariate Skew-Elliptical Distributions", *Journal of Multivariate Analysis*, vol. 79, no. 1, pp. 99–113, 2001.

[86] A. Azzalini and A. Capitanio, "Distributions Generated by Perturbation of Symmetry with Emphasis on a Multivariate Skew t-Distribution", *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 65, no. 2, pp. 367–389, 2003.

[87] A. K. Gupta, "Multivariate skew t-distribution", *Statistics*, vol. 37, no. 4, pp. 359–363, 2003.

[88] T.-I. Lin, "Robust mixture modeling using multivariate skew t distributions", *Statistics and Computing*, vol. 20, pp. 343–356, 2010.

[89] S. X. Lee and G. J. McLachlan, "Finite mixtures of canonical fundamental skew t-distributions – The unification of the restricted and unrestricted skew t-mixture models", *Statistics and Computing*, vol. 26, no. 3, pp. 573–589, 2016.

[90] G. Tallis, "The Moment Generating Function of the Truncated Multi-normal Distribution", *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 23, no. 1, pp. 223–119, 1961.

[91] J. P. Cunningham, P. Hennig, and S. Lacoste-Julien, *Gaussian Probabilities and Expectation Propagation*, Arxiv, 2013. [Online]. Available: `http://arxiv.org/abs/1111.6832`.

[92] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Z. Win, "Ranging With Ultrawide Bandwidth Signals in Multipath Environments", *Proceedings of the IEEE*, vol. 97, no. 2, pp. 404–426, 2009.

[93] R. Piché, S. Särkkä, and J. Hartikainen, "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate Student-t distribution", in *2012 IEEE International Workshop on Machine Learning for Signal Processing*, 2012.

[94] C. Ascher, C. Kessler, R. Weis, and G. Trommer, "Multi-floor map matching in indoor environments for mobile platforms", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012.

[95] H. Nurminen and T. Ardeshiri, *Approximate Recursive Identification of Autoregressive Systems with Skewed Innovations*, Arxiv, 2016. [Online]. Available: `http://arxiv.org/abs/1612.03761`.

[96] A. Azzalini and A. Capitanio, *The Skew-Normal and Related Families*. Cambridge University Press, 2014.

Appendix

A Probability distributions

Univariate Gaussian distribution $\mathrm{N}(\mu, \sigma^2)$

support	$\mathbb{R}$
parameters	location $\mu \in \mathbb{R}$
	scale $\sigma \in \mathbb{R}_+$
PDF	$\mathrm{N}(x;\mu,\sigma^2) = \frac{1}{(2\pi)^{\frac{1}{2}}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} = \frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)$
CDF	$F_{\mathrm{N}(\mu,\sigma^2)}(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{2}\left(1+\mathrm{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma}\right)\right)$
mean	μ
variance	σ^2
skewness	0
exc. kurtosis	0

Truncated univariate Gaussian distribution $\mathrm{N}_{[a,b]}(\mu,\sigma^2)$

support	$[a,b]$
parameters	location $\mu\in\mathbb{R}$
	scale $\sigma\in\mathbb{R}_+$
	truncation interval $[a,b]$
PDF	$\mathrm{N}_{[a,b]}(x;\mu,\sigma^2)=\frac{1}{\alpha}\mathrm{N}(x;\mu,\sigma^2)\cdot[\![x\in[a,b]]\!],\quad \alpha=\Phi\left(\frac{b-\mu}{\sigma}\right)-\Phi\left(\frac{a-\mu}{\sigma}\right)$
CDF	$F_{\mathrm{N}_{[a,b]}(\mu,\sigma^2)}(x)=\frac{1}{\alpha}\left(\Phi\left(\frac{x-\mu}{\sigma}\right)-\Phi\left(\frac{a-\mu}{\sigma}\right)\right)$ when $x\in[a,b]$
mean	$\mu+\frac{\phi((a-\mu)/\sigma)-\phi((b-\mu)/\sigma)}{\alpha}\sigma$
variance	$\sigma^2\left(1+\frac{\frac{a-\mu}{\sigma}\phi((a-\mu)/\sigma)-\frac{b-\mu}{\sigma}\phi((b-\mu)/\sigma)}{\alpha}-\left(\frac{\phi((a-\mu)/\sigma)-\phi((b-\mu)/\sigma)}{\alpha}\right)^2\right)$

Univariate Gaussian distribution truncated into positive numbers $\mathrm{N}_+(\mu,\sigma^2)$

support	$\mathbb{R}_+$
parameters	location $\mu\in\mathbb{R}$
	scale $\sigma\in\mathbb{R}_+$
PDF	$\mathrm{N}_+(x;\mu,\sigma^2)=\frac{1}{\alpha}\mathrm{N}(x;\mu,\sigma^2)\cdot[\![x>0]\!],\quad \alpha=\Phi\left(\frac{\mu}{\sigma}\right)$
CDF	$F_{\mathrm{N}_+(0,\sigma^2)}(x)=\frac{1}{\alpha}\left(\Phi\left(\frac{x-\mu}{\sigma}\right)+\Phi\left(\frac{\mu}{\sigma}\right)-1\right)$
mean	$\mu+\frac{\phi(\mu/\sigma)}{\alpha}\sigma$
variance	$\sigma^2\left(1-\frac{\mu}{\sigma}\frac{\phi(\mu/\sigma)}{\alpha}-\left(\frac{\phi(\mu/\sigma)}{\alpha}\right)^2\right)$
skewness	$\frac{4-\pi}{2}\frac{b^3}{(1-b^2)^{\frac{3}{2}}},\quad b=\sqrt{\frac{2}{\pi}}$
exc. kurtosis	$2(\pi-3)\frac{b^4}{(1-b^2)^2}$

Student's t-distribution $\mathrm{t}(\mu, \sigma^2, \nu)$

support	$\mathbb{R}$
parameters	location $\mu \in \mathbb{R}$
	scale $\sigma \in \mathbb{R}_+$
	degrees of freedom $\nu \in \mathbb{R}_+$
PDF	$\mathrm{t}(x;\mu,\sigma^2,\nu) = \frac{1}{(\pi\nu)^{\frac{1}{2}}\sigma} \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \left(1 + \frac{1}{\nu}\frac{(x-\mu)^2}{\sigma^2}\right)^{-\frac{\nu+1}{2}}$
mean	μ, if $\nu > 1$
variance	$\frac{\nu}{\nu-2}\sigma^2$, if $\nu > 2$
skewness	0, if $\nu > 3$
exc. kurtosis	$\frac{6}{\nu-4}$, if $\nu > 4$

Skew Gaussian distribution $\mathrm{SN}(\mu, r, \delta)$

support	$\mathbb{R}$
parameters	location $\mu \in \mathbb{R}$
	shape $r \in \mathbb{R}_+$ (spread)
	shape $\delta \in \mathbb{R}$ (skewness)
PDF	$\mathrm{SN}(x;\mu,r,\delta) = 2\,\mathrm{N}(x;\mu,r+\delta^2)\,F_{\mathrm{N}(\mu,r+\delta^2)}(x)$
mean [96, Ch. 2.1]	$\mu + b\delta, \quad b = \sqrt{\frac{2}{\pi}}$
variance [96]	$r + \delta^2 - (b\delta)^2$
skewness [96]	$\frac{4-\pi}{2}\frac{(b\delta)^3}{(r+\delta^2-(b\delta)^2)^{\frac{3}{2}}}$
exc. kurtosis [96]	$2(\pi-3)\frac{(b\delta)^4}{(r+\delta^2-(b\delta)^2)^2}$

Skew t-distribution $\mathrm{ST}(\mu, r, \delta, \nu)$

support	$\mathbb{R}$
parameters	location $\mu \in \mathbb{R}$
	shape $r \in \mathbb{R}_+$ (spread)
	shape $\delta \in \mathbb{R}$ (skewness)
	degrees of freedom $\nu \in \mathbb{R}_+$
PDF	$\mathrm{ST}(x; \mu, r, \delta, \nu)$
	$= 2\,\mathrm{t}(x; \mu, r+\delta^2, \nu)\, F_{\mathrm{t}(0,1,\nu+1)}\left(\frac{\delta}{\sqrt{r}} \frac{x-\mu}{\sqrt{r+\delta^2}} \sqrt{\frac{\nu+1}{\nu+\frac{(x-\mu)^2}{r+\delta^2}}}\right)$
mean [96, Ch. 4.3]	$\mu + b\delta, \quad b = \sqrt{\frac{\nu}{\pi}} \frac{\Gamma(\frac{\nu-1}{2})}{\Gamma(\frac{\nu}{2})}, \quad \text{if } \nu > 1$
variance [96]	$\frac{\nu}{\nu-2}(r+\delta^2) - (b\delta)^2, \quad \text{if } \nu > 2$
skewness [96]	$b\delta \frac{\frac{\nu}{\nu-3}(3r+2\delta^2) - \frac{3\nu}{\nu-2}(r+\delta^2)+2(b\delta)^2}{\left(\frac{\nu}{\nu-2}(r+\delta^2)-(b\delta)^2\right)^{\frac{3}{2}}}, \quad \text{if } \nu > 3$
exc. kurtosis [96]	$\frac{\frac{3\nu^2}{(\nu-2)(\nu-4)}(r+\delta^2)^2 - \frac{4\nu}{\nu-3}(b\delta)^2(3r+2\delta^2) + \frac{6\nu}{\nu-2}(b\delta)^2(r+\delta^2) - 3(b\delta)^4}{\left(\frac{\nu}{\nu-2}(r+\delta^2)-(b\delta)^2\right)^2} - 3, \quad \text{if } \nu > 4$

Gamma distribution $\mathrm{G}(\alpha, \beta)$

support	$\mathbb{R}_+$
parameters	shape $\alpha \in \mathbb{R}_+$
	rate $\beta \in \mathbb{R}_+$
PDF	$\mathrm{G}(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad \Gamma \text{ is gamma function}$
mean	$\frac{\alpha}{\beta}$
variance	$\frac{\alpha}{\beta^2}$

Inverse-gamma distribution $\mathrm{G}^{-1}(\alpha,\beta)$

support	$\mathbb{R}_+$
parameters	shape $\alpha \in \mathbb{R}_+$
	rate $\beta \in \mathbb{R}_+$
PDF	$\mathrm{G}^{-1}(x;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} e^{-\frac{\beta}{x}}$, Γ is gamma function
mean	$\frac{\beta}{\alpha-1}$, if $\alpha > 1$
variance	$\frac{\beta^2}{(\alpha-1)^2(\alpha-2)}$, if $\alpha > 2$

Generalised inverse-Gaussian distribution $\mathrm{GIG}(\lambda,\delta,\gamma)$

support	$\mathbb{R}_+$
parameters	$\lambda \in \mathbb{R}$, $\delta \in \mathbb{R}_+$, $\gamma \in \mathbb{R}_+$
PDF	$\mathrm{GIG}(x;\lambda,\delta,\gamma) = \left(\frac{\gamma}{\delta}\right)^\lambda \frac{x^{\lambda-1}}{2K_\lambda(\gamma\delta)} e^{-\frac{1}{2}(\frac{\delta^2}{x}+\gamma^2 x)}$,
	K_α is the modified Bessel function of the 2nd kind of order α

Categorical distribution $\mathrm{cat}(\alpha^{(1:k)})$

support	$\{1,2,\ldots,k\}$
parameters	number of categories $k \in \mathbb{N}$
	category probabilities $\alpha^{(1:k)} \in [0,1]$, $\sum_{i=1}^k \alpha^{(k)} = 1$
prob. mass function	$\mathrm{cat}(x;\alpha^{(1:k)}) = \sum_{i=1}^k [\![x = i]\!] \cdot \alpha^{(i)}$

Multivariate Gaussian distribution $\mathrm{N}(\boldsymbol{\mu},\Sigma)$

support	$\mathbb{R}^n$
parameters	location $\boldsymbol{\mu} \in \mathbb{R}^n$
	squared-scale matrix $\Sigma \in \mathbb{R}^{n\times n}$
	(symmetric positive definite (spd))
PDF	$\mathrm{N}(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} \det(\Sigma)^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$
mean	$\boldsymbol{\mu}$
covariance m.	Σ

Multivariate Gaussian distribution truncated into positive orthant $\mathrm{N}_+(\boldsymbol{\mu}, \Sigma)$

support	$\mathbb{R}_+^n$
parameters	location $\boldsymbol{\mu} \in \mathbb{R}^n$
	squared-scale matrix $\Sigma \in \mathbb{R}^{n\times n}$ (spd)
PDF	$\mathrm{N}_+(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\alpha}\,\mathrm{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) \cdot \prod_{i=1}^n [\![x_i > 0]\!], \quad \alpha = \int_{\mathbb{R}_+^n} \mathrm{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma)\,\mathrm{d}\boldsymbol{x}$

Multivariate t-distribution $\mathrm{t}(\boldsymbol{\mu}, \Sigma, \nu)$

support	$\mathbb{R}^n$
parameters	location $\boldsymbol{\mu} \in \mathbb{R}^n$
	squared-scale matrix $\Sigma \in \mathbb{R}^{n\times n}$ (spd)
	degrees of freedom $\nu \in \mathbb{R}_+$
PDF	$\mathrm{t}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma, \nu) = \frac{1}{(\pi\nu)^{\frac{n}{2}} \det(\Sigma)^{\frac{1}{2}}} \frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{\nu}{2})} \left(1 + \frac{1}{\nu}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)^{-\frac{\nu+n}{2}}$
mean	$\boldsymbol{\mu}$, if $\nu > 1$
covariance m.	$\frac{\nu}{\nu-2}\Sigma$, if $\nu > 2$

Multivariate skew Gaussian distribution $\mathrm{MSN}(\boldsymbol{\mu}, \mathrm{R}, \Delta)$

support	$\mathbb{R}^n$
parameters	location $\boldsymbol{\mu} \in \mathbb{R}^n$
	shape $\mathrm{R} \in \mathbb{R}^{n\times n}$ (spread) (spd)
	shape $\Delta \in \mathbb{R}^{n\times n}$ (skewness)
PDF	$\mathrm{MSN}(\boldsymbol{x}; \boldsymbol{\mu}, \mathrm{R}, \Delta) = 2^n\,\mathrm{N}(\boldsymbol{x}; \boldsymbol{\mu}, \Omega)\, F_{\mathrm{N}(\mathbf{0}, \mathrm{I} - \Delta^{\mathrm{T}}\Omega^{-1}\Delta)}\left(\Delta\Omega^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$, $\Omega = \mathrm{R} + \Delta\Delta^{\mathrm{T}}$
mean	$\boldsymbol{\mu} + b\Delta\mathbf{1}, \quad b = \sqrt{\frac{2}{\pi}}$
covariance m.	$\mathrm{R} + b^2\Delta\Delta^{\mathrm{T}}$

Multivariate skew t-distribution $\mathrm{MST}(\boldsymbol{\mu}, \mathrm{R}, \Delta, \nu)$

support	$\mathbb{R}^n$
parameters	location $\boldsymbol{\mu} \in \mathbb{R}^n$
	shape $\mathrm{R} \in \mathbb{R}^{n\times n}$ (spread) (spd)
	shape $\Delta \in \mathbb{R}^{n\times n}$ (skewness)
	degrees of freedom $\nu \in \mathbb{R}_+$
PDF	$\mathrm{MVST}(\boldsymbol{x}; \boldsymbol{\mu}, \mathrm{R}, \Delta, \nu) = 2^n\, \mathrm{t}(\boldsymbol{x}; \boldsymbol{\mu}, \Omega, \nu)$
	$\times F_{\mathrm{t}(\mathbf{0}, \mathrm{I}-\Delta^{\mathrm{T}}\Omega^{-1}\Delta, \nu+n)}\Big(\Delta^{\mathrm{T}}\Omega^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\sqrt{\frac{\nu+n}{\nu+(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\Omega^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}}\Big)$
mean [89]	$\boldsymbol{\mu} + b\Delta\mathbf{1}, \quad b = \sqrt{\frac{\nu}{\pi}}\frac{\Gamma(\frac{\nu-1}{2})}{\Gamma(\frac{\nu}{2})}, \quad \text{if } \nu > 1$
covariance m. [89]	$\frac{\nu}{\nu-2}\big(\mathrm{R} + (1-\frac{2}{\pi})\Delta\Delta^{\mathrm{T}}\big) - \big(\frac{2}{\pi}\frac{\nu}{\nu-2} - b^2\big)\Delta\mathbf{1}\mathbf{1}^{\mathrm{T}}\Delta^{\mathrm{T}}, \quad \text{if } \nu > 2$

Gaussian mixture distribution $\mathrm{GM}(\alpha^{(1:N_{\mathrm{GM}})}, \boldsymbol{\mu}^{(1:N_{\mathrm{GM}})}, \Sigma^{(1:N_{\mathrm{GM}})})$

support	$\mathbb{R}^n$
parameters	component weights $\alpha^{(1:N_{\mathrm{GM}})} \in [0,1]$, $\sum_{i=1}^{N_{\mathrm{GM}}} \alpha^{(N_{\mathrm{GM}})} = 1$
	locations $\boldsymbol{\mu}^{(1:N_{\mathrm{GM}})} \in \mathbb{R}^n$
	squared-scale matrices $\Sigma^{(1:N_{\mathrm{GM}})} \in \mathbb{R}^{n\times n}$ (spd)
PDF	$\mathrm{GM}(\boldsymbol{x}; \alpha^{(1:N_{\mathrm{GM}})}, \boldsymbol{\mu}^{(1:N_{\mathrm{GM}})}, \Sigma^{(1:N_{\mathrm{GM}})}) = \sum_{i=1}^{N_{\mathrm{GM}}} \alpha^{(i)}\, \mathrm{N}(\boldsymbol{x}; \boldsymbol{\mu}^{(i)}, \Sigma^{(i)})$
CDF	$F_{\mathrm{GM}(\alpha^{(1:N_{\mathrm{GM}})}, \boldsymbol{\mu}^{(1:N_{\mathrm{GM}})}, \Sigma^{(1:N_{\mathrm{GM}})})} = \sum_{i=1}^{N_{\mathrm{GM}}} \alpha^{(i)} F_{\mathrm{N}(\boldsymbol{\mu}^{(i)}, \Sigma^{(i)})}$
mean	$\sum_{i=1}^{N_{\mathrm{GM}}} \alpha^{(i)} \boldsymbol{\mu}^{(i)}$
covariance m.	$\sum_{i=1}^{N_{\mathrm{GM}}} \alpha^{(i)} \big(\Sigma^{(i)} + (\boldsymbol{\mu}^{(i)} - \mathbb{E}[x])(\boldsymbol{\mu}^{(i)} - \mathbb{E}[x])^{\mathrm{T}}\big)$

B Derivation of variational Bayes filter and smoother

In this Appendix the variational Bayes (VB) smoother and filter for conditionally Gaussian state-space models (SSMs) are derived.

The logarithm of the smoothing posterior of (21) is

$$
\begin{aligned}
&\log p(\boldsymbol{x}_{1:K}, \boldsymbol{\theta}_{1:K} | \boldsymbol{y}_{1:K}) \\
=& -\frac{1}{2}(\boldsymbol{x}_0 - \boldsymbol{x}_{0|0})^{\mathrm{T}} \mathrm{P}_{0|0}^{-1} (\boldsymbol{x}_0 - \boldsymbol{x}_{0|0}) - \frac{1}{2}\sum_{k=1}^{K}\Big[\log(\det(\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k))) + \log(\det(\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k))) \\
&+ (\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_k - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} (\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_k - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k)) \\
&+ (\boldsymbol{x}_k - \mathrm{A}\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} (\boldsymbol{x}_k - \mathrm{A}\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k)) - 2\log p(\boldsymbol{\theta}_k) \Big] + c,
\end{aligned}
\tag{43}
$$

where c denotes in this Appendix any constant that does not depend on the argument of the considered probability density function (PDF). Thus, given the approximative smoothing distribution $q_i(\boldsymbol{\theta}_{1:K,i})$ for $i = 1, \ldots, M$, the VB update (17) for $\boldsymbol{x}_{0:K}$ is

$$
\begin{aligned}
&\log q(\boldsymbol{x}_{0:K}) \\
=&-\frac{1}{2}(\boldsymbol{x}_0 - \boldsymbol{x}_{0|0})^{\mathrm{T}} \mathrm{P}_{0|0}^{-1} (\boldsymbol{x}_0 - \boldsymbol{x}_{0|0}) \\
&-\frac{1}{2}\sum_{k=1}^{K} \mathop{\mathbb{E}}_{q_{1:M}} \Big[(\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_k - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} (\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_k - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k)) \\
&+ (\boldsymbol{x}_k - \mathrm{A}\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} (\boldsymbol{x}_k - \mathrm{A}\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k)) \Big] + c \qquad (44) \\
=&-\frac{1}{2}(\boldsymbol{x}_0 - \boldsymbol{x}_{0|0})^{\mathrm{T}} \mathrm{P}_{0|0}^{-1} (\boldsymbol{x}_0 - \boldsymbol{x}_{0|0}) \\
&-\frac{1}{2}\sum_{k=1}^{K} \Bigg[(\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_k - \mathop{\mathbb{E}}_{q_{1:M}} [\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1}]^{-1} \mathop{\mathbb{E}}_{q_{1:M}} [\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k)])^{\mathrm{T}} \mathop{\mathbb{E}}_{q_{1:M}} [\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1}] \\
&\quad \times (\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_k - \mathop{\mathbb{E}}_{q_{1:M}} [\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1}]^{-1} \mathop{\mathbb{E}}_{q_{1:M}} [\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k)]) \\
&+ (\boldsymbol{x}_k - \mathrm{A}\boldsymbol{x}_{k-1} - \mathop{\mathbb{E}}_{q_{1:M}} [\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1}]^{-1} \mathop{\mathbb{E}}_{q_{1:M}} [\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k)])^{\mathrm{T}} \mathop{\mathbb{E}}_{q_{1:M}} [\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1}] \\
&\quad \times (\boldsymbol{x}_k - \mathrm{A}\boldsymbol{x}_{k-1} - \mathop{\mathbb{E}}_{q_{1:M}} [\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1}]^{-1} \mathop{\mathbb{E}}_{q_{1:M}} [\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k)]) \Bigg] + c \qquad (45) \\
=&\log \mathrm{N}(\boldsymbol{x}_0; \boldsymbol{x}_{0|0}, \mathrm{P}_{0|0}) \\
&-\frac{1}{2}\sum_{k=1}^{K} \Big[\log \mathrm{N}(\boldsymbol{y}_k; \mathrm{C}\boldsymbol{x}_k + \overline{\boldsymbol{\epsilon}}_k, \overline{\mathrm{R}}_k) + \log \mathrm{N}(\boldsymbol{x}_k; \mathrm{A}\boldsymbol{x}_{k-1} + \overline{\boldsymbol{\omega}}_{k-1}, \overline{\mathrm{Q}}_{k-1}) \Big] + c, \qquad (46)
\end{aligned}
$$

where $\mathbb{E}_{q_{1:M}}$ is short-hand for $\mathbb{E}_{\prod_{i=1}^{M} q_i(\boldsymbol{\theta}_{1:K,i})}$, and $\overline{\boldsymbol{\epsilon}}_k$, $\overline{\mathrm{R}}_k$, $\overline{\boldsymbol{\omega}}_{k-1}$, and $\overline{\mathrm{Q}}_{k-1}$ follow the notation of Algorithm 5. Thus, $q(\boldsymbol{x}_{0:K})$ is the smoothing posterior of the linear–Gaussian SSM with the process noise distribution $\boldsymbol{w}_k \sim \mathrm{N}(\overline{\boldsymbol{\omega}}_k, \overline{\mathrm{Q}}_k)$ and the measurement noise distribution $\boldsymbol{e}_k \sim \mathrm{N}(\overline{\boldsymbol{\epsilon}}_k, \overline{\mathrm{R}}_k)$. A marginal smoothing posterior of two consecutive states is therefore

$$
q(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}) = \mathrm{N}\left(\begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{x}_{k-1} \end{bmatrix}; \begin{bmatrix} \boldsymbol{x}_{k|K} \\ \boldsymbol{x}_{k-1|K} \end{bmatrix}, \begin{bmatrix} \mathrm{P}_{k|K} & \mathrm{P}_{k|K} \mathrm{G}_{k-1}^{\mathrm{T}} \\ \mathrm{G}_{k-1} \mathrm{P}_{k|K} & \mathrm{P}_{k-1|K} \end{bmatrix} \right), \qquad (47)
$$

where $\boldsymbol{x}_{k|K}$ and $\mathrm{P}_{k|K}$ as well as $\boldsymbol{x}_{k-1|K}$ and $\mathrm{P}_{k-1|K}$ are the smoothing means and covariance matrices of the Rauch–Tung–Striebel

smoother (RTSS), and G_{k-1} is the gain matrix in the RTSS [18, Ch. 8]. This RTSS is computed in Algorithm 5. Let us denote $\overline{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}_{k|K} \\ \boldsymbol{x}_{k-1|K} \end{bmatrix}$ and $\overline{\mathrm{P}} = \begin{bmatrix} \mathrm{P}_{k|K} & \mathrm{P}_{k|K}\mathrm{G}_{k-1}^{\mathrm{T}} \\ \mathrm{G}_{k-1}\mathrm{P}_{k|K} & \mathrm{P}_{k-1|K} \end{bmatrix}$. The VB approximation of the latent parameter $\boldsymbol{\theta}_{1:K,i}$ is then by (17)

$$\begin{aligned}
&\log q(\boldsymbol{\theta}_{1:K,i}) \\
&= -\frac{1}{2}\sum_{k=1}^{K} \mathop{\mathbb{E}}_{q_{-i}} \Bigg[\mathop{\mathbb{E}}_{q(\boldsymbol{x}_k, \boldsymbol{x}_{k-1})} \big[\log(\det(\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k))) + \log(\det(\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k))) \\
&\quad + (\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_k - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} (\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_k - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k)) \\
&\quad + (\boldsymbol{x}_k - \mathrm{A}\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} (\boldsymbol{x}_k - \mathrm{A}\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k)) \big] \\
&\quad - 2\log p(\boldsymbol{\theta}_k) \Bigg] + c
\end{aligned} \tag{48}$$

$$\begin{aligned}
&= -\frac{1}{2}\sum_{k=1}^{K} \mathop{\mathbb{E}}_{q_{-i}} \Big[\log(\det(\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k))) + \log(\det(\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k))) \\
&\quad + (\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_{k|K} - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} (\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_{k|K} - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k)) \\
&\quad + \mathrm{tr}\left\{ \mathrm{P}_{k|K}\mathrm{C}^{\mathrm{T}} \Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} \mathrm{C} \right\} \\
&\quad + ([\mathrm{I} \; -\mathrm{A}]\overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} ([\mathrm{I} \; -\mathrm{A}]\overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k)) \\
&\quad + \mathrm{tr}\left\{ \overline{\mathrm{P}} [\mathrm{I} \; -\mathrm{A}]^{\mathrm{T}} \Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} [\mathrm{I} \; -\mathrm{A}] \right\} - 2\log p(\boldsymbol{\theta}_k) \Big] + c,
\end{aligned} \tag{49}$$

where q_{-i} is short-hand for $\prod_{j \neq i} q_j(\boldsymbol{\theta}_{1:K,j})$. These distributions are solved in Algorithm 5.

A VB filter step requires the distributions $p(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, \boldsymbol{\theta}_k | \boldsymbol{y}_{1:k})$, and

$$\begin{aligned}
&\log p(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, \boldsymbol{\theta}_k | \boldsymbol{y}_{1:k}) \\
&= -\frac{1}{2} \Big[(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1|k-1})^{\mathrm{T}} \mathrm{P}_{k-1|k-1}^{-1} (\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1|k-1}) \\
&\quad + \log(\det(\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k))) + \log(\det(\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k))) \\
&\quad + (\boldsymbol{x}_k - \mathrm{A}\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1} (\boldsymbol{x}_k - \mathrm{A}\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k)) \\
&\quad + (\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_k - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))^{\mathrm{T}} \Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1} (\boldsymbol{y}_k - \mathrm{C}\boldsymbol{x}_k - \boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k)) - 2\log p(\boldsymbol{\theta}_k) \Big] + c.
\end{aligned} \tag{50}$$

This gives

$$\begin{aligned}
&\log q(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k) \\
&= -\frac{1}{2} \mathop{\mathbb{E}}_{q_{1:M}} \Big[(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1|k-1})^{\mathrm{T}} \mathrm{P}_{k-1|k-1}^{-1} (\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1|k-1})
\end{aligned}$$

$$+(\boldsymbol{x}_k-\mathrm{A}\boldsymbol{x}_{k-1}-\boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))^{\mathrm{T}}\Sigma_{\boldsymbol{w}}(\boldsymbol{\theta}_k)^{-1}(\boldsymbol{x}_k-\mathrm{A}\boldsymbol{x}_{k-1}-\boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{\theta}_k))$$
$$+(\boldsymbol{y}_k-\mathrm{C}\boldsymbol{x}_k-\boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))^{\mathrm{T}}\Sigma_{\boldsymbol{e}}(\boldsymbol{\theta}_k)^{-1}(\boldsymbol{y}_k-\mathrm{C}\boldsymbol{x}_k-\boldsymbol{\mu}_{\boldsymbol{e}}(\boldsymbol{\theta}_k))\Big]+c \tag{51}$$
$$=-\frac{1}{2}\Big[(\boldsymbol{x}_{k-1}-\boldsymbol{x}_{k-1|k-1})^{\mathrm{T}}\mathrm{P}_{k-1|k-1}^{-1}(\boldsymbol{x}_{k-1}-\boldsymbol{x}_{k-1|k-1})$$
$$+(\boldsymbol{x}_k-\mathrm{A}\boldsymbol{x}_{k-1}-\overline{\boldsymbol{\omega}}_k)^{\mathrm{T}}\overline{\mathrm{Q}}_k^{-1}(\boldsymbol{x}_k-\mathrm{A}\boldsymbol{x}_{k-1}-\overline{\boldsymbol{\omega}}_k)$$
$$+(\boldsymbol{y}_k-\mathrm{C}\boldsymbol{x}_k-\overline{\boldsymbol{\epsilon}}_k)^{\mathrm{T}}\overline{\mathrm{R}}_k^{-1}(\boldsymbol{y}_k-\mathrm{C}\boldsymbol{x}_k-\overline{\boldsymbol{\epsilon}}_k)\Big]+c \tag{52}$$
$$=-\frac{1}{2}\Bigg[\left(\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{x}_{k-1}\end{bmatrix}-\begin{bmatrix}\mathrm{A}\boldsymbol{x}_{k-1|k-1}+\overline{\boldsymbol{\omega}}_{k-1}\\ \boldsymbol{x}_{k-1|k-1}\end{bmatrix}\right)^{\mathrm{T}}$$
$$\times\begin{bmatrix}\mathrm{AP}_{k-1|k-1}\mathrm{A}^{\mathrm{T}}+\overline{\mathrm{Q}}_{k-1} & \mathrm{AP}_{k-1|k-1}\\ \mathrm{P}_{k-1|k-1}\mathrm{A}^{\mathrm{T}} & \mathrm{P}_{k-1|k-1}\end{bmatrix}^{-1}\left(\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{x}_{k-1}\end{bmatrix}-\begin{bmatrix}\mathrm{A}\boldsymbol{x}_{k-1|k-1}+\overline{\boldsymbol{\omega}}_{k-1}\\ \boldsymbol{x}_{k-1|k-1}\end{bmatrix}\right)$$
$$+\left(\boldsymbol{y}_k-\begin{bmatrix}\mathrm{C} & \mathrm{O}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{x}_{k-1}\end{bmatrix}-\overline{\boldsymbol{\epsilon}}_k\right)^{\mathrm{T}}\overline{\mathrm{R}}_k^{-1}\left(\boldsymbol{y}_k-\begin{bmatrix}\mathrm{C} & \mathrm{O}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{x}_{k-1}\end{bmatrix}-\overline{\boldsymbol{\epsilon}}_k\right)\Bigg]+c \tag{53}$$
$$=\log\mathrm{N}\left(\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{x}_{k-1}\end{bmatrix};\begin{bmatrix}\boldsymbol{x}_{k|k-1}\\ \boldsymbol{x}_{k-1|k-1}\end{bmatrix},\begin{bmatrix}\mathrm{P}_{k|k-1} & \mathrm{AP}_{k-1|k-1}\\ \mathrm{P}_{k-1|k-1}\mathrm{A}^{\mathrm{T}} & \mathrm{P}_{k-1|k-1}\end{bmatrix}\right)$$
$$+\log\mathrm{N}\left(\boldsymbol{y}_k-\overline{\boldsymbol{\epsilon}}_k;\begin{bmatrix}\mathrm{C} & \mathrm{O}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{x}_{k-1}\end{bmatrix},\overline{\mathrm{R}}_k\right)+c, \tag{54}$$

where $\mathbb{E}_{q_{1:M}}$ is short-hand for $\mathbb{E}_{\prod_{i=1}^{M}q_i(\boldsymbol{\theta}_{1:K,i})}$. Thus, $q(\boldsymbol{x}_{k-1},\boldsymbol{x}_k)$ is the Gaussian distribution whose mean and covariance matrix are solved using a Kalman filter (KF) update where the state variable is $\left[\begin{smallmatrix}\boldsymbol{x}_k\\ \boldsymbol{x}_{k-1}\end{smallmatrix}\right]$. This KF update is computed in Algorithm 4. The derivation of the filtering approximation $q_i(\boldsymbol{\theta}_{k,i})$ is similar to that of the smoother case in (49).

PUBLICATION 1

Henri Nurminen, Anssi Ristimäki, Simo Ali-Löytty, and Robert Piché: Particle filter and smoother for indoor localization. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, October 2013.
DOI: 10.1109/IPIN.2013.6817903

2013 International Conference on Indoor Positioning and Indoor Navigation, 28–31st October 2013

Particle filter and smoother for indoor localization

Henri NURMINEN, Anssi RISTIMÄKI, Simo ALI-LÖYTTY and Robert PICHÉ
Tampere University of Technology, Tampere, Finland
Emails: {henri.nurminen, anssi.ristimaki, simo.ali-loytty, robert.piche}@tut.fi

***Abstract*—We present a real-time particle filter for 2D and 3D hybrid indoor positioning. It uses wireless local area network (WLAN) based position measurements, step and turn detection from a hand-held inertial sensor unit, floor plan restrictions, altitude change measurements from barometer and possibly other measurements such as occasional GNSS fixes. We also present a particle smoother, which uses future measurements to improve the position estimate for non-real-time applications. A light-weight fallback filter is run in the background for initialization, divergence monitoring and possibly re-initialization. In real-data tests the particle filter is more accurate and consistent than the methods that do not use floor plans. An example is shown on how smoothing helps to improve the filter estimate. Moreover, a floor change case is presented, in which the filter is capable of detecting the floor change and improving the 2D accuracy using the floor change information.**

keywords: indoor positioning; framework for hybrid positioning; particle filtering; particle smoothing; signal strength based methods

I. Introduction

Accurate real-time indoor positioning relying solely on existing architecture that has not been built for positioning purposes remains a challenging technical problem. An inexpensive wide-availability solution is wireless local area network (WLAN) positioning, which, however, suffers from poor accuracy in many cases especially if no positioning-specific modifications are done to the WLAN network. Inertial motion sensors (IMS) such as accelerometer and gyroscope are usable for pedestrian dead reckoning (PDR), which estimates the current position given the previous position. If the initial state is given, IMS can typically estimate the user's path accurately only over a short period because the biased error accumulates over time. Using time-series algorithms, WLAN positioning and PDR can be combined so that the absolute position information of WLANs is used to correct the drift of the PDR. One way to improve the accuracy further by time-series is to incorporate floor plans. In indoor positioning the floor plans include information about forbidden paths, such as paths crossing walls or other obstacles. The floor plan information results in non-linear motion models, whose estimation is very challenging in analytical methods.

One solution for the hybrid time-series positioning problem with nonlinear models is the particle filter, which is based on Bayesian statistical theory and Monte Carlo (MC) simulation. Particle filters provide a set of weighted MC samples of the state at each time instant [1, Ch. 2.5.4]. These samples are called particles. In conventional hybrid indoor positioning, the MC samples of position and possibly other position-related components are updated using the PDR information. WLANs and floor plan are then used to enhance the estimate by updating the particle weights. This approach with various modifications has been used in [2–8], among others.

We have implemented a particle filter that uses accelerometers, gyroscope, WLAN received signal strengths (RSS), floor plans and barometer. One of the novel features of our filter is divergence detection and re-initialization based on running a light-weight fallback filter in the background of the particle filter. This scheme aims to remedy some shortcomings of the approaches in [7] and [9]. Another novel feature is that, because the positioning is intended for hand-held rather than foot-mounted devices, we use the IMS only for step detection and measuring heading changes, and estimate the step length in the particle filter. We also present a method and some results of hybrid multi-floor indoor positioning using particle filter.

We also pay special attention to the use of *consistent* measurement models, that is, the dispersions of the measurement distributions are required to be realistic. We infer the WLAN position using RSS measurements and simple path loss (PL) models whose parameters as well as the access point positions are estimated off-line using learning data. We also infer the measurement uncertainty based on the WLAN environment and the uncertainties of the PL parameters. This approach enables better consistency than some previous papers that assume constant covariance for the WLAN position measurements, e.g. [2, 5, 10].

Particle smoothing for non-realtime estimation of the track of the user is also covered in this paper. The approach of [4] stores the particle histories and uses the weights of the current particles also for historical particles. A problem with this approach is that approximations of the smoothed distributions tend to be degenerate. Our smoothing method is based on forward-backward recursions [11, p. 204], and this algorithm does not suffer as much from the degeneracy problem.

We test our filter and smoother in a building at Tampere University of Technology, Finland using the existing WLAN architecture and a hand-held inertial sensor unit including barometer. No compass is used. To evaluate the importance of map information, we test filters operating with and without floor plans. Furthermore, based on statistics of the Monte Carlo error of the filter we find that 400 particles is enough to achieve reasonable estimation accuracy.

This paper is organized as follows. In section II it is described how PDR, WLAN, barometer and floor plan information are used for positioning and in section III particle filter algorithms for 2D and 3D cases are presented. Section IV covers particle smoothing for 2D positioning. In section

V the test results are presented and section VI concludes the paper.

Notations: $t_3(\mathbf{m}, \mathrm{P})$ refers to the non-central scaled Student's t distribution with 3 degrees of freedom and with mean $\mathbf{m}$ and covariance matrix P (shape matrix $\frac{1}{3}\mathrm{P}$), and $t_3(\mathbf{x}|\mathbf{m}, \mathrm{P})$ refers to its probability density function (pdf) evaluated at $\mathbf{x}$.

II. Used measurements

A. Pedestrian dead reckoning

Our 2D filter implements an inertial pedestrian dead reckoning (PDR) system using MEMS-based (Micro Electro Mechanical Systems) accelerometer, gyroscope and barometer data. In many sources, the sensor unit is assumed to be attached to the user's shoe. However, this is not practical in many applications, in which the user is a private person, who cannot be assumed to attach any extra devices to his or her clothing. For example, if the positioning system is designed to function in a mobile device, very little can be assumed of the sensors' placements. Therefore, in the test cases presented in this paper the sensor unit is held in the hand. It is only assumed that the sensor unit's orientation with respect to the user does not change or the orientation can be estimated continuously.

Since the sensor unit is hand-held, the accelerometer data are much noisier than for shoe-mounted systems. Thus, our PDR algorithm is not based on computing displacements by double integration but on detecting the footsteps and estimating each step's length. Steps are detected based on the norm of the low-pass filtered acceleration as suggested by Leppäkoski et al. in [2], and the change of the user's heading during the step is inferred from the average angular velocity in the 2D plane indicated by the gyro. The authors of [2] use the step duration to infer the step length, and other methods have also been proposed [12]. However, we found that this method is not reliable enough, and in this paper the step length is estimated on the higher layer by the particle filter.

For example, [4] uses magnetometer (i.e. compass) data to obtain absolute heading information, but we do not use it because its reliability is questionable in many indoor environments [13]. Instead, the absolute heading is inferred only by the particle methods.

The barometer measures the air pressure from which the changes in altitude can be inferred accurately within short time intervals. However, the reliability may suffer from changes in the indoor air conditioning systems and in longer time periods from changes in the outdoor air pressure. Thus, continual recalibration of the mapping between barometric pressure and altitude is required. Barometer information is used by the 3D particle filter to detect floor changes.

B. Positioning with WLAN measurements

Besides floor plans, WLANs are the only source of absolute position information in the system, and WLAN is the only information channel that can produce a static snapshot estimate of the position. The position information contained by the WLANs is based on training data that must be collected from each floor of each building beforehand. It is assumed that the collected fingerprints contain accurately known position, list of the identifiers of the heard access points (AP) and the corresponding received signal strength indicators (RSSI). Furthermore, it is assumed that the RSSI readings of different devices can be calibrated so that they are comparable and that the received signal strength (RSS) in dBm can be computed based on the RSSI [14, 15]. All the data used in this paper's tests are collected with similar devices.

There are several ways to infer the user position using WLAN signals. This paper uses only RSS, since their usage does not require any external hardware modifications, and they can be measured by the receiving terminal alone. To average out noise and to keep the number of stored parameters small, the standard logarithmic path loss model is used to model the dependency of RSS from user position:

$$P = A - 10n \log_{10}(||\mathbf{r} - \mathbf{m}||) + w_{\mathrm{PL}}, \tag{1}$$

where P is the measured RSSI level, $\mathbf{r}$ is the user's position, $\mathbf{m}$ the AP's position, and $w_{\mathrm{PL}} \sim \mathrm{N}\left(0, \sigma^2\right)$ is a normally distributed shadowing component. For the shadowing standard deviation we used the constant $6\,\mathrm{dB}$. The model parameters A and n as well as the location of the AP are estimated by the Gauss–Newton method as in [16].

The positioning algorithm is also a Gauss–Newton method presented in [16]. Since the WLAN measurements are in this paper fused with other types of position measurements in time-series, it is critical that the variance of the position estimate is estimated correctly. Therefore, the uncertainties of the path loss parameters and AP positions are also estimated and used in the positioning phase as suggested in [16]. For example [2, 5, 10] use a global uncertainty parameter for the position coordinates, but we have found this assumption to be unsatisfactory, since the achievable precision in the position domain depends heavily on the WLAN environment; it can vary significantly even within one building.

C. Floor plan information in positioning

Another means to compensate the sensor drifts and to make the estimate more accurate is to use floor plans. This paper assumes that the floor plans of every floor for every building of interest are available. A floor plan is a set of thin walls, each described by five numbers: the floor and the coordinates of the end points of the wall. Doors are modeled as gaps in the walls. Furthermore, some additional information is included such as floor heights and the locations of spaces that allow floor change, such as staircases and elevators. Information on furniture or other movable objects is not used due to its changeable nature.

Inaccuracies must be taken into account. There may be errors in the maps, such as missing or nonexistent doors or walls. Nonexistent walls and missing doors in the maps can be coped with by allowing a small probability of going through a wall [10]. This is a trade-off between accuracy and robustness of the probabilistic model, since some information is unavoidably lost if wall penetration is allowed.

This paper assumes that the building is known. Building detection is left for future research.

III. Particle filtering

A. State model

A particle filter is a Monte Carlo algorithm that approximates the posterior distributions $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ provided that certain Markov assumptions hold and the probability distributions $p(\mathbf{x}_0)$, $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and $p(\mathbf{y}_k|\mathbf{x}_k)$ are known and their density values are computable for each time index k. The components of the vector $\mathbf{x}$ are the state variables, and the vector $\mathbf{y}$ contains the measurements. No assumptions of linearity or Gaussianity have to be made.

In this article, the state consists of 2-dimensional position $\mathbf{r}$, heading α, step length ℓ and possibly altitude z:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{r}_k \\ \alpha_k \\ \ell_k \\ z_k \end{bmatrix}$$

The PDR output that contains step detection, heading change and altitude change is involved in the state transition model. The process noise is chosen to be Student's t distributed, which choice is discussed later in this section. Thus, the state transition density for each step index k is

$$\begin{aligned} & p(\mathbf{x}_{k+1}|\mathbf{x}_k) \\ =& p(\mathbf{r}_{k+1}, \alpha_{k+1}, \ell_{k+1}, z_{k+1}|\mathbf{r}_k, \alpha_k, \ell_k, z_k) \\ =& p(\alpha_{k+1}, \ell_{k+1}, z_{k+1}|\mathbf{r}_k, \alpha_k, \ell_k, z_k) \\ & \cdot p(\mathbf{r}_{k+1}|\alpha_{k+1}, \ell_{k+1}, z_{k+1}, \mathbf{r}_k, \alpha_k, \ell_k, z_k) \\ =& t_3\left(\begin{bmatrix} \alpha_{k+1} \\ \ell_{k+1} \\ z_{k+1} \end{bmatrix} | \begin{bmatrix} \alpha_k + \Delta_k \\ \ell_k \\ z_k + b_k \end{bmatrix}, \mathrm{Q}_k\right) \\ & \cdot t_3\left(\mathbf{r}_{k+1}|\mathbf{r}_k + \ell_{k+1} \cdot \begin{bmatrix} \cos(\alpha_{k+1}) \\ \sin(\alpha_{k+1}) \end{bmatrix}, \mathrm{P}_k\right) \end{aligned} \tag{2}$$

where Δ is the heading change indicated by the PDR and b is the altitude change indicated by the barometer. The footstep length's distribution is restricted *a priori* so that it is always a sensible step length. The process noise covariance matrix for position is $\mathrm{P}_k = \sigma_{x,y}^2 \cdot \mathrm{I}$, where $\sigma_{x,y}$ is a configuration parameter. The process noise covariance matrix for the rest of the state parameters α, ℓ and z is assumed to be diagonal, so it is

$$\mathrm{Q}_k = \begin{bmatrix} \Delta t_k \cdot \sigma_\Delta^2 & & \mathrm{O} \\ & \sigma_\ell^2 & \\ \mathrm{O} & & \sigma_b^2 \end{bmatrix},$$

with $\sigma_b = 0$ in the 2-dimensional method with known altitude.

The process noise variances are configuration parameters that are set off-line. We emphasize the significance of the variance parameters, since they determine the magnitude of the smoothing effect of the filter: if this is too low, useful time-series information is neglected, but if the filter is over-smoothing, the particle cloud may fail to cover the whole interesting state-space area. A filter with too small process noise variance might e.g. not find a correct door or narrow corridor when a turn happens after a long time since the last WLAN measurement. We found that it is especially recommendable to use somewhat larger noise variances than the actual accuracy of the PDR would indicate, if there is a danger that WLAN measurements contain outliers with respect to the assumed measurement uncertainties. If an outlying WLAN measurement biases the particle cloud, the state-space area of actual interest might become a low-probability area that is inadequately or not at all covered by the particles. Large process variances and the heavy-tailed Student's t distribution then increase the coverage of the particle cloud more rapidly. In the literature, this *ad-hoc* uncertainty increase is called jittering or roughening [17].

One aspect in the process noise variance is the ratio of the position noise to the noise of the motion model variables heading and step length. Because only position is measured directly, too large $\sigma_{x,y}$ with respect to σ_α and σ_ℓ results in incapability of estimating heading and step length. Furthermore, too small ratio results in over-learning heading and step length; their estimates tend to become biased in order to correct a bias in position estimate, which eventually leads to overcorrecting the position estimate, i.e. correcting the position estimate beyond what is needed. Therefore, more sophisticated PDR solutions with as little noise as possible are of great value.

The state estimate is corrected using two kinds of measurement models: floor plan and WLAN. Based on the floor plan information, the probability of the wall penetrating transitions is zero or at least small. Formally, the floor plan measurement model is expressed as

$$\mathbb{P}(C_k|\mathbf{r}_k, \mathbf{r}_{k-1}) = \epsilon, \tag{3}$$

where

$C_k =$ "There is a wall in the map that crosses the kth step".

Thus, the number ϵ is a permeability coefficient which models the probability that in the map there is a step-crossing wall in the map is actually nonexistent. The inequality $0 \leq \epsilon \ll 1$ should hold. In our tests, the particles that enter inaccessible areas as well as the particles that move out of the building are given weight zero.

Having impermeable walls i.e. having small ϵ tends to introduce some inefficiency to the system, since lots of particles are eliminated regularly, these particles thus becoming useless until the next resampling. Therefore, [7] proposes a retry procedure: if the generated particle collides with wall, it is regenerated several times before deletion. However, one must be careful with this. Firstly, wall collision checking is computationally the most expensive part of the filter. Secondly, retry changes the statistical model: If step length is being estimated by the particles, retry increases the probability of short steps in the state transition model. If step length is taken directly from the PDR as in [7], it reduces the weight of PDR step length measurement and gives more probability to a short step. Instead of this, we consider wall collisions to

indicate that the particles original state was improbable and the particle deserves to be given a lower weight. Retry could only be applied as a heuristic robustness-increasing method in limited cases where only a few particles are in the interesting state-space area, but this is not considered further in this paper.

The choice of parameter ϵ has significant influence on the filter behaviour: Allowing wall penetration, i.e. setting $\epsilon > 0$, adds robustness against errors in the map and, for example, against inconsistent WLAN measurement which tend to result in a too small particle cloud that does not cover the true position. On the other hand, compared to the model $\epsilon = 0$ somewhat less map information is used, and if the particle cloud is inconsistent, the choice $\epsilon > 0$ may allow the whole particle cloud to penetrate a wall without any effect on the weighting. If resampling is done when part of the cloud has penetrated a wall and the rest will penetrate at the next step, there is a risk that the particle estimate starts to lag from the true estimate. This phenomenon is due to the Monte Carlo approximation of the distribution, and techniques for avoiding it are a topic for future research.

The true likelihood of one WLAN measurement is not Gaussian, but it is approximated with a Gaussian by the Gauss–Newton method as described in [16]. Thus, the used WLAN measurement model is

$$p(\mathbf{y}_k|\mathbf{r}_k, \alpha_k, \ell_k, z_k) \overset{\mathbf{r}_k}{\propto} \mathrm{N}\left(\mathbf{r}_k|\mu_k(\mathbf{y}_k), \Sigma_k(\mathbf{y}_k)\right). \quad (4)$$

In our method the measurement covariance matrix $\Sigma_k(\mathbf{y}_k)$ is not a configuration parameter but it is returned by the Gauss–Newton method. For fusion with other measurement information, it is crucial that the covariance matrices are in accordance with the real precision of the WLAN measurements.

B. Particle filter algorithm

A particle filter approximates the presented model's posterior distributions with a set of weighted particles $\{(\mathbf{x}_k^i, W_k^i)|i \in \{1, \ldots, N\}\}$ that are random realizations of the state space [1, 18]. The random samples $\mathbf{x}_k^i$ follow the importance distribution, also known as the proposal distribution, and the weights W_k^i, also known as importance weights, make the sample approximate the true posterior in the sense that every moment of the distribution are approximated by weighted sums over the particle set:

$$\mathrm{E}(\mathbf{g}(\mathbf{x}_k)|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N} W_k^i \cdot \mathbf{g}(\mathbf{x}_k^i)$$

where $\mathbf{g}$ is an almost arbitrary Borel measurable function and N is the number of particles. This is conventionally denoted by the Dirac delta notation

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N} W_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i).$$

Thus, the posterior is approximated by a set of sharp distribution peaks.

In the initial phase the particle values are generated from the initial prior $p(\mathbf{x}_0)$ with equal weights. The particle filter is updated whenever a footstep is detected or a WLAN measurement is received. As for the Kalman-type filters, one update consists of two stages: the prediction stage, in which the particles of new time instant are generated from the pre-specified proposal distributions $q(\mathbf{x}_k|\mathbf{x}_{1:k-1}^i, \mathbf{y}_{1:k})$, and the update stage, in which the measurements are used to update the weights of the particles. Given the previous timestep's weights W_{k-1}^i, the current time instant's unnormalized weights $\tilde{W}_k^i$ are obtained using the formula

$$\tilde{W}_k^i = p(\mathbf{y}_k|\mathbf{x}_k^i)\mathbb{P}(C_k|\mathbf{x}_{k-1:k}^i) \cdot \frac{p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{1:k-1}^i, \mathbf{y}_{1:k})} \cdot W_{k-1}^i, \quad (5)$$

in which the first terms represent the measurement likelihood and the middle term the ratio of state model and proposal distribution. The normalization factors of the pdf's $p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)$ and $q(\mathbf{x}_k^i|\mathbf{x}_{1:k-1}^i, \mathbf{y}_{1:k})$ are not required, since the normalization constants are often well approximated by

$$W_k^i = \frac{\tilde{W}_k^i}{\sum_{i=1}^{N} \tilde{W}_k^i} \quad (6)$$

[1, Ch. 2.5.2].

The choice of the proposal distribution is a crucial ingredient of any particle filter. A rule of thumb is that the proposal distribution should be as close to the final posterior as possible. In this paper the state transition distribution $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is used as a proposal distribution.

A third stage is required by any particle filter to avoid all the weight concentrating to one particle: resampling [18, Ch. 3.3]. At the resampling stage the particles' weights are equalized by sampling with replacement N new particles from the old particle set with the old particle weights as probabilities. One drawback of this resampling method is the danger of sample impoverishment, which means that most of the particles are resampled to one or a few locations, which may lead to inadequate coverage of the interesting state space areas [18]. A conventional method for avoiding this is to resample only at some time steps. A standard approach is to resample only when the effective number of particles (ENP), $N_{\mathrm{eff},k} = 1/\sum_{i=1}^{N}(W_k^i)^2$, goes below some threshold [18, Ch. 3.3]. Since resampling increases the Monte Carlo variance of the estimate, the estimate reported before the resampling, and with step detection the resampling is only performed when a footstep is detected and the ENP condition is fulfilled.

With the model of this paper, the prediction stage updates the particle values with PDR readings and randomly generated noise modifying the weights of wall-crossing particles, and the update stage corrects the particle weights based on the WLAN measurement. In case a WLAN scan is not made at some time instant, the update stage is simply omitted. The detailed algorithm description is present in Algorithm 1.

C. Initialization and divergence monitoring

To ensure that the particle filter estimate converges to the true posterior in a reasonable number of time steps with a reasonable number of particles, it is crucial that the initial prior

Algorithm 1 Particle filter for 2D indoor positioning

1) For each $i \in \{1, \ldots, N\}$ set $W_0^i := \frac{1}{N}$ and generate $\mathbf{x}_0^i \leftarrow p(\mathbf{x}_0)$. Set the time index $k := 1$.
2) Set $\tilde{W}_k^i := W_k^i$. If no footstep is detected at time window index k, go to Phase 5. Otherwise, if $\frac{1}{\sum_{i=1}^N (W_{k-1}^i)^2} < N_{\text{eff,lim}}$, perform resampling.
3) For each $i \in \{1, \ldots, N\}$ generate
$$\begin{bmatrix} \alpha_k^i \\ \ell_k^i \end{bmatrix} \leftarrow t_3 \left(\begin{bmatrix} \alpha_{k-1}^i + \Delta_{k-1} \\ \ell_{k-1}^i \end{bmatrix}, \mathrm{Q}_{k-1} \right)$$
$$\mathbf{r}_k^i \leftarrow t_3 \left(\mathbf{r}_{k-1}^i + \ell_k^i \cdot \begin{bmatrix} \cos(\alpha_k^i) \\ \sin(\alpha_k^i) \end{bmatrix}, \mathrm{P}_{k-1} \right).$$
4) Set
$$\tilde{W}_k^{i^*} := \frac{\epsilon}{1-\epsilon} \tilde{W}_{k-1}^{i^*}$$
for all i^* such that there is a wall between $\mathbf{r}_{k-1}$ and $\mathbf{r}_k$.
5) If no WLAN measurement is obtained at time index k, go to Phase 6. Otherwise, perform the Gauss–Newton algorithm to obtain the WLAN position's mean μ_k and covariance matrix Σ_k. Set
$$\tilde{W}_k^i := \mathrm{N}\left(\mu_k | \mathbf{r}_k^i, \Sigma_k\right) \cdot \tilde{W}_{k-1}^i$$
for each $i \in \{1, \ldots, N\}$.
6) Normalize the weights by
$$W_k^i := \frac{\tilde{W}_k^i}{\sum_{i=1}^N \tilde{W}_k^i}.$$
Report the mean and covariance matrix for position
$$\hat{\mu}_k := \sum_{i=1}^N W_k^i \cdot \mathbf{r}_k^i, \;\; \hat{\Sigma}_k := \sum_{i=1}^N W_k^i \cdot (\mathbf{r}_k^i - \hat{\mu}_k)(\mathbf{r}_k^i - \hat{\mu}_k)^{\mathrm{T}}$$
If the positioning ends, stop. Otherwise, set $k := k + 1$ and go to Phase 2.

is accurate enough for each component of the state, i.e. it does not have a significant bias and the variance is small. Otherwise especially the convergence of heading and step length may be very slow, since they are not measured directly, but only through position. Especially if WLAN accuracy is low, there is also a danger that large open areas are overweighted, since inaccurate particles are less likely to be eliminated by wall collisions there than in more confined spaces.

The accuracy of the initial prior and thus the applied particle initialization method depends on the scenario. If the user comes from outdoors and has been using GPS or some other accurate positioning method, there is likely to be a feasible prior distribution for position and heading and even for step length. If, however, the positioning device is switched on indoors without any history of positioning data, there is no prior information.

One initialization scenario is that the particles have got stuck in an area that does not fit well with the WLAN measurements. Due to the "hard" nature of wall constraints even a relatively small bias in position may result in getting stuck, and the recovery may be slow or almost impossible. Allowing the particles to penetrate walls, i.e. setting $\epsilon > 0$ in Eq. (3) may help but does not completely resolve the issue, since wall penetrations delay the estimate, or in the case of multimodal cloud they may result in overweighting large open areas. Since the standard form of particle filter does not perform a global state space search, recovery from filter divergence requires that all or some of the particles are re-initialized.

Our solution to both the initialization problems is to maintain a robust light-weight fallback filter in the background. The fallback filter should perform global state space search and be independent of the floor plan constraints. For this we use the Kalman filter (KF) described by Raitoharju et al in [19]. This algorithm has 2D position and 2D step vector in the state, and the step detection and heading given by the PDR are fed in at the prediction stage and WLAN measurements at the update stage. Thus, the algorithm estimates simultaneously position, heading and footstep length using a linear state model. To make the algorithm more robust against outlying WLAN estimates, the measurements are assumed to have Student's t noise distribution. This model can be solved approximatively using a Variational Bayes-based version of KF [20].

In the case of unknown initial state the fallback KF is initialized with the mean of position at an arbitrary position such as the middle point of the building and large variance. As soon as the KF estimate has converged, i.e. its variance for the step is small enough, the KF estimate is used for initializing the particles. Generating the particles from the normal distribution defined by the KF output is straightforward.

After the first initialization and in the case of known initial state, the fallback KF is used for monitoring the quality of the particle cloud, i.e. for checking the re-initialization criterion and for estimating the distribution of the re-initialized particles. At re-initialization, it might be advantageous to re-initialize only $100 \cdot \lambda\,\%$ of the particle cloud and sample the rest from the old particle set, λ being a configuration parameter for which $0 \ll \lambda \leq 1$.

The criterion for when to re-initialize the existing particles needs some consideration. In [9], re-initialization is done when the Kullback–Leibler divergence (KLd) between the normalized measurement likelihood and the prior implied by the particles is large. However, we did not find this a suitable criterion, since the KLd measures also the difference in the uncertainties of the distributions, and the accuracy of the measurement is never dependent on the accuracy of prior. In [7] the re-initialization decision is based on the range measurements' deviations from the least deviating particle's prediction, and in [21] the unnormalized particle likelihoods are monitored. However, there might be outliers among the WLAN measurements, and in such a case the re-initialization might result in significant information losses. Therefore, we do not use the static WLAN measurement but the more robust KF estimate, which also considers the heading and step length of the particles. If the best particle is improbable given the

4-dimensional KF distribution, the particles are re-initialized based on the KF estimate. "Best" refers here to the particle that is closest to the KF estimate of all positively weighted particles. This algorithm is described in Algorithm 2.

Algorithm 2 Re-initialization of particle filter

1) After Phase 5 of Algorithm 1, feed the Kalman filter (KF) with PDR and WLAN estimate to get the KF estimate (μ_k^K, Σ_k^K).
2) Compute the step vector for each positively-weighted particle $i \in \{i|\tilde{W}_k^i > 0\}$

$$\mathbf{s}_k^i := \ell_k^i \cdot \begin{bmatrix}\cos(\alpha_k^i)\\ \sin(\alpha_k^i)\end{bmatrix}.$$

and compute the KF deviances

$$d_k^i := \left(\begin{bmatrix}\mathbf{r}_k^i\\ \mathbf{s}_k^i\end{bmatrix} - \mu_k^K\right)^{\mathrm{T}} (\Sigma_k^K)^{-1} \left(\begin{bmatrix}\mathbf{r}_k^i\\ \mathbf{s}_k^i\end{bmatrix} - \mu_k^K\right).$$

Select the smallest KF deviance

$$d_k^* := \min_{i\in\{i|\tilde{W}_k^i>0\}} (d_k^i).$$

3) If d_k^* exceeds the threshold value, sample $\mathrm{round}(\lambda N)$ new particles from the KF distribution and sample $N - \mathrm{round}(\lambda N)$ particles from the old weighted particle set with replacement. Give all the sampled particles equal weights. Go to Phase 6 of Algorithm 1.

D. 3D indoor positioning

Currently, an active field of indoor positioning research is floor estimation. For the end user, it is important to see the correct floor's map, and the choices of floor plan and WLAN models influence 2D positioning performance. A calibrated barometer is accurate in short-term positioning, but it drifts over time due to changes in the air pressure. Furthermore, it may also be biased by air conditioning systems.

Static floor estimation based on WLAN fingerprinting has been investigated by [22–24], among others. Article [25] also incorporates barometer information, and calibrates the barometer with WLAN. Results in multifloor particle filter positioning have been presented at least by [6, 26].

Provided that the barometer is in use, the extension of the presented particle filter algorithm into 3D position space is straightforward in principle. Given the means of the barometric pressures over the previous and latest time intervals ρ_{k-1} and ρ_k, the mean of the predicted altitude change in Eq. (2) is

$$b_k = \frac{RT}{gM} \ln\left(\frac{\rho_k}{\rho_{k-1}}\right), \tag{7}$$

where R is the molar gas constant, T is the temperature for which 293 K was used in our tests, g is the acceleration of gravity, and M is the molar mass of air [27, Ch. 2.1]. Provided that the floor elevations are known, a particle's floor is always implied by the particle's altitude, so the probability of each floor can be computed by summing the weights of the particles that are located in the floor. The reported position estimate is computed using only the particles of the most probable floor.

The WLAN measurements are used straightforwardly in the 3D method provided that the floor is always known in the learning phase: the path loss parameters are learnt separately for each floor, and in the positioning phase the used parameters are chosen separately for each particle according to the floor of the particle. Because the Gauss–Newton method returns only the mean and covariance matrix for position, this method normalizes the likelihood for each floor with respect to user position. However, the normalization constants are different for different floors, so some of the absolute floor information is neglected by this method. This is another topic for future research.

The floor plans provide extra information for the floor change cases, if staircases, elevators and other spaces that connect different floors are labeled there. The floor forms a similar kind of map restriction as the walls in planar positioning: there are "doors" that connect two floors, and otherwise the floor change is only possible with small probability ϵ_f, which is the probability that the used connector is missing from the map.

Similarly to the 2D setup, there is a possibility that the particle filter diverges, that is, it cannot find the true floor e.g. due to failure in finding the connector. Moreover, if the whole particle cloud changes floor without finding a connector, the weighting is not affected when $\epsilon_f > 0$, except that if the cloud is resampled during the penetration, the altitude estimate starts to lag. There could be a connector-independent fallback filtering algorithm for the altitude dimension also, and the estimate of this algorithm could be used to measure the adequacy of the particle cloud coverage and possibly for multifloor re-initialization. However, this idea was not implemented for this paper, but it is left for future research.

IV. Particle smoothing

A. Algorithm derivation

The goal of Bayesian smoothing is to compute the marginal smoothed distributions $p(\mathbf{x}_k|\mathbf{y}_{1:T})$ where $T > k$. Compared with Bayesian filtering also future measurements $\mathbf{y}_{k+1}, ..., \mathbf{y}_T$ are used to estimate the state $\mathbf{x}_k$, and hence it is possible to achieve better estimation accuracy. A computationally efficient way to do particle smoothing is to do it while filtering and keeping in memory the particle trajectories. This can be done by computing first the joint distribution for the whole time interval of interest $k \in \{0, \ldots, T\}$

$$p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) \approx \sum_{i=1}^{N} W_T^i \delta(\mathbf{x}_{0:T} - \mathbf{x}_{0:T}^i), \tag{8}$$

and then using the latest weights W_T^i and particles from each time step to approximate the smoothed marginals

$$p(\mathbf{x}_k|\mathbf{y}_{1:T}) \approx \sum_{i=1}^{N} W_T^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{9}$$

[28, p. 662, 674]. This type of smoothing is used in [4] where the Backtracking Particle Filter (BPF) is introduced.

BPF uses the latest weights to correct the estimate of the past state $\mathbf{x}_{k-m}$. In [4, p. 210, 212] the value of m is established empirically.

A problem with this smoothing method is that it produces degenerate approximations of smoothed distributions at times $k \ll T$ [28, p. 698], [29]. The smoothing method used in this paper is the Forward Filtering-Backward Smoothing (FFBS) which does not suffer that much from the degeneracy problem. In the FFBS the reweighing of the particles is based on the equation

$$p(\mathbf{x}_k|\mathbf{y}_{1:T}) = p(\mathbf{x}_k|\mathbf{y}_{1:k}) \int \frac{p(\mathbf{x}_{k+1}|\mathbf{x}_k)}{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})} p(\mathbf{x}_{k+1}|\mathbf{y}_{1:T}) d\mathbf{x}_{k+1}, \tag{10}$$

which implies the smoothing formula for the weights

$$W^i_{k|T} = W^i_k \cdot \left[\sum_{j=1}^{N} W^j_{k+1|T} \frac{p(\mathbf{x}^j_{k+1}|\mathbf{x}^i_k)}{\sum_{l=1}^{N} W^l_k p(\mathbf{x}^j_{k+1}|\mathbf{x}^l_k)} \right], \tag{11}$$

where W^i_k are the filtering weights [11, p. 204-205]. In a practical FFBS implementation particle filtering is done first, and then new weights are computed by (11). The smoothed distributions are then represented by the new weights and the particles from the filtering phase. A disadvantage of FFBS is that it requires $O(N^2T)$ operations to approximate $p(\mathbf{x}_k|\mathbf{y}_{1:T})$ [28, p. 700]. The algorithm for FFBS is given in Algorithm 3.

Algorithm 3 Forward Filtering-Backward Smoothing

1) Perform PF with Algorithm 1 and store particles $\mathbf{x}^i_t$ and weights W^i_t for each time window index $k \in \{1, \ldots, T\}$.
2) For time window index T, report the filter estimate. Set $k := T - 1$ and $W^i_{T|T} := W^i_T$.
3) Compute new weight for each particle $\mathbf{x}^i_k$ by

$$W^i_{k|T} := W^i_k \cdot \left[\sum_{j=1}^{N} W^j_{k+1|T} \frac{p(\mathbf{x}^j_{k+1}|\mathbf{x}^i_k)}{\sum_{l=1}^{N} W^l_k p(\mathbf{x}^j_{k+1}|\mathbf{x}^l_k)} \right]$$

4) Report the mean and covariance matrix for position

$$\hat{\mu}_k := \sum_{i=1}^{N} W^i_{k|T} \mathbf{r}^i_k, \;\; \hat{\Sigma}_k := \sum_{i=1}^{N} W^i_{k|T} (\mathbf{r}^i_k - \hat{\mu}_k)(\mathbf{r}^i_k - \hat{\mu}_k)^{\mathrm{T}}$$

If the positioning ends, stop. Otherwise, set $k := k - 1$ and go to Phase 3.

B. Particle smoother for indoor positioning

FFBS is implemented and tested in 2D indoor positioning scenarios. A fixed-interval smoothing approach is presented using the whole time series as the interval, but fixed-lag smoothing would also be possible using the same formulas. The map information is not used in the smoothing phase in order to make the computations lighter. The map information is, however, taken into account in the filter weights W^i_k.

The state model of (2) can be written as a stochastic difference equation

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{w}_k), \tag{12}$$

where $\mathbf{w}_k \sim t_3\left(\mathbf{0}, \begin{bmatrix} \mathrm{P}_k & \mathrm{O} \\ \mathrm{O} & \mathrm{Q}_k \end{bmatrix}\right)$ is the process noise and $\mathbf{f}_k$ is the state transition function. For computing the state transition densities $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$, the state model is linearized by

$$\begin{aligned}
&\mathbf{f}_k(\mathbf{x}_k, \mathbf{w}_k) \\
&= \begin{bmatrix} r_{x,k} + (\ell_k + w_{4,k})\cos(\alpha_k + \Delta_k + w_{3,k}) + w_{1,k} \\ r_{y,k} + (\ell_k + w_{4,k})\sin(\alpha_k + \Delta_k + w_{3,k}) + w_{2,k} \\ \alpha_k + \Delta_k + w_{3,k} \\ \ell_k + w_{4,k} \end{bmatrix} \\
&\approx \begin{bmatrix} r_{x,k} + \ell_k \cos(\alpha_k + \Delta_k) \\ r_{y,k} + \ell_k \sin(\alpha_k + \Delta_k) \\ \alpha_k + \Delta_k \\ \ell_k \end{bmatrix} \\
&\quad + \begin{bmatrix} 1 & 0 & -\ell_k \sin(\alpha_k + \Delta_k) & \cos(\alpha_k + \Delta_k) \\ 0 & 1 & \ell_k \cos(\alpha_k + \Delta_k) & \sin(\alpha_k + \Delta_k) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_{1,k} \\ w_{2,k} \\ w_{3,k} \\ w_{4,k} \end{bmatrix} \\
&\triangleq \mathbf{f}^*_k(\mathbf{x}_k) + \mathrm{C}_k \mathbf{w}_k.
\end{aligned} \tag{13}$$

Note that $\mathrm{C}_k\mathbf{w}_k \sim t_3\left(\mathbf{0}, \mathrm{C}_k \begin{bmatrix} \mathrm{P}_k & \mathrm{O} \\ \mathrm{O} & \mathrm{Q}_k \end{bmatrix} \mathrm{C}_k^{\mathrm{T}}\right)$, because if $\mathbf{w} \sim t_v(\mu, \Sigma)$ and C is invertible, $\mathrm{C}\mathbf{w} \sim t_v(\mathrm{C}\mu, \mathrm{C}\Sigma\mathrm{C}^{\mathrm{T}})$ holds [30, Ch. 1.9].

V. Testing

A. Equipment and environment

Particle filter and FFBS particle smoother are tested in Tampere University of Technology campus building Tietotalo. The presented test tracks are located in corridors that are surrounded by offices. Only the existing WLAN architecture is used. The inertial sensor unit is Xsens MTx. Acer Iconia Tab W500 tablet PC with Windows 7 OS is used to log the WLAN measurements in the learning and positioning phase. The used indoor maps are HERE Destination Maps. The reference locations are set manually by the user by tapping the floor plan figure on the tablet's screen. Positioning algorithms are computed with MATLAB using MEX-files in the most critical parts of the code to speed up the computation.

The results of two positioning tracks are shown in this paper. The first one is in one floor (the 2D track), and the other one contains a floor change from floor 2 to floor 3 (the 3D track). The WLAN Learning data have been collected from each floor of the five-story building. At the time of the test track collection, the learning data were about three months old. Floor 2 has path loss models for 103 APs and floor 3 for 126 APs. Typically, 15–40 APs are observed at each scan, 25 on average. During the data collection, the inertial sensor unit was held in the hand avoiding rotations with respect to the user. WLANs were scanned every ten seconds. The floor change in the 3D track was made using stairs.

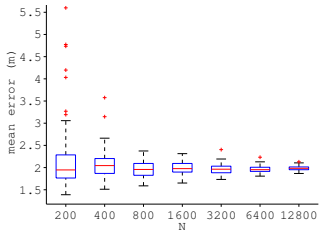
The solvers are given the correct initial position and heading with variances $(2\,\mathrm{m})^2 \cdot \mathrm{I}_{2\times 2}$ and $(10^\circ)^2$. The step length prior is $\mathrm{N}\left(0.7\,\mathrm{m}, (0.015\,\mathrm{m})^2\right)$. In the 3D test, the initial floor is assumed to be known, and the prior for altitude is the

normal distribution with mean in the correct floor's altitude and variance depending on the floor's elevation.

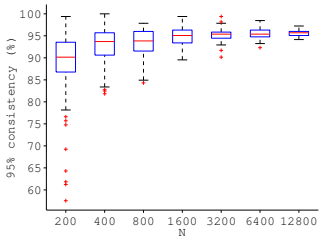
B. Results and discussion

Figure 1 shows box plots of the mean errors of 100 particle filter runs on the 2D track with different numbers of particles. From the plot we infer that 400 particles is enough to achieve the best possible median performance. However, if the reliability of the filter is to be improved, the number can be increased. The convergence rate of Monte Carlo integration is known to be $O(\sqrt{N})$ [18, Ch. 3.1]. Adding dimensions, such as altitude, to the state tends to slow up convergence.

Our measurement device is capable of processing the 400-particle filter in real time. Our implementation is not highly optimized. However, the set of walls has been divided into groups so that wall crossing is checked only with the walls that are close to the moving particle. Thus, the complexity of the bottleneck phase does not depend on the total number of walls and floors.



(a)



(b)

Figure 1. Boxplots of the mean errors (1a) and 95 % consistencies (1b) of 100 particle filter runs on the 2D track varying the particle number N

For the 2D track, the error statistics for filters and smoother are presented in Table I. Particle filter (PF) and smoother (FFBS) are the only methods using floor plans. The reference methods are static WLAN, the Kalman filter (KF) described in Section III-C, and particle filter without map information. The errors of the particle filters and FFBS are averages of the error statistics of 100 Monte Carlo simulations with 400 particles. The error statistics are empirical mean, median and 95 % quantile of the weighted particle mean's errors with respect to the reference trajectory. Note that the error of the static WLAN algorithm is computed only at the time instants when WLAN measurement is received. For all the other algorithms, the error is computed with 0.5 s interval. The 95 % consistency is determined using the Gaussian consistency test [31, p. 235]: A solver is deemed to be consistent at a certain time step if the true position is within the 95 %-ellipse of the posterior distribution, assuming normality of the posterior. The closer this number is to 95 %, the better the weighted covariance matrix of the particle cloud corresponds to the realized error. More rigorous consistency evaluation of non-Gaussian distributions is left for future research.

TABLE I. Positioning results for static WLAN, KF, PF with and without floor plan information and FFBS on the 2D track

	mean error (m)	median error(m)	95 % quant. of errors (m)	95 % cons. (%)
static WLAN	7.5	4.7	20.7	80
Kalman filter	4.8	3.2	16.3	83
PF (no map)	5.2	3.3	17.7	83
PF	2.0	1.3	6.0	92
FFBS	1.4	0.8	4.9	61

The KF combines WLAN and PDR in time-series. Based on the error statistics of Table I, the KF reduces the error by about 25 %. PF without floor plans uses the same measurements and gives similar results with the KF, which validates the linear motion model of the KF. Incorporating the floor plans (map) to the PF yields further significant decrease in errors. Also, the estimate crosses walls more rarely, which provides a better user experience. The particle smoother FFBS decreases the errors further. The performance differences of PF and FFBS in one example case are analyzed in the following paragraph.

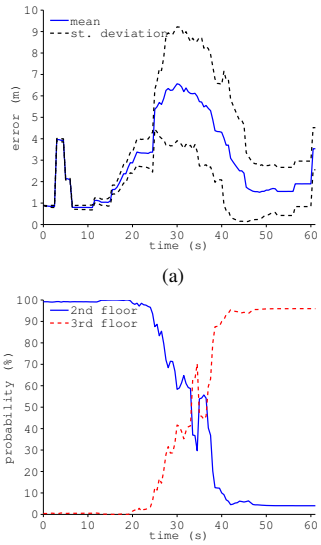
Estimated tracks for one Monte Carlo simulation are in Figure 2. In subfigures 3a and 3b there are snapshots of the particle cloud and particle filter estimate moving from one corridor to another. In subfigure 3a the weighted mean is penetrating through the walls but there are particles also in the corridor where the reference position is. In subfigure 3b, most of the particles are resampled in the corridor and the weighted mean also moves there. Thus, due to the future information the particles that went along the corridors are given more weight in the smoothing phase. The improvement provided by smoothing is also visible in Figure 2.

Figure 2. Estimated tracks for PF and FFBS

(a) (b)

Figure 3. Particle cloud moving from one corridor to another and the estimated particle filter track

Figure 4 shows the distribution of 2D error as well as the average floor probabilities estimated by the particles for the 3D track. The plots are based on 100 Monte Carlo iterations, and the initial prior is slightly biased and has somewhat more uncertainty than in the 2D case, the variances being $(5\,\mathrm{m})^2 \cdot \mathrm{I}_{2\times 2}$ for position and $(20^\circ)^2$ for heading. In this case the filter is capable of detecting the floor change and also reducing the 2D error using the floor change information. At first, the 2D error increases, because apart from the particles that are in the new floor inside a connector also those that have stayed in the old floor may survive. However, the first WLAN measurement after the floor change fits better with the particles in the new floor, so the estimate tends to converge to the correct sub-cloud.



(a)

(b)

Figure 4. Distribution for 2D error of position (4a) and average floor probabilities (4b) in the 3D track for 100 Monte Carlo simulations

VI. Conclusions

This article presented a particle filtering and smoothing algorithms for the hybrid indoor positioning problem involving measurements from accelerometer, gyroscope, barometer, WLAN and floor plans. It is also straightforward to add other measurement sources, such as occasional GNSS measurements. The particle algorithms estimate user's position, heading, footstep length and altitude. The particles can be initialized and, in the case of filter divergence, re-initialized using a light-weight fallback filter.

It was shown that floor plans provide a significant improvement to the positioning accuracy and consistency and that an adequate number of particles might be 400, which should be feasible for modern high-end mobile devices. Cases were presented where smoothing helps to improve the filter estimate. Moreover, a floor change case was presented, which showed that the filter was capable of detecting the floor change and improving also the 2D accuracy using this information.

Indoor positioning using particle methods still provides several future challenges. Footstep length estimation using inertial motion sensors will be studied: the coming method might include a user-specific constant, which could then be estimated by the particle filter. The wall and floor permeability models will be considered further to remedy the wall-penetration problem discussed in Section III-A. Building detection will be tested. Seamlessly 3D particle methods will be developed further by introducing multifloor fallback filter.

Acknowledgment

This research was funded by Nokia Inc.. The authors are grateful to Jari Syrjärinne and Lauri Wirola for support and advice. Toni Fadjukoff, Jussi Parviainen, Jussi Collin and Matti Raitoharju are thanked for their participation in development, implementation and testing of the algorithms.

References

[1] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[2] H. Leppäkoski, J. Collin, and J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals," *Journal of Signal Processing Systems*, vol. 71, no. 3, pp. 287–296, June 2013.

[3] H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U. D. Hanebeck, "WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors," in *4th Workshop on Positioning, Navigation and Communication, 2007. WPNC '07.*, 2007, pp. 1–7.

[4] Widyawan, M. Klepal, and S. Beauregard, "A Backtracking Particle Filter for Fusing Building Plans with PDR Displacement Estimates," in *5th Workshop on Positioning, Navigation and Communication, 2008. WPNC 2008.*, 2008, pp. 207–212.

[5] F. Evennou, M. Francois, and E. Novakov, "Map-aided indoor mobile positioning system using particle filter," in *2005 IEEE Wireless Communications and Networking Conference*, vol. 4, March 2005, pp. 2490–2494.

[6] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th inter-*

national conference on Ubiquitous computing, 2008, pp. 114–123.

[7] P. Blanchart, L. He, and F. Le Gland, "Information fusion for indoor localization," in *12th International Conference on Information Fusion, 2009. FUSION '09.*, July 2009, pp. 2083–2090.

[8] J. Pinchin, C. Hide, and T. Moore, "A particle filter approach to indoor navigation using a foot mounted inertial navigation system and heuristic heading information," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, 2012, pp. 1–10.

[9] B. Turgut and R. P. Martin, "Restarting particle filters: an approach to improve the performance of dynamic indoor localization," in *IEEE Global Telecommunications Conference, 2009. GLOBECOM 2009.*, November 2009, pp. 1–7.

[10] P. Davidson, J. Collin, and J. Takala, "Application of particle filters for indoor positioning using floor plans," in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010*, October 2010, pp. 1–4.

[11] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[12] W. Chen, R. Chen, Y. Chen, H. Kuusniemi, and J. Wang, "An effective pedestrian dead reckoning algorithm using a unified heading error model," in *2010 IEEE/ION Position Location and Navigation Symposium (PLANS)*, May 2010, pp. 340–347.

[13] E. R. Bachmann, X. Yun, and A. Brumfield, "Limitations of attitude estimation algorithms for inertial/magnetic sensor modules," *IEEE Robotics Automation Magazine*, vol. 14, no. 3, pp. 76–87, 2007.

[14] C. Laoudias, R. Piché, and C. G. Panayiotou, "Device signal strength self-calibration using histograms," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, 2012, pp. 1–8.

[15] M. B. Kjærgaard, "Indoor location fingerprinting with heterogeneous clients," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 31–43, 2011.

[16] H. Nurminen, J. Talvitie, S. Ali-Löytty, P. Müller, E.-S. Lohan, R. Piché, and M. Renfors, "Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, November 2012, pp. 1–9.

[17] P. Fearnhead, "Sequential Monte Carlo methods in filter theory," Ph.D. dissertation, Merton College, University of Oxford, 1998.

[18] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter, Particle Filters for Tracking Applications*. Boston, London: Artech House, 2004.

[19] M. Raitoharju, H. Nurminen, and R. Piché, "A linear state model for PDR+WLAN positioning," in *Proceedings of of the Conference on Design & Architectures for Signal and Image Processing (DASIP 2013)*, in press.

[20] R. Piché, S. Särkkä, and J. Hartikainen, "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate Student-t distribution," in *2012 IEEE International Workshop on Machine Learning for Signal Processing*, September 2012.

[21] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, July 2010.

[22] F. Alsehly, T. Arslan, and Z. Sevak, "Indoor positioning with floor determination in multi story buildings," in *2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN2011)*, 2011, pp. 1–7.

[23] H.-H. Liu and Y.-N. Yang, "Wifi-Based Indoor Positioning for Multi-Floor Environment," in *TENCON 2011*, 2011, pp. 597–601.

[24] S. Gansemer, S. Hakobyan, S. Püschel, and U. Großmann, "3D WLAN indoor positioning in multi-storey buildings," in *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2009. IDAACS 2009.*, 2009, pp. 669–672.

[25] H. Wang, H. Lenz, A. Szabo, U. D. Hanebeck, and J. Bamberger, "Fusion of barometric sensors, WLAN signals and building information for 3-D indoor/campus localization," in *Proceedings of International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, 2006, pp. 426–432.

[26] C. Ascher, C. Kessler, R. Weis, and G. Trommer, "Multi-floor map matching in indoor environments for mobile platforms," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, 2012, pp. 1–8.

[27] E. U. Condon and H. Odishaw, Eds., *Handbook of Physics*. York, Pennsylvania: McGraw-Hill Book Company, 1958.

[28] A. Doucet and A. M. Johansen, "A Tutorial on Particle Filtering and Smoothing: 15 years later," in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovskiĭ, Eds. Oxford University Press, 2011, ch. 24, pp. 656–704.

[29] P. Fearnhead, D. Wyncoll, and J. Tawn, "A sequential smoothing algorithm with linear computational cost," *Biometrika*, vol. 97, no. 2, pp. 447–464, 2010.

[30] S. Kotz and S. Nadarajah, *Multivariate t Distributions and Their Applications*. Cambridge University Press, 2001.

[31] Y. Bar-Shalom, R. X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation, Theory Algorithms and Software*. John Wiley & Sons, 2001.

PUBLICATION 2

Henri Nurminen, Mike Koivisto, Simo Ali-Löytty, and Robert Piché: Motion model for positioning with graph-based indoor map. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, October 2014.
DOI: 10.1109/IPIN.2014.7275539

Motion Model for Positioning with Graph-Based Indoor Map

Henri NURMINEN, Mike KOIVISTO, Simo ALI-LÖYTTY, and Robert PICHÉ
Tampere University of Technology, Tampere, Finland
Emails: {henri.nurminen, mike.koivisto, simo.ali-loytty, robert.piche}@tut.fi

***Abstract*—This article presents a training-free probabilistic pedestrian motion model that uses indoor map information represented as a set of links that are connected by nodes. This kind of structure can be modelled as a graph. In the proposed model, as a position estimate reaches a link end, the choice probabilities of the next link are proportional to the total link lengths (TLL), the total lengths of the subgraphs accessible by choosing the considered link alternative. The TLLs can be computed off-line using only the graph, and they can be updated if training data are available. A particle filter in which all the particles move on the links following the TLL-based motion model is formulated. The TLL-based motion model has advantageous theoretical properties compared to the conventional models. Furthermore, the real-data WLAN positioning tests show that the positioning accuracy of the algorithm is similar or in many cases better than that of the conventional algorithms. The TLL-based model is found to be advantageous especially if position measurements are used infrequently, with 10-second or more time intervals.**

***Keywords*—indoor positioning; particle filter; motion model; map-matching; graph**

I. Introduction

This paper is concerned with wireless local area network (WLAN) positioning combined with a graph-based floor plan representation. In the graph-based floor plan, the set of expected user paths is condensed into links (edges). The links are undirected line segments that are connected by nodes (vertices) according to their real-world connectivity. It is assumed that the user can be anywhere on the links.

A conventional scheme would be to combine WLAN positioning, pedestrian dead reckoning (PDR) based on e.g. inertial navigation system (INS), and floor plan information in some form [1–6]. However, many consumer-grade mobile devices that one may want to use as indoor positioning devices are equipped with only low-quality INS or no INS at all, which makes reliable PDR challenging or impossible. Absence of PDR emphasises the need for a realistic statistical model for the users motion, since the motion model combines the measurements of different time instants and can, to some extent, be used to maintain location awareness also between WLAN scans. Floor plans can be used to make the motion model more precise and realistic, but their usage typically results in complex and highly nonlinear motion and/or measurement models.

This research was funded by TUT Graduate School, Finnish Doctoral Programme in Computational Sciences (FICS), and HERE, a Nokia business.

The most straightforward and flexible solution to nonlinear estimation problems is particle filters, which are based on Bayesian statistical theory and Monte Carlo (MC) simulation. Particle filters provide a set of weighted MC samples from the posterior distribution of the hidden state at each time instant. These samples are called particles. A practical limitation of particle filters is that they require a precise enough proposal distribution for propagating the particles. If the dispersion of the particle cloud is high compared to the measurement noise variance, the filter requires a large number particles for accurate modelling of the posterior distribution.

A traditional approach in indoor positioning is to use the motion model as a proposal distribution, and then apply wall constraints as a measurement that reduces the weight of wall-colliding particles. Evennou et al. show, however, that if the motion model is only a random-walk model, the required number of particles is huge [7]. Evennou et al. propose using the allowed user paths instead of the prohibited ones, i.e. propagating the particles on the links of the graph. This reduces the degrees of freedom from the particles' movement, which makes the state space easier to be modelled with small numbers of particles.

Furthermore, a graph-based motion model is potentially more realistic than random-walk-based models. Typical pedestrian movement is in a more determined way directed towards some destination than random-walk based models predict. A well-tuned graph-based motion model assumes more continuity for the user's direction than random-walk models, still allowing sharp turns at corridor junctions. The ideas of "destination" and map-based motion modelling have also been proposed for the wall constraint based map representation by Khider et al., but their method requires storing large tables of pre-computed probabilities [8].

The link transition rule plays an important part in making the graph-based model realistic. This paper proposes a novel model for the link transition probabilities, i.e. a rule for distributing the user position's probability from one link to the others. In the proposed model, the probability to choose a link is proportional to the size of the subgraph that is accessible through each link option. This number is called the total link length (TLL). The more potential destinations a direction allows, the more probably this direction is chosen. Extraction of the proposed model's parameters can be done off-line, and it does not require any training data, but the probabilities can be updated if suitable data are available. This

model can straightforwardly be incorporated in a particle filter, which estimates both position and velocity of the user.

Real-data tests are done in a campus building using the existing WLAN infrastructure. The compared methods are particle filters with three different link transition probability rules, the particle filter with wall-collision model, and the Kalman filter. It is found that 400 particles enables reliable positioning with the proposed TLL-based algorithm. Furthermore, the proposed method is found to outperform the comparison methods especially if WLAN scan rate is low.

In section II the proposed link transition model is explained in detail and compared with the methods in the literature. Section III presents the proposed positioning algorithm, and Section IV presents the real-data tests. Section V presents the conclusions.

Notations: $\mathrm{N}(\mathbf{m}, \mathrm{P})$ refers to the (multivariate) normal distribution with mean $\mathbf{m}$ and covariance matrix P, and $\mathrm{N}(\mathbf{x}|\mathbf{m}, \mathrm{P})$ refers to its probability density function (pdf) evaluated at $\mathbf{x}$. $\mathrm{cat}(p_1, \ldots, p_n)$ is the categorical distribution.

II. Graph-based indoor map

A. Related work

The choice of the indoor map's representation depends on the availability of the map and on the requirements of the positioning method. In outdoor vehicle positioning, the node–link model is a natural choice as map representation because the roads usually strictly restrict the movement of the vehicle and such maps are commonly available [9]. In indoor environments, many maps are in the wall representation format. However, e.g. the topological skeleton of the wall representation, sometimes also called the Voronoi diagram, has been proposed to be used as the graph representation [7], and the skeleton of a monochrome digital image can be created automatically [10]. In this paper it is assumed that a graph is available and it is a reliable representation of the building's floor plan.

Indoors, the graph representation is also a crude approximation of the wall representation because in reality a pedestrian can have arbitrary heading and motion patterns. The density of the graph sets a limit for the positioning accuracy especially in large open areas. However, if the measurement accuracy is low compared to the sizes of the geometrical patterns in the building plan, the graph representation removes unnecessary degrees of freedom; in a corridor, a one-dimensional subspace in the middle of the corridor is an adequate approximation when no better accuracy can be expected from the measurement system. Link density should thus be in accordance with the accuracy of the used measurements.

Link transition is the event of a position estimate reaching the end of one link and moving to another. The link transition is modelled with transition probabilities that are assigned to each link connected to the node. In many cases this probability distribution is not uniform. For instance, the user is in general more likely to continue walking on a corridor than to turn to a small room.

It is possible to define some *a priori* rules for the link transition probabilities even without inertial measurements, training data, or knowledge of the functions of different parts of the building. However, only a little discussion of this topic appears in indoor positioning literature. Liao et al. [11] use a uniform link transition prior distribution, but also propose an algorithm for updating the probabilities using a database of measurement sequences. With uniform link transition probabilities, the probability of reaching a certain destination given a starting point is inversely proportional to the number of link branches that could have been chosen during the transition. For a general case, this is counter-intuitive.

In the approach of Yu et al. [12] the distribution is uniform except that links that result in changing the user's room are given lower weights. This method gives somewhat more weight to smooth motion patterns, but for example when a corridor ends to a room, the user might be more likely to enter the room than to turn back. This method cannot model walking in corridors, either. The articles [13, 14] mention the possibility of having non-uniform link transition distribution, but do not propose any automatic rule for setting the probabilities. Maybe a more realistic automatic approach for setting the link transition probabilities has been proposed by Evennou et al. in [7]. In their model the link transition probabilities are based on the continuity of heading: the probability of moving from link λ_i to link λ_j is defined to be

$$\mathbb{P}(\lambda_j|\lambda_i) \propto 1 + \cos(\Delta\theta_{i,j}), \tag{1}$$

where $\Delta\theta_{i,j}$ is angle between links λ_i and λ_j. In an end node of the graph, the user is assumed to make a U-turn. Although this approach models walking in a corridor, for example, in a more realistic way than the uniform distribution model, it still gives lower probabilities to graph areas behind many junctions even if they have similar heading difference histories. Furthermore, this method will fail if there is a sharp turn in the corridor and walking straight takes you into a small room which is visited relatively rarely.

Graph-based indoor maps have been used in assisting indoor positioning also with inertial navigation systems (INS), for example in [6, 15, 16]. In these articles, probability is first propagated according to the INS output, and then projected to the most probable link. Link transition probability is defined by various criteria related to spatial and heading differences. In principle, these articles use the uniform link transition prior probabilities, but the INS measurements give such a strong update to the probabilities that the prior becomes negligible in most cases.

B. Link transition probabilities based on total link lengths

In this subsection, it is assumed that there is a graph-based indoor map. A node ν_m's basic attributes are three-dimensional position $\textsc{pos}(\nu_m)$ that also contains the altitude. A link λ_i has two end nodes $\textsc{endn}(\lambda_i, 1)$ and $\textsc{endn}(\lambda_i, 2)$, whose numbering is arbitrary, and link length $\textsc{length}(\lambda_i)$, which is the Euclidean distance of the end nodes' x and y

coordinates:

$$\text{LENGTH}(\lambda_i) = \left\| \text{POS}(\text{ENDN}(\lambda_i, 2)) - \text{POS}(\text{ENDN}(\lambda_i, 1)) \right\|.$$

Assume now that the user is in a node ν_m of the map, coming from the link λ_k. Let us denote the options for the new link with $\lambda_{k_1}, \ldots, \lambda_{k_n}$; the old link λ_k is also an option. The motion model determines probabilities for continuing in the direction of each of the links attached to the current node. Without any prior knowledge, the user's destination of travel can be modelled to follow a uniform distribution over the accessible parts of the building. Assuming that each link corresponds to certain constant width, which is an idealisation, this distribution corresponds to a discrete uniform distribution over all the links. Thus, the probabilities $\mathbb{P}(\lambda_{k_i}|\lambda_k)$ are determined by the following principles:

1) All possible link points are equally probable destinations
2) The user moves from the current location to the destination using the shortest possible path

Furthermore, the following simplifications are made to allow efficient practical algorithms:

3) The destination is within some along-graph distance ℓ_{MAX} of the current position
4) Link history is forgotten except for the latest link λ_k

The assumption 4 means that the link history is not used to update the distribution of the destination; it is uniformly distributed also in the areas that could have been reached with a shorter path from an earlier point of the user trajectory.

Principle 1 implies that the probability of the destination being in an arbitrary link λ_i is

$$\mathbb{P}(\text{IS\_DESTINATION}(\lambda_i)) = \frac{[i \in \mathcal{I}_k] \cdot \text{LENGTH}(\lambda_i)}{\sum_{j \in \mathcal{I}_k} \text{LENGTH}(\lambda_j)}, \quad (2)$$

where $[\cdot]$ is the Iverson bracket and $\mathcal{I}_k$ is the set of possible destination link indices. Since the user uses the shortest path and the latest link is in the memory, the user cannot choose to go back to the latest link. Thus,

$$\mathcal{I}_k = \{j | \text{shortest path from } \nu_m \text{ to } \lambda_j \text{ is shorter than } \ell_{\text{MAX}} \text{ and does not use } \lambda_k\}.$$

Furthermore, the principles 2 and 4 imply

$$\mathbb{P}(\lambda_{k_i}|\lambda_k) = \sum_{j \in \mathcal{J}_{k,i}} \mathbb{P}(\text{IS\_DESTINATION}(\lambda_j)), \quad (3)$$

where

$$\mathcal{J}_{k,i} = \{j | \text{shortest path from } \nu_m \text{ to } \lambda_j \text{ uses } \lambda_{k_i}\}.$$

Combining (2) and (3) gives

$$\begin{aligned} &\mathbb{P}(\lambda_{k_i}|\lambda_k) \\ &= \sum_{j \in \mathcal{I}_k \cap \mathcal{J}_{k,i}} \text{LENGTH}(\lambda_j) / \sum_{j \in \mathcal{I}_k} \text{LENGTH}(\lambda_j) \\ &= \frac{[k_i \neq k] \cdot \text{TLL}(\lambda_{k_i}, 1 + [\text{ENDN}(\lambda_{k_i}, 2) = \nu_m])}{\sum_{j=1}^{n} [k_j \neq k] \cdot \text{TLL}(\lambda_{k_j}, 1 + [\text{ENDN}(\lambda_{k_j}, 2) = \nu_m])}, \end{aligned} \quad (4)$$

where $\text{TLL}(\lambda_i, d)$ is the total link length (TLL) behind the link λ_i starting from the end node $\text{ENDN}(\lambda_i, d)$:

$$\text{TLL}(\lambda_i, d) = \sum_{j \in \mathcal{K}_{i,d}} \text{LENGTH}(\lambda_j), \quad (5)$$

where

$$\mathcal{K}_{i,d} = \{j | \text{shortest path from } \text{ENDN}(\lambda_i, d) \text{ to } \lambda_j \text{ is shorter than } \ell_{\text{MAX}} \text{ and uses } \lambda_i\}.$$

Using this definition, the TLLs can be computed efficiently with the off-line algorithm listed in Algorithm 1. The function GRAPHDIST returns the shortest along-link distances of any two nodes of the distance-weighted graph. They can be computed using Dijkstra's algorithm [17, Ch. 4.8]. A link has two TLLs, one for both end nodes.

A special case in this model is the dead end nodes. Because there is in fact no direction choice possibility, the probability of moving beyond a dead end node is reflected backwards in the same link. This U-turn approach is also used in [7].

Note that this scheme does not model static periods or destination shifts. For those, there have to be separate models, especially for modelling stopping and U-turns.

The principle 2 enables fast and straightforward computation of the link transition probabilities. In principle, detours and loops could be taken into account in TLLs with smaller weights, but this is not considered in this paper. Because the history information does not affect the link transition probabilities, detours and loops do in fact have a non-zero probability. The assumption 3 is made to limit the domination of the major passageways so that less probable options can also be modelled by Monte Carlo approximations. In the tests, $\ell_{\text{MAX}} = 40\,\text{m}$ was used. The history forgetting principle 4 is used to allow off-line computation of the probabilities and recursive positioning algorithms.

The presented model generalises straightforwardly to interfloor links, i.e. vertical links connecting different floors at staircases or elevators etc.. In TLL modelling, a vertical link should be given a length higher than the true length to avoid staircases absorbing too much probability.

One possible drawback of the presented model is the requirement that the level of detail is uniform everywhere in the graph-based floor plan. If an area has exceptionally high link density without denser activity in the area, this area gets unjustifiably high probabilities. For instance, if in two similar office rooms one has a link to each desk in the room but the other has only one link leading to the room, the TLL model is not viable with this graph-based floor plan. This is a challenge especially in large open areas. Furthermore, the model may sometimes generate so low or high transition probabilities that their modelling with Monte Carlo algorithms is inefficient. Therefore, some maximum and minimum values should be used for the used transition probabilities.

C. Parameter learning

Liao et al. present an EM (Expectation–Maximisation) algorithm for determining the link transition probabilities using

Algorithm 1 Total link length computation

input: links $\lambda_{1:n_\lambda}$, destination distance ℓ_{MAX}
output: matrix of total link lengths $\mathrm{L} \in \mathbb{R}^{n_\lambda \times 2}$ for each link for both directions

1) Set node index $k := 1$.
2) Compute the vector of the shortest node distances $\delta^{\text{min}} := \textsc{graphdist}(\lambda_{1:n_\lambda}, k)$.
3) For each link index $I \in \{i | \lambda_i \text{ connected to } \nu_k\}$, choose direction $d = 1 + [\textsc{endn}(\lambda_I, 2) = \nu_k]$ and compute $\mathrm{L}_{I,d} := \textsc{tll\_recursive}(I, d, \lambda_{1:n_\lambda}, \delta^{\text{min}}, \ell_{\text{MAX}}, 0)$.
4) If all nodes handled, stop. Otherwise, set $k := k + 1$ and go to step 2.

function $\ell = \textsc{tll\_recursive}(I, d, \lambda_{1:n_\lambda}, \delta^{\text{min}}, \ell_{\text{MAX}}, \ell_{\text{used}})$
$\ell := \textsc{length}(\lambda_I)$;
$d' := \mod(d, 2) + 1$;
if $\ell_{\text{used}} + \ell > \delta^{\text{min}}_{\textsc{endn}(\lambda_I, d')}$ OR $\ell_{\text{used}} + \ell > \ell_{\text{MAX}}$ **then**
  $\ell := \min\left\{\frac{1}{2}(\ell + \delta^{\text{min}}_{\textsc{endn}(\lambda_I, d')} - \ell_{\text{used}}), \ell_{\text{MAX}} - \ell_{\text{used}}\right\}$;
  return;
end if
$\ell_{\text{used}} := \ell_{\text{used}} + \ell$;
for all $J \in \{j | j \neq I, \textsc{endn}(\lambda_j, 1) = \textsc{endn}(\lambda_I, d')\}$ **do**
  $\ell := \ell + \textsc{tll\_recursive}(J, 1, \lambda_{1:n_\lambda}, \delta^{\text{min}}, \ell_{\text{MAX}}, \ell_{\text{used}})$;
end for
for all $J \in \{j | j \neq I, \textsc{endn}(\lambda_j, 2) = \textsc{endn}(\lambda_I, d')\}$ **do**
  $\ell := \ell + \textsc{tll\_recursive}(J, 2, \lambda_{1:n_\lambda}, \delta^{\text{min}}, \ell_{\text{MAX}}, \ell_{\text{used}})$;
end for
end function

training data [11]. Their method requires iteration of fixed-interval particle smoothing varying the value of the highly multidimensional parameter vector, which is not practical in large-scale systems. It is also not possible to learn and store specific parameters for each user.

In this article it is assumed that the training data contain the information on which direction the users chose after being close to a certain node. Let the link options from the node be $\lambda_1, \lambda_2, \ldots, \lambda_{n_\lambda}$. The number of users that chose the link λ_i is now denoted with y_i, the link transition probability random variable with p_i, and the normalised TLL with π_i. Assuming that the choices of different users are independent, the vector $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{n_\lambda} \end{bmatrix}^{\mathrm{T}}$ follows the multinomial distribution

$$p(\mathbf{y}|\mathbf{p}) = \mathrm{lhd}(\mathbf{p}) \propto \prod_{i=1}^{n_\lambda} p_i^{y_i}. \tag{6}$$

The TLL probabilities can be interpreted as a Dirichlet prior

$$p(\mathbf{p}) = \mathrm{Dirichlet}(\mathbf{p}|\boldsymbol{\pi}, a) \propto \prod_{i=1}^{n_\lambda} p_i^{a \cdot \pi_i - 1}, \tag{7}$$

where $a \in \mathbb{R}^+$ is a parameter that determines how many observations the prior corresponds to. This gives the Dirichlet posterior

$$p(\mathbf{p}|\mathbf{y}) = \mathrm{Dirichlet}\left(\mathbf{p}|\left[\begin{matrix} \frac{a\pi_1 + y_1}{a+n} & \cdots & \frac{a\pi_{n_\lambda} + y_{n_\lambda}}{a+n} \end{matrix}\right], a + n\right), \tag{8}$$

where $n = \sum_{i=1}^{n_\lambda} y_i$ is the total number of observations. Thus, the update formula for the link transition probabilities is

$$p_i' = \frac{a \cdot p_i + y_i}{a + n}. \tag{9}$$

Further updates can be made using the same formula as more data become available. The U-turn probability can be set to zero in the positioning phase.

III. Positioning algorithm

A. Speed model

Indoor positioning Kalman filters conventionally assume a Gaussian random-walk model for either velocity or position. Either of these is problematic: Velocity is seldom random-walk due to relatively frequent sharp turns and halts. Position is not likely to be random-walk either, since people tend to have transient tendencies in their movement and a position-random-walk does not model these tendencies. However, the graph-based motion models have more potential to take both abrupt changes in direction and tendencies in velocity into account: sharp turns are given probability only at corridor junctions and bends, and when the direction is known, the user's speed can be modelled as random-walk, except for halts.

The motion model described here is similar to the motion model presented by Liao et al. [11], except for the link transition probabilities. A two-mode motion model is used, the modes m_k being random-walk speed combined with the link transition probabilities ('motion'), and static ('static'). The state vector $\mathbf{x}_k$ at time instant t_k is four-dimensional, containing the link index I_k, position on the link $p_k \in [0, 1]$, direction $d_k \in \{1, 2\}$, and speed v_k. Additionally, the state contains the mode m_k. A state vector thus contains all the information needed to calculate the position in Cartesian coordinates as well as the floor. The model for position propagation expressed in probabilistic notation is then

$$p(s_k, v_k | v_{k-1}, m_k) = \begin{cases} \mathrm{N}\left(\begin{bmatrix} s_k \\ v_k \end{bmatrix} | \begin{bmatrix} (\Delta t)_k v_{k-1} \\ v_{k-1} \end{bmatrix}, \mathrm{Q}_k\right) & \text{, if } m_k = \text{'motion'} \\ \textsc{Dirac}(s_k) \cdot \textsc{Dirac}(v_k) & \text{, if } m_k = \text{'static'} \end{cases}, \tag{10}$$

where s_k is the translated distance within time interval $[t_{k-1}, t_k]$, Dirac is Dirac delta function, $(\Delta t)_k$ is the length of the discretisation, interval, and

$$\mathrm{Q}_k = \sigma_v^2 \begin{bmatrix} \frac{1}{3}(\Delta t)_k^3 & \frac{1}{2}(\Delta t)_k^2 \\ \frac{1}{2}(\Delta t)_k^2 & (\Delta t)_k \end{bmatrix} \tag{11}$$

is the process noise covariance matrix. The normal distribution should be truncated so that only feasible pedestrian speeds are possible. The distance s_k is not in the state because it only depends on its previous value through the state variable v_k. Together with the link transition probabilities, the distribution of s_k describes how probability propagates on the links. The evolution model of motion mode m_k has two parameters that describe the mode transition probabilities.

B. WLAN positioning

WLAN is the only measurement source used in this paper, in addition to the map. Since the motion model is based on the map, the map cannot give feedback information on the state's distribution, unlike WLAN. WLAN is also the only information channel that can produce a static snapshot estimate of the position; the information contained by the indoor map is mainly useful in a filtering (time-series) context.

WLAN positioning is based on training data ("fingerprints") that are collected from each floor of the building beforehand. It is assumed that the fingerprints contain accurate position. Furthermore, it is assumed that the received signal strength indicator (RSSI) reported by the device can be mapped to the actual received signal strength (RSS) in dBm units so that the measurements of each data collection device are comparable. A method for this is presented in [18], for example.

The WLAN positioning method used in this article's tests is based only on RSS measurements because their usage does not require any hardware modifications and they can be measured by the receiving mobile terminal alone. To average out noise and to keep the number of stored parameters small, the standard logarithmic path loss model is used to model the dependency of the RSS from user position:

$$P = A - 10n \log_{10}(\|\mathbf{r} - \mathbf{r}_{\mathrm{BS}}\|) + w_{\mathrm{PL}}, \tag{12}$$

where P is the measured RSS level, $\mathbf{r}$ is the user's position, $\mathbf{r}_{\mathrm{BS}}$ is the access point's position, and $w_{\mathrm{PL}} \sim \mathrm{N}\left(0, \sigma^2_{\mathrm{PL}}\right)$ is a normally distributed shadowing noise component. The model parameters A and n, and the access point positions are estimated by the Gauss–Newton method as in [19].

The positioning algorithm is another Gauss–Newton method described in [19]. These methods estimate and take into account variances of the path loss parameters and access point positions. Article [19] shows that this approach gives more realistic variance information for the position measurement. Consistent uncertainty estimation is important when combining measurements from different sources: if the used variances are too small, the system might rely too much on possibly erroneous measurements, and if too large, the information is not used efficiently [20].

The true likelihood of one WLAN measurement is not Gaussian, but it is approximated with a Gaussian in the Gauss–Newton method as described in [19]. Thus, the used WLAN measurement model is

$$p(\mathbf{y}_k | I_k, p_k, v_k, m_k) \propto \mathrm{N}\left(\mathbf{r}_k(I_k, p_k) | \mu_k(\mathbf{y}_k), \Sigma_k(\mathbf{y}_k)\right). \tag{13}$$

As Liao et al. point out in [11], the exact likelihood for a link point is the integral of the two-dimensional likelihood over the space that is mapped to the considered link point. However, the implementation of this idea might be computationally infeasible, and it would require using the wall representation of the floor plan also. This is to be avoided at least for mobile applications involving map data transfer via wireless networks.

C. Particle filter

A particle filter is a Monte Carlo algorithm that approximates the posterior distributions $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ provided that certain Markov assumptions hold and the probability distributions $p(\mathbf{x}_0)$, $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and $p(\mathbf{y}_k|\mathbf{x}_k)$ are known and their density values are computable for each time index k. The components of the vector $\mathbf{x}$ are the state variables, and the vector $\mathbf{y}$ contains the measurements. No assumptions of linearity or Gaussianity have to be made.

A particle filter approximates the presented model's posterior distributions with a set of weighted particles $\{(\mathbf{x}^i_k, W^i_k) | i \in \{1, \ldots, N\}\}$ that are random realisations of the statistical model [21]. The random samples $\mathbf{x}^i_k$ are generated using the pre-specified proposal distributions, also known as the importance distributions, and the weights W^i_k, also known as importance weights, make the sample set approximate the true posterior in the sense that the moments of the distribution are approximated by the weighted sum over the particle set:

$$\mathrm{E}(\mathbf{g}(\mathbf{x}_k)|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N} W^i_k \cdot \mathbf{g}(\mathbf{x}^i_k) \tag{14}$$

where $\mathbf{g}$ is an almost arbitrary Borel measurable function.

In the initial phase the particle values are generated from the initial prior $p(\mathbf{x}_0)$ with equal weights. A filter update has two stages: the prediction stage, in which the particles of the current time instant are generated from the proposal distributions $q(\mathbf{x}_k|\mathbf{x}^i_{1:k-1}, \mathbf{y}_{1:k})$, and the update stage, in which the measurements are used to update the weights of the particles. Given the previous time instant's weights W^i_{k-1}, the current time instant's unnormalized weights $\tilde{W}^i_k$ are obtained using the formula

$$\tilde{W}^i_k = \frac{p(\mathbf{y}_k|\mathbf{x}^i_k) p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})}{q(\mathbf{x}^i_k|\mathbf{x}^i_{1:k-1}, \mathbf{y}_{1:k})} \cdot W^i_{k-1}. \tag{15}$$

The normalization factors of the pdf's $p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})$ and $q(\mathbf{x}^i_k|\mathbf{x}^i_{1:k-1}, \mathbf{y}_{1:k})$ are not required, since the normalisation constants are approximated rigorously by the ordinary normalisation $W^i_k = \tilde{W}^i_k / \sum_{j=1}^{N} \tilde{W}^j_k$ [22, Ch. 7.2].

The choice of the proposal distribution is a crucial ingredient of any particle filter. A rule of thumb is that the proposal distribution should be as close to the final posterior as possible. In this paper the state transition distribution $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is used as a proposal distribution.

A third stage is required by any particle filter to avoid all the weight concentrating to one particle: resampling. At the resampling stage the particles' weights are equalised by sampling with replacement N new particles from the old particle set with the old particle weights as probabilities. One drawback of this resampling method is the danger of sample impoverishment, which means that most of the particles are resampled to one or few locations. This can be avoided by resampling only occasionally, e.g. only when the effective number of particles, $N_{\mathrm{eff},k} = 1/\sum_{i=1}^{N} (W^i_k)^2$, goes below a threshold. Since resampling increases the Monte Carlo

variance of the estimate, the estimate is reported before the resampling. [21, Ch. 3.3]

Particle filters have the drawback that they are unable to detect increase of probability in areas where there are no particles. The proposal distribution should be such that this is improbable. However, getting stuck in a local maximum is possible, and this risk is exacerbated by the floor plan model, which may make points that are close in Euclidean sense distant. Map errors such as missing links may have similar effect, since the graph-based particle filter cannot model map errors, unlike the wall-collision particle filter presented in [5]. To recover from filter divergences, the divergence monitoring and re-initialisation procedures proposed in [5] are used. After the re-initialised particles have been generated from the Gaussian distribution, they are moved to the closest link points.

With the model of this article, the prediction stage updates the particle values with the graph-based motion model and randomly generated noise modifying the weights of wall-crossing particles, and the update stage corrects the particle weights based on the WLAN measurement. In case a WLAN scan is not made at some time instant, the update stage is simply omitted. The algorithm description is in Algorithm 2.

D. Point estimator

It would be convenient for the end user that the reported point estimate of the position would be in a feasible pedestrian area. The links of the graph-based floor plan provide a definition for the feasible area because they should contain the most likely locations. However, the weighted mean of the particles, which minimises the weighted sum of squares, can even be outside the building if the distribution is multimodal or L-shaped, for example.

The Constrained Mean algorithm proposed in [23] can be used to enforce the reported position to be in one of the links. The used estimator is the minimiser of the weighted sum of squares in the set of the link positions:

$$\hat{\mu}_k = \arg\min_{\boldsymbol{\xi} \in C} \sum_{i=1}^{N} W_k^i \left\| \mathbf{r}_k^i - \boldsymbol{\xi} \right\|^2, \tag{16}$$

where C is the set of all link positions. A straightforward derivation shows that the weighted sum of squares function is a strictly increasing function of the distance to the ordinary weighted mean. Thus, the Constrained Mean estimator for the graph-based model equals choosing the link point that is closest to the weighted mean. The detailed description of the used point estimator is in Algorithm 3.

Algorithm 2 Particle filter for 2D indoor positioning with graph-based indoor map

1) For each $i \in \{1, \ldots, N\}$ set $W_0^i := \frac{1}{N}$, and generate $\mathbf{x}_0^i \leftarrow p(\mathbf{x}_0)$. Set the time index $k := 1$.
2) Generate motion states m_k^i based on the previous motion states m_{k-1}^i and the transition probabilities. For the particles $i \in \{i | m_{k-1}^i = \text{'static' and } m_k^i = \text{'motion'}\}$ generate $v_{k-1}^i \leftarrow p(v_o)$ and $d_{k-1}^i \leftarrow \mathrm{cat}(0.5, 0.5)$. For each $i \in \{i | m_k^i = \text{'motion'}\}$ generate
$$\begin{bmatrix} s_k^i \\ v_k^i \end{bmatrix} \leftarrow \mathrm{N}\left(\begin{bmatrix} (\Delta t)_k v_{k-1}^i \\ v_{k-1}^i \end{bmatrix}, \mathrm{Q}_{k-1} \right),$$
set $v_k^i := \min(\max(v_k^i, v_{\min}), v_{\max})$ and set $d_k^i := d_{k-1}^i$. The matrix Q_{k-1} is the one in Eq. (11). For each $i \in \{i | m_k^i = \text{'static'}\}$ set $s_k^i := 0$ and $v_k^i := 0$.
3) For each $i \in \{1, \ldots, N\}$ move the particle to the direction d_k^i at most to the end of the link $\lambda_{I_{k-1}^i}$ to distance s' no more than s_k^i. Set $\tilde{s} := s_k^i - s'$ and $I_k^i = I_{k-1}^i$, and compute
   while $\tilde{s} > 0$ **do**
      Generate the new link index I_k^i from the categorical distribution $\mathrm{cat}(\mathbb{P}(\lambda_{k_1} | \lambda_{I_k^i}), \ldots, \mathbb{P}(\lambda_{k_{n_\lambda}} | \lambda_{I_k^i}))$
      Update the direction d_k^i
      Move particle i to the direction d_k^i at most to the end of the current link to distance s' no more than $\tilde{s}$
      Set $\tilde{s} := \tilde{s} - s'$
   end while
   Compute the particle's position $\mathbf{r}_k^i$ in the Cartesian coord..
4) Set $\tilde{W}_k^i := W_{k-1}^i$ for each $i \in \{1, \ldots, N\}$. Perform divergence monitoring. If re-initialised, go to phase 6.
5) If no WLAN measurement is obtained at time index k, go to Phase 6. Otherwise, run the Gauss–Newton algorithm to obtain the WLAN estimate's mean $\boldsymbol{\mu}_k$ and covariance matrix Σ_k . Set
$$\tilde{W}_k^i := \mathrm{N}\left(\mathbf{r}_k^i | \boldsymbol{\mu}_k, \Sigma_k\right) \cdot \tilde{W}_k^i,$$
and normalize by $W_k^i := \tilde{W}_k^i / \sum_{j=1}^N \tilde{W}_k^j$.
6) Compute the reported estimate $\hat{\boldsymbol{\mu}}_k$ using Algorithm 3.
7) If $1/\sum_{i=1}^N (W_k^i)^2 < N_{\text{eff,lim}}$, perform resampling and equalise the weights.

IV. Testing

A. Test setup

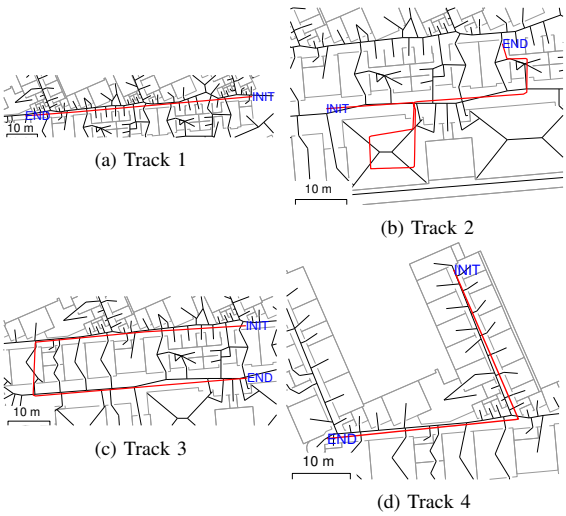
The experimental tests are carried out in the building Tietotalo of Tampere University of Technology. Only the existing communication WLAN infrastructure is used. A Nexus 7 tablet with Android 4.4.2 OS is used to log the WLAN measurements. All training and positioning data are collected with the same device. The reference locations are set manually by the user by tapping a floor plan figure on the tablet's screen. Positioning algorithms are computed with MATLAB. The filters are updated and errors computed with 0.5 s intervals.

The test tracks are shown in Fig. 1. Track 1 in Fig. 1a is a straight corridor that has offices and classrooms around. There are also some rooms that have access to another long corridor, as well as some branching corridors. Track 2 in Fig. 1b contains two visits to smaller rooms around the corridor. Track 3 in Fig. 1c contains two long corridors and a room that has access to both corridors. Track 4 in Fig. 1d uses only main corridors but makes a 90-degree turn. Additionally, a longer test track that combines all the shorter tracks is collected for evaluating the required number of particles.

Algorithm 3 Point estimate computation
(Link-constrained weighted least squares)

Input: particle positions $\mathbf{r}^i$ with weights W^i, $i \in \{1, \ldots, N\}$
Output: point estimate $\hat{\boldsymbol{\mu}}$.

1) Compute the weighted mean $\hat{\boldsymbol{\mu}}' := \sum_{i=1}^{N} W^i \cdot \mathbf{r}^i$.
2) For each interesting link j (e.g. all links of the building part), find the link point $\mathbf{z}_j$ that is closest to $\hat{\boldsymbol{\mu}}'$. If the orthogonal projection of $\hat{\boldsymbol{\mu}}'$ to the line corresponding to the link is between the end points of the link, choose this point. Otherwise, choose the closest end point.
3) Set $s_j := \|\mathbf{z}_j - \hat{\boldsymbol{\mu}}'\|$ for each interesting link j.
4) Set $j^* := \arg\min_j s_j$ and $\hat{\boldsymbol{\mu}} := \mathbf{z}_{j^*}$.



(a) Track 1 (b) Track 2 (c) Track 3 (d) Track 4

Figure 1. The test tracks. Dim colour represents the walls, black colour the links, and red colour the ground truth. Initial points and end points are labelled.

The compared methods are three graph-based particle filters, the wall-collision particle filter (WCPF), and the Kalman filter (KF). The particle filters use the uniform link transition model, link angle difference based model (Evennou et al.), and the proposed total link length (TLL) model for determining the link transition probabilities. The WCPF and the KF use the random-walk position model. In the WCPF the particles that collide with walls are given zero weight, and the quality monitoring as well as re-initialisation are done similarly as in the other particle filters.

B. Results and discussion

To evaluate Monte Carlo errors and to find a suitable number of particles, the filter is run repeatedly for the long test track. The box plot of mean errors of 100 replications are presented in Fig. 2. The different levels of the boxes represent 5 %, 25 %, 50 %, 75 %, and 95 % empirical quantiles. The figure shows that the proposed filter performs reasonably well already with small numbers of particles, such as 20 or 50 with a 5-second measurement interval. After 400 particles the performance

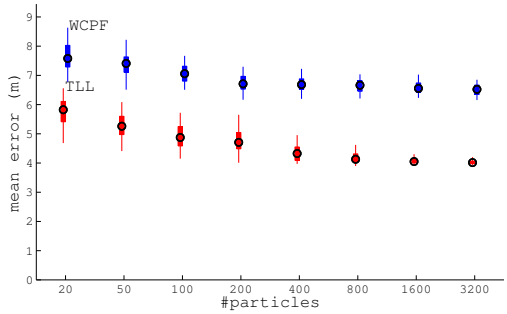


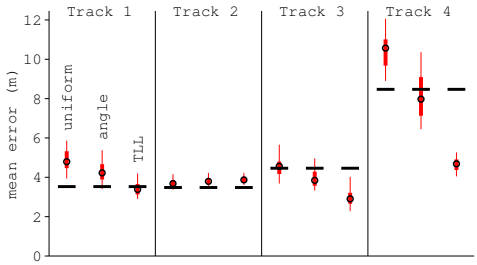
Figure 2. Mean error distribution as a function of particle number for the graph-based total link length particle filter (TLL) and for the wall collision particle filter (WCPF). WLAN scan interval is 5 s.

does not improve significantly. The WCPF also achieves stable performance with 400 particles, but it is clearly worse than that of the TLL-based method.

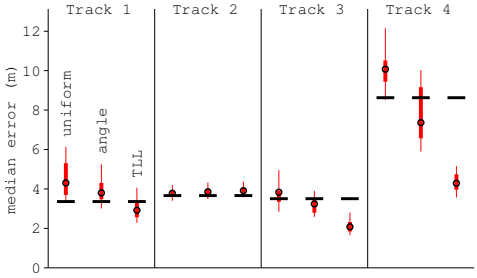
In the actual positioning test, each particle filter is run 100 times with 400 particles for each of the four test tracks. Figure 3 compares the obtained error distributions. The black horizontal lines show the KF's errors. The different subfigures represent different error statistics: mean errors, median errors, 95 % quantiles of errors, and room detection rates.

In track 1, the straight corridor, the proposed TLL method performs slightly better in positioning errors than the other particle filters. This can be explained by the fact that in the TLL method most particles typically continue on the corridor links, which have high TLLs. Thus, the TLL-based filter infers the motion pattern that happens to be correct for this case. The other particle filters, especially the uniform distribution method, distribute large proportions of particles to branching links, and thus their motion models are closer to that of the KF. In fact, their performances are slightly worse than the KF's, which might be explained by the Monte Carlo errors, i.e. low particle densities or poor coverage in interesting state space areas. Track 2 is not expected to favour the TLL method, since it contains two visits to low-TLL branches and lacks stationary motion patterns. However, in the test results the TLL method is still only slightly worse than the others for this track. In track 3, the TLL method outperforms especially the KF. This might reflect the fact that the motion pattern inferred by the TLL method, the speed estimate, is still usable after the 90-degree turns. The KF, in contrast, cannot infer any velocity information. The performance of the TLL method is superior also in track 4. This shows that the TLL methods performs well in corridors even if the corridor is branched or has turns.

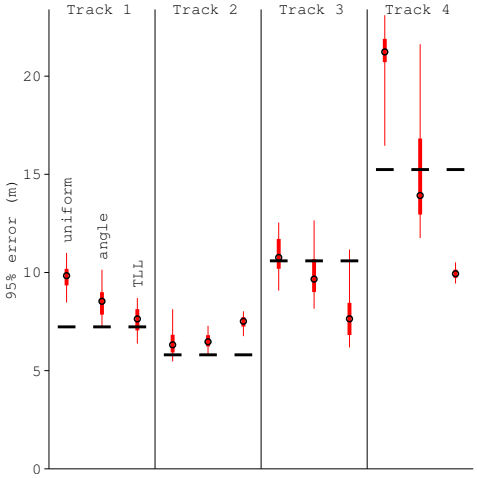
In some applications, it is enough to know the room of the user correctly. Therefore, the box plot in Fig. 3d shows the percentages of how often the position estimates are inside the correct room. The corridor system is counted as one room. According to the figure, the graph-based particle filters perform significantly better than the KF. The difference is clearer in room detection rate than in positioning accuracy. Furthermore, the TLL-based filter has slightly better room detection rates than the comparison methods.



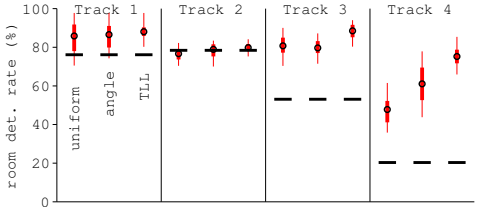
(a) Mean errors



(b) Median errors

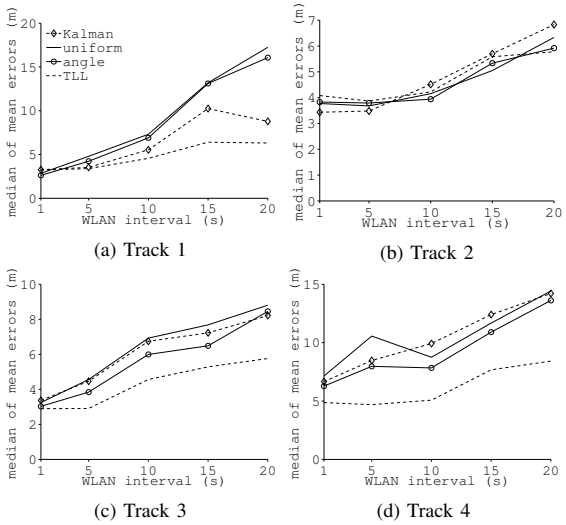


(c) 95 % quantiles of errors



(d) Room detection rates

Figure 3. Error statistics and room detection rates of the four test tracks as a function of the link transition probability rule. The black horizontal line segments indicate the Kalman filter results. The WLAN scan interval is 5 s.



(a) Track 1 (b) Track 2

(c) Track 3 (d) Track 4

Figure 4. Errors as a function of the WLAN scan interval with different link transition probability rules and the Kalman filter

Fig. 4 shows the medians of the mean errors obtained in 100 Monte Carlo replications as a function of the WLAN measurement time interval. An interesting result is that the TLL-based particle filter outperforms the other methods more and more clearly when WLAN interval is increased. The only exception is track 2, where the unpredictable turns to small rooms make the performance deteriorate with large WLAN intervals regardless of the algorithm.

V. Conclusions

A novel statistical training-free motion model for indoor positioning with graph-based indoor map was presented. The proposed model gives more probability to the directions that allow access to larger or more branched areas in the graph. The probability parameters can be updated if training data are available. A particle filter using the proposed model and WLAN-based position measurements was also presented and compared with the conventional methods. Moreover, an optimal method for enforcing the particle cloud based position estimate to be located on a link of the graph was presented.

Compared to the other models, the proposed model is intuitive and is the only one allowing most of the position's probability mass to proceed along a corridor instead of dispersing it to side corridors and rooms. With the other models, the probabilities that continue on the corridor are so low that the particle filters were not able to distribute particles in the true position's area. Still, the method is also able to handle occasional visits in small rooms. With the proposed model, the longer side corridors also get significant weight, as is natural. The proposed model assigns more particles to the graph's most branched areas whose modelling with small numbers of particles is most difficult.

The presented real-data tests showed that the link transition probability rule has significant influence on the filter performance and the particle filter relying on the proposed model

outperforms the comparison methods in office environments. The difference in positioning accuracy is significant especially if most of the test track is in straight or intersecting corridors or if the WLAN measurement time interval is large. Room detection rate, which may be important for the user experience, is also the highest for the proposed method.

In future, methods for ensuring the floor plan graph's uniform level of detail, which is an assumption in the proposed model, will be studied. Moreover, suitability of the graph models to different environments, such as shopping centres, will be tested. Particle smoother using the proposed model will be an interesting topic since the incorporation of future measurements is also based on realistic state modelling. Furthermore, using the graph structure for routing and navigation and incorporating the destination information in the link transition model can be studied. Testing the models with large authentic measurement data sets would be important for making final conclusions of the statistical behaviour of each method and the presented parameter learning algorithm.

Acknowledgment

We thank the HERE team and our colleagues (especially Matti Raitoharju) for their valuable comments.

References

[1] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *10th international conference on Ubiquitous computing (UbiComp'08)*, September 2008, pp. 114–123.

[2] S. Beauregard, Widyawan, and M. Klepal, "Indoor PDR performance enhancement using minimal map information and particle filters," in *2008 IEEE/ION Position, Location and Navigation Symposium (PLANS 2008)*, May 2008, pp. 141–147.

[3] P. Blanchart, L. He, and F. Le Gland, "Information fusion for indoor localization," in *12th International Conference on Information Fusion, 2009. (FUSION'09)*, July 2009, pp. 2083–2090.

[4] H. Leppäkoski, J. Collin, and J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals," *Journal of Signal Processing Systems*, vol. 71, no. 3, pp. 287–296, June 2013.

[5] H. Nurminen, A. Ristimäki, S. Ali-Löytty, and R. Piché, "Particle filter and smoother for indoor localization," in *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN2013)*, October 2013, pp. 137–146.

[6] M. I. Khan and J. Syrjärinne, "Investigating effective methods for integration of building's map with low cost inertial sensors and wifi-based positioning," in *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN2013)*, October 2013, pp. 884–891.

[7] F. Evennou, M. François, and E. Novakov, "Map-aided indoor mobile positioning system using particle filter," in *2005 IEEE Wireless Communications and Networking Conference (WCNC 2005)*, vol. 4, March 2005, pp. 2490–2494.

[8] M. Khider, S. Kaiser, and P. Robertson, "A novel three dimensional movement model for pedestrian navigation," *Journal of Navigation*, vol. 65, pp. 245–264, April 2012.

[9] P. Davidson, J. Collin, and J. Takala, "Application of particle filters to a map-matching algorithm," *Gyroscopy and Navigation*, vol. 2, no. 4, pp. 285–292, October 2011.

[10] L. Lam, S.-W. Lee, and C. Y. Suen, "Thinning methodologies–A comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869–885, September 1992.

[11] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz, "Voronoi tracking: location estimation using sparse and noisy sensor data," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS 2003)*, October 2003, pp. 723–728.

[12] J. Yu, W.-S. Ku, M.-T. Sun, and H. Lu, "An RFID and particle filter-based indoor spatial query evaluation system," in *16th International Conference on Extending Database Technology (EDBT'13)*, March 2013, pp. 263–274.

[13] B. Ferris, D. Hähnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *2006 Robotics: Sciences and Systems Conference (RSS 2006)*, August 2006.

[14] I. Kim, E. Choi, and H. Oh, "Indoor user tracking with particle filter," in *The Fourth International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE 2012)*, July 2012, pp. 59–62.

[15] P.-Y. Gilliéron, I. Spassov, and B. Merminod, "Indoor navigation enhanced by map-matching," *European Journal of Navigation*, vol. 3, no. 3, 2005.

[16] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, "Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning," in *2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2014)*, September 2014, pp. 147–158.

[17] S. S. Skiena, *The Algorithm Design Manual*. New York, NY, USA: Springer-Verlag, 1998.

[18] C. Laoudias, R. Piché, and C. G. Panayiotou, "Device signal strength self-calibration using histograms," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, November 2012, pp. 272–279.

[19] H. Nurminen, J. Talvitie, S. Ali-Löytty, P. Müller, E.-S. Lohan, R. Piché, and M. Renfors, "Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, November 2012, pp. 461–469.

[20] P. Ivanov, S. Ali-Löytty, and R. Piché, "On consistency of the estimation," in *International Conference on Localization and GNSS (ICL-GNSS)*, June 2014.

[21] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter, Particle Filters for Tracking Applications*. Boston, London: Artech House, 2004.

[22] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, UK: Cambridge University Press, 2013.

[23] R. Piché and M. Koivisto, "A method to enforce map constraints in a particle filter's position estimate," in *11th IEEE Workshop on Positioning, Navigation and Communication (WPNC'14)*, March 2014.

PUBLICATION 3

Mike Koivisto, Henri Nurminen, Simo Ali-Löytty, and Robert Piché: Graph-based map matching for indoor positioning. In *10th International Conference on Information, Communications and Signal Processing (ICICS)*, December 2015.
DOI: 10.1109/ICICS.2015.7459983

Graph-Based Map Matching for Indoor Positioning

Mike Koivisto, Henri Nurminen, Simo Ali-Löytty, and Robert Piché
Tampere University of Technology
Tampere, Finland
Emails: {mike.koivisto, henri.nurminen, simo.ali-loytty, robert.piche}@tut.fi

***Abstract*—This article presents a probabilistic motion model that is based on an economical graph-based indoor map representation, such that the motion of the user is constrained according to the floor plan of a building. The floor plan is modeled as a combination of links and open space polygons that are connected by nodes. In the authors' earlier work the link transition probabilities in this graph are proportional to the total link lengths that are the total lengths of the subgraphs accessible by choosing the considered link option, and this article extends this model to include open space polygons as well. A particle filter using the extended motion model in which all particles are constrained according to the map structure is presented. Furthermore, wireless local area network and Bluetooth Low Energy positioning tests show that the proposed algorithm outperforms comparison methods especially if the measurement rate is low.**

***Keywords*—indoor positioning; particle filter; motion model; map matching; graph**

I. Introduction

Wireless local area network (WLAN) and Bluetooth Low Energy (BLE) systems are commonly used for indoor positioning, and their accuracy can be improved by filtering measurements over time using statistical models of the user's motion. When inertial navigation is not available, the user's position or velocity is typically modeled as a random walk. However, these motion models neglect the motion constraints imposed by the walls in the building. Using the floor plan information in the motion model enables more efficient particle filtering than the conventional particle filter [1] that uses the random-walk motion model and treats wall constraints as measurements [2].

The floor plan information can be included in the motion model using a graph-based floor plan representation [3], [4], [5], [6], [2]. In the graph-based floor plan, the expected user paths are represented by links (edges). The links are undirected line segments that are connected by nodes (vertices) according to their real-world connectivity. It is assumed that the user can be anywhere on the links. A graph-based motion model is potentially more realistic than random-walk-based models because typical pedestrian movement is oriented towards a destination in a more determined way than random-walk models predict. A well-tuned graph-based motion model assumes more continuity for the user's direction than random-walk models, while allowing sharp turns at corridor junctions [2].

Graphs usually represent well corridors and small rooms. However, large open spaces, wide corridors, and outdoor spaces are in a more economical and natural way represented by two-dimensional polygons. Inside these polygons moving is free in two dimensions, so the positioning accuracy is not limited by the map representation. Thus, the map used in this article contains two types of map objects (MO): links and open space polygons (OS). Such a combined map representation has been proposed by Ferris et al. [5], but they do not give an automatic method for assigning the MO transition rules.

This paper extends the authors' earlier work [2], where a particle filter using a graph-based motion model for WLAN positioning is proposed. The novelty of this paper is extending the link transition rule proposed in [2] to the combined map representation that includes the OSs. In the proposed algorithm, the MO transition probability is proportional to the total link length (TLL) that describes the total size of the area accessible by choosing the considered MO option.

This paper also presents real-data tests in open spaces and at indoor–outdoor transitions. The tests are done in a campus building using the existing WLAN infrastructure and a BLE network built for positioning research. The proposed method is found to outperform the comparison methods especially if measurement rate is low.

II. Graph with open areas

A. Description and definition

In this paper, the map structure similar to [5] is expressed as a combination of MOs and nodes such that $G = (\Lambda, \mathrm{N})$. The set Λ contains the arbitrarily indexed MOs denoted by λ_k, which can be either links that represent small rooms and corridors, or OSs that represent larger open spaces. The other set N contains the arbitrarily indexed nodes ν_n that connect MOs and have a three-dimensional position.

Large OSs are defined as polygons inside which it is possible to place a circle with a 4-meter radius so that the circle does not cross any walls of a floor plan map. An OS λ_k contains information about the boundaries of the polygon and also accessor nodes (AN). These ANs are the only points where the user can enter the OS polygon λ_k from the graph or vice versa.

The rest of the map structure is represented as links such that each link λ_k has two end nodes. Different floors of a building are connected with vertical links in places where floor transitions are possible, such as in elevators and stair cases. The map structure for one floor is shown in Fig. 1.

B. Map object transition rule

Transitions between MOs can be modeled using different types of transition probabilities [3], [4]. To extend the TLL-based link transition rule of [2], corresponding MO lengths need to be defined for OSs.

Fig. 1: Graph with OSs in the test building. Links and nodes as well as ANs are shown with black. OSs are modeled as polygons which are shown with light blue. The outdoor area is also considered as an OS.

The user can move to another destination through an OS, which affects the computation of the TLLs. Therefore, temporary links $\lambda_{k,1}, \ldots, \lambda_{k,n}$ are created between the ANs of an OS λ_k to represent the shortest possible distances between different ANs. Such temporary links can be created in an off-line phase using Dijkstra's algorithm [7, Ch. 4.8], for instance, and they need not be stored in the permanent map data structure; the temporary links are used only to determine the TLLs in the off-line phase.

For adapting to the definition of TLL in [2], the parts of the OSs that are not covered by the temporary links are also given a size measure that is here referred to as MO length. The MO length should depend on the area of the OS, and the lengths of the temporary links which are parts of the shortest paths along the map structure ares subtracted from the OSs' MO length. The MO length of an OS λ_k is thus defined as

$$\text{LENGTH}(\lambda_k) = \frac{\text{AREA}(\lambda_k)}{\text{CORRIDOR\_WIDTH}} - \sum_{i \in \mathcal{A}} \text{LENGTH}(\lambda_{k,i}), \tag{1}$$

where $\text{AREA}(\lambda_k)$ is the area of the polygon, CORRIDOR_WIDTH is a configuration parameter that describes a typical corridor width, for example 3 m, and

$\mathcal{A} = \{j \mid$ shortest path from initial node ν_m to or through λ_k is shorter than ℓ_{MAX} and contains $\lambda_{k,j}\}$.

The MO transition probabilities $\mathbb{P}(\lambda_{k_i}|\lambda_k)$ are determined by extending the principles in the authors' earlier work [2] with an assumption that considers temporary links also as parts of the shortest paths. This assumption together with the principles in [2] imply that the user uses the shortest possible path to reach the destination. Thus, the old MO is not one of the possible destinations and U-turns are only allowed by an additional motion model as explained in section III-A. Using these assumptions the MO transition probability of arriving to node ν_m from link λ_k can be expressed by extending the transition rule as follows

$$\mathbb{P}(\lambda_{k_i}|\lambda_k) = \frac{\sum_{j \in \mathcal{I}_{k,i}} \text{LENGTH}(\lambda_j) + \sum_{j \in \mathcal{K}_{k,i}} \text{LENGTH}(\lambda_{p,j})}{\sum_{j \in \mathcal{I}_k} \text{LENGTH}(\lambda_j) + \sum_{j \in \mathcal{K}_k} \text{LENGTH}(\lambda_{p,j})}, \tag{2}$$

where

$\mathcal{I}_k = \{j \mid$ shortest path from ν_m to λ_j is shorter than ℓ_{MAX} and does not use $\lambda_k\}$,

$\mathcal{I}_{k,i} = \{j \mid$ shortest path from ν_m to λ_j is shorter than ℓ_{MAX} and uses λ_{k_i} but does not use $\lambda_k\}$,

$\mathcal{K}_k = \{j \mid$ shortest path from ν_m to temporary link $\lambda_{p,j}$ is shorter than ℓ_{MAX} and does not use $\lambda_k\}$,

$\mathcal{K}_{k,i} = \{j \mid$ shortest path from ν_m to temporary link $\lambda_{p,j}$ is shorter than ℓ_{MAX} and uses λ_{k_i} but does not use $\lambda_k\}$.

The transition probabilities defined in (2) can be computed efficiently by using Algorithm 1 of [2] and considering temporary links as ordinary links and open spaces as destinations with lengths determined in Eq. (1).

At MO transitions, this model gives most weight to corridors and major open areas, and a high TLL can be considered as an indication of a link being such a major pathway [2]. The assumptions limit the probability of the main routes with the variable ℓ_{MAX} such that options with shorter lengths get large enough probabilities. Although the variable ℓ_{MAX} ensures that less probable options get small probabilities, some lower limit for the probabilities can be also set. In the real-data tests, the limiting variable ℓ_{MAX} is set to 40 m and the lower limit for the transition probabilities is set to 5 %.

The proposed TLL algorithm does not use any information about the functions of different building parts; the TLLs are computed off-line using only the map structure G as input. However, if real data of people's behavior are available, the MO transition probabilities can be updated using the TLLs as a prior. This learning process is similar to the one described in [2].

III. Positioning algorithm

A. Motion model

Instead of being a random-walk, the motion of a user usually contains sharp turns and it tends to be oriented towards some destination. The graph-based motion models take these sharp turns into account at junction points as well as tendencies in velocity when the direction of the user is known.

In this paper, a two-mode motion model is used to model the constrained motion of the user as in [3], [5] and [2]. The mode m_k is either a combination of the MO transition probabilities and random-walk speed ($m_k =$ 'motion') or is static ($m_k =$ 'static') in both OSs and links.

The state vector $\mathbf{x}_k$ at time instant t_k depends on the MOs such that

$$\mathbf{x}_k = \begin{cases} [I_k, p_k, d_k, v_k, m_k]^T & \mathbf{x}_k \text{ is on a link} \\ [I_k, x_k, y_k, d_k, v_k, m_k]^T & \mathbf{x}_k \text{ is in an OS,} \end{cases} \tag{3}$$

where I_k is the MO index, $p_k \in [0, 1]$ is the one-dimensional location on the link and $[x_k, y_k]^T$ is the two-dimensional location in the OS. Furthermore, $d_k \in \{-1, 0, 1\}$ indicates the direction of the user on the link, $d_k \in [0, 2\pi]$ denotes the heading in the OS, and v_k the speed (magnitude of velocity).

The motion model of the user contains four parts: motion in links and in OSs, and transitions between links and OSs. The motion model used in links is presented in [2]. Because

the user can go through OSs, straight motion is preferred in the motion model inside OSs as in [5]. Thus, using the probabilistic notation and assuming that the heading of the user is random-walk in OSs, the motion model in an OS is

$$p(s_k, v_k, d_k | v_{k-1}, d_{k-1}, m_k) = \begin{cases} \mathrm{N}\left(\begin{bmatrix} s_k \\ v_k \\ d_k \end{bmatrix} \mid \begin{bmatrix} (\Delta t)_k v_{k-1} \\ v_{k-1} \\ d_{k-1} \end{bmatrix}, \mathrm{P}_k \right), & \text{if } m_k = \text{'motion'} \\ \textsc{Dirac}(s_k, v_k, d_k), & \text{if } m_k = \text{'static'} \end{cases}, \tag{4}$$

where s_k is the distance travelled by the user within the time interval $[t_{k-1}, t_k]$ and DIRAC denotes the multi-dimensional Dirac delta function. Furthermore, $(\Delta t)_k = t_k - t_{k-1}$ is the length of the discretisation interval and the covariance matrix of the process noise is now

$$\mathrm{P}_k = \begin{bmatrix} \mathrm{Q}_k & 0 \\ 0 & \sigma_d^2 \end{bmatrix} \tag{5}$$

where the matrix Q_k is similar to one in [2] and σ_d is another configuration parameter.

The transition from a link to an OS occurs when the user is in an accessor node coming from a graph. If the user chooses to enter the OS, the heading of the user is the heading of the latest link with some uncertainty. The transition from an OS to a graph occurs if the user is inside the OS and crosses a boundary of the OS close to an accessor node. Then the probability mass is set to the accessor node and the motion continues along the graph.

B. WLAN positioning

In this paper, most of the tests are based on WLAN positioning. Fingerprints that are collected beforehand from each floor of the test building are used in the experimental tests presented in section IV. The same measurement model based on the standard logarithmic path loss model with Gaussian shadowing noise is used as in [2]. A Gauss–Newton method is used to obtain a Gaussian approximation of the actual likelihood taking the parameter uncertainties into account as in [8]. Thus, the measurement model for the particle weighting is

$$p(\mathbf{y}_k | \mathbf{x}_k) \propto \mathrm{N}(\mathbf{r}_k(\mathbf{x}_k) | \boldsymbol{\mu}_k(\mathbf{y}_k), \Sigma_k(\mathbf{y}_k)), \tag{6}$$

where $\mathbf{y}_k$ is the vector of received signal strengths (RSS), and $\mathbf{r}_k$ is the position of the user in Cartesian coordinates. $\boldsymbol{\mu}_k$ and Σ_k are the mean and covariance matrix of the likelihood approximation, respectively.

In this paper, the current floor of the user is estimated by choosing the most probable floor in the building based on the measurement likelihood.

C. Bluetooth Low Energy positioning

The proposed method with the current map structure was also tested using measurements from StickNFind BLE beacons (Fig. 2). The beacons were placed inside the building such that the average number of heard beacons in one fingerprint location around the test track was 11, which is more than the threshold proposed in [9]. Since received measurements are RSS-values as in WLAN positioning, the BLE measurement model is similar to the measurement model (6).

Fig. 2: A BLE beacon

D. Particle filter

A particle filter is a Monte Carlo algorithm that approximates the posterior distribution of the state given the measurement history, when the measurement and the state transition models as well as prior information of the state is known, and certain Markovian assumptions hold. Although the continuity of the distributions is usually assumed, it is not necessary because the existence of the corresponding probability measures is a sufficient condition for convergence of a particle filter [10].

A particle filter approximates the posterior distribution $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ with a set of weighted particles $\{(\mathbf{x}_k^i, w_k^i) \mid i \in \{1, \ldots, N\}\}$. Initially, particles are generated according to prior distribution $p(\mathbf{x}_0)$. In the prediction phase, the particles are generated according to the proposal distribution $q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{y}_{1:k})$, and in the update phase importance weights w_k^i are updated using the measurement likelihood $p(\mathbf{y}_k | \mathbf{x}_k^i)$. Finally, resampling ensures that the weight does not concentrate to one particle. [11]

In this paper, the state transition distribution $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ is used as a proposal distribution, and the effective sample size is used to determine when resampling is needed. The particle filter is presented in detail in Algorithm 1, where the re-initialization method presented in [1] is also used.

The algorithm for moving a particle from an OS to a link is presented in detail in Algorithm 2. When a particle is inside an OS, it might reach the walls far away from any AN. Instead of bouncing the particle back by changing the heading as in [5], the weight of the colliding particle is set to zero. When the particle exits the OS close to an AN, the particle's weight is reduced by a coefficient that depends on the distance to the closest AN.

A natural choice for the point estimate would be the weighted mean of the particles. However, the estimate might be in a region that is inaccessible via the MOs even when all the particles are in the feasible region. Choosing the best particle [5] or using the MAP (*maximum a posteriori*) estimate [12] are possible options to constrain the estimate according to the map structure, but the estimate trajectories might display jumpiness due to the multimodality of the posterior distribution.

The continuous Constrained Mean algorithm [13] enforces the estimate to the map structure by minimizing the weighted sum of squares over all possible MOs. The Constrained Mean estimator for the graph-based model is equivalent to choosing

the MO point that is closest to the weighted mean. The detailed description of this point estimator can be found in [2].

It is still possible that the solution of the Constrained Mean algorithm is inside a MO that does not contain any particles. Such estimates can be avoided by constraining the set of possible MOs at each time step to include only specific accessible areas.

IV. Tests

A. Test setup

The tests are carried out in Tietotalo building of Tampere University of Technology campus using Samsung Galaxy S5 phone. Three tests are done using only WLAN measurements from existing WLAN infrastructure and one test is done using only BLE measurements from a set of installed beacons. Actual positioning algorithms are computed offline using MATLAB software.

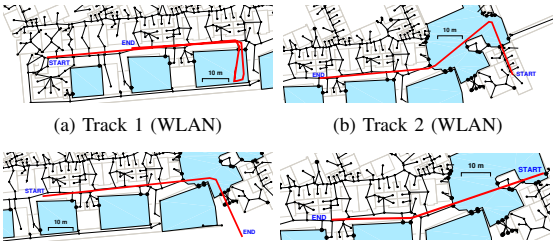
The test tracks are shown in Fig. 3. Track 1 (Fig. 3a) starts from a corridor, enters a lecture hall and stays inside for a while before leaving the lecture hall back to the corridor. Track 2 (Fig. 3b) starts from a narrow corridor and goes through an irregularly shaped open area ending up to the main corridor. Third WLAN track 3 (Fig. 3c) illustrates a movement from indoors to outdoors. Track 4 (Fig. 3d) is a BLE track that starts from an open area and goes to the main corridor by crossing an open area on its way.

Results from the experimental tests are compared with the Kalman filter (KF) and the similar particle filter with uniform MO transition probabilities. Since it is more convenient to model open areas such as outdoors with polygons instead of links, the proposed method is not compared with the authors' earlier method [2]. The KF uses a random-walk motion model.

B. Results and discussion

In the experimental tests, 400 particles are used for both particle filters. Each filter is run 100 times for all four test tracks and the measurement scanning intervals are set to 5 and 10 seconds. RMS-error (root mean square error) statistics from the tests are shown in Fig. 4.

The proposed TLL-based particle filter performs slightly better than the KF in many cases and better than the particle filter with uniform transition probabilities in every test case. In track 1, which goes through a lecture hall, the TLL-based method is giving less probability to the lecture hall according to the TLLs. However, the proposed method performs well with both scan rates because less probable options have enough



(a) Track 1 (WLAN) (b) Track 2 (WLAN)

(c) Track 3 (WLAN) (d) Track 4 (Bluetooth)

Fig. 3: The real-data test tracks with walls, links, nodes, and the ground truth of test tracks. Starting and ending points are labelled, and accessor nodes are shown as larger black points.

Algorithm 1 Particle filter for 2D indoor positioning

1) For each $i \in \{1, \ldots, N\}$ set $w_0^i := \frac{1}{N}$, and generate $\mathbf{x}_0^i \leftarrow p(\mathbf{x}_0)$. Set the time index $k := 1$.
2) Generate motion states m_k^i based on the previous motion states m_{k-1}^i and the transition probabilities. For the particles $i \in \{i | m_{k-1}^i = \text{'static'} \text{ and } m_k^i = \text{'motion'}\}$ generate $v_{k-1}^i \leftarrow p(v_o)$ and $d_{k-1}^i \leftarrow \text{cat}(0.5, 0.5)$. For each $i \in \{i | m_k^i = \text{'motion'}\}$ generate
$$\begin{bmatrix} s_k^i \\ v_k^i \end{bmatrix} \leftarrow \text{N}\left(\begin{bmatrix} (\Delta t)_k v_{k-1}^i \\ v_{k-1}^i \end{bmatrix}, \text{Q}_{k-1} \right),$$
set $v_k^i := \min(\max(v_k^i, v_{\min}), v_{\max})$ and set $d_k^i := d_{k-1}^i$. The matrix Q_{k-1} is the one in [2]. For each $i \in \{i | m_k^i = \text{'static'}\}$ set $s_k^i := 0$ and $v_k^i := 0$.
3) **for** $i \in \{1, \ldots, N\}$ **do**
   Set $\tilde{w}_k^i := w_{k-1}^i$ and $\tilde{s} := s_k^i$.
   if $\lambda_{I_{k-1}^i}$ is a link **then**
      Move the ith particle to direction d_k^i at most to the end of $\lambda_{I_{k-1}^i}$ to distance $s' \leq s_k^i$.
      Set $\tilde{s} := \tilde{s} - s'$.
   end if
   Set $I_k^i := I_{k-1}^i$.
   while $\tilde{s} > 0$ **do**
      Generate new MO index I_k^i from categorical distribution $\text{cat}(\mathbb{P}(\lambda_{k_1} | \lambda_{I_k^i}), \ldots, \mathbb{P}(\lambda_{k_{n_\lambda}} | \lambda_{I_k^i}))$
      if $\lambda_{I_k^i}$ is a link **then**
         Update the direction d_k^i.
         Move particle i to the direction d_k^i at most to the end of $\lambda_{I_k^i}$ to distance $s' \leq \tilde{s}$.
      else
         Generate direction d_k^i using Eq. (4) and move particle i to distance $s' \leq \tilde{s}$ not farther than the OS boundary.
         Run the algorithm `OS_to_link`.
      end if
      Set $\tilde{s} := \tilde{s} - s'$
   end while
   Find the particle's position $\mathbf{r}_k^i$ in Cartesian coord.
end for
4) If a measurement is obtained at time index k, run Gauss–Newton to obtain mean $\boldsymbol{\mu}_k$ and covariance Σ_k [8]. Set
$$\tilde{w}_k^i := \text{N}\left(\mathbf{r}_k^i | \boldsymbol{\mu}_k, \Sigma_k\right) \cdot \tilde{w}_k^i.$$
5) Normalize the weights by $w_k^i := \tilde{w}_k^i / \sum_{j=1}^N \tilde{w}_k^j$.
6) Perform divergence monitoring as in [2].
7) Compute the estimate $\hat{\boldsymbol{\mu}}_k$ using the Constr. Mean [13].
8) If $1/\sum_{i=1}^N (w_k^i)^2 < N_{\text{eff,lim}}$, perform resampling and equalize the weights. Set $k := k + 1$, and go to phase 2.

Algorithm 2 OS_to_link

if The particle is on an OS boundary **then**
  $\text{dist}_{\text{AN}} :=$ distance to the closest AN
  if $0 \leq \text{dist}_{\text{AN}} \leq 0.5 \cdot \text{width}_{\text{door}}$ **then**
    Move the particle to the AN.
  else if $0.5 \cdot \text{width}_{\text{door}} < \text{dist}_{\text{AN}} \leq 5\,\text{m}$ **then**
    Move the particle to the AN and update the weight: $w^i_{k-1} := 2^{-4(\text{dist}_{\text{AN}} - 0.5 \cdot \text{width}_{\text{door}})^2} \cdot w^i_{k-1}$.
  else
    Set $s' := \tilde{s}$ and $w^i_{k-1} := 0$.
  end if
end if

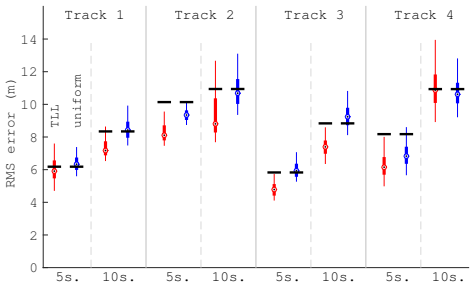


Fig. 4: Error statistics from the real-data tests for each method and track when the measurement scanning rates are 5 and 10 seconds. Black line segments are the KF results

probability to attract particles. The KF is almost as good as the TLL-method with the scanning interval of 5 seconds, but when the scans are made less frequently, the TLL-based method outperforms both comparison methods clearly. Despite the fact that track 2 has one sharp turn in the large OS, the proposed particle filter performs well because the variance of the random-walk direction is large enough. It also outperforms the comparison methods significantly.

The KF as well as the particle filter with uniform transition probabilities are less accurate than TLL-method also in track 3 which contains an indoor–outdoor transition. In track 4 both particle filters outperform the KF with the 5-second scanning interval. The difference between the particle filters is smaller than with the other tracks, which is probably because most of the track is in an open area.

V. Conclusions

A novel statistical motion model for indoor positioning with the map structure that contains a graph with open area regions was proposed. The proposed model constrains the motion of the user based on the map structure such that larger and more branched areas in the map have more probability at map object transitions. A particle filter algorithm using the proposed model and WLAN or BLE measurements was presented and compared with two other methods.

The user's motion typically takes place in corridors and major open spaces (OS) and is oriented towards some destination instead of being random-walk. The proposed motion model prefers motion in OSs and corridors but is also able to handle less probable areas such as small rooms. The motion model also prefers straight motion inside OSs where the motion of the particles is constrained by the OS boundaries.

The presented experimental tests indicate that the proposed motion model is advantageous especially if the measurement scanning rate is low. In addition to the good performance in the corridors, the particle filter with the proposed motion model performs well also in OSs and outdoor spaces, and outperforms the comparison methods in most of the cases.

Acknowledgments

This research was partly funded by HERE, a Nokia business. Henri Nurminen receives funding from Tampere University of Technology Graduate School, Finnish Doctoral Programme in Computational Sciences, the Foundation of Nokia Corporation, and Tekniikan edistämissäätiö.

References

[1] H. Nurminen, A. Ristimäki, S. Ali-Löytty, and R. Piché, "Particle filter and smoother for indoor localization," in *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN2013)*, October 2013, pp. 137–146.

[2] H. Nurminen, M. Koivisto, S. Ali-Löytty, and R. Piché, "Motion model for positioning with graph-based indoor map," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN2014)*, October 2014, pp. 646–655.

[3] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz, "Voronoi tracking: location estimation using sparse and noisy sensor data," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS 2003)*, October 2003, pp. 723–728.

[4] F. Evennou, M. François, and E. Novakov, "Map-aided indoor mobile positioning system using particle filter," in *2005 IEEE Wireless Communications and Networking Conference (WCNC 2005)*, vol. 4, March 2005, pp. 2490–2494.

[5] B. Ferris, D. Hähnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *2006 Robotics: Sciences and Systems Conference (RSS 2006)*, August 2006.

[6] M. I. Khan and J. Syrjärinne, "Investigating effective methods for integration of building's map with low cost inertial sensors and wifi-based positioning," in *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN2013)*, October 2013, pp. 884–891.

[7] S. S. Skiena, *The Algorithm Design Manual*. New York, NY, USA: Springer-Verlag, 1998.

[8] H. Nurminen, J. Talvitie, S. Ali-Löytty, P. Müller, E.-S. Lohan, R. Piché, and M. Renfors, "Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, November 2012, pp. 461–469.

[9] R. Faragher and R. Harle, "Location Fingerprinting with Bluetooth Low Energy Beacons," in *IEEE Journal on Selected Areas in Communications*, 2015.

[10] M. Koivisto, "Graph-based particle filter in indoor positioning," Master's thesis, Tampere University of Technology, 2015.

[11] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[12] H. Driessen and Y. Boers, "MAP estimation in particle filter tracking," in *IET Seminar on Target Tracking and Data Fusion: Algorithms and Applications*, April 2008, pp. 41–45.

[13] R. Piché and M. Koivisto, "A method to enforce map constraints in a particle filter's position estimate," in *11th IEEE Workshop on Positioning, Navigation and Communication (WPNC'14)*, March 2014.

PUBLICATION 4

Henri Nurminen, Matti Raitoharju, and Robert Piché: An efficient indoor positioning particle filter using a floor-plan based proposal distribution. In *19th International Conference on Information Fusion (FUSION)*, July 2016.

An efficient indoor positioning particle filter using a floor-plan based proposal distribution

Henri Nurminen, Matti Raitoharju, and Robert Piché
Department of Automation Science and Engineering
Tampere University of Technology
Tampere, Finland
Emails: {henri.nurminen, matti.raitoharju, robert.piche}@tut.fi

***Abstract*—We present a novel floor-plan and PDR (pedestrian dead reckoning) based proposal distribution for indoor positioning particle filtering. Including floor-plan information in the proposal distribution makes the particle filtering more efficient than using the map only in the measurement model, because the proposal distribution becomes more accurate and the measurement model less accurate. The method uses offline-computed distances from each point of a regular grid to the closest wall in each direction. Our simulations show that the novel proposal distribution combined with a floor-plan and PDR based motion model improves the positioning accuracy with small numbers of particles and noisy PDR compared to the particle filters that use the floor-plan only for particle weighting.**

I. Introduction

Wireless network based positioning is an attractive indoor positioning technology due to relatively low costs and wide coverage of wireless communication networks. However, the accuracy is limited by complicated radio environments with details too numerous to be modelled and stored into databases. Therefore, wireless network based measurements are typically complemented with other measurements such as inertial measurements, barometers, and map information, i.e. floor-plan. This article proposes a novel particle filter (PF) algorithm that uses the floor-plan information with improved computational efficiency. Our PF fuses radio positioning, inertial measurement based pedestrian dead reckoning (PDR), and floor-plans.

We formulate indoor positioning as a Bayesian filtering problem that consists of a motion model (dynamical model, state evolution model) and a measurement model. Indoor map measurements are highly non-linear and non-Gaussian, so their application with the Kalman filter (KF), a conventional computationally light and easy-to-implement estimation algorithm, is challenging, and the KF can only use part of the information. Therefore, computationally more challenging methods such as grid filters [1] and PFs [2] have attracted interest in indoor positioning community.

The PF is a Monte Carlo based time series estimation algorithm that generates weighted pseudo-random samples (aka particles) of the state's posterior probability distribution [3, Ch. 3] [4]. The algorithm is very flexible in the sense that the assumptions of the dynamic and measurement models are not restrictive; in particular, Gaussianity of errors or linearity of models is not required. As the number of particles is increased, the PF solution approaches the minimum mean-square error solution for the given statistical model. However, for some models the number of particles required for achieving reasonable accuracy is prohibitively high.

In addition to the model structure and parameters, the accuracy of PF is affected by filter design choices such as (a) the choice of the proposal distribution and (b) the choice of the resampling method [4]. This article concentrates on (a). The proposal distribution, also known as the importance distribution, is a probability distribution that is used for the generation of the new particles based on the existing particles. It is important to generate particles with high density in the relevant regions of the state space so that the particle set will be an accurate representation of the posterior probability distribution. A common choice is to use the motion model of the state as the proposal distribution; this is the bootstrap filter.

In the bootstrap filter the particles' prediction locations do not reflect the newest measurement, so if the measurement is much more precise than the prediction or conflicts with the prediction, the measurement is not taken into account properly. This results in particle degeneracy, whereby the weight concentrates to only a few particles [4]. This causes frequent resampling which introduces additional Monte Carlo error. Therefore, it is advantageous to make use of the newest measurement already in the proposal distribution [4].

In indoor positioning, the PDR distribution is typically used as both the motion model and proposal distribution, and the floor-plan is used as a measurement. That is, the particles are propagated using the PDR, and the particles that collide with a wall are given small or zero weights [2], [5]–[7]. This might lead to degeneracy especially if the PDR is low-quality, because large portions of the particles are colliding with the walls and do not contribute to the estimation.

We propose including some map information in the PF's proposal distribution. The method is based on an angular PDF (probability density function) modified from that of Kaiser et al. [8]. We distort the PDR distribution by giving more probability to the directions where the distance to the closest walls is larger. This way, fewer particles collide with the walls

H. Nurminen receives funding from Tampere University of Technology Graduate School, the Foundation of Nokia Corporation, and Tekniikan edistämissäätiö.

M. Raitoharju works in OpenKin project that is funded by the Academy of Finland.

and more particles are modelling the most probable areas.

We test the proposed method by simulating test tracks with PDR and absolute position measurements upon real indoor floor-plans. The simulations show that the proposed method improves accuracy at least when the PDR quality is low.

In this article we first explain the angular PDF and compare it with the approach of Kaiser et al. Then, we introduce the other measurements sources. Section III explains the conventional wall collision PF and explains the novel features of our filter in detail. Section IV presents the simulation results, and Section V summarises the conclusions.

Notations: $\mathrm{N}(\mathbf{m}, \mathrm{P})$ is the (multivariate) normal distribution with mean $\mathbf{m}$ and covariance matrix P, and $\mathrm{N}(\mathbf{x}|\mathbf{m}, \mathrm{P})$ is its PDF evaluated at $\mathbf{x}$.

II. Measurements

We use a standard probabilistic state-space model of a time-varying state $\mathbf{x}_k$ with associated measurements $\mathbf{y}_k$. The index k is the time index. A state-space model is defined by three probability distributions: the initial prior $p(\mathbf{x}_0)$, the motion model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and the measurement model $p(\mathbf{y}_k|\mathbf{x}_k)$. Furthermore, the standard conditional independences are assumed [9, Ch. 4.1]. In this article the state vector is

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{r}_k \\ \varphi_k \\ \ell_k \end{bmatrix}, \tag{1}$$

where $\mathbf{r}_k \in \mathbb{R}^2$ is the user position, $\varphi_k \in \mathbb{R}$ is the heading angle, and $\ell_k \in \mathbb{R}$ is the footstep length.

A. Pedestrian dead reckoning

Pedestrian dead reckoning (PDR) means measuring the displacement of the user from a fixed starting point. In this article it is assumed that the PDR gives reliable footstep detection and noisy measurements of the footstep length and change of heading in the horizontal plane.

In personal indoor positioning, PDR is based on inertial navigation systems (INS). An indoor positioning INS includes three-axis accelerometers and gyroscopes. The footsteps can be detected using low-pass filtered norm of acceleration [2]. The direction of gravity can also be inferred from the accelerometer data, which gives the horizontal plane, and the change of heading is then obtained by projecting the gyroscopes' angular velocity vector to the horizontal plane [10]. The footstep length can be assumed to be inversely proportional to the footstep duration with a known or fitted proportionality constant [11]. Another method for INS-based PDR would be double integration of acceleration with zero-velocity updates, but this is not reliable if the positioning device can be hand-held instead of foot-mounted.

The strengths of the PDR method in positioning are low infrastructure cost and high short-term accuracy. Its weaknesses are the need of external initial position information and its low long-term accuracy due to sensor drift. Sensor drift means that the heading obtained by integrating the gyroscope output tends to drift away from the true value because the gyroscopes always have some systematic error. Therefore, PDR with low-cost INS needs to be complemented with additional position information such as map and/or absolute position measurements.

A conventional indoor positioning motion model that uses the PDR and no map information is

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathrm{N}(\varphi_k|\varphi_{k-1} + \overline{\Delta}_k, q^{\overline{\Delta}}) \cdot \mathrm{N}(\ell_k|\overline{\ell}_k, q^{\overline{\ell}}) \\ \times \mathrm{N}(\mathbf{r}_k|\mathbf{r}_{k-1} + \ell_{k-1} \cdot \begin{bmatrix} \cos(\varphi_k) \\ \sin(\varphi_k) \end{bmatrix}, q^{\mathbf{r}} \cdot \mathrm{I}_{2\times 2}), \end{aligned} \tag{2}$$

where $\overline{\Delta}_k$ is the heading change measurement (positive angle for anticlockwise) and $\overline{\ell}_k$ is the footstep length measurement, and q^{Δ} and q^{ℓ} are their variances. The matrix $q^{\mathbf{r}}\mathrm{I}$ is the covariance matrix of the user position's independent process noise; this parameter mainly affects the robustness of the PF, and it should be close to zero [7].

B. Radio network positioning

Wireless radio networks such as WLAN (wireless local area network), BLE (Bluetooth Low Energy), and UWB (ultra-wideband) can provide absolute position information, i.e. they can be used as a standalone positioning technology. However, their positioning accuracy can be relatively low especially if only the communication infrastructure WLAN without positioning-specific modifications is used. Furthermore, frequent radio scanning consumes battery power. These features make radio positioning a suitable complement for PDR: the short-term accuracy can be highly improved by the PDR, while the radio positioning is capable of giving the initial position estimate, and its quality does not degrade over time. Furthermore, radio positioning can be used for monitoring the integrity of the fusion estimate [7], [12].

For simplicity, we assume that the radio positioning system gives a position estimate with a multivariate normally distributed measurement error, i.e. the measurement model is

$$\mathbf{z}_k|\mathbf{r}_k \sim \mathrm{N}(\mathbf{r}_k, \Sigma_k), \tag{3}$$

where $\mathbf{z}_k$ is the 2-dimensional user position estimate, $\mathbf{r}_k$ is the 2-dimensional user position, and Σ_k is the measurement noise covariance matrix. The position measurements can be e.g. the outcome of the coverage area positioning method [13]. It is also straightforward to inject other measurement types to a PF, such as received signal strength (RSS) of WLAN and/or BLE, time of arrival ranging of UWB, or pseudo-ranges of a satellite positioning system.

C. Map matching

In indoor positioning, map matching means using the floor-plan to exclude trajectories that cross walls or floor levels. In this article the floor-plan is a set of thin wall segments on each floor of the building. The statistical model can give a small non-zero probably for walking through a wall of the map:

$$\mathbb{P}(C_k|\mathbf{r}_k, \mathbf{r}_{k-1}) = \begin{cases} \epsilon, & C_k = \text{“a step-crossing wall in the map”} \\ 1-\epsilon, & C_k = \text{“no step-crossing wall”} \end{cases}, \tag{4}$$

where $0 \leq \epsilon \ll 1$ holds, and $\mathbb{P}$ denotes probability. A non-zero wall-permeability ϵ makes the estimation algorithm more

robust to small positioning errors and floor-plan errors; with $\epsilon = 0$ all the particles can easily get stuck in a wrong room due to an erroneous position measurement or the map showing a wall that does not exist in reality [7]. Information on furniture or other movable objects is not used due to its changeable nature. In this work the floor-plans are HERE Venue Maps.

In addition, our algorithm uses the map information of distances from each candidate position to the closest obstacle in each direction. Formulated as a probabilistic model, this approach can be called the angular PDF. We use the angular PDF in the proposal distribution so that the more open space a particle has in a direction, the more likely the particle is moved to this direction. Thus, fewer particles will collide with the walls. The angular PDF can also be included in the motion model or the measurement likelihood.

Kaiser et al. consider a PF with a similar likelihood function in [8]. However, they use an angular PDF for particle weighting, while the particle propagation uses the PDR alone. In their nomenclature PDR model is part of the measurement likelihood while the angular PDF is called motion model. We adopt the a common convention and consider the PDR as a motion model and the angular PDF as part of either motion or measurement model depending on the PDF normalisation.

Kaiser et al. argue that the angular PDF helps in balancing between open areas and more narrow spaces [8]. They demonstrate how a particle subcloud in a narrow corridor will eventually disappear in the conventional PF due to wall collisions if there is another subcloud in a more open area, e.g. outdoors. The approach of [8] indeed gives more weight to areas where the PDR track is a close match to the building layout, i.e. the PDR direction is one of few directions allowed by the map, while open areas are underweighted. This feature is justified in some cases as demonstrated in [8], but results in erroneous outcomes in other cases. For example, when the user walks straight, narrow corridors are favoured over wider ones, which we do not consider a realistic model in general. Particle weighting with a likelihood that is not used in the proposal distribution can also result in more frequent resampling.

In [8], the measure of open space in a direction is the distance to certain contour plots of the gas diffusion distribution or to the closest wall, whichever is smaller. That is, the current waypoint (a point of a grid) is used as a source for a free gas diffusion and full wall absorption model. Instead of the diffusion contour distance, we use an increasing function of the distance to the closest wall. We chose this approach mainly to reduce computational burden and to make the implementation simpler; in our approach one needs to implement the crossing point of two line segments, while the diffusion algorithm includes that and other computations in addition. The gas diffusion model can be linked to the diffusion of probability, but its use as part of the likelihood is still heuristic, and [8] does not give any justification to the gas diffusion model compared to other models. Furthermore, the diffusion approach results in unwanted phenomena such as the fact that narrow long corridors are weighted less than wide long corridors because the gas-absorbing walls are closer.

Due to high computational requirements, we compute the angular PDFs offline in a server. We discretise the area into a regular square grid with a 0.5 m spacing. From each grid point we compute the distance to the closest wall in each direction with a 5-degree discretisation interval. If the distance is more than 10 m, we set the distance to 10 m because we assume that differences beyond that do not affect the heading distribution. Because the grid is regular, the grid point coordinates need not be stored and the grid density does not affect critically the computational heaviness of particle–grid point matching. The database size is the same as in [8].

If floor-plan is not available for an area, the angular PDF becomes uniform giving the standard PF. PF can also be transformed into a computationally faster KF by computing the mean and covariance matrix of the KF state variables. Furthermore, KF can be transformed into a PF by sampling from the KF distribution when a map is again available.

III. Particle filtering solution

This section first explains the conventional wall collision PF and then presents our novel modifications.

A. Wall collision particle filter

The particle filter (PF) is an importance sampling approximation of the Bayesian filter for the state-space model [3]. Let $\mathbf{x}_k^i$ denote the state of the ith particle at the kth time instant, W_k^i its weight, and N the number of particles. Initially, the particles are equal-weighted samples from the initial prior $p(\mathbf{x}_0)$. At each time instant, they are propagated by generating new samples from the proposal distribution $q_k(\mathbf{x}_k|\mathbf{x}_{0:k-1}^i, \mathbf{y}_{1:k})$, where the conditioning is on all the previous states of the particle and all the measurements up to and including the newest measurement. The proposal distribution can be chosen freely given that one can easily sample from it and that its support covers the posterior distribution's support. However, the PF algorithm will be the more efficient in estimation accuracy the closer the proposal distribution is to the actual posterior, and the distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{y}_k)$ is the optimal proposal in the sense that it minimises the variance of the weight W_k^i given $\mathbf{x}_{0:k-1}^i$ and $\mathbf{y}_{1:k}$ [14].

The particle weights are affected by the motion model, measurement likelihood, and proposal distribution. The weight update of the wall collision PF is

$$\widetilde{W}_k^i = \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)\,\mathbb{P}(C_k|\mathbf{x}_k^i, \mathbf{x}_{k-1}^i)\,p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i, \mathbf{y}_{1:k})} \cdot W_{k-1}^i, \quad (5)$$

where $\mathbf{z}_k$ is the absolute position measurement and $\mathbf{y}_{1:k}$ includes absolute position, map information, and possible other types of measurements. The normalisation factor of the posterior is not required because the normalisation to unity

$$W_k^i = \widetilde{W}_k^i / \textstyle\sum_{j=1}^{N_\mathrm{p}} \widetilde{W}_k^j \quad (6)$$

tends to approximate it well [9, Ch. 7.2].

The resampling step ensures that weight does not eventually concentrate to one or few particles. In this article, the particles are resampled after the measurement update whenever

Algorithm 1 Computation of the wall distances

Input: grid points $\mathbf{g}_{1:N_{\text{grid points}}}$, angle discretisation $N_\alpha = 72$, $\alpha = \{0°, 5°, 10°, \ldots, 355°\} \cdot \pi/180°$, walls
Output: matrix of wall distances $\mathrm{S} \in \mathbb{R}^{N_{\text{grid points}} \times N_\alpha}$
for $m \in \{1 : N_{\text{grid points}}\}$ **do**
  for $j \in \{1 : N_\alpha\}$ **do**
    for $n \in \{\text{all walls within } 10\,\text{m radius}\}$ **do**
      $\ell :=$ line segment $(\mathbf{g}_m, \mathbf{g}_m + (10\,\text{m}) \cdot \begin{bmatrix}\cos(\alpha_j)\\ \sin(\alpha_j)\end{bmatrix})$
      if ℓ and wall n cross **then**
        $\mathbf{r}^* :=$ crossing point of ℓ and wall n
        $\mathbf{d}_n := ||\mathbf{r}^* - \mathbf{g}_m||$
      else
        $\mathbf{d}_n := 10\,\text{m}$
      end if
    end for
    $[\mathrm{S}]_{m,j} := \min_n \mathbf{d}_n$
  end for
end for

$1/\sum_{i=1}^{N_\text{p}} (W_k^i)^2 < 0.1 \cdot N_\text{p}$, which is the standard approach based on the effective particle number [3, Ch. 3.3]. In the tests, the multinomial resampling is used, where the new particles are generated with replacement from the discrete distribution defined by the previous particle states and weights [15].

As a fourth step, the PF's integrity is monitored by running a fallback KF in parallel with the PF [7], [12]. This monitor detects when the whole particle cloud gets stuck behind walls in a wrong area, and restarts the PF. This article uses the PDR-Kalman of Raitoharju et al. [12] that uses a linear motion model. The fallback KF uses the PDR and absolute position measurements, but it is independent of the floor-plan and particles to avoid getting stuck.

The conventional wall collision PF such as Algorithm 1 of [7] is a bootstrap PF, i.e. the motion model is used as the proposal distribution, so the term $p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i) \,/\, q(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i, \mathbf{y}_{1:k})$ vanishes from (5). This choice can be inefficient when the INS has high noise level and the positioning area is dense in walls, i.e. the process noise variance is large compared to the measurement noise variance.

B. Particle generation using PDR and floor-plan

This subsection explains a novel proposal distribution that includes some map information already in the particle propagation phase. We propose using an angular motion model modified from the angular PDF of [8] to particle propagation.

In the offline phase, we begin by defining a regular square grid for the building. For each grid point and for each direction with a fixed angle discretisation, we compute distances to the closest wall. These are stored in a database for the online phase. The details are given in Algorithm 1.

In the PF's particle propagation phase we choose the closest grid point m^i to each particle, which is straightforward to find because the grid is a regular square grid. It would be justified to limit the grid point search to the particle's room, but this could be computationally expensive, so we leave this for future research and assume that the wall collision weighting corrects errors caused by choosing the grid point from a wrong room. The wall distances of the chosen grid point are converted into the angular PDF using a monotonically increasing function. We use the logistic function and normalisation to unity

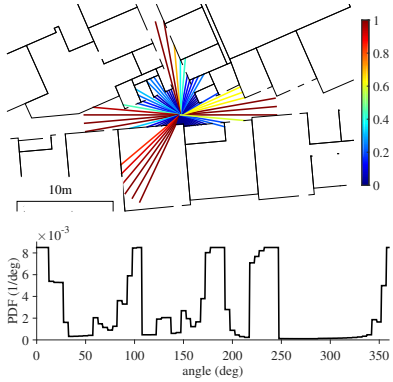


Fig. 1. The proposed angular PDF. In the upper figure the lengths of the radial line segments represent the wall distances $[S]_{m,j}$, while the colours represent the angular PDF (9) with 0.7 m footstep length.

$$\tilde{s}_{m^i}(\alpha_j) = \frac{1}{1 + 99\exp(-0.8 \cdot ([\mathrm{S}]_{m^i,j} - \ell_k^i))}, \tag{7}$$

$$s_{m^i}(\alpha_j) = \tilde{s}_{m^i}(\alpha_j) / \textstyle\sum_{j=1}^{N_\alpha} \tilde{s}_{m^i}(\alpha_j), \tag{8}$$

which give small non-zero weights to distances shorter than the footstep length and a small slope with distances of several meters. The result is the piecewise-constant PDF

$$p_\alpha(\varphi_k|\mathbf{r}_{k-1}^i, \ell_k^i) = \tfrac{N_\alpha}{2\pi} \sum_{j=1}^{N_\alpha} s_{m^i}(\alpha_j) \cdot I_{[\alpha_j - \pi/N_\alpha, \alpha_j + \pi/N_\alpha]}(\varphi_k), \tag{9}$$

where $I_S(x)$ is the indicator function for set S. An example of an angular PDF is given in Fig. 1.

We want the proposal distribution to be the product of the floor-plan-based and PDR-based PDFs. Thus, our novel proposal distribution for the user's heading is

$$\begin{aligned} &q(\varphi_k|\mathbf{r}_{k-1}^i, \varphi_{k-1}^i, \ell_k^i, \overline{\Delta}_k, \text{map}) \\ =&\frac{1}{Z^i}\, p_\alpha(\varphi_k|\mathbf{r}_{k-1}^i, \ell_k^i)\, \mathrm{N}(\varphi_k|\varphi_{k-1}^i + \overline{\Delta}_k, q^{\overline{\Delta}}) &(10)\\ =&\frac{1}{Z^i} \sum_{j=1}^{N_\alpha} w_j^i\, \mathrm{N}_{[\alpha_j - \pi/N_\alpha, \alpha_j + \pi/N_\alpha]}(\varphi_k|\varphi_{k-1}^i + \overline{\Delta}_k, q^{\overline{\Delta}}), &(11)\end{aligned}$$

where $\mathrm{N}_{[a,b]}$ is normal distribution truncated to interval $[a, b]$

$$\mathrm{N}_{[a,b]}(x|\mu, \sigma^2) = \tfrac{1}{\Phi((b-\mu)/\sigma) - \Phi((a-\mu)/\sigma)} \mathrm{N}(x|\mu, \sigma^2) \cdot I_{[a,b]}(x),$$

and

$$\begin{aligned} w_j^i =& \tfrac{N_\alpha}{2\pi} \cdot s_{m^i}(\alpha_j) \cdot [\Phi(\Delta^\angle(\alpha_j + \pi/N_\alpha, \varphi_{k-1}^i + \overline{\Delta}_k)/(q^{\overline{\Delta}})^{\frac{1}{2}}) \\ &- \Phi(\Delta^\angle(\alpha_j - \pi/N_\alpha, \varphi_{k-1}^i + \overline{\Delta}_k)/(q^{\overline{\Delta}})^{\frac{1}{2}})], &(12)\end{aligned}$$

$$Z^i = \sum_{j=1}^{N_\alpha} w_j^i, \tag{13}$$

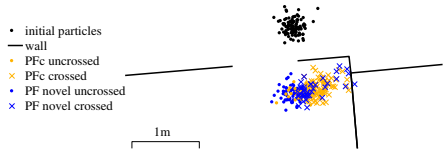


Fig. 2. Particle propagation with conventional PF (PFc) and the proposed method (PF novel) with heading $\varphi \sim \mathrm{N}(290°, (10°)^2)$. Map guides 60 % of PF novel particles through a door, while only 20 % of PFc particles survive.

$\Delta_\angle(\alpha, \beta)$ being the difference angle $\alpha - \beta$ translated by a multiple of 2π to the interval $(-\pi, \pi]$, and $\Phi(x)$ being the cumulative distribution function (CDF) of the standard normal distribution. If the heading change measurement noise follows a non-normal distribution, Φ can be replaced by the appropriate CDF.

Sampling from the distribution (11) is straightforward. First, one generates an index j^i by sampling from the categorical distribution $\mathrm{cat}(w_1^i/Z^i, \ldots, w_{N_\alpha}^i/Z^i)$, i.e. the discrete distribution of $\{1, 2, \ldots, N_\alpha\}$ with the probabilities $\{w_1^i/Z^i, \ldots, w_{N_\alpha}^i/Z^i\}$. Then, the sample of (11) is generated from $\mathrm{N}_{[\alpha_{j^i} - \pi/N_\alpha, \alpha_{j^i} + \pi/N_\alpha]}(\varphi_{k-1}^i + \overline{\Delta}_k, q^{\overline{\Delta}})$, for which efficient methods exist, see e.g. [16]. Fig. 2 shows an example where the novel proposal distribution guides most of the particles through a door while few particles survive with the conventional PDR-only proposal distribution.

Our novel proposal distribution could also be used without any PDR. However, in this case a graph-based map matching method might be more efficient unless the absolute position measurements are very accurate and are made frequently [17].

By modifying the weighting function (7), the proposal distribution could be tuned so that none of the particles would collide with walls (given $q^{\mathbf{r}} = 0$). However, floor-plan errors would not then be modelled, and wall constraints would not be used as measurements that remove particle subclouds where PDR contradicts with the map. Thus, we allow some wall collisions. Notice that the more accurate the PDR, the less influence the map-based angular PDF has. With very accurate PDR few particles should collide with walls anyway.

C. Angular PDF in motion model and/or likelihood

The floor-plan based proposal distribution inspires three PF algorithms: the angular PDF is used only in the proposal distribution, the angular PDF is included in both motion model and proposal distribution, or it is included in measurement likelihood and proposal distribution. The three algorithms differ in the particle weight update formula. A detailed description of the three PFs is presented in Algorithm 2.

PF1: If the angular PDF is used only in the proposal, the proposal PDF and motion model PDF are different, so they do not cancel each other in (5). The update becomes

$$\widetilde{W}_k^i = p(\mathbf{z}_k|\mathbf{x}_k^i)\, \mathbb{P}(C_k|\mathbf{x}_k^i, \mathbf{x}_{k-1}^i) \frac{Z^i}{s_{m^i}(\alpha_{j^i})} \cdot W_{k-1}^i, \qquad (14)$$

where j^i is the angle discretisation index generated for the ith particle. The resulting PF is a solution to the same problem as the conventional wall collision PF, but the particles will collide with the walls less frequently due to the modified proposal distribution. However, the difference in proposal and motion model can again increase the resampling rate.

PF2: If the angular PDF is used in both proposal distribution and motion model, the term $\mathrm{N}(\varphi_k|\varphi_{k-1} + \overline{\Delta}_k, q^{\overline{\Delta}})$ in (2) is replaced by the distribution of (11), so the proposal and motion model cancel out each other in the weight update (5) which then simplifies to

$$\widetilde{W}_k^i = p(\mathbf{z}_k|\mathbf{x}_k^i)\, \mathbb{P}(C_k|\mathbf{x}_k^i, \mathbf{x}_{k-1}^i) \cdot W_{k-1}^i. \qquad (15)$$

This weight update is similar to that of the conventional wall collision PF. However, the motion model is different due to different particle propagation. The proposal PDF cancels out completely in the weight update, so this should provide the lowest resampling rate of the proposed three algorithms. This motion model should be advantageous when the PDR is inaccurate, i.e. $q^{\overline{\Delta}}$ is large, because the probability is not spread to random directions but more probability will be assigned to corridor and open space directions, which can be considered more likely. A possible drawback is that the influence of the map measurements is reduced: particles of wrong areas are not eliminated so often by wall collisions, but the filter relies more on the absolute position measurements.

PF3: If the angular PDF is used in both proposal distribution and measurement likelihood, the proposal distribution's normalisation factor does not cancel out in (5), so the weight update is

$$\widetilde{W}_k^i = p(\mathbf{z}_k|\mathbf{x}_k^i)\, \mathbb{P}(C_k|\mathbf{x}_k^i, \mathbf{x}_{k-1}^i) \cdot Z^i \cdot W_{k-1}^i. \qquad (16)$$

This approach is based on the same motion and measurement models as the method of [8]. Compared to PF2, the proposal distribution's normalisation factor Z^i gives more weight to the particles where the heading matches best with the map.

A major motivation of Kaiser et al. is the scenario where there is imprecise PDR and a bimodal particle cloud with one subcloud in a narrow corridor and another subcloud in open space [8]. In PF3 all the weight will eventually concentrate in the corridor. In PF1 the open space will eventually be more probable, but the proposal distribution improves the estimation in narrow corridors so that the corridor subcloud will die out slower than in the conventional wall collision PF. The approach PF2 gives more weight to the corridor than PF1 but the weight is not moved from open space to corridor. In summary, only PF3 meets the requirement that in multimodal situations the weight should eventually concentrate to narrow corridors, but PF1 and PF2 attempt to make the modelling of corridors more accurate and let the absolute positioning decide in cases with multimodal distributions.

IV. Testing

A. Simulation setting

We test the proposed algorithms with simulated indoor positioning data. The tests were implemented with MATLAB. We used the floor-plan of a campus building of Tampere

Algorithm 2 PF with map & PDR based proposal distribution

Input: prior $p(\mathbf{x}_0)$; number of particles N_p, PDR $\{\overline{\Delta}_k, \overline{\ell}_k\}$ and position meas. $\mathbf{z}_k$, $k \in \{1, \dots\}$; map; angular PDF $s_m(\alpha_j)$
Output: position estimate $\hat{\mathbf{r}}_k$ and covariance matrix $\hat{\Sigma}_k$

1) For each $i = \{1, \dots, N_\mathrm{p}\}$ set $W_0^i := \frac{1}{N_\mathrm{p}}$ and generate $\mathbf{x}_0^i \leftarrow p(\mathbf{x}_0)$. Set the time index $k := 1$.
2) If no footstep is detected at time index k, go to step 5. Otherwise, find the closest grid point m^i for each $i = \{1, \dots, N_\mathrm{p}\}$, and generate
$$\ell_k^i \leftarrow \mathrm{N}(\overline{\ell}_k, q^{\overline{\ell}})$$
$$j^i \leftarrow \mathrm{cat}(w_1^i/Z^i, \dots, w_{N_\alpha}^i/Z^i),\ Z^i = \textstyle\sum_{j=1}^{N_\alpha} w_j^i$$
$$\varphi_k^i \leftarrow \mathrm{N}_{[\alpha_{j^i} - \pi/N_\alpha, \alpha_{j^i} + \pi/N_\alpha]}(\varphi_{k-1}^i + \overline{\Delta}_k, q^{\overline{\Delta}})$$
$$\mathbf{r}_k^i \leftarrow \mathrm{N}(\mathbf{r}_{k-1}^i + \ell_k^i \cdot \begin{bmatrix} \cos(\varphi_k^i) \\ \sin(\varphi_k^i) \end{bmatrix}, q^{\mathbf{r}} \cdot \mathrm{I}_{2\times 2})$$
3) Perform angular PDF weighting $\widetilde{W}_k^i := \mathcal{W}_\alpha^i \cdot W_{k-1}^i$ with
PF1: $\mathcal{W}_\alpha^i = Z^i / s_{m^i}(\alpha_{j^i})$
PF2: $\mathcal{W}_\alpha^i = 1$
PF3: $\mathcal{W}_\alpha^i = Z^i$
4) Set $\widetilde{W}_k^i := \frac{\epsilon}{1-\epsilon} \widetilde{W}_k^i$ for all i such that there is a wall between $\mathbf{r}_{k-1}^i$ and $\mathbf{r}_k^i$, where ϵ is defined in (4).
5) Perform integrity monitoring using the PDR-Kalman. If re-initialised, go to step 7.
6) If no absolute position measurement is obtained at time index k, go to step 7. Otherwise, set $\widetilde{W}_k^i := \mathrm{N}(\mathbf{r}_k^i | \mathbf{z}_k, \Sigma_k) \cdot \widetilde{W}_k^i$ for each $i \in \{1, 2, \dots, N_\mathrm{p}\}$.
7) Normalise the weights by $W_k^i := \widetilde{W}_k^i / \sum_{j=1}^{N_\mathrm{p}} \widetilde{W}_k^j$ for each $i \in \{1, 2, \dots, N_\mathrm{p}\}$.
8) $\hat{\mathbf{r}}_k := \sum_{i=1}^{N_\mathrm{p}} W_k^i \mathbf{r}_k^i$, $\hat{\Sigma}_k := \sum_{i=1}^{N_\mathrm{p}} W_k^i (\mathbf{r}_k^i - \hat{\mathbf{r}}_k)(\mathbf{r}_k^i - \hat{\mathbf{r}}_k)^\mathrm{T}$
9) If $1/\sum_{i=1}^{N_\mathrm{p}} (W_k^i)^2 < 0.1 \cdot N_\mathrm{p}$, resample, and set $W_k^i := 1/N_\mathrm{p}$. Set $k := k + 1$, and go to step 2.

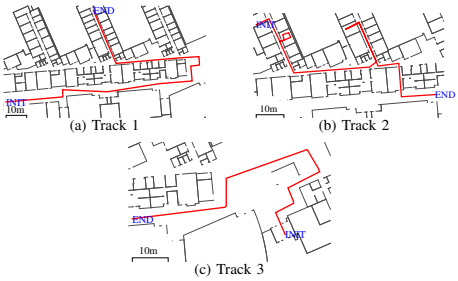


Fig. 3. The test tracks. Track 1 tests behaviour in corridors, track 2 tests doors and rooms, and track 3 tests open space.

University of Technology. We designed three different tracks to test different properties of the algorithms. The tracks are depicted in Fig. 3. Track 1 tests the algorithms' behaviours in corridors, track 2 tests doors and rooms, while track 3 tests open spaces and transition from an open space to a corridor.

The test tracks' paths were defined by hand, but the footstep lengths ℓ_k were simulated from the model

$$v_0 \sim \mathrm{N}(0, 0.2718^2), \tag{17}$$
$$\begin{bmatrix} \ell_k \\ v_k \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} 0.7 + 0.9748 v_{k-1} \\ 0.95 v_{k-1} \end{bmatrix}, \begin{bmatrix} 0.3208 & 0.4751 \\ 0.4751 & 0.9504 \end{bmatrix} \right) \tag{18}$$

This model guarantees that the marginal distribution of each ℓ_k is $\mathrm{N}(0.7, 0.2718^2)$. The step detection was assumed perfect, and the PDR measurements were generated by

$$\overline{\ell}_k \sim \mathrm{N}(||\mathbf{r}_k - \mathbf{r}_{k-1}||,\ (0.7\,\mathrm{m} \cdot 2^\circ \tfrac{\pi}{180^\circ})^2) \tag{19}$$
$$\overline{\Delta}_k \sim \mathrm{N}(\Delta_k - 0.3^\circ \tfrac{\pi}{180^\circ},\ q^{\overline{\Delta}} - (0.3^\circ \tfrac{\pi}{180^\circ})^2), \tag{20}$$
$$\Delta_k = \Delta^\angle(\mathrm{atan}_2([\mathbf{r}_k - \mathbf{r}_{k-1}]_2, [\mathbf{r}_k - \mathbf{r}_{k-1}]_1), \mathrm{atan}_2([\mathbf{r}_{k-1} - \mathbf{r}_{k-2}]_2, [\mathbf{r}_{k-1} - \mathbf{r}_{k-2}]_1)).$$

The model includes a gyro bias of -0.3° per step. The absolute position measurements were generated by

$$\mathbf{z}_k \sim \mathrm{N}(\mathbf{r}_k, (4\,\mathrm{m})^2 \cdot \mathrm{I}_{2\times 2}), \tag{21}$$

and the measurements were received every 20 steps.

B. Filter details

The compared methods are the PDR Kalman filter (KF), the conventional wall collision PF (PFc), the wall collision PF with the conventional (PDR-based) proposal distribution and angular PDF likelihood weighting (PFw), the PF with the novel proposal distribution (PF1), the PF with the novel proposal distribution and the angular PDF included in the motion model (PF2), and the PF with the novel proposal distribution and the angular PDF included in the measurement likelihood (PF3).

The filters are given the correct initial position with covariance matrix $\mathrm{I}_{2\times 2}$ and the correct initial heading with variance $(3^\circ)^2$. In a real scenario, if the initial state were unknown, the KF could be used in the beginning to improve the initial prior of the PF [12]. The KF is based on the motion model

$$\begin{bmatrix} \mathbf{r}_k \\ \mathbf{v}_k \end{bmatrix} | \begin{bmatrix} \mathbf{r}_{k-1} \\ \mathbf{v}_{k-1} \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \mathrm{I} & \mathrm{R}_k \\ \mathrm{O} & \mathrm{R}_k \end{bmatrix} \begin{bmatrix} \mathbf{r}_{k-1} \\ \mathbf{v}_{k-1} \end{bmatrix}, \begin{bmatrix} q^{\mathbf{r}} \cdot \mathrm{I} & \mathrm{O} \\ \mathrm{O} & \sigma_v^2 \cdot \mathrm{I} \end{bmatrix} \right), \tag{22}$$

where $\mathbf{r}$ is user position, $\mathbf{v}$ is step vector, $q^{\mathbf{r}} = (0.01\,\mathrm{m})^2$, and

$$\mathrm{R}_k = \begin{bmatrix} \cos\overline{\Delta}_k & -\sin\overline{\Delta}_k \\ \sin\overline{\Delta}_k & \cos\overline{\Delta}_k \end{bmatrix}, \sigma_v^2 = \max\{(\tfrac{\pi}{90})^2, q^{\overline{\Delta}}\} \cdot (0.7\,\mathrm{m})^2.$$

The KF is thus a version of the PDR-Kalman of [12]. Notice that the KF uses neither footstep length nor map measurements. The same KF is also used as a fallback of the PFs, so that half of the particles are re-initialised if none of the non-zero-weighted particles are in the 99 % probability ellipse of the KF-posterior, similarly to [7].

For robustness, the PFs' propagation step adds independent noise to position with the variance parameter $q^{\mathbf{r}} = (0.01\,\mathrm{m})^2$. The PFs do not take the gyro bias into account, i.e. the PDR model is $\varphi_k | \varphi_{k-1} \sim \mathrm{N}(\varphi_{k-1} + \overline{\Delta}_k, q^{\overline{\Delta}})$. The wall collision checking of the particles is implemented so that each square of a regular grid is assigned with the walls that cross this square, and only the grid squares that are crossed by

the particle trajectory are checked. This is important for the computational efficiency [2]. If a particle crosses a wall, its weight is multiplied by $\frac{\epsilon}{1-\epsilon} = 10^{-4}$.

C. Results and discussion

Fig. 4 shows boxplots of the simulated empirical distributions of the root-mean-square-errors (RMSE) of the filters for the three tracks, for three different values of the gyro noise parameter $q^{\overline{\Delta}}$, and for different numbers of particles N_p. The boxplots show the $5\,\%$, $25\,\%$, $50\,\%$, $75\,\%$, and $95\,\%$ quantiles. The results are based on 100 Monte Carlo replications.

Fig. 4 shows that PF2 has the lowest errors of the novel filters. PF2 converges to the same or slightly better RMSE than the conventional wall collision filter PFc, but with small numbers of particles N_p PF2 is significantly more accurate in corridor tracks 1 and 2 and has similar accuracy in the open space track 3. This can be explained by the fact that the novel proposal distribution makes the filter more efficient in corridors and small rooms. The advantage of PF2 is also clearer when PDR is imprecise, i.e. when $q^{\overline{\Delta}}$ is large, which was expectable because the map measurements have a fixed resolution: when PDR is very precise, the map does not help.

PF1 converges to the same results as PFc, but with noisy PDR and low N_p PF1 outperforms PFc. PFw and PF3 are also based on the same model, and PF3 gives slightly better results with small N_p. Notice that the angular PDF as a part of the measurement likelihood in PFw and PF3 behaves as expected: the accuracy is high on track 1 which consists of corridors, but low on tracks 2 (doors, rooms) and 3 (open space).

Fig. 5 shows the resampling rates of the algorithms, i.e. the number of resamplings divided by the number of footsteps. The results show that the proposed method PF2 has clearly the lowest resampling rate especially when the PDR is imprecise and when the track contains doors and narrow corridors (track 2). Low resampling rate indicates reduced particle degeneracy, which is one explanation for the good performance of PF2.

Based on this simulation, the PF with the novel proposal distribution and angular PDF-affected motion model provides the best accuracy with a small number of particles. Notice that the novel filters require more offline and more online computation per particle as well as a larger map database than the conventional filter PFc. However, the differences in online computation are small compared to the differences in the required N_p; in our MATLAB implementation the online computational requirements of PF1, PF2, and PF3 were roughly $50\,\%$ higher than that of the PFc with the same N_p, and roughly $15\,\%$ higher than that of the PFw.

V. Conclusions

We have presented a novel floor-plan and PDR based proposal distribution for indoor positioning particle filtering. Three versions of the proposed particle filter were compared by computer simulations with the conventional wall collision particle filter and with the particle filter that uses the angular PDF only for particle weighting. Our simulations showed that using floor-plan information in the particle filter's proposal distribution improves accuracy and reduces particle degeneracy especially when computational resources are limited and PDR measurements are noisy. Furthermore, our simulations showed that the angular PDF should also be included in the motion model so that motion model and proposal distribution become the same distributions.

An important topic for future work is generalising the proposed model to multifloor buildings. Another open problem is how to compress the size of the map database: the proposed method requires a grid where each grid point contains an offline-computed discrete distribution with 72 parameters. This might be reduced for example by fitting a continuous distribution to each grid point instead of the discrete distribution.

References

[1] J. Liu, R. Chen, L. Pei, R. Guinness, and H. Kuusniemi, "A hybrid smartphone indoor positioning solution for mobile LBS," *Sensors*, vol. 12, no. 12, pp. 17 208–17 233, December 2012.

[2] H. Leppäkoski, J. Collin, and J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals," *Journal of Signal Processing Systems*, vol. 71, no. 3, pp. 287–296, June 2013.

[3] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter, Particle Filters for Tracking Applications*. Boston, London: Artech House, 2004.

[4] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, July 2010.

[5] F. Evennou, F. Marx, and E. Novakov, "Map-aided indoor mobile positioning system using particle filter," in *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2005, pp. 2490–2494.

[6] Widyawan, M. Klepal, and S. Beauregard, "A Backtracking Particle Filter for Fusing Building Plans with PDR Displacement Estimates," in *5th Workshop on Positioning, Navigation and Communication (WPNC)*, 2008, pp. 207–212.

[7] H. Nurminen, A. Ristimäki, S. Ali-Löytty, and R. Piché, "Particle filter and smoother for indoor localization," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, October 2013, pp. 137–146.

[8] S. Kaiser, M. Khider, and P. Robertson, "A human motion model based on maps for navigation systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 60, 2011.

[9] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, UK: Cambridge University Press, 2013.

[10] J. Collin, O. Mezentsev, and G. Lachapelle, "Indoor positioning system using accelerometry and high accuracy heading sensors," in *GPS/GNSS 2003 Conference*, September 2003.

[11] W. Chen, R. Chen, Y. Chen, H. Kuusniemi, and J. Wang, "An effective pedestrian dead reckoning algorithm using a unified heading error model," in *2010 IEEE/ION Position Location and Navigation Symposium (PLANS)*, May 2010, pp. 340–347.

[12] M. Raitoharju, H. Nurminen, and R. Piché, "Kalman filter with a linear state model for PDR+WLAN positioning and its application to assisting a particle filter," *EURASIP Journal on Advances in Signal Processing*, vol. 1, no. 33, December 2015.

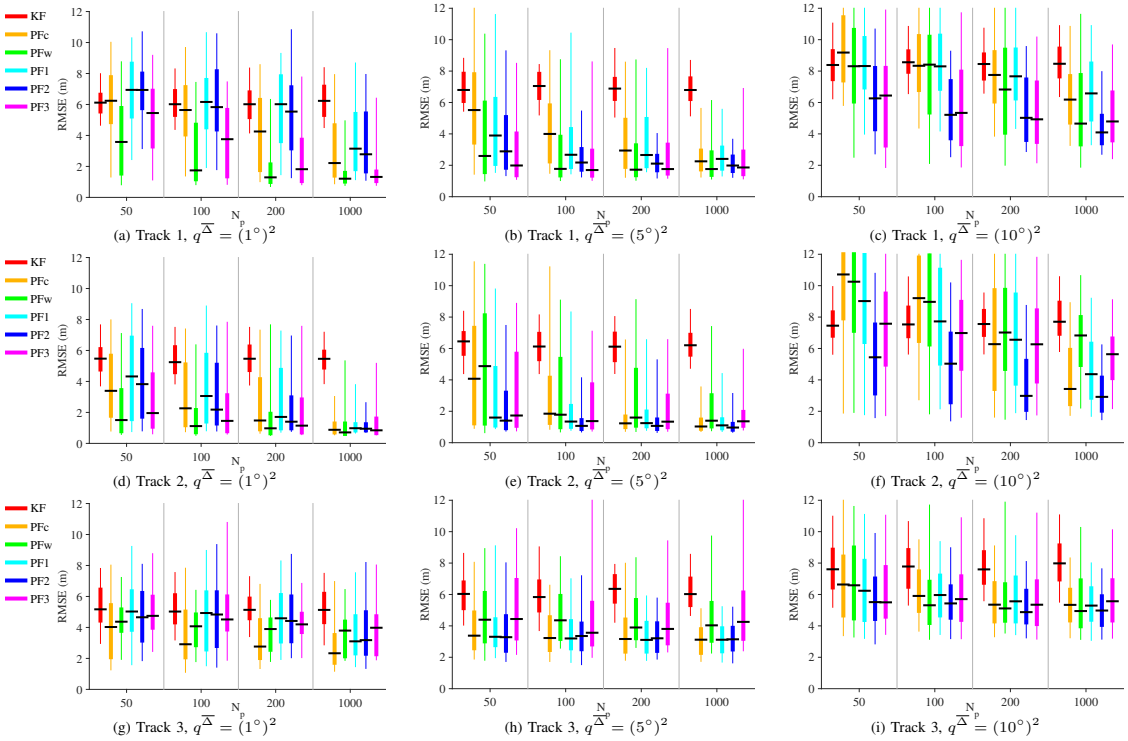
[13] L. Koski, T. Perälä, and R. Piché, "Indoor positioning using WLAN coverage area estimates," in *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, September 2010.

[14] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[15] J. D. Hol, T. B. Schön, and F. Gustafsson, "On resampling algorithms for particle filters," in *IEEE Nonlinear Statistical Signal Processing Workshop (NSSPW)*, September 2006, pp. 79–82.

[16] N. Chopin, "Fast simulation of truncated Gaussian distributions," *Statistics and Computing*, vol. 21, no. 2, pp. 275–288, April 2011.

[17] M. Koivisto, H. Nurminen, S. Ali-Löytty, and R. Piché, "Graph-based map matching for indoor positioning," in *10th International Conference on Information, Communications and Signal Processing (ICICS)*, December 2015.

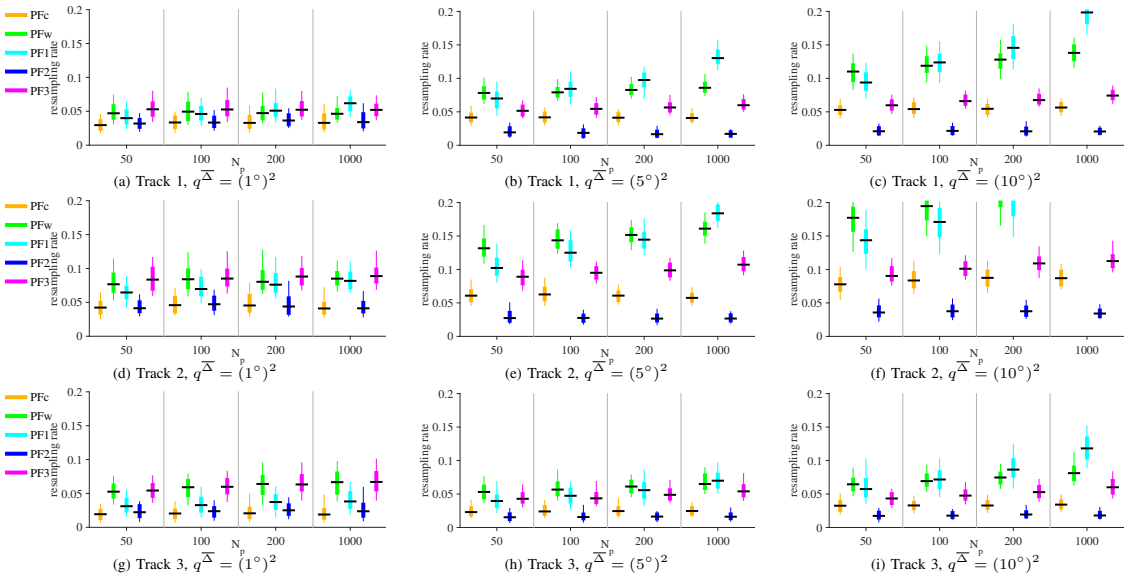


(a) Track 1, $q^{\overline{\Delta}} = (1°)^2$ (b) Track 1, $q^{\overline{\Delta}} = (5°)^2$ (c) Track 1, $q^{\overline{\Delta}} = (10°)^2$

(d) Track 2, $q^{\overline{\Delta}} = (1°)^2$ (e) Track 2, $q^{\overline{\Delta}} = (5°)^2$ (f) Track 2, $q^{\overline{\Delta}} = (10°)^2$

(g) Track 3, $q^{\overline{\Delta}} = (1°)^2$ (h) Track 3, $q^{\overline{\Delta}} = (5°)^2$ (i) Track 3, $q^{\overline{\Delta}} = (10°)^2$

Fig. 4. Simulated RMSE distributions for three test tracks and different values of gyro noise variance $q^{\overline{\Delta}}$ and the number of particles N_p. PF2 outperforms the others especially when the number of particles N_p is low, $q^{\overline{\Delta}}$ is large, and the track contains narrow corridors and doors (track 2).



(a) Track 1, $q^{\overline{\Delta}} = (1°)^2$ (b) Track 1, $q^{\overline{\Delta}} = (5°)^2$ (c) Track 1, $q^{\overline{\Delta}} = (10°)^2$

(d) Track 2, $q^{\overline{\Delta}} = (1°)^2$ (e) Track 2, $q^{\overline{\Delta}} = (5°)^2$ (f) Track 2, $q^{\overline{\Delta}} = (10°)^2$

(g) Track 3, $q^{\overline{\Delta}} = (1°)^2$ (h) Track 3, $q^{\overline{\Delta}} = (5°)^2$ (i) Track 3, $q^{\overline{\Delta}} = (10°)^2$

Fig. 5. Simulated resampling rate distributions. PF2 has the lowest resampling rate, which reduces Monte Carlo error and thus explains good performance.

PUBLICATION 5

Henri Nurminen, Tohid Ardeshiri, Robert Piché, and Fredrik Gustafsson: Robust inference for state-space models with skewed measurement noise. In *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1898–1902, November 2015.
DOI: 10.1109/LSP.2015.2437456

Appendix:
Tohid Ardeshiri, Henri Nurminen, Robert Piché, and Fredrik Gustafsson: Variational Iterations for Filtering and Smoothing with skew-t measurement noise. Department of Electrical Engineering, Linköping University, Sweden, Technical report. March 2015. [Online] http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-115741

Robust Inference for State-Space Models with Skewed Measurement Noise

Henri Nurminen, Tohid Ardeshiri, Robert Piché, *Member, IEEE*, and Fredrik Gustafsson, *Fellow, IEEE*

***Abstract*—Filtering and smoothing algorithms for linear discrete-time state-space models with skewed and heavy-tailed measurement noise are presented. The algorithms use a variational Bayes approximation of the posterior distribution of models that have normal prior and skew-t-distributed measurement noise. The proposed filter and smoother are compared with conventional low-complexity alternatives in a simulated pseudorange positioning scenario. In the simulations the proposed methods achieve better accuracy than the alternative methods, the computational complexity of the filter being roughly 5 to 10 times that of the Kalman filter.**

***Index Terms*—skew t, skewness, t-distribution, robust filtering, Kalman filter, RTS smoother, variational Bayes**

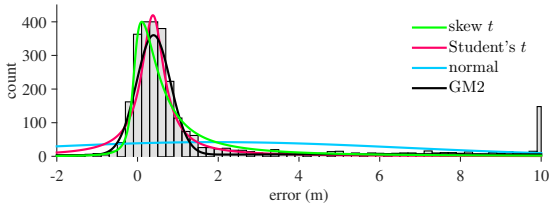


Figure 1. Skewed distributions fit better than symmetric distributions to the time-of-flight measurement errors. BIC values for 2905 data points are 9600 for skew t, 10500 for Student's t, 17200 for normal, and 10000 for GM2.

I. Introduction

The Kalman filter (KF) [1] is the linear minimum mean-square-error filter for linear state-space models, but it is optimal within the set of all filters only when the noise processes are normally distributed [2]. However, the normal distribution has small tail probabilities, and real-world data typically contain large errors ("outliers") more often than the normal distribution predicts [3]. Therefore, the KF is prone to large estimation errors when outliers occur. Hence, there is a need for filtering and smoothing algorithms that mitigate the outlier measurements' influence.

Many applications involve noise processes that have both heavy-tailed (high-kurtosis) and asymmetric (skewed) distributions. In radio signal based distance estimation [4], [5], for example, non-line-of-sight causes large positive errors [6], [7]. Fig. 1 shows the error histogram of a time-of-flight based ultra-wideband distance measurement experiment[1] and maximum likelihood fits of some probability distribution families. By the Bayesian information criterion (BIC) [8], the skewed distributions skew t [9, Ch. 4.3] and two-component Gaussian mixture (GM2) model the data better than the symmetric Student's t [10, Ch. 28] and normal. Other applications for asymmetric distributions have emerged in biostatistics [11], psychiatry [12], environmetrics [13], and economics [14].

Despite these applications, a computationally efficient estimation algorithm for time-series data with heavy-tailed and asymmetric noise has been missing. Robust algorithms that model the heavy-tailed noise with a t-distribution are proposed in [15]–[17], but these do not use the skewness information. A GM2 can model skewness, but the number of mixture components in the posterior increases exponentially with the number of measurements. Furthermore, the GM2 has heavy tails only within a limited range near the component locations, and it has five parameters, while four suffices for modeling location, spread, skewness and kurtosis. Particle filters (PF) [18] can cope with a wide range of models including skewed noise processes, but their computational complexity increases rapidly as the state dimension increases.

This letter proposes approximations to the Bayesian filter and smoother that retain the computational efficiency of the KF while introducing more modeling flexibility for skewed and heavy-tailed measurement noise. The measurement noise is modelled by the skew t-distribution, and the proposed algorithms use a variational Bayes (VB) approximation of the posterior. The proposed filter and smoother are evaluated by numerical pseudorange positioning simulations, where they are compared with the state-of-the-art computationally light algorithms and a PF. To our knowledge, the only earlier work applying VB approximations to the skew t-distribution is that of Wand et al. [19]. However, Wand et al. do not consider state-space models and time-series estimation.

II. Skew t-distribution

Skewed extensions of the well-known unimodal symmetric distributions have been studied since the introduction of the skew normal distribution by Azzalini in [20]. The univariate skew t-distribution is parametrized by its location parameter $\mu \in \mathbb{R}$, spread parameter $\sigma > 0$, shape parameter $\delta \in \mathbb{R}$ and degrees of freedom $\nu > 0$, and has a probability density function (PDF) of the form

$$\mathrm{ST}(z; \mu, \sigma^2, \delta, \nu) = 2\,\mathrm{t}(z; \mu, \delta^2 + \sigma^2, \nu)\,\mathrm{T}(\widetilde{z}; 0, 1, \nu + 1), \tag{1}$$

where

$$\mathrm{t}(z; \mu, \sigma^2, \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{(z-\mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}} \tag{2}$$

H. Nurminen and R. Piché are with the Department of Automation Science and Engineering, Tampere University of Technology (TUT), PO Box 692, 33101 Tampere, Finland (e-mails: henri.nurminen@tut.fi, robert.piche@tut.fi).
H. Nurminen receives funding from TUT Graduate School, Finnish Doctoral Programme in Computational Sciences (FICS), and the Foundation of Nokia Corporation.

T. Ardeshiri and F. Gustafsson are with the Department of Electrical Engineering, Linköping University, 58183 Linköping, Sweden, (e-mails: tohid@isy.liu.se, fredrik@isy.liu.se). T. Ardeshiri receives funding from Swedish research council (VR), project ETT (621-2010-4301)

[1]High accuracy reference measurements are provided through the use of the Vicon real-time tracking system courtesy of the UAS Technologies Lab, Artificial Intelligence and Integrated Computer Systems Division (AIICS) at the Department of Computer and Information Science (IDA). http://www.ida.liu.se/divisions/aiics/aiicssite/index.en.shtml

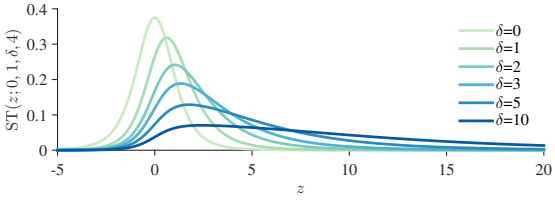


Figure 2. The PDF $\mathrm{ST}(z;0,1,\delta,4)$ for different shape parameter values δ.

is the PDF of Student's t-distribution, $\Gamma(\cdot)$ is the gamma function, and $\widetilde{z} = \frac{(z-\mu)\delta}{\sigma}\left(\frac{\nu+1}{\nu(\delta^2+\sigma^2)+(z-\mu)^2}\right)^{\frac{1}{2}}$. Also, $\mathrm{T}(\cdot;0,1,\nu)$ denotes the cumulative distribution function (CDF) of Student's t-distribution with degrees of freedom ν. The PDF $\mathrm{ST}(z;0,1,\delta,4)$ is plotted for six different values of shape parameter δ in Fig. 2. The skew t-distribution approaches normal distribution when $\nu \to \infty$ and $\delta \to 0$. Expressions for the first two moments of the univariate skew t-distribution with the parametrisation (1) can be found in [21].

Following the introduction of the multivariate skew normal distribution in [22], multivariate skew t-distributions have been proposed in [23]–[25]. In these versions, the PDF of the skew t-distribution involves only the univariate CDF of t-distribution, while the definition of skew t-distribution given in [26]–[28] involves the multivariate CDF, but a single kurtosis factor. In this letter the measurement noise distribution is a product of independent univariate skew t-distributions. This less general model is justified in applications where one-dimensional data from different sensors can be assumed to be statistically independent.

III. Problem formulation

Consider the linear state-space model with skew-t-distributed measurement noise

$$x_{k+1} = Ax_k + w_k, \qquad w_k \overset{\text{iid}}{\sim} \mathcal{N}(w_k;0,Q), \tag{3a}$$

$$y_k = Cx_k + e_k, \qquad [e_k]_i \overset{\text{iid}}{\sim} \mathrm{ST}([e_k]_i;0,R_{ii},\Delta_{ii},\nu_i) \tag{3b}$$

where $\mathcal{N}(\cdot;\mu,\Sigma)$ denotes a (multivariate) normal PDF with mean μ and covariance Σ; $A \in \mathbb{R}^{n_x \times n_x}$ is the state transition matrix; $x_k \in \mathbb{R}^{n_x}$ indexed by $1 \le k \le K$ is the state to be estimated with prior distribution

$$p(x_1) = \mathcal{N}(x_1; x_{1|0}, P_{1|0}); \tag{4}$$

where the subscript "$a|b$" is read "at time a using measurements up to time b"; $y_k \in \mathbb{R}^{n_y}$ also indexed by $1 \le k \le K$ are the measurements and the elements of y_k are conditionally independently skew-t-distributed; $R \in \mathbb{R}^{n_y \times n_y}$ is a diagonal matrix whose diagonal elements R_{ii} are the squares of the spread parameters of (3b); $\Delta \in \mathbb{R}^{n_y \times n_y}$ is a diagonal matrix whose diagonal elements Δ_{ii} are the shape parameters of (3b); $\nu \in \mathbb{R}^{n_y}$ is a vector whose elements ν_i are the degrees of freedom of (3b); $C \in \mathbb{R}^{n_y \times n_x}$ is the measurement matrix; $\{w_k \in \mathbb{R}^{n_x} | 1 \le k \le K\}$ and $\{e_k \in \mathbb{R}^{n_y} | 1 \le k \le K\}$ are mutually independent noise sequences; and the operator $[\cdot]_{ij}$ gives the (i,j) entry of its argument.

The aim of this letter is to derive a Bayesian filter and a Bayesian smoother using the VB method that computes an approximation of the filtering distribution $p(x_k|y_{1:k})$ and smoothing distribution $p(x_k|y_{1:K})$.

IV. Variational solution

The likelihood function implied from (3b) has the hierarchical representation [27]

$$y_k|x_k,u_k,\Lambda_k \sim \mathcal{N}(Cx_k + \Delta u_k, \Lambda_k^{-1}R), \tag{5a}$$

$$u_k|\Lambda_k \sim \mathcal{N}_+(0,\Lambda_k^{-1}), \tag{5b}$$

$$[\Lambda_k]_{ii} \sim \mathcal{G}\left(\frac{\nu_i}{2},\frac{\nu_i}{2}\right). \tag{5c}$$

Λ_k is a diagonal matrix with independent random diagonal elements $[\Lambda_k]_{ii}$, and $\mathcal{N}_+(\mu,\Sigma)$ denotes the (multivariate) truncated normal distribution with closed positive orthant as support, location parameter μ, and squared-scale matrix Σ. Furthermore, $\mathcal{G}(\alpha,\beta)$ denotes the gamma distribution with shape parameter α and rate parameter β.

Using Bayes' theorem, the likelihood (5) and the prior (4), the joint smoothing posterior PDF can be written as

$$\begin{aligned} p(x_{1:K},u_{1:K},\Lambda_{1:K}|y_{1:K}) &\propto p(x_1)\prod_{l=1}^{K-1}p(x_{l+1}|x_l) \\ &\quad\times\prod_{k=1}^{K}p(y_k|x_k,u_k,\Lambda_k)p(u_k|\Lambda_k)p(\Lambda_k) \qquad (6) \\ &= \mathcal{N}(x_1;x_{1|0},P_{1|0})\prod_{l=1}^{K-1}\mathcal{N}(x_{l+1};Ax_l,Q) \\ &\quad\times\prod_{k=1}^{K}\mathcal{N}(y_k;Cx_k+\Delta u_k,\Lambda_k^{-1}R)\mathcal{N}_+(u_k;0,\Lambda_k^{-1}) \\ &\quad\times\prod_{k=1}^{K}\prod_{i=1}^{n_y}\mathcal{G}\left([\Lambda_k]_{ii};\frac{\nu_i}{2},\frac{\nu_i}{2}\right). \qquad (7) \end{aligned}$$

This posterior is not analytically tractable. We seek an approximation in the form

$$p(x_{1:K},u_{1:K},\Lambda_{1:K}|y_{1:K}) \approx q_x(x_{1:K})q_u(u_{1:K})q_\Lambda(\Lambda_{1:K}). \tag{8}$$

In the VB approach, the Kullback-Leibler divergence (KLD) [29] of the true posterior from the factorized approximation is minimized;

$$\hat{q}_x,\hat{q}_u,\hat{q}_\Lambda = \underset{q_x,q_u,q_\Lambda}{\operatorname{argmin}} D_{\mathrm{KL}}(q_x(x_{1:K})q_u(u_{1:K})q_\Lambda(\Lambda_{1:K})||p(x_{1:K},u_{1:K},\Lambda_{1:K}|y_{1:K}))$$

where $D_{\mathrm{KL}}(q(\cdot)||p(\cdot)) \triangleq \int q(x)\log\frac{q(x)}{p(x)}\,\mathrm{d}x$ is the KLD. The analytical solutions for $\hat{q}_x$, $\hat{q}_u$ and $\hat{q}_\Lambda$ can be obtained by cyclic iteration of

$$\log q_x(x_{1:K}) \leftarrow \underset{q_u q_\Lambda}{\mathbb{E}}[\log p(y_{1:K},x_{1:K},u_{1:K},\Lambda_{1:K})] + c_x \tag{9a}$$

$$\log q_u(u_{1:K}) \leftarrow \underset{q_x q_\Lambda}{\mathbb{E}}[\log p(y_{1:K},x_{1:K},u_{1:K},\Lambda_{1:K})] + c_u \tag{9b}$$

$$\log q_\Lambda(\Lambda_{1:K}) \leftarrow \underset{q_x q_u}{\mathbb{E}}[\log p(y_{1:K},x_{1:K},u_{1:K},\Lambda_{1:K})] + c_\Lambda \tag{9c}$$

where the expected values on the right hand sides of (9) are taken with respect to the current q_x, q_u and q_Λ [30, Chapter 10] [31], [32]. Also, c_x, c_u and c_Λ are constants with respect to the variables x_k, u_k and Λ_k, respectively. This recursion is convergent to a local optimum [30, Chapter 10]. When the iterations converge, approximate densities q_u and q_Λ are integrated out from the right hand side of (8) by simply discarding them. Then, the approximate marginal smoothing density $q_x(x_k)$ is obtained, and it turns out to be a normal

Table I
SMOOTHING FOR SKEW-t MEASUREMENT NOISE

Inputs: A, C, Q, R, Δ, ν, $x_{1|0}$, $P_{1|0}$ and $y_{1:K}$
initialization
$\overline{\Lambda_k} \leftarrow I_{n_y}$ for $k = 1 \cdots K$
$\overline{u_k} \leftarrow 0$ for $k = 1 \cdots K$
repeat
  update $q_x(x_{1:K})$ *given* $q_u(u_{1:K})$ *and* $q_\Lambda(\Lambda_{1:K})$
  for k = 1 to K **do**
    $K_x \leftarrow P_{k|k-1}C^{\mathrm{T}}(CP_{k|k-1}C^{\mathrm{T}} + \overline{\Lambda_k}^{-1}R)^{-1}$
    $x_{k|k} \leftarrow x_{k|k-1} + K_x(y_k - Cx_{k|k-1} - \Delta\overline{u_k})$
    $P_{k|k} \leftarrow (I - K_xC)P_{k|k-1}$
  predict $q_x(x_{k+1})$
    $x_{k+1|k} \leftarrow Ax_{k|k}$
    $P_{k+1|k} \leftarrow AP_{k|k}A^{\mathrm{T}} + Q$
  end for
  for k = K-1 down to 1 **do**
    $G_k \leftarrow P_{k|k}A^{\mathrm{T}}P^{-1}_{k+1|k}$
    $x_{k|K} \leftarrow x_{k|k} + G_k(x_{k+1|K} - Ax_{k|k})$
    $P_{k|K} \leftarrow P_{k|k} + G_k(P_{k+1|K} - P_{k+1|k})G_k^{\mathrm{T}}$
  end for
  update $q_u(u_{1:K})$ *and* $q_\Lambda(\Lambda_{1:K})$ *given* $q_x(x_{1:K})$
  for k = 1 to K **do**
    update $q_u(u_k) = \mathcal{N}_+(u_k; u_{k|K}, U_{k|K})$
    $\widetilde{u}_k = y_k - Cx_{k|K}$
    $K_u \leftarrow \Delta(\Delta^2 + R)^{-1}$
    $u_{k|K} \leftarrow K_u\widetilde{u}_k$
    $U_{k|K} \leftarrow (I - K_u\Delta)\overline{\Lambda_k}^{-1}$
    $\overline{u_k} \leftarrow \mathbb{E}_{\mathcal{N}_+(u_{k|K}, U_{k|K})}[u_k]$ ▷ see [34] for the formula
    for i = 1 to n_y **do**
      $\Upsilon_{ii} \leftarrow \mathbb{E}_{\mathcal{N}_+(u_{k|K}, U_{k|K})}[[u_k]_i^2]$ ▷ see [34] for the formula
    end for
  update $q_\Lambda(\Lambda_k) = \prod_{i=1}^{n_y} \mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu_i}{2} + 1, \frac{\nu_i + [\Psi_k]_{ii}}{2}\right)$
    $\Psi_k \leftarrow R^{-1}(\widetilde{u}_k\widetilde{u}_k^{\mathrm{T}} + CP_{k|K}C^{\mathrm{T}}) + (\Delta R^{-1}\Delta + I)\Upsilon$
      $-R^{-1}\Delta\overline{u_k}\widetilde{u}_k^{\mathrm{T}} - \Delta R^{-1}\widetilde{u}_k\overline{u_k}^{\mathrm{T}}$
    $[\overline{\Lambda_k}]_{ii} \leftarrow \frac{\nu_i + 2}{\nu_i + [\Psi_k]_{ii}}$
  end for
until converged
Outputs: $x_{k|K}$ **and** $P_{k|K}$ **for** $k = 1 \cdots K$

Table II
FILTERING FOR SKEW-t MEASUREMENT NOISE

Inputs: A, C, Q, R, Δ, ν, $x_{1|0}$, $P_{1|0}$ and $y_{1:K}$
for k = 1 to K **do**
  initialization
    $\overline{\Lambda_k} \leftarrow I_{n_y}$
    $\overline{u_k} \leftarrow 0$
  repeat
    update $q_x(x_k) = \mathcal{N}(x_k; x_{k|k}, P_{k|k})$ *given* $q_u(u_k)$ *and* $q_\Lambda(\Lambda_k)$
      $K_x \leftarrow P_{k|k-1}C^{\mathrm{T}}(CP_{k|k-1}C^{\mathrm{T}} + \overline{\Lambda_k}^{-1}R)^{-1}$
      $x_{k|k} \leftarrow x_{k|k-1} + K_x(y_k - Cx_{k|k-1} - \Delta\overline{u_k})$
      $P_{k|k} \leftarrow (I - K_xC)P_{k|k-1}$
    update $q_u(u_k) = \mathcal{N}_+(u_k; u_{k|k}, U_{k|k})$ *given* $q_x(x_k)$ *and* $q_\Lambda(\Lambda_k)$
      $K_u \leftarrow \Delta(\Delta^2 + R)^{-1}$
      $\widetilde{u}_k = y_k - Cx_{k|k}$
      $u_{k|k} \leftarrow K_u\widetilde{u}_k$
      $U_{k|k} \leftarrow (I - K_u\Delta)\overline{\Lambda_k}^{-1}$
      $\overline{u_k} \leftarrow \mathbb{E}_{\mathcal{N}_+(u_{k|k}, U_{k|k})}[u_k]$ ▷ see [34] for the formula
      for i = 1 to n_y **do**
        $\Upsilon_{ii} \leftarrow \mathbb{E}_{\mathcal{N}_+(u_{k|k}, U_{k|k})}[[u_k]_i^2]$ ▷ see [34] for the formula
      end for
    update $q_\Lambda(\Lambda_k) = \prod_{i=1}^{n_y} \mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu_i}{2} + 1, \frac{\nu_i + [\Psi_k]_{ii}}{2}\right)$
    given $q_u(u_k)$ *and* $q_x(x_k)$
      $\Psi_k \leftarrow R^{-1}(\widetilde{u}_k\widetilde{u}_k^{\mathrm{T}} + CP_{k|k}C^{\mathrm{T}}) + (\Delta R^{-1}\Delta + I)\Upsilon$
        $-R^{-1}\Delta\overline{u_k}\widetilde{u}_k^{\mathrm{T}} - \Delta R^{-1}\widetilde{u}_k\overline{u_k}^{\mathrm{T}}$
      $[\overline{\Lambda_k}]_{ii} \leftarrow \frac{\nu_i + 2}{\nu_i + [\Psi_k]_{ii}}$
    until converged
  predict $q_x(x_{k+1})$
    $x_{k+1|k} \leftarrow Ax_{k|k}$
    $P_{k+1|k} \leftarrow AP_{k|k}A^{\mathrm{T}} + Q$
end for
Outputs: $x_{k|k}$ **and** $P_{k|k}$ **for** $k = 1 \cdots K$

distribution $q_x(x_k) = \mathcal{N}(x_k; x_{k|K}, P_{k|K})$ where the parameters $x_{k|K}$ and $P_{k|K}$ are the output of the smoothing algorithm given in Table I. The filtering algorithm and the parameters of the filtering posterior $q_x(x_k) = \mathcal{N}(x_k; x_{k|k}, P_{k|k})$ can be found in Table II. The derivations for the expectations given in (9) are relegated to [33] because of space constraints.

V. SIMULATIONS

Numerical simulations are carried out to evaluate the performance of the proposed algorithms Skew-t variational Bayes filter (STVBF) and Skew-t variational Bayes smoother (STVBS). The compared filters are t variational Bayes filter (TVBF) [16], the bootstrap Particle filter (PF), the Kalman filter (KF), and the KF with measurement validation gating (KF-G) [35, Ch. 5.7.2] that discards the individual measurement components whose normalized squared innovation is larger than the χ^2_1-distribution's 99 % quantile. The smoothers are t variational Bayes smoother (TVBS) [16], and Rauch-Tung-Striebel smoother with gating (RTSS-G) [36]. KF and RTSS use the true mean and covariance of the measurement noise distribution, and the TVBF and TVBS use the true mean and $(\nu - 2)/\nu$ times the true covariance as the shape matrix. The computations are done using MATLAB.

A. One-dimensional positioning

The simulation consists of 1000 100-step random-walks of model (3) with parameters $A = 1$, $Q = 1$, $C = \mathbf{1}_{3\times1}$, $R = I_{3\times3}$, $\nu = 4 \cdot \mathbf{1}_{3\times1}$, and $\Delta = 5 \cdot I_{3\times3}$, where $\mathbf{1}$ is a vector of ones. The VB iterations of STVBF and TVBF are terminated when the change in the estimate is less than 0.01.

Some statistics of the estimation error are in Table III, and Fig. 3 shows an example of the error processes. Table III shows that the STVBF has the lowest root-mean-square error (RMSE), the TVBF and KF-G have negative bias, and the KF's error process has the highest standard deviation and positive skew. As illustrated by Fig. 3, the TVBF and KF-G react relatively slowly to positive errors, interpreting them as outliers to be discounted. The KF error's skewness is caused by excessive sensitivity to the large positive measurement errors.

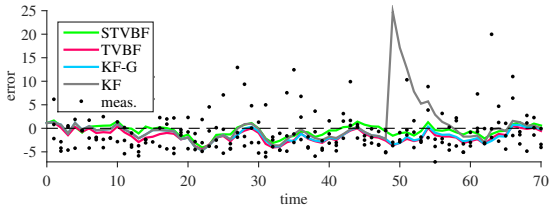
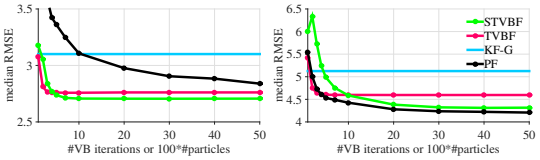


Figure 3. One-dimensional positioning example illustrates TVBF estimate's negative bias and KF's sensitivity to outliers. Measurement error of 300 at time instant 49 is not shown.

Table III
ERROR STATISTICS IN ONE-DIMENSIONAL POSITIONING

Filter	RMSE	Mean	Standard deviation	Skewness
STVBF	1.2	0.1	1.2	0.0
TVBF	1.5	-0.8	1.3	0.2
KF-G	1.5	-0.5	1.4	0.1
KF	1.6	0.0	1.6	0.5



(a) $\delta = 2$ (b) $\delta = 5$

Figure 4. Convergence of the STVBF with $q = 10$. Ten STVBF iterations is enough to outperform TVBF. One additional VB iteration gives the same accuracy gain as 100 additional PF particles.

B. Pseudorange positioning

GNSS-type (global navigation satellite system) pseudorange measurements are simulated from the model

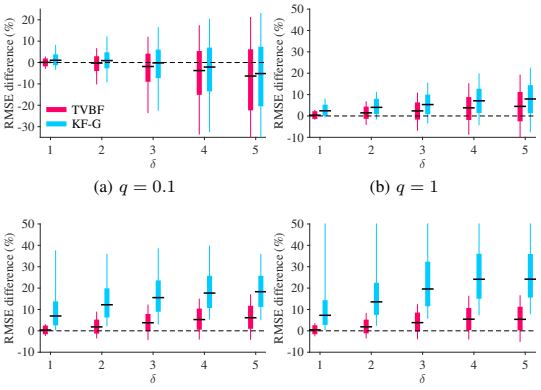
$$[y_k]_i = \|s_i - [x_k]_{1:3}\| + [x_k]_4 + [e_k]_i, \;\; [e_k]_i \overset{\text{iid}}{\sim} \text{ST}(0, 1, \delta, 4) \tag{10}$$

where s_i is the ith satellite's position, $[x_k]_4$ is bias, e_k is noise, and δ is varied. The model is linearized, and the linearization error is negligible because the satellites are far from the receiver. The state model is a three-dimensional random walk with process noise covariance matrix $Q = \text{diag}(q^2, q^2, 0.5^2)$, where q is a parameter. The constant bias $[x_k]_4$ has prior $\mathcal{N}(0, 0.75^2)$. Satellite constellations of Global Positioning System provided by the International GNSS service [37] are used, and on average 7.6 satellites are measured. The results are based on 1000 Monte Carlo replications of a 100-step trajectory. The RMSE is computed for the components $[x_k]_{1:3}$.

1) Evaluation of the filter: Fig. 4 studies the convergence of the STVBF's VB iteration with $q = 10$. The speed of convergence depends on the parameters of the model; the larger δ, the slower convergence, and large q and a high number of sensors can also increase the required number of iterations. The RMSE reduction is fastest for the first iterations, 10 iterations is enough to outperform TVBF, and after 30 iterations the RMSE reduction is negligible. Thus, the STVBF is slower than the TVBF that requires 5 iterations. In this example, one additional VB iteration gives the same accuracy gain as 100 additional PF particles. In the remaining numerical examples, STVBF's VB iteration is terminated after 30 iterations, and TVBF's after 10 iterations.

Fig. 5 shows the distributions of the RMSE differences of the comparison methods from the STVBF's RMSE as percentages of the STVBF's RMSE. The levels of the boxes are 5 %, 25 %, 50 %, 75 %, and 95 % quantiles. With $q \geq 1$, the STVBF outperforms the comparison methods in significant majority of the replications. The problems with $q = 0.1$ are explained by the model structure: only sums of x_k and u_k are measured, so x_k and u_k are correlated *a posteriori*, which makes the VB approximation underestimate the posterior variance [30, Ch. 10.1.2]. The STVBF works well only when the process noise has enough dispersion to dominate in the prior's variance, i.e. when the signal-to-noise ratio (SNR) is not very low.

2) Real-world noise: The robustness of the STVBF is evaluated by generating the noise in Eq. (10) from the histogram distribution of the time-of-flight data set of Fig. 1 and using $q = 10$. The histogram of the RMSE differences of TVBF from the RMSE of STVBF is in Fig. 6. The proposed method has lower RMSE than the TVBF in 61 % of the 1000 Monte Carlo replications. This indicates that the proposed filter is robust to small deviations from the model that appear in real data.



(a) $q = 0.1$ (b) $q = 1$

(c) $q = 10$ (d) $q = 40$

Figure 5. RMSE differences per cent of the STVBF's RMSE. The proposed STVBF outperforms the comparison methods with skewed measurements when the signal-to-noise ratio is high enough.

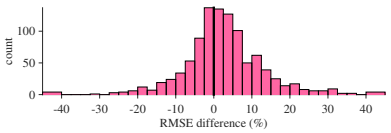
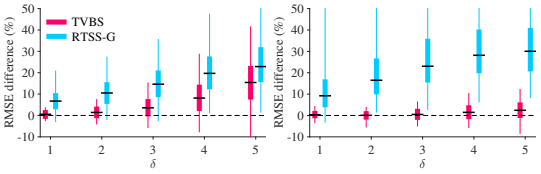


Figure 6. RMSE difference of TVBF per cent of the STVBF's RMSE with noise generated from real time-of-flight measurements' error histogram. STVBF has lower RMSE than the TVBF in 61 % of the 1000 replications.

3) Evaluation of the smoother: The smoother versions of the compared algorithms are evaluated in the same simulation of Eq. (10) with skew-t noise. The STVBS uses 30 and the TVBS 10 VB iterations, which were observed to provide convergence. Fig. 7 shows that the STVBS outperforms the TVBS also at low SNR, but the percentile differences at high SNR are smaller than those of the corresponding filters.



(a) $q = 1$ (b) $q = 40$

Figure 7. Smoothers' RMSE differences per cent of the STVBS's RMSE. STVBS performs well also at low SNR, but difference to TVBS is smaller than the difference between the corresponding filters.

VI. Conclusions

A filter and a smoother that take into account the skewness and heavy-tailedness of the measurement noise are proposed. The algorithms use the variational Bayes approximation. In the presented computer simulations the proposed methods outperform the conventional symmetric Kalman-type algorithms when skewness is present. The computational burden depends on the measurement dimension and model parameters. In the presented simulations the proposed filter has roughly 5 to 10 times the Kalman filter's computational cost.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[2] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, ser. Prentice-Hall information and system sciences. Prentice-Hall, 1979.

[3] R. K. Pearson, "Outliers in process modeling and identification," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 55–63, January 2002.

[4] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, July 2005.

[5] B.-S. Chen, C.-Y. Yang, F.-K. Liao, and J.-F. Liao, "Mobile location estimator in a rough wireless environment using Extended Kalman-based IMM and data fusion," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1157–1169, March 2009.

[6] K. Kaemarungsi and P. Krishnamurthy, "Analysis of WLAN's received signal strength indication for indoor location fingerprinting," *Pervasive and Mobile Computing*, vol. 8, no. 2, pp. 292–316, 2012, special Issue: Wide-Scale Vehicular Sensor Networks and Mobile Sensing.

[7] M. Kok, J. D. Hol, and T. B. Schön, "Indoor positioning using ultrawideband and inertial measurements," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1293–1303, April 2015.

[8] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[9] A. Azzalini and A. Capitanio, *The Skew-Normal and Related Families*. Cambridge University Press, 2014.

[10] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions, Vol. 2*. Wiley, May 1995.

[11] S. Frühwirth-Schnatter and S. Pyne, "Bayesian inference for finite mixtures of univariate and multivariate skew-normal and skew-t distributions," *Biostatistics*, vol. 11, no. 2, pp. 317–336, 2010.

[12] M. Eling, "Fitting insurance claims to skewed distributions: Are the skew-normal and skew-student good models?" *Insurance: Mathematics and Economics*, vol. 51, no. 2, pp. 239–248, 2012.

[13] N. Counsell, M. Cortina-Borja, A. Lehtonen, and A. Stein, "Modelling psychiatric measures using skew-normal distributions," *European Psychiatry*, vol. 26, no. 2, pp. 112–114, 2010.

[14] Y. V. Marchenko, "Multivariate skew-t distributions in econometrics and environmetrics," Ph.D. dissertation, Texas A&M University, December 2010.

[15] G. Agamennoni, J. Nieto, and E. Nebot, "Approximate inference in state-space models with heavy-tailed noise," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5024–5037, October 2012.

[16] R. Piché, S. Särkkä, and J. Hartikainen, "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate Student-t distribution," in *2012 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, September 2012.

[17] M. Roth, E. Özkan, and F. Gustafsson, "A Student's t filter for heavy tailed process and measurement noise," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 5770–5774.

[18] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, July 2000.

[19] M. P. Wand, J. T. Ormerod, S. A. Padoan, and R. Frühwirth, "Mean field variational Bayes for elaborate distributions," *Bayesian Analysis*, vol. 6, no. 4, pp. 847–900, 2011.

[20] A. Azzalini, "A class of distributions which includes the normal ones," *Scandinavian Journal of Statistics*, vol. 12, no. 2, pp. 171–178, 1985.

[21] S. K. Sahu, D. K. Dey, and M. D. Branco, "Erratum: A new class of multivariate skew distributions with applications to Bayesian regression models," *Canadian Journal of Statistics*, vol. 37, no. 2, pp. 301–302, 2009.

[22] A. Azzalini and A. Dalla Valle, "The multivariate skew-normal distribution," *Biometrika*, vol. 83, no. 4, pp. 715–726, 1996.

[23] M. D. Branco and D. K. Dey, "A general class of multivariate skew-elliptical distributions," *Journal of Multivariate Analysis*, vol. 79, no. 1, pp. 99–113, October 2001.

[24] A. Azzalini and A. Capitanio, "Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t-distribution," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 65, no. 2, pp. 367–389, 2003.

[25] A. K. Gupta, "Multivariate skew t-distribution," *Statistics*, vol. 37, no. 4, pp. 359–363, 2003.

[26] S. K. Sahu, D. K. Dey, and M. D. Branco, "A new class of multivariate skew distributions with applications to Bayesian regression models," *Canadian Journal of Statistics*, vol. 31, no. 2, pp. 129–150, 2003.

[27] T.-I. Lin, "Robust mixture modeling using multivariate skew t distributions," *Statistics and Computing*, vol. 20, pp. 343–356, 2010.

[28] S. X. Lee and G. J. McLachlan, "EMMIXuskew: An R package for fitting mixtures of multivariate skew t distributions via the EM algorithm," *Journal of Statistical Software*, vol. 55, no. 12, pp. 1–22, November 2013.

[29] T. M. Cover and J. Thomas, *Elements of Information Theory*. John Wiley and Sons, 2006.

[30] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[31] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 131–146, November 2008.

[32] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, Gatsby Computational Neuroscience Unit, University College London, 2003.

[33] T. Ardeshiri, H. Nurminen, R. Piché, and F. Gustafsson, "Variational iterations for filtering and smoothing with skew-t measurement noise," Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, Tech. Rep. LiTH-ISY-R-3076, March 2015. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-115741

[34] D. R. Barr and E. T. Sherrill, "Mean and variance of truncated normal distributions," *The American Statistician*, vol. 53, no. 4, pp. 357–361, 1999.

[35] Y. Bar-Shalom, R. X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation, Theory Algorithms and Software*. John Wiley & Sons, 2001.

[36] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum Likelihood Estimates of Linear Dynamic Systems," *Journal of the American Institute of Aeronautics and Astronautics*, vol. 3, no. 8, pp. 1445–1450, August 1965.

[37] J. M. Dow, R. Neilan, and C. Rizos, "The international GNSS service in a changing landscape of global navigation satellite systems," *Journal of Geodesy*, vol. 83, no. 7, p. 689, February 2009.

Variational Iterations for Filtering and Smoothing with skew-t measurement noise

Tohid Ardeshiri, Henri Nurminen*, Robert Piché*
and Fredrik Gustafsson

2015-03-24

Abstract

In this technical report, some derivations for the filter and smoother proposed in [1] are presented. More specifically, the derivations for the cyclic iteration needed to solve the variational Bayes filter and smoother for state space models with skew t likelihood proposed in [1] are presented.

1 Problem formulation

A Bayesian filter and a Bayesian smoother using the variational Bayes method for normal prior and skew-t measurement noise are given in [1]. These algorithms compute an approximation of the filtering distribution $p(x_k|y_{1:k})$ and smoothing distribution $p(x_k|y_{1:K})$, respectively. Here, we derive the expectations needed for the cyclic iterations of the variational Bayes smoother which approximates the joint smoothing posterior density given in [1]. The joint smoothing posterior density

$$p(x_{1:K}, u_{1:K}, \Lambda_{1:K}|y_{1:K}) \propto p(x_{1:K}, u_{1:K}, \Lambda_{1:K}, y_{1:K}) \tag{1}$$

$$= p(x_1) \prod_{l=1}^{K-1} p(x_{l+1}|x_l) \prod_{k=1}^{K} p(y_k|x_k, u_k, \Lambda_k) p(u_k|\Lambda_k) p(\Lambda_k) \tag{2}$$

$$= \mathcal{N}(x_1; x_{1|0}, P_{1|0}) \prod_{l=1}^{K-1} \mathcal{N}(x_{l+1}; Ax_l, Q)$$

$$\times \prod_{k=1}^{K} \left\{ \mathcal{N}(y_k; Cx_k + \Delta u_k, \Lambda_k^{-1} R) \mathcal{N}_+(u_k; 0, \Lambda_k^{-1}) \prod_{i=1}^{n_y} \mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu_i}{2}, \frac{\nu_i}{2}\right) \right\} \tag{3}$$

is approximated in [1] by a factorized probability density function (PDF) in the form

$$p(x_{1:K}, u_{1:K}, \Lambda_{1:K}|y_{1:K}) \approx q_x(x_{1:K}) q_u(u_{1:K}) q_\Lambda(\Lambda_{1:K}). \tag{4}$$

*H. Nurminen and R. Piché are with the Department of Automation Science and Engineering, Tampere University of Technology (TUT), PO Box 692, 33101 Tampere, Finland (e-mails: henri.nurminen@tut.fi, robert.piche@tut.fi). H. Nurminen receives funding from TUT Graduate School, Finnish Doctoral Programme in Computational Sciences (FICS), and the Foundation of Nokia Corporation.

The analytical solutions for $\hat{q}_x$, $\hat{q}_u$ and $\hat{q}_\Lambda$ can be obtained by cyclic iteration of

$$\log q_x(x_{1:K}) \leftarrow \mathop{\mathbb{E}}_{q_u q_\Lambda} [\log p(y_{1:K}, x_{1:K}, u_{1:K}, \Lambda_{1:K})] + c_x \tag{5a}$$

$$\log q_u(u_{1:K}) \leftarrow \mathop{\mathbb{E}}_{q_x q_\Lambda} [\log p(y_{1:K}, x_{1:K}, u_{1:K}, \Lambda_{1:K})] + c_u \tag{5b}$$

$$\log q_\Lambda(\Lambda_{1:K}) \leftarrow \mathop{\mathbb{E}}_{q_x q_u} [\log p(y_{1:K}, x_{1:K}, u_{1:K}, \Lambda_{1:K})] + c_\Lambda \tag{5c}$$

where the expected values on the right hand sides of (5) are taken with respect to the current q_x, q_u and q_Λ and c_x, c_u and c_Λ are constants with respect to the variables x_k, u_k and Λ_k, respectively [2, Chapter 10] [3].

2 Derivations for the smoother

In sections 2.1, 2.3 and 2.2 the derivations for the variational solution (5) are given. For brevity all constant values are denoted by c in the derivation. The logarithm of the joint smoothing posterior which is needed for the derivations is given as

$$\begin{aligned}\log p(x_{1:K}, u_{1:K}, \Lambda_{1:K}, y_{1:K}) = \log \mathcal{N}(x_1; x_{1|0}, P_{1|0}) + \sum_{l=1}^{K-1} \log \mathcal{N}(x_{l+1}; Ax_l, Q) \\ + \sum_{k=1}^{K} \left\{ \log \mathcal{N}(y_k; Cx_k + \Delta u_k, \Lambda_k^{-1} R) + \log \mathcal{N}_+(u_k; 0, \Lambda_k^{-1}) \right\} \\ + \sum_{k=1}^{K} \sum_{i=1}^{n_y} \log \mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu_i}{2}, \frac{\nu_i}{2} \right)\end{aligned} \tag{6}$$

2.1 Derivations for q_x

Using equation (5a) we obtain

$$\begin{aligned}
\log q_x(x_{1:K}) &= \log \mathcal{N}(x_1; x_{1|0}, P_{1|0}) + \sum_{l=1}^{K-1} \log \mathcal{N}(x_{l+1}; Ax_l, Q) \\
&\quad + \sum_{k=1}^{K} \mathop{\mathbb{E}}_{q_u q_\Lambda} \left[\log \mathcal{N}(y_k; Cx_k + \Delta u_k, \Lambda_k^{-1} R) \right] + c \qquad (7) \\
&= \log \mathcal{N}(x_1; x_{1|0}, P_{1|0}) + \sum_{l=1}^{K-1} \log \mathcal{N}(x_{l+1}; Ax_l, Q) \\
&\quad - \frac{1}{2} \sum_{k=1}^{K} \mathop{\mathbb{E}}_{q_u q_\Lambda} \left[(y_k - Cx_k - \Delta u_k)^{\mathrm{T}} R^{-1} \Lambda_k (y_k - Cx_k - \Delta u_k) \right] + c \qquad (8) \\
&= \log \mathcal{N}(x_1; x_{1|0}, P_{1|0}) + \sum_{l=1}^{K-1} \log \mathcal{N}(x_{l+1}; Ax_l, Q) \\
&\quad - \frac{1}{2} \sum_{k=1}^{K} (y_k - Cx_k - \Delta \overline{u_k})^{\mathrm{T}} R^{-1} \overline{\Lambda_k} (y_k - Cx_k - \Delta \overline{u_k}) + c \qquad (9) \\
&= \log \mathcal{N}(x_1; x_{1|0}, P_{1|0}) + \sum_{l=1}^{K-1} \log \mathcal{N}(x_{l+1}; Ax_l, Q) \\
&\quad + \sum_{k=1}^{K} \log \mathcal{N}(y_k - \Delta \overline{u_k}; Cx_k, \overline{\Lambda_k}^{-1} R) + c \qquad (10)
\end{aligned}$$

where $\overline{u_k} \triangleq \mathbb{E}_{q_u}[u_k]$ and $\overline{\Lambda_k} \triangleq \mathbb{E}_{q_\Lambda}[\Lambda_k]$ are derived in sections 2.2 and 2.3, respectively. Hence, $\log q_x(x_{1:K})$ in (10) has the same form as logarithm of the joint posterior of a linear state-space model with measurements $\widetilde{y_k} \triangleq y_k - \Delta \overline{u_k}$ and Gaussian measurement noise covariance $\widetilde{R} \triangleq \overline{\Lambda_k}^{-1} R$. Therefore, $q_x(x_{1:K})$ can be computed using the Rauch-Tung-Striebel smoother's recursion [4]. The approximate marginal distribution of x_k turns out to be

$$q_x(x_k) = \mathcal{N}(x_k; x_{k|K}, P_{k|K}), \qquad (11)$$

where expressions for $x_{k|K}$, $P_{k|K}$ are given in [1, Table I].

2.2 Derivations for q_u

Using equation (5b) we obtain

$$\log q_u(u_{1:K}) = \sum_{k=1}^{K} \mathop{\mathbb{E}}_{q_\Lambda q_x} \left[\log \mathcal{N}(y_k; Cx_k + \Delta u_k, \Lambda_k^{-1} R) + \log \mathcal{N}_+(u_k; 0, \Lambda_k^{-1})\right] + c. \tag{12}$$

Therefore, $q_u(u_{1:K}) = \prod_{k=1}^{K} q_u(u_k)$ where

$$\log q_u(u_k) = \mathop{\mathbb{E}}_{q_\Lambda q_x} \left[\log \mathcal{N}(y_k; Cx_k + \Delta u_k, \Lambda_k^{-1} R) + \log \mathcal{N}_+(u_k; 0, \Lambda_k^{-1})\right] + c \tag{13}$$

$$= -\frac{1}{2} \mathop{\mathbb{E}}_{q_\Lambda q_x} \left[(y_k - Cx_k - \Delta u_k)^{\mathrm{T}} R^{-1} \Lambda_k (y_k - Cx_k - \Delta u_k)\right] - \frac{1}{2} \mathop{\mathbb{E}}_{q_\Lambda} [u_k^{\mathrm{T}} \Lambda_k u_k] + c \tag{14}$$

$$= -\frac{1}{2}(y_k - C\overline{x_k} - \Delta u_k)^{\mathrm{T}} R^{-1} \overline{\Lambda_k} (y_k - C\overline{x_k} - \Delta u_k) - \frac{1}{2} u_k^{\mathrm{T}} \overline{\Lambda_k} u_k + c, \tag{15}$$

where, $\overline{x_k} \triangleq \mathbb{E}_{q_x}[x_k] = x_{k|K}$. Therefore,

$$q_u(u_k) = \mathcal{N}_+(u_k; u_{k|K}, U_{k|K}) \tag{16}$$

where

$$u_{k|K} = K_u(y_k - Cx_{k|K}), \tag{17}$$

$$U_{k|K} = (I - K_u \Delta)\overline{\Lambda_k}^{-1}, \tag{18}$$

$$K_u = \overline{\Lambda_k}^{-1} \Delta (\Delta \overline{\Lambda_k}^{-1} \Delta + \overline{\Lambda_k}^{-1} R)^{-1} = \Delta(\Delta^2 + R)^{-1}. \tag{19}$$

The expectation $\overline{u_k}$ is needed in (10) and can be calculated using e.g., [5]. Note that the cumulative distribution function of univariate normal distribution (or some approximation of it) is required in the computation of the moments of the truncated normal distribution.

2.3 Derivations for q_Λ

Using equation (5c) we obtain

$$\begin{aligned}\log q_\Lambda(\Lambda_{1:K}) = &\sum_{k=1}^{K} \left\{ \mathop{\mathbb{E}}_{q_u q_x} \left[\log \mathcal{N}(y_k; Cx_k + \Delta u_k, \Lambda_k^{-1} R) + \log \mathcal{N}_+(u_k; 0, \Lambda_k^{-1})\right] \right\} \\ &+ \sum_{k=1}^{K} \sum_{i=1}^{n_y} \log \mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu_i}{2}, \frac{\nu_i}{2}\right) + c. \end{aligned} \tag{20}$$

Therefore, $q_\Lambda(\Lambda_{1:K}) = \prod_{k=1}^K q_\Lambda(\Lambda_k)$ where

$$\begin{aligned}
\log q_\Lambda(\Lambda_k) = &\mathop{\mathbb{E}}_{q_u q_x}\left[\log \mathcal{N}(y_k; Cx_k + \Delta u_k, \Lambda_k^{-1} R) + \log \mathcal{N}_+(u_k; 0, \Lambda_k^{-1})\right] \\
&+ \sum_{i=1}^{n_y} \log \mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu_i}{2}, \frac{\nu_i}{2}\right) + c \qquad (21) \\
= &-\frac{1}{2} \mathop{\mathbb{E}}_{q_u q_x}\left[\mathrm{tr}(\Lambda_k R^{-1}(y_k - Cx_k - \Delta u_k)(y_k - Cx_k - \Delta u_k)^{\mathrm{T}})\right] \\
&- \frac{1}{2} \mathop{\mathbb{E}}_{q_u}[\mathrm{tr}(\Lambda_k u_k u_k^{\mathrm{T}})] + \sum_{i=1}^{n_y} \left(\frac{\nu_i}{2} \log[\Lambda_k]_{ii} - \frac{\nu_i}{2}[\Lambda_k]_{ii}\right) + c \qquad (22) \\
= &-\frac{1}{2} \mathop{\mathbb{E}}_{q_x}[\mathrm{tr}(\Lambda_k R^{-1}(y_k - Cx_k)(y_k - Cx_k)^{\mathrm{T}})] - \frac{1}{2} \mathop{\mathbb{E}}_{q_u}[\mathrm{tr}(\Lambda_k \Delta R^{-1} \Delta u_k u_k^{\mathrm{T}})] \\
&+ \frac{1}{2}(y_k - C\overline{x_k})^{\mathrm{T}} \Lambda_k R^{-1} \Delta \overline{u_k} + \frac{1}{2} \overline{u_k}^{\mathrm{T}} \Delta \Lambda_k R^{-1}(y_k - C\overline{x_k}) \\
&- \frac{1}{2} \mathop{\mathbb{E}}_{q_u}[\mathrm{tr}(\Lambda_k u_k u_k^{\mathrm{T}})] + \sum_{i=1}^{n_y} \left(\frac{\nu_i}{2} \log[\Lambda_k]_{ii} - \frac{\nu_i}{2}[\Lambda_k]_{ii}\right) + c \qquad (23) \\
= &-\frac{1}{2} \mathrm{tr}(\Lambda_k R^{-1}((y_k - Cx_{k|K})(y_k - Cx_{k|K})^{\mathrm{T}} + CP_{k|K}C^{\mathrm{T}})]) \\
&- \frac{1}{2} \mathrm{tr}(\Lambda_k(\Delta R^{-1}\Delta + I) \mathop{\mathbb{E}}_{q_u}[u_k u_k^{\mathrm{T}}]) + \frac{1}{2} \mathrm{tr}(\Lambda_k R^{-1} \Delta \overline{u_k}(y_k - Cx_{k|K})^{\mathrm{T}}) \\
&+ \frac{1}{2} \mathrm{tr}(\Lambda_k \Delta R^{-1}(y_k - Cx_{k|K})\overline{u_k}^{\mathrm{T}}) + \sum_{i=1}^{n_y} \left(\frac{\nu_i}{2} \log[\Lambda_k]_{ii} - \frac{\nu_i}{2}[\Lambda_k]_{ii}\right) + c \qquad (24) \\
= &\sum_{i=1}^{n_y} \left(\frac{\nu_i}{2} \log[\Lambda_k]_{ii} - \frac{\nu_i + [\Psi_k]_{ii}}{2}[\Lambda_k]_{ii}\right) + c, \qquad (25)
\end{aligned}$$

where the commutative property of product of diagonal matrices Δ, Λ_k and R is used in several occasions and

$$\begin{aligned}
\Psi_k = &R^{-1}((y_k - Cx_{k|K})(y_k - Cx_{k|K})^{\mathrm{T}} + CP_{k|K}C^{\mathrm{T}}) + (\Delta R^{-1}\Delta + I) \mathop{\mathbb{E}}_{q_u}[u_k u_k^{\mathrm{T}}] \\
&- R^{-1}\Delta \overline{u_k}(y_k - Cx_{k|K})^{\mathrm{T}} - \Delta R^{-1}(y_k - Cx_{k|K})\overline{u_k}^{\mathrm{T}}. \qquad (26)
\end{aligned}$$

Therefore,

$$q_\Lambda(\Lambda_k) = \prod_{i=1}^{n_y} \mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu_i}{2} + 1, \frac{\nu_i + [\Psi_k]_{ii}}{2}\right). \qquad (27)$$

Note that only the diagonal elements of the matrix Ψ_k are required. As a consequence, provided that Δ and R are diagonal, only the diagonal elements of $\mathbb{E}[u_k u_k^{\mathrm{T}}]$ are required. These are second moments of univariate truncated normal distributions that can be computed using e.g., [5]. In the derivations above $\mathbb{E}_{q_\Lambda}[\Lambda_k]$ is required. The diagonal elements of $\mathbb{E}_{q_\Lambda}[\Lambda_k]$ are

$$\mathop{\mathbb{E}}_{q_\Lambda}[[\Lambda_k]_{ii}] = \frac{\nu_i + 2}{\nu_i + [\Psi_k]_{ii}}. \qquad (28)$$

3 Derivations for the Filter

The filtering recursions are similar to those of the smoother given in section 2. However, since the notation used in the filtering algorithm [1, Table II] is different from the notation used for smoothing algorithm, the derivation for the filter will be given separately.

Suppose that at time index k the measurement y_k is available, and the prediction PDF $p(x_k|y_{1:k-1})$ is

$$p(x_k|y_{1:k-1}) = \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1}). \tag{29}$$

Then, using Bayes' theorem the joint filtering posterior PDF can be written as

$$\begin{aligned} p(x_k, u_k, \Lambda_k|y_{1:k}) \propto & p(y_k, x_k, u_k, \Lambda_k|y_{1:k-1}) && (30)\\ = & p(y_k|x_k, u_k, \Lambda_k)p(x_k|y_{1:k-1})p(u_k|\Lambda_k)p(\Lambda_k) && (31)\\ = & \mathcal{N}(y_k; Cx_k + \Delta u_k, \Lambda_k^{-1}R)\mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1}) \\ & \times \mathcal{N}_+(u_k; 0, \Lambda_k^{-1}) \prod_{i=1}^{n_y} \mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu_i}{2}, \frac{\nu_i}{2}\right). && (32) \end{aligned}$$

This posterior is not analytically tractable. We seek an approximation in the form

$$p(x_k,u_k, \Lambda_k|y_{1:k}) \approx q_x(x_k)q_u(u_k)q_\Lambda(\Lambda_k). \tag{33}$$

In the VB approach, the Kullback-Leibler divergence (KLD) of the true posterior from the factorized approximation is minimized;

$$\hat{q}_x, \hat{q}_u, \hat{q}_\Lambda = \underset{q_x, q_u, q_\Lambda}{\operatorname{argmin}} D_{\mathrm{KL}}(q_x(x_k)q_u(u_k)q_\Lambda(\Lambda_k)||p(x_k, u_k, \Lambda_k|y_{1:k}))$$

where $D_{\mathrm{KL}}(q(\cdot)||p(\cdot)) \triangleq \int q(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$ is the KLD [6]. The analytical solutions for $\hat{q}_x$, $\hat{q}_u$ and $\hat{q}_\Lambda$ can be obtained by cyclic iteration of

$$\log q_x(x_k) \leftarrow \underset{q_u q_\Lambda}{\mathbb{E}} [\log p(y_k, x_k, u_k, \Lambda_k|y_{1:k-1})] + c_x \tag{34a}$$

$$\log q_u(u_k) \leftarrow \underset{q_x q_\Lambda}{\mathbb{E}} [\log p(y_k, x_k, u_k, \Lambda_k|y_{1:k-1})] + c_u \tag{34b}$$

$$\log q_\Lambda(\Lambda_k) \leftarrow \underset{q_x q_u}{\mathbb{E}} [\log p(y_k, x_k, u_k, \Lambda_k|y_{1:k-1})] + c_\Lambda \tag{34c}$$

where the expected values on the right hand sides of (34) are taken with respect to the current q_x, q_u and q_Λ and c_x, c_u and c_Λ are constants with respect to the variables x_k, u_k and Λ_k, respectively [2, Chapter 10] [3]. This recursion is convergent to a local optimum [2, Chapter 10]. In sections 3.1, 3.3 and 3.2 the derivations for the variational solution (34) are given. For brevity all constant values are denoted by c in the derivation. The logarithm of the joint filtering posterior which is needed for the derivations is given by

$$\begin{aligned} \log p(y_k, x_k, u_k, \Lambda_k|y_{1:k-1}) = & -\frac{1}{2}(y_k - Cx_k - \Delta u_k)^{\mathrm{T}}\Lambda_k R^{-1}(y_k - Cx_k - \Delta u_k) \\ & -\frac{1}{2}(x_k - x_{k|k-1})^{\mathrm{T}}P_{k|k-1}^{-1}(x_k - x_{k|k-1}) \\ & -\frac{1}{2}u_k^{\mathrm{T}}\Lambda_k u_k + \sum_{i=1}^{n_y} \frac{\nu_i}{2} \log[\Lambda_k]_{ii} - \frac{\nu_i}{2}[\Lambda_k]_{ii} + c. \end{aligned} \tag{35}$$

3.1 Derivations for q_x

Using equation (34a) we obtain

$$\begin{aligned}\log q_x(x_k) = &-\frac{1}{2}\mathop{\mathbb{E}}_{q_u q_\Lambda}\left[(y_k - Cx_k - \Delta u_k)^{\mathrm{T}}\Lambda_k R^{-1}(y_k - Cx_k - \Delta u_k)\right] \\ &-\frac{1}{2}(x_k - x_{k|k-1})^{\mathrm{T}}P_{k|k-1}^{-1}(x_k - x_{k|k-1}) + c \qquad (36)\\ = &-\frac{1}{2}(y_k - Cx_k - \Delta\overline{u_k})^{\mathrm{T}}\overline{\Lambda_k}R^{-1}(y_k - Cx_k - \Delta\overline{u_k}) \\ &-\frac{1}{2}(x_k - x_{k|k-1})^{\mathrm{T}}P_{k|k-1}^{-1}(x_k - x_{k|k-1}) + c, \qquad (37)\end{aligned}$$

where $\overline{u_k} \triangleq \mathbb{E}_{q_u}[u_k]$ and $\overline{\Lambda_k} \triangleq \mathbb{E}_{q_\Lambda}[\Lambda_k]$ are derived in sections 3.2 and 3.3, respectively. Hence,

$$q_x(x_k) \propto \mathcal{N}(y_k - \Delta\overline{u_k}; Cx_k, \overline{\Lambda_k}^{-1}R)\mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1}) \qquad (38)$$

which can be computed using the Kalman filter's [7] measurement update. Therefore,

$$q_x(x_k) = \mathcal{N}(x_k; x_{k|k}, P_{k|k}) \qquad (39)$$

where,

$$\begin{aligned}x_{k|k} &= x_{k|k-1} + K_x(y_k - Cx_{k|k-1} - \Delta\overline{u_k}), &(40)\\ P_{k|k} &= (I - K_x C)P_{k|k-1}, &(41)\\ K_x &= P_{k|k-1}C^{\mathrm{T}}(CP_{k|k-1}C^{\mathrm{T}} + \overline{\Lambda_k}^{-1}R)^{-1}. &(42)\end{aligned}$$

3.2 Derivations for q_u

Using equation (34b) we obtain

$$\begin{aligned}\log q_u(u_k) = &-\frac{1}{2}\mathop{\mathbb{E}}_{q_x q_\Lambda}\left[(y_k - Cx_k - \Delta u_k)^{\mathrm{T}}\Lambda_k R^{-1}(y_k - Cx_k - \Delta u_k)\right] \\ &-\frac{1}{2}\mathop{\mathbb{E}}_{q_\Lambda}[u_k^{\mathrm{T}}\Lambda_k u_k] + c \qquad (43)\\ = &-\frac{1}{2}(y_k - C\overline{x_k} - \Delta u_k)^{\mathrm{T}}\overline{\Lambda_k}R^{-1}(y_k - C\overline{x_k} - \Delta u_k) \\ &-\frac{1}{2}u_k^{\mathrm{T}}\overline{\Lambda_k}u_k + c, \qquad (44)\end{aligned}$$

where, $\overline{x_k} \triangleq \mathbb{E}_{q_x}[x_k] = x_{k|k}$. Therefore,

$$q_u(u_k) = \mathcal{N}_+(u_k; u_{k|k}, U_{k|k}) \qquad (45)$$

where

$$\begin{aligned}u_{k|k} &= K_u(y_k - Cx_{k|k}), &(46)\\ U_{k|k} &= (I - K_u\Delta)\overline{\Lambda_k}^{-1}, &(47)\\ K_u &= \overline{\Lambda_k}^{-1}\Delta(\Delta\overline{\Lambda_k}^{-1}\Delta + \overline{\Lambda_k}^{-1}R)^{-1} = \Delta(\Delta^2 + R)^{-1}. &(48)\end{aligned}$$

The expectation $\overline{u_k}$ is needed in (40) and can be calculated using e.g., [5].

3.3 Derivations for q_Λ

Using equation (34c) we obtain

$$\begin{aligned}
\log q_\Lambda(\Lambda_k) = &-\frac{1}{2}\mathop{\mathbb{E}}_{q_u q_x}[\mathrm{tr}(\Lambda_k R^{-1}(y_k - Cx_k - \Delta u_k)(y_k - Cx_k - \Delta u_k)^{\mathrm{T}})] \\
&- \frac{1}{2}\mathop{\mathbb{E}}_{q_u}[\mathrm{tr}(\Lambda_k u_k u_k^{\mathrm{T}})] + \sum_{i=1}^{n_y}\left(\frac{\nu_i}{2}\log[\Lambda_k]_{ii} - \frac{\nu_i}{2}[\Lambda_k]_{ii}\right) + c \qquad (49)\\
= &-\frac{1}{2}\mathop{\mathbb{E}}_{q_x}[\mathrm{tr}(\Lambda_k R^{-1}(y_k - Cx_k)(y_k - Cx_k)^{\mathrm{T}})] - \frac{1}{2}\mathop{\mathbb{E}}_{q_u}[\mathrm{tr}(\Lambda_k \Delta R^{-1}\Delta u_k u_k^{\mathrm{T}})] \\
&+ \frac{1}{2}(y_k - C\overline{x_k})^{\mathrm{T}}\Lambda_k R^{-1}\Delta\overline{u_k} + \frac{1}{2}\overline{u_k}^{\mathrm{T}}\Delta\Lambda_k R^{-1}(y_k - C\overline{x_k}) \\
&- \frac{1}{2}\mathop{\mathbb{E}}_{q_u}[\mathrm{tr}(\Lambda_k u_k u_k^{\mathrm{T}})] + \sum_{i=1}^{n_y}\left(\frac{\nu_i}{2}\log[\Lambda_k]_{ii} - \frac{\nu_i}{2}[\Lambda_k]_{ii}\right) + c \qquad (50)\\
= &-\frac{1}{2}\mathrm{tr}(\Lambda_k R^{-1}((y_k - Cx_{k|k})(y_k - Cx_{k|k})^{\mathrm{T}} + CP_{k|k}C^{\mathrm{T}})]) \\
&- \frac{1}{2}\mathrm{tr}(\Lambda_k(\Delta R^{-1}\Delta + I)\mathop{\mathbb{E}}_{q_u}[u_k u_k^{\mathrm{T}}]) + \frac{1}{2}\mathrm{tr}(\Lambda_k R^{-1}\Delta\overline{u_k}(y_k - Cx_{k|k})^{\mathrm{T}}) \\
&+ \frac{1}{2}\mathrm{tr}(\Lambda_k \Delta R^{-1}(y_k - Cx_{k|k})\overline{u_k}^{\mathrm{T}}) + \sum_{i=1}^{n_y}\left(\frac{\nu_i}{2}\log[\Lambda_k]_{ii} - \frac{\nu_i}{2}[\Lambda_k]_{ii}\right) + c \qquad (51)\\
= &\sum_{i=1}^{n_y}\left(\frac{\nu_i}{2}\log[\Lambda_k]_{ii} - \frac{\nu_i + [\Psi_k]_{ii}}{2}[\Lambda_k]_{ii}\right) + c, \qquad (52)
\end{aligned}$$

where

$$\begin{aligned}
\Psi_k = &R^{-1}((y_k - Cx_{k|k})(y_k - Cx_{k|k})^{\mathrm{T}} + CP_{k|k}C^{\mathrm{T}}) + (\Delta R^{-1}\Delta + I)\mathop{\mathbb{E}}_{q_u}[u_k u_k^{\mathrm{T}}] \\
&- R^{-1}\Delta\overline{u_k}(y_k - Cx_{k|k})^{\mathrm{T}} - \Delta R^{-1}(y_k - Cx_{k|k})\overline{u_k}^{\mathrm{T}}. \qquad (53)
\end{aligned}$$

Therefore,

$$q_\Lambda(\Lambda_k) = \prod_{i=1}^{n_y}\mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu_i}{2} + 1, \frac{\nu_i + [\Psi_k]_{ii}}{2}\right). \qquad (54)$$

Note that only the diagonal elements of the matrix Ψ_k are required. As a consequence, provided that Δ and R are diagonal, only the diagonal elements of $\mathbb{E}[u_k u_k^{\mathrm{T}}]$ are required. These are second moments of univariate truncated normal distributions that can be computed using e.g., [5]. In the derivations above $\mathbb{E}_{q_\Lambda}[\Lambda_k]$ is required. The diagonal elements of $\mathbb{E}_{q_\Lambda}[\Lambda_k]$ are

$$\mathop{\mathbb{E}}_{q_\Lambda}[[\Lambda_k]_{ii}] = \frac{\nu_i + 2}{\nu_i + [\Psi_k]_{ii}}. \qquad (55)$$

References

[1] H. Nurminen, T. Ardeshiri, R. Piché, and F. Gustafsson, "Robust inference for state-space models with skewed measurement noise," *Signal Processing Letters*, p. submitted, 2015. [Online]. Available: http://arxiv.org/abs/1503.06606

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[3] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 131–146, Nov. 2008.

[4] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum Likelihood Estimates of Linear Dynamic Systems," *Journal of the American Institute of Aeronautics and Astronautics*, vol. 3, no. 8, pp. 1445–1450, Aug 1965.

[5] D. R. Barr and E. T. Sherrill, "Mean and variance of truncated normal distributions," *The American Statistician*, vol. 53, no. 4, pp. 357–361, 1999.

[6] T. M. Cover and J. Thomas, *Elements of Information Theory*. John Wiley and Sons, 2006.

[7] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

PUBLICATION 6

Henri Nurminen, Tohid Ardeshiri, Robert Piché, and Fredrik Gustafsson: A NLOS-robust TOA positioning filter based on a skew-t measurement noise model. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, October 2015.
DOI: 10.1109/IPIN.2015.7346786

2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 13-16 October 2015, Banff, Alberta, Canada

A NLOS-robust TOA positioning filter based on a skew-t measurement noise model

Henri NURMINEN*, Tohid ARDESHIRI†, Robert PICHÉ* , and Fredrik GUSTAFSSON†

*Department of Automation Science and Engineering, Tampere University of Technology, Finland
{henri.nurminen, robert.piche}@tut.fi
†Division of Automatic Control, Linköping University, Sweden
{tohid, fredrik}@isy.liu.se

***Abstract*—A skew-t variational Bayes filter (STVBF) is applied to indoor positioning with time-of-arrival (TOA) based distance measurements and pedestrian dead reckoning (PDR). The proposed filter accommodates large positive outliers caused by occasional non-line-of-sight (NLOS) conditions by using a skew-t model of measurement errors. Real-data tests using the fusion of inertial sensors based PDR and ultra-wideband based TOA ranging show that the STVBF clearly outperforms the extended Kalman filter (EKF) in positioning accuracy with the computational complexity about three times that of the EKF.**

***Keywords*—indoor positioning, TOA, UWB, NLOS, robust filtering, skewness, skew t, variational Bayes**

I. Introduction

In line-of-sight (LOS) conditions ultra-wideband (UWB) radio's time-of-arrival (TOA) measurements provide ranging accuracy of tens of centimeters. However, non-line-of-sight (NLOS) can reduce the estimation accuracy drastically by introducing a positive measurement bias whose magnitude can be several meters [1]. This is a serious problem in UWB-based indoor positioning because indoor environments typically contain various obstacles that cause NLOS situations whose modeling is challenging.

The conventional extended Kalman filter (EKF) algorithm for TOA positioning gives large location estimation errors in the NLOS condition [1]. Therefore, several works have proposed using NLOS identification and mitigation procedures that attempt to recognize NLOS from the shape of the received UWB pulse [2]–[4]. However, these methods use more than just the TOA information; they process various features of the underlying UWB pulse at an extra computational cost. Furthermore, NLOS identification always has uncertainties due to complexity of the propagation environments. Monitoring the variance of the ranging error within a sliding time window gives information on changes in the LOS condition [5], but these methods are not recursive. The NLOS condition can also be augmented to the filter state and estimated only through the TOA measurements using e.g. particle filters [6]–[9], or interacting multiple model filters [10], [11]. However, the computational costs of these methods increase rapidly as the number of UWB beacons increases.

The sensitivity of the EKF to large measurement errors is due to the underlying assumption of normally distributed measurement errors. The normal distribution is light-tailed, that is, errors of several sigmas are very improbable. Therefore, when an outlier measurement is encountered, it overwhelms the motion model's state prediction, which causes large estimation errors. One approach for robustifying TOA positioning against NLOS is to assume a heavy-tailed measurement noise distribution [12].

This article proposes applying the recursive skew-t variational Bayes filter (STVBF) [13] to TOA positioning in mixed LOS/NLOS condition fused with inertial measurement based pedestrian dead reckoning (PDR). To our knowledge this is the first work to implement time-series TOA positioning and sensor fusion with PDR using the skew-t model and real-world data. The skew t-distribution is also applied to range-based positioning in [14], but that work considers static estimation with the expectation–maximization algorithm. Neither [13] nor [14] validate the models and methods with real-data tests.

The proposed filter assumes that the measurement noise follows the skew t-distribution, and approximates the posterior distribution using a variational Bayes (VB) approximation [15, Ch. 10]. The skew t-distribution produces occasional large outliers and is skewed, allowing asymmetric distribution of errors around the mean value. Furthermore, the skew t-distribution allows some negative outliers too, thus accounting also for outliers due to e.g. false pulse detections. The STVBF does not require a separate NLOS identification method, but relies only on comparison with the other measurements and the filter prior distribution.

The structure of this paper is the following: First, the skew t-distribution and the STVBF algorithm are explained in detail. Second, it is explained how the STVBF can be used for positioning with UWB and inertial measurements. Finally, the proposed method is tested in indoor environments with real UWB data that are fused with an inertial measurement based motion model. The proposed algorithm is shown to outperform the conventional EKF algorithm at the cost of a moderately increased computational burden.

II. Skew t-distribution for TOA measurement

This paper assumes that the TOA measurements from different beacons are independently univariate skew-t-distributed. This definition of the univariate skew t-distribution was originally proposed in multivariate forms by [16]–[18]. The univariate skew t-distribution is parametrized by its location parameter $\mu \in \mathbb{R}$, spread parameter $\sigma \in \mathbb{R}^+$, shape parameter $\delta \in \mathbb{R}$ and degrees of freedom $\nu \in \mathbb{R}^+$, and has a probability density function (PDF)

$$\mathrm{ST}(z; \mu, \sigma^2, \delta, \nu) = 2\,\mathrm{t}(z; \mu, \delta^2 + \sigma^2, \nu)\,\mathrm{T}(\widetilde{z}; 0, 1, \nu + 1), \tag{1}$$

H. Nurminen receives funding from Tampere University of Technology Graduate School, Finnish Doctoral Programme in Computational Sciences, the Foundation of Nokia Corporation, and Tekniikan edistämissäätiö.
T. Ardeshiri receives funding from Swedish research council (VR), project ETT (621-2010-4301)

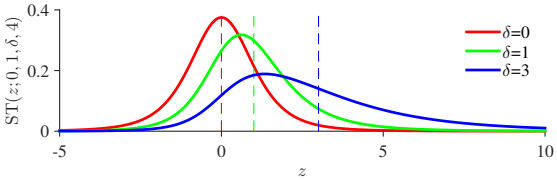


Figure 1. The PDF of $\mathrm{ST}(z; 0, 1, \delta, 4)$ for different values of δ. Dashed lines show the mean values. $\delta = 0$ gives Student's t-distribution, and increasing the absolute value of δ increases skewness.

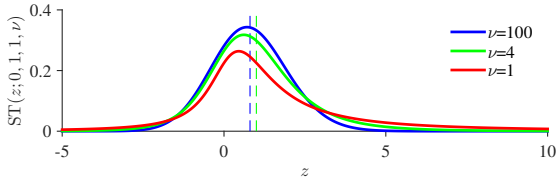


Figure 2. The PDF of $\mathrm{ST}(z; 0, 1, 1, \nu)$ for different values of ν. Dashed lines show the mean values for $\nu > 1$. $\nu = \infty$ gives the skew normal distribution, and decreasing ν increases heavy-tailedness. As for the Student's t-distribution the mean does not exist for $\nu \leq 1$.

where

$$\mathrm{t}(z; \mu, \sigma^2, \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1 + \frac{(z-\mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}} \tag{2}$$

is the PDF of Student's t-distribution, $\Gamma(\cdot)$ is the gamma function, $\widetilde{z}$ is defined by

$$\widetilde{z} \triangleq \frac{(z-\mu)\delta}{\sigma}\left(\frac{\nu+1}{\nu(\delta^2+\sigma^2)+(z-\mu)^2}\right)^{\frac{1}{2}}, \tag{3}$$

and $\mathrm{T}(\cdot; 0, 1, \nu)$ is the cumulative distribution function (CDF) of Student's t-distribution with degrees of freedom ν and scale 1.

In Fig. 1 the PDF of $\mathrm{ST}(z; 0, 1, \delta, 4)$ is plotted for three values of δ, and in Fig. 2 the PDF of $\mathrm{ST}(z; 0, 1, 1, \nu)$ is plotted for three values of ν. Figures 1 and 2 also show the mean values of the skew t-distributions to demonstrate the fact that the mean is not smaller than than the mode that is again not smaller than the location parameter μ. With $\nu \leq 1$ the mean does not exist. The expressions for the first two moments of the univariate skew t-distribution with the parametrization (1) can be found in [19], [20]. The sign of δ is the sign of skewness, and the skew t-distribution becomes the normal distribution when $\nu \to \infty$ and $\delta \to 0$,

A useful representation of the skew t-distribution is the hierarchical representation [21]

$$z|u, \lambda \sim \mathcal{N}(\mu + \delta u, \lambda^{-1}\sigma^2), \tag{4a}$$
$$u|\lambda \sim \mathcal{N}_+(0, \lambda^{-1}), \tag{4b}$$
$$\lambda \sim \mathcal{G}\left(\frac{\nu}{2}, \frac{\nu}{2}\right), \tag{4c}$$

where u and λ are scalar random variables and $\mathcal{N}_+(m, s^2)$ denotes the truncated normal distribution with closed positive orthant as support, location parameter m, and scale-parameter s. Furthermore, $\mathcal{G}(\alpha, \beta)$ denotes the gamma distribution with shape parameter α and rate parameter β.

The hierarchical representation (4) introduces two latent variables, u and λ. The representation shows that a skew-t-distributed random variable is a sum of a conditionally normal random variable and an independent conditionally truncated-normal random variable where the conditioning is on the gamma-distributed factor λ. Roughly speaking, u being always positive produces the asymmetric deviation, and small λ realisations generate outliers that may be several standard deviations away from the mean, which produces the heavy-tailedness.

III. Variational Bayes for skew-t measurement noise

The formulas of a filter based on a VB approximation for state-space models with skew-t-distributed measurement noise are presented in [13], and the derivations are given in [22]. The filter computes an approximation of the posterior distribution of x_k for the dynamical model

$$x_{k+1} = A_k x_k + w_k, \qquad w_k \overset{\text{iid}}{\sim} \mathcal{N}(0, Q), \tag{5a}$$
$$y_k = C_k x_k + e_k, \qquad [e_k]_i \overset{\text{iid}}{\sim} \mathrm{ST}(\mu_i, \sigma^2, \delta, \nu) \tag{5b}$$

where $A_k \in \mathbb{R}^{n_x \times n_x}$ is the state transition matrix; $x_k \in \mathbb{R}^{n_x}$ is the state to be estimated with the prior distribution

$$p(x_1) = \mathcal{N}(x_1; x_{1|0}, P_{1|0}), \tag{6}$$

where the subscript "$a|b$" means "at time a using measurements up to time b" and $\mathcal{N}(\cdot; m, S)$ denotes a (multivariate) normal PDF with mean m and covariance matrix S; $y_k \in \mathbb{R}^{n_y}$ are the measurements, and the elements of the measurement noise vector $e_k \in \mathbb{R}^{n_y}$ are independently skew-t-distributed; $\mu \in \mathbb{R}^{n_y}$ is the vector of location parameters of the measurement noise distribution; $\sigma \in \mathbb{R}^+$ is the spread parameter; $\delta \in \mathbb{R}$ is the shape parameter; $\nu \in \mathbb{R}^+$ is the degrees of freedom; and $C_k \in \mathbb{R}^{n_y \times n_x}$ is the measurement model matrix.

The filtering posterior $p(x_k|y_{1:k})$ is not analytically tractable. However, when the predicted filtering density $p(x_k|y_{1:k-1})$ is normal, the hierarchical representation of the likelihood in (4) enables the VB approximation of the full filtering posterior given by

$$\begin{aligned} p(x_k, u_k, \Lambda_k|y_{1:k}) \propto\ & p(x_k|y_{1:k-1})p(y_k|x_k, u_k, \Lambda_k) \\ & \times p(u_k|\Lambda_k)p(\Lambda_k). \end{aligned} \tag{7}$$

In (7), $u_k \in \mathbb{R}^{n_y}$ is a vector and $\Lambda_k \in \mathbb{R}^{n_y \times n_y}$ is a diagonal matrix whose diagonal elements $[\Lambda_k]_{ii}$ have independent gamma-priors of (4c). The VB method [15, Ch. 10] finds the approximation for the joint posterior in the form

$$p(x_k, u_k, \Lambda_k|y_{1:k}) \approx q_x(x_k)q_u(u_k)q_\Lambda(\Lambda_k) \tag{8}$$

from which an approximation of the marginal $p(x_k|y_{1:k})$ can be obtained. The approximate distributions $q_x(x_k)$, $q_u(u_k)$ and $q_\Lambda(\Lambda_k)$ are chosen such that they minimize the Kullback-Leibler divergence (KLD) [23] of the true posterior from the factorized approximation on the right hand side of (8):

$$\begin{aligned} \hat{q}_x, \hat{q}_u, \hat{q}_\Lambda = & \operatorname*{argmin}_{q_x, q_u, q_\Lambda} \\ & D_{\mathrm{KL}}(q_x(x_k)q_u(u_k)q_\Lambda(\Lambda_k)||p(x_k, u_k, \Lambda_k|y_{1:k})) \end{aligned} \tag{9}$$

where $D_{\mathrm{KL}}(q(\cdot)||p(\cdot)) \triangleq \int q(x) \log \frac{q(x)}{p(x)}\, \mathrm{d}x$ is the KLD. The VB method results in an iterative algorithm presented in Table I. With the skew-t measurement noise the approximative marginal posterior of x_k is a normal distribution with the parametrization $q_x(x_k) = \mathcal{N}(x_k; x_{k|k}, P_{k|k})$, where the parameters $x_{k|k}$ and $P_{k|k}$ are the output of the STVBF algorithm of Table I. Because the algorithm uses a normal prior and

Table I. SKEW-t VARIATIONAL BAYES FILTER

Inputs: A_k, C_k, Q, μ, σ^2, δ, ν, $x_{1|0}$, $P_{1|0}$, $y_{1:K}$ and N_{VB}
for k = 1 to K **do**
  initialization
  $\overline{\Lambda_k} \leftarrow I_{n_y \times n_y}$
  $\overline{u_k} \leftarrow 0_{n_y \times 1}$
  $\Upsilon \leftarrow 0_{n_y \times n_y}$
  $K_u \leftarrow \delta(\delta^2 + \sigma^2)^{-1} \cdot I_{n_y \times n_y}$
  for j = 1 to N_{VB} **do**
    update $q_x(x_k) = \mathcal{N}(x_k; x_{k|k}, P_{k|k})$ *given* $q_u(u_k)$ *and* $q_\Lambda(\Lambda_k)$
    $K_x \leftarrow P_{k|k-1} C_k^{\mathrm{T}} (C_k P_{k|k-1} C_k^{\mathrm{T}} + \sigma^2 \overline{\Lambda_k}^{-1})^{-1}$
    $x_{k|k} \leftarrow x_{k|k-1} + K_x(y_k - \mu - C_k x_{k|k-1} - \delta\overline{u_k})$
    $P_{k|k} \leftarrow (I - K_x C_k) P_{k|k-1}$
    update $q_u(u_k) = \mathcal{N}_+(u_k; u_{k|k}, U_{k|k})$ *given* $q_x(x_k)$ *and* $q_\Lambda(\Lambda_k)$
    $\widetilde{u}_k \leftarrow y_k - \mu - C_k x_{k|k}$
    $u_{k|k} \leftarrow K_u \widetilde{u}_k$
    $U_{k|k} \leftarrow (I - \delta K_u)\overline{\Lambda_k}^{-1}$
    for i = 1 to n_y **do**
      $\xi \leftarrow \frac{[u_{k|k}]_i}{[U_{k|k}]_{ii}}$
      $\epsilon \leftarrow \frac{\phi(\xi)}{\Phi(\xi)}$, ϕ is the PDF and Φ the CDF of $\mathcal{N}(0,1)$
      If $\Phi(\xi)$ underflows to zero, set $[\overline{u_k}]_i \leftarrow 0$ and $\Upsilon_{ii} \leftarrow 0$
      $[\overline{u_k}]_i \leftarrow [u_{k|k}]_i + \epsilon\sqrt{[U_{k|k}]_{ii}}$
      $\Upsilon_{ii} \leftarrow [U_{k|k}]_{ii} \cdot (1 - \xi\epsilon - \epsilon^2) + [\overline{u_k}]_i^2$
    end for
    update $q_\Lambda(\Lambda_k) = \prod_{i=1}^{n_y} \mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu}{2} + 1, \frac{\nu + [\Psi_k]_{ii}}{2}\right)$ *given* $q_u(u_k)$ *and* $q_x(x_k)$
    $\Psi \leftarrow \frac{1}{\sigma^2}(\widetilde{u}_k \widetilde{u}_k^{\mathrm{T}} + C_k P_{k|k} C_k^{\mathrm{T}}) + (\frac{\delta^2}{\sigma^2} + 1)\Upsilon - \frac{\delta^2}{\sigma^2}(\overline{u_k}\widetilde{u}_k^{\mathrm{T}} + \widetilde{u}_k \overline{u_k}^{\mathrm{T}})$
    $[\overline{\Lambda_k}]_{ii} \leftarrow \frac{\nu+2}{\nu + \Psi_{ii}}$
  end for
  predict $q_x(x_{k+1})$
  $x_{k+1|k} \leftarrow A_k x_{k|k}$
  $P_{k+1|k} \leftarrow A_k P_{k|k} A_k^{\mathrm{T}} + Q$
end for
Outputs: $x_{k|k}$ **and** $P_{k|k}$ **for** $k = 1 \cdots K$

results in a normal posterior approximation, it provides a Kalman filter -type recursive solution to the inference problem outlined in (5).

IV. FUSION OF UWB RANGING AND PDR

This section explains how the measurements of UWB ranging and inertial measurement based PDR can be used in the general filtering framework of Table I to produce a fused estimate for the user's position. Inertial measurement based PDR produces a continuous and relatively accurate estimate of the change in the user's position. However, due to the need of initial position estimate and the sensor drift it has to be complemented by some absolute position information such as UWB. Correspondingly, the UWB ranging has high absolute accuracy, but the precise motion model provided by the PDR helps in detecting NLOS outliers.

A. Motion model from PDR

The proposed positioning system uses a PDR solution based on inertial sensors, i.e. three-axis accelerometers and gyroscopes. The used PDR solution's output is footstep detection and measurements of the horizontal-plane heading change ψ_k.

The footsteps are detected using the accelerometer output's norm [24]. The accelerometers also show the direction of gravity which gives the horizontal plane, and the user's heading change during a footstep is then estimated from the gyroscope's angular velocity measurements projected to the horizontal plane [25].

The sensor fusion algorithm uses a linear motion model based on this PDR solution. The method is proposed for 2-dimensional positioning in [26], and in this paper the altitude of the user equipment is also in the state. The state vector is $x_k = \left[\begin{smallmatrix} l_k \\ s_k \end{smallmatrix}\right]$, where $l_k \in \mathbb{R}^3$ is the user equipment's 3-dimensional position and $s_k \in \mathbb{R}^2$ is the horizontal footstep vector. At a time update, the step vector is rotated by the heading change ψ_k and affected by process noise, and the x and y coordinates are updated by the step vector. The altitude process is modeled as a random walk. The state transition matrix and the process noise covariance matrix in (5a) are thus

$$A_k = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \cos\overline{\psi_k} & -\sin\overline{\psi_k} \\ 0 & 0 & 0 & \sin\overline{\psi_k} & \cos\overline{\psi_k} \end{bmatrix}, \tag{10}$$

$$Q = \mathrm{Diag}(0, 0, \sigma_h^2, \sigma_s^2, \sigma_s^2), \tag{11}$$

respectively [26]. Here $\overline{\psi_k}$ is the heading change estimate given by the PDR solution, σ_s^2 is a variance parameter that models the uncertainty of the PDR measurements and the step length's process noise, and σ_h^2 is the altitude's process noise variance. Because this motion model is linear, it is free from linearization errors that occur when the velocity is unknown, but the model cannot model the uncertainties of the PDR as flexibly as some conventional non-linear models [26].

If PDR is not used, positioning can be done with UWB alone by adopting a less informative motion model. This can be a white noise based kinematic model [27, Ch. 6], for example.

B. Measurement model from UWB ranging

A TOA measurement gives the distance traveled by the radio wave between the UWB beacon and the user. To account for occasional large positive measurement errors due to non-line-of-sight conditions, the measurement noise of all sensors is assumed to be identically and independently skew-t-distributed:

$$[y_k]_i = \|l_k - b_i\| + [e_k]_i, \;\; [e_k]_i \overset{\text{iid}}{\sim} \mathrm{ST}(\mu, \sigma^2, \delta, \nu), \tag{12}$$

where $y_k \in \mathbb{R}^{n_y}$ is the TOA-based distance vector, l_k is the user position, $b_i \in \mathbb{R}^3$ is the 3-dimensional position of the ith UWB beacon, and e_k is measurement noise.

The model (5) and the STVBF assume a linear measurement model, while (12) is nonlinear. This limitation can be overcome by linearizing the model at the prior mean $x_{k|k-1} = \left[\begin{smallmatrix} l_{k|k-1} \\ s_{k|k-1} \end{smallmatrix}\right]$, giving

$$y_k = C_k x_k + e_k, \tag{13}$$

where the ith row of C_k is

$$[C_k]_i = \begin{bmatrix} \frac{(l_{k|k-1} - b_i)^{\mathrm{T}}}{\|l_{k|k-1} - b_i\|} & \mathrm{O}_{1\times 2} \end{bmatrix}, \tag{14}$$

and

$$[e_k]_i \sim \mathrm{ST}\left(\mu + \left\|l_{k|k-1} - b_i\right\| - [C_k]_i x_{k|k-1}, \sigma^2, \delta, \nu\right). \tag{15}$$

Hence, the approximative measurement equations (13) and (15) are in the form of (5b).

Figure 3. A SpoonPhone and a BeSpoon UWB tag.

V. Real-data tests

Real UWB measurements were collected to evaluate the proposed method's performance. The measurement equipment was a Spoonphone smartphone [28] with Android 4.2 operating system and UWB channel 2 pulse radio (3993.6 MHz, 500 MHz bandwidth), and six BeSpoon UWB tags. Fig. 3 shows a picture of the equipment. The system uses two-way TOA ranging, and so it does not depend on clock synchronization. The used inertial sensors were the Spoonphone's built-in sensors. The UWB measurement update was done with 2 Hz frequency, and the error computation for obtaining the error statistics was also done with 2 Hz frequency. Five test tracks were collected in a laboratory environment and three test tracks in a real indoor environment of a university campus. The algorithms were computed with MATLAB.

The estimation algorithm for the skew-t noise is the STVBF in Table I. The parameters of the skew-t noise distribution, $\{\mu, \sigma^2, \delta, \nu\}$, were optimized to minimize the average root-mean-square error (RMSE) of the 2-dimensional positioning error for all the test tracks. The parameter optimization was done using 30 VB iterations (N_{VB}=30), which should ensure convergence [13]. The optimized parameter values are in Table II. The same data were used for both parameter calibration and positioning tests to obtain a fair comparison of the optimal setups of each filtering algorithm. This eliminates the effect of possible differences in calibration and positioning data.

The used motion model parameter values for 2 Hz updating frequency are $\sigma_h = 5 \cdot 10^{-4}$ m and $\sigma_s = 3 \cdot 10^{-2}$ m. Due to the small value of σ_h floor changes need a separate model, but this is out of the scope of this paper.

The STVBF is compared with the EKF that is based on the normal measurement noise model

$$[y_k]_i = \|l_k - b_i\| + e_k, \;\; [e_k]_i \overset{\text{iid}}{\sim} \mathcal{N}(\tau, \rho^2). \qquad (16)$$

The noise parameters τ and ρ^2 were also optimized for the test track set, and the optimized parameter values are in Table II.

Fig. 5 shows the average RMSE of the STVBF as a function of the number of VB iterations, and its comparison with the EKF's RMSE. Note that the methods use different noise parameter values according to Table II. The figure shows that 2 VB iterations already outperforms the best possible EKF estimate, and 4 VB iterations is enough to achieve the converged state's average RMSE in this scenario.

Table II. The used parameter values obtained by optimizing with respect to the test tracks' average RMSE

Filter	STVBF				EKF	
Param.	μ (m)	σ (m)	δ (m)	ν	τ (m)	ρ (m)
Value	-0.1	0.3	0.6	4	1.3	1.6

Table III. The error statistics in the laboratory environment with optimized noise parameters

Filter	RMSE (m)	mean (m)	median (m)	95 % quant. (m)	95 % cons. (%)	running time
EKF	1.36	1.16	1.02	2.58	94	1
STVBF	0.56	0.46	0.38	1.16	91	3

A. Laboratory measurements with high-accuracy reference

One data set was measured in a laboratory environment to obtain high-accuracy reference position[1]. To acquire a realistic simulation of an office-like indoor environment, a whiteboard was placed in the middle of the testing area, and lab personnel moved about in the area to simulate passersby. These obstacles caused occasional NLOS conditions for some UWB beacons. Fig. 4 depicts the testing area and the used test tracks. Fig. 6 compares the distance measurements of one track for one UWB beacon with the reference distance given by the laboratory equipment, and shows that occasional NLOS measurements are visible as positive peaks in the measurement error. Fig. 7 shows that the histogram distribution of these errors is positively skewed.

The error statistics of the STVBF with 4 VB iterations and the EKF are in Table III. The shown accuracy measures are the average RMSE of the five tracks, mean error, median error, empirical 95 % quantile of errors, 95 % consistency, and relative running time averaged over the five test tracks. Consistency is the percentage of time instants when the filter was consistent with respect to the Gaussian NEES (normalized estimation error squared) consistency test [27, Ch. 5.4.2]; the closer the consistency is to 95 %, the more accurately the filter reports the estimate's uncertainty. The results show that both the filters are fairly consistent, but the STVBF has significantly better accuracy; the STVBF's errors are less than 50 % of the EKF's errors. The computational burden of our STVBF implementation is only three times that of the EKF's, which we consider reasonable for the improvement in performance.

B. Measurements in campus environment

Three test tracks were measured in real indoor environment in a campus building of Tampere University of Technology to ensure that the conclusions based on the laboratory measurements also hold in realistic indoor environment. In these tests the reference position of the user was obtained by matching certain reference points of the track with an indoor map and interpolating the positions between these reference points. Five UWB beacons were used, and the beacon positions were also obtained by matching with a map. These reference positions are not as accurate as those in section V-A, but the accuracy is in the order of tens of centimeters, which is more accurate than the expected positioning accuracy. The maps of the test areas are shown in Figures 8, 9, and 10 for tracks 1, 2, and 3, respectively.

The beacon positions at the tracks 1 and 2 were chosen such that NLOS conditions occur every now and then. Track 1 consists of walking in corridors and turning at two corridor

[1]High accuracy reference measurements are provided through the use of the Vicon real-time tracking system courtesy of the UAS Technologies Lab, Artificial Intelligence and Integrated Computer Systems Division (AIICS) at the Department of Computer and Information Science (IDA). http://www.ida.liu.se/divisions/aiics/aiicssite/index.en.shtml

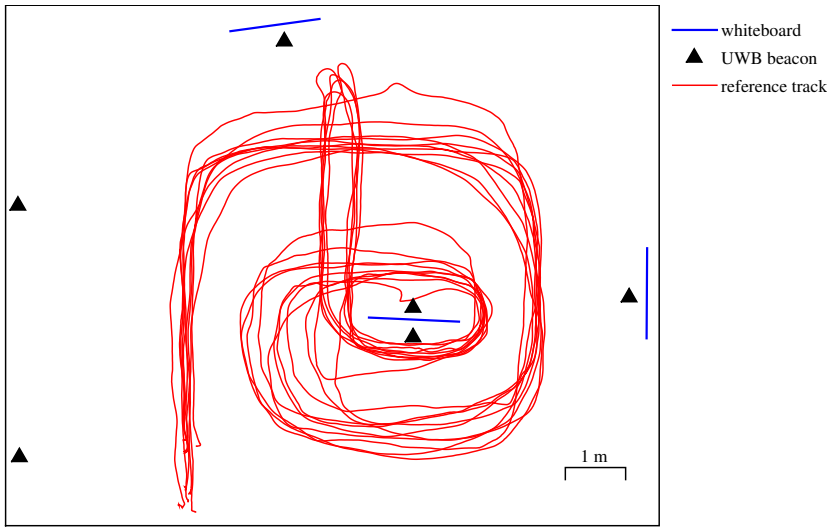


Figure 4. Layout of the testing area. NLOS condition is created by the whiteboard at the center of the testing area.

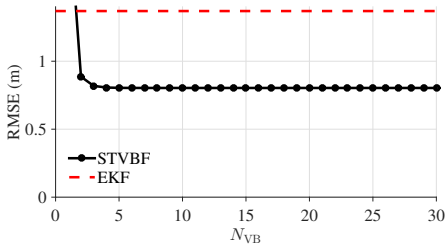


Figure 5. STVBF's average RMSE for all the test tracks as a function of the number of VB iterations compared with the EKF's average RMSE. 4 VB iterations is enough for the STVBF in the PDR & UWB-TOA positioning.

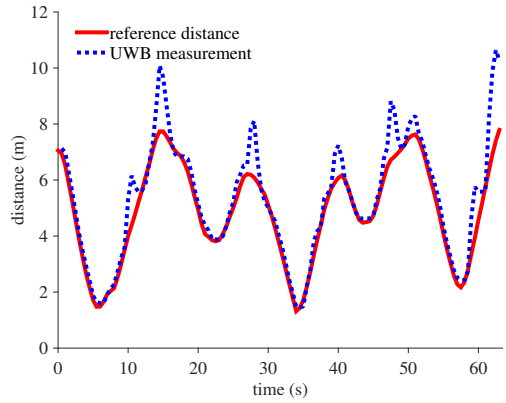


Figure 6. Distance measurement of one laboratory track for one UWB beacon compared with the true distance.

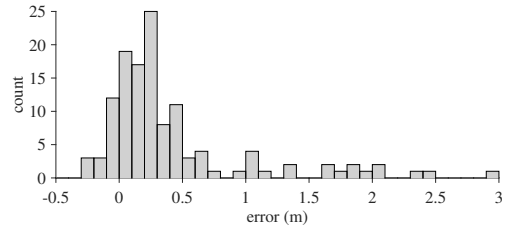


Figure 7. Error histogram of the measurements depicted by Fig. 6.

junctions. Track 2 consists of walking in a corridor and visiting an office room. Track 3 contains only walking in a straight corridor, and the time under NLOS condition was minimized in this track to evaluate the algorithms' performance also in pure LOS condition.

The error statistics of the positioning tests in the campus environment are shown in Table IV. In tracks 1 and 2 the STVBF clearly outperforms the EKF in positioning accuracy. In track 3 the performances of the STVBF and EKF are much closer, which is expectable because track 3 mainly contains LOS measurements. However, the STVBF is still slightly more accurate that the EKF in track 3, which might partly be explained by occasional NLOS caused by the body of the person conducting the experiments. The test results indicate that the STVBF outperforms the EKF in mixed LOS/NLOS condition, and performs equally well or better in LOS condition.

VI. Conclusions

The use of the skew-t variational Bayes filter (STVBF) in indoor positioning with inertial sensors and UWB ranging in mixed LOS/NLOS conditions is proposed. The proposed filter is more robust against outliers than the conventional EKF that is based on assumed normality of the measurement noise. Due to the assumption of positively skewed measurement noise, the proposed algorithm is also capable of accounting for the fact that the NLOS phenomena typically cause positive outlier

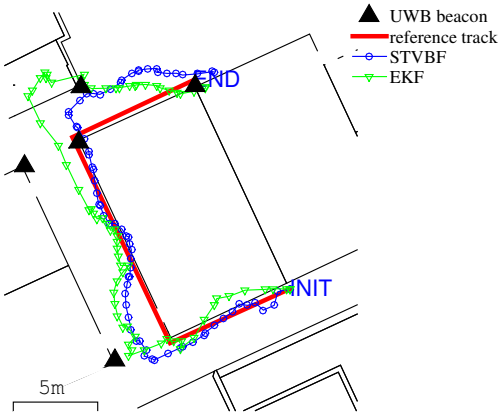


Figure 8. Campus test track 1 consists of corridors and turns at corridor junctions.

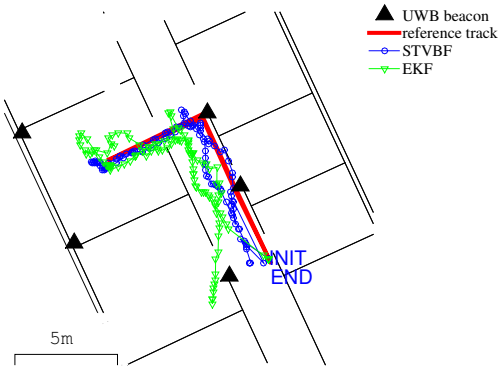


Figure 9. Campus test track 2 consists of a visit in an office room.

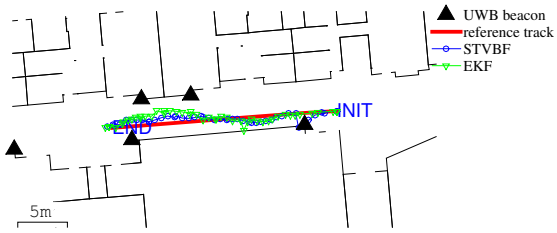


Figure 10. In campus test track 3 the tags are in LOS condition most of the time.

Table IV. THE ERROR STATISTICS IN THE CAMPUS ENVIRONMENT WITH OPTIMIZED NOISE PARAMETERS

Track	Filter	RMSE (m)	mean (m)	median (m)	95 % quant. (m)	95 % cons. (%)
1	EKF	1.91	1.57	1.20	4.02	83
	STVBF	1.23	1.11	1.04	1.94	41
2	EKF	1.41	1.24	1.21	2.79	95
	STVBF	0.81	0.70	0.63	1.45	59
3	EKF	0.83	0.72	0.67	1.43	98
	STVBF	0.74	0.68	0.73	1.07	55

measurements more frequently than negative ones. Real-data tests showed that STVBF enables dramatic improvement in positioning accuracy compared to the EKF with the computational burden of about three EKFs.

REFERENCES

[1] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, July 2005.

[2] I. Güvenç, C.-C. Chong, F. Watanabe, and H. Inamura, "NLOS identification and weighted least-squares localization for UWB systems using multipath channel statistics," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, January 2008.

[3] J. Khodjaev, Y. Park, and A. S. Malik, "Survey of NLOS identification and error mitigation problems in UWB-based positioning algorithms for dense environments," *Annals of Telecommunications*, vol. 65, no. 5–6, pp. 301–311, June 2010.

[4] S. Marano, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1026–1035, September 2010.

[5] C.-D. Wann and C.-S. Hsueh, "NLOS mitigation with biased Kalman filters for range estimation in UWB systems," in *TENCON 2007 - 2007 IEEE Region 10 Conference*, October 2007, pp. 1–4.

[6] B. Denis, L. Ouvry, B. Uguen, and F. Tchoffo-Talom, "Advanced Bayesian filtering techniques for UWB tracking systems in indoor environments," in *2005 IEEE International Conference on Ultra-Wideband (ICU 2005)*, September 2005, pp. 638–643.

[7] J. González, J. Blanco, C. Galindo, A. Ortiz-de Galisteo, J. Fernández-Madrigal, F. Moreno, and J. Martínez, "Mobile robot localization based on ultra-wide-band ranging: A particle filter approach," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 496–507, May 2009.

[8] L. Chen, R. Piché, H. Kuusniemi, and R. Chen, "Adaptive mobile tracking in unknown non-line-of-sight conditions with application to digital TV networks," *EURASIP Journal on Advances in Signal Processing*, vol. 22, no. 1, p. 10 pp., February 2014.

[9] E. Özkan, F. Lindsten, C. Fritsche, and F. Gustafsson, "Recursive maximum likelihood identification of jump Markov nonlinear systems," *IEEE Transactions on Signal Processing*, vol. 63, no. 3, pp. 754–765, February 2015.

[10] J.-F. Liao and B.-S. Chen, "Robust mobile location estimator with NLOS mitigation using Interacting Multiple Model algorithm," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3002–3006, November 2006.

[11] C. Fritsche, U. Hammes, A. Klein, and A. M. Zoubir, "Robust mobile terminal tracking in NLOS environments using interacting multiple model algorithm," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, April 2009, pp. 3049–3052.

[12] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, July 2005.

[13] H. Nurminen, T. Ardeshiri, R. Piché, and F. Gustafsson, "Robust inference for state-space models with skewed measurement noise," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1898–1902, November 2015.

[14] P. Müller and R. Piché, "Statistical trilateration with skew-t errors," in *International Conference on Localization and GNSS (ICL-GNSS 2015)*, June 2015.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[16] M. D. Branco and D. K. Dey, "A general class of multivariate skew-elliptical distributions," *Journal of Multivariate Analysis*, vol. 79, no. 1, pp. 99–113, October 2001.

[17] A. Azzalini and A. Capitanio, "Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t-distribution," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 65, no. 2, pp. 367–389, 2003.

[18] A. K. Gupta, "Multivariate skew t-distribution," *Statistics*, vol. 37, no. 4, pp. 359–363, 2003.

[19] S. K. Sahu, D. K. Dey, and M. D. Branco, "A new class of multivariate skew distributions with applications to Bayesian regression models," *Canadian Journal of Statistics*, vol. 31, no. 2, pp. 129–150, 2003.

[20] ——, "Erratum: A new class of multivariate skew distributions with applications to Bayesian regression models," *Canadian Journal of Statistics*, vol. 37, no. 2, pp. 301–302, 2009.

[21] T.-I. Lin, "Robust mixture modeling using multivariate skew t distributions," *Statistics and Computing*, vol. 20, pp. 343–356, 2010.

[22] T. Ardeshiri, H. Nurminen, R. Piché, and F. Gustafsson, "Variational iterations for filtering and smoothing with skew-t measurement noise," Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, Tech. Rep. LiTH-ISY-R-3076, March 2015. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-115741

[23] T. M. Cover and J. Thomas, *Elements of Information Theory*. John Wiley and Sons, 2006.

[24] H. Leppäkoski, J. Collin, and J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals," *Journal of Signal Processing Systems*, vol. 71, no. 3, pp. 287–296, June 2013.

[25] J. Collin, O. Mezentsev, and G. Lachapelle, "Indoor positioning system using accelerometry and high accuracy heading sensors," in *GPS/GNSS 2003 Conference (Session C3)*, Portland, OR, September 2003.

[26] M. Raitoharju, H. Nurminen, and R. Piché, "Kalman filter with a linear state model for PDR+WLAN positioning and its application to assisting a particle filter," *EURASIP Journal on Advances in Signal Processing*, no. 33, April 2015.

[27] Y. Bar-Shalom, R. X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation, Theory Algorithms and Software*. John Wiley & Sons, 2001.

[28] The SpoonPhone. [Online]. Available: http://spoonphone.com/en/

7 UNPUBLISHED MANUSCRIPT

Henri Nurminen, Tohid Ardeshiri, Robert Piché, and Fredrik Gustafsson: Skew-t filter and smoother with improved covariance matrix approximation.

Skew-t Filter and Smoother with Improved Covariance Matrix Approximation

Henri Nurminen, *Student Member, IEEE*, Tohid Ardeshiri, Robert Piché, *Senior Member, IEEE*, and Fredrik Gustafsson, *Fellow, IEEE*

***Abstract*—Filtering and smoothing algorithms for linear discrete-time state-space models with skew-t-distributed measurement noise are proposed. The algorithms use a variational Bayes based posterior approximation where the location and skewness variables are coupled to reduce the error caused by the variational approximation. Although the variational update is done suboptimally using an expectation propagation algorithm, our simulations show that the proposed method gives a more accurate approximation of the posterior covariance matrix than an earlier proposed variational algorithm. Consequently, the novel filter and smoother outperform the earlier proposed robust filter and smoother and other existing low-complexity alternatives in accuracy and speed. We present both simulations and tests based on real-world navigation data, in particular GPS data in an urban area, to demonstrate the performance of the novel methods. Moreover, the extension of the proposed algorithms to cover the case where the distribution of the measurement noise is multivariate skew-t is outlined. Finally, the paper presents a study of theoretical performance bounds for the proposed algorithms.**

***Index Terms*— skew t, t-distribution, robust filtering, Kalman filter, RTS smoother, variational Bayes, expectation propagation, truncated normal distribution, Cramér–Rao lower bound**

I. Introduction

Asymmetric and heavy-tailed noise processes are present in many inference problems. In radio signal based distance estimation [1]–[3], for example, obstacles cause large positive errors that dominate over symmetrically distributed errors from other sources [4]. An example of this is the error histogram of time-of-flight in distance measurements collected in an indoor environment given in Fig. 1. The asymmetric distributions cannot be predicted by the normal or t-distributions that are equivalent in second order moments, because normal and t-distributions are symmetric distributions. The skew t-distribution [5]–[7] is a generalization of the t-distribution that has the modeling flexibility to capture both skewness and heavy-tailedness of such noise processes. To illustrate this, Fig. 2 shows the contours of the likelihood function for three range measurements where some of the measurements include large positive errors. In this example, skew-t, t, and normal measurement noise models are compared. Due to the additional modeling flexibility, the skew-t based likelihood provides a more apposite spread of the probability mass than the normal and t based likelihoods.

H. Nurminen and R. Piché are with the Laboratory of Automation and Hydraulics, Tampere University of Technology (TUT), PO Box 692, 33101 Tampere, Finland (e-mails: henri.nurminen@tut.fi, robert.piche@tut.fi). H. Nurminen receives funding from TUT Graduate School, the Foundation of Nokia Corporation, Tekniikan edistämissäätiö, and Emil Aaltonen Foundation.

T. Ardeshiri was with the Division of Automatic Control, Department of Electrical Engineering, Linköping University, 58183, Linköping, Sweden and received funding from Swedish research council (VR), project scalable Kalman filters for this work. T. Ardeshiri is currently with the Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, UK, (e-mail: ta417@cam.ac.uk).

F. Gustafsson is with the Division of Automatic Control, Department of Electrical Engineering, Linköping University, 58183 Linköping, Sweden, (e-mail: fredrik@isy.liu.se).

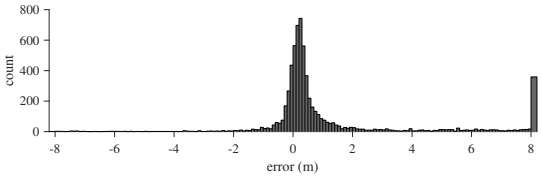


Fig. 1. The error histogram in an ultra-wideband (UWB) ranging experiment described in [8] shows positive skewness. The edge bars show the errors outside the figure limits.

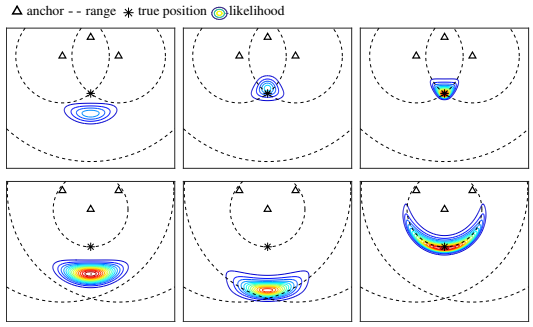


Fig. 2. The likelihood contours of distance measurements from three known anchors for the normal (left), t (middle) and skew-t (right) measurement noise models. The t and skew-t based likelihoods handle one large positive error (upper row), while only the skew-t model handles the two large positive errors (bottom row) due to its asymmetry. The measurement model parameters are selected such that the degrees-of-freedom values and the first two moments coincide.

The applications of the skew distributions are not limited to radio signal based localization. In biostatistics skewed distributions are used as a modeling tool for handling heterogeneous data involving asymmetric behaviors across subpopulations [9]. In psychiatric research skew normal distribution is used to model asymmetric data [10]. Further, in economics skew normal and skew t-distributions are used as models for describing claims in property-liability insurance [11]. More examples describing approaches for analysis and modeling using multivariate skew normal and skew t-distributions in econometrics and environmetrics are presented in [12].

There are various algorithms dedicated to statistical inference of time series when the data exhibit asymmetric distribution. Particle filters [13] can easily be adapted to skew noise distributions, but the computational complexity of these filters increases rapidly as the state dimension increases. A skew Kalman filter is proposed in [14], and in [15] this filter is extended to a robust scale-mixture filter using Monte Carlo integration. These solutions are based on state-space models where the measurement noise is a dependent process

with skewed marginals. The article [16] proposes filtering of independent skew measurement and process noises with the cost of increasing the filter state's dimension over time. In all the skew filters of [14]–[16], sequential processing requires numerical evaluation of multidimensional integrals. The inference problem with skew likelihood distributions can also be cast into an optimization problem; [3] proposes an approach to model the measurement noise in an ultra-wideband (UWB) based positioning problem using a tailored half-normal–half-Cauchy distribution. Skewness can also be modeled by a mixture of normal distributions (Gaussian mixtures, GM) [1]. There are many filtering algorithms for GM distributions such as Gaussian sum filter [17] and interactive multiple model (IMM) filter [18]. However, GMs have exponentially decaying tails and can thus be too sensitive to outlier measurements. Furthermore, in order to keep the computational cost of a Gaussian sum filter practicable, a mixture reduction algorithm (MRA) [19] is required, and these MRAs can be computationally expensive and involve approximations to the posterior density. Filtering and smoothing algorithms for linear discrete-time state-space models with skew-t measurement noise using a variational Bayes (VB) method are presented in [20]. The VB approach avoids the increasing filter state dimensionality and numerical integrations by finding an optimal approximation with the constraint that the state is independent of the non-dynamic latent variables; this makes analytical marginalisation straightforward. In tests with real UWB indoor localization data [8], this filter was shown to be accurate and computationally inexpensive.

This paper proposes improvements to the robust filter and smoother proposed in [20]. Analogous to [20], the measurement noise is modeled by the skew t-distribution, and the proposed filter and smoother use VB approximations of the filtering and smoothing posteriors. However, the main contributions of this paper are (1) a novel VB factorization of the posterior and showing that at highly skewed models this factorization provides major improvement in both convergence speed of the VB iterations and accuracy of the estimate and covariance matrix, (2) application of an existing expectation propagation (EP) based algorithm for approximating the statistics of a truncated multivariate normal distribution (TMND) that appears in the proposed VB algorithm, (3) a derivation for a greedy approach for a truncation ordering in the EP approximation of the TMND's moments, (4) derivation of Cramér–Rao lower bound (CRLB) for the proposed filter and smoother, and (5) the variational lower bound for the proposed VB factorization. A TMND is a multivariate normal distribution whose support is restricted (truncated) by linear constraints and that is re-normalized to integrate to unity. The aforementioned contributions improve the estimation performance of the skew-t filter and smoother by reducing the covariance underestimation common to most VB inference algorithms [21, Chapter 10]. To our knowledge, VB approximations have been applied to the skew t-distribution only in our earlier works [8], [20], and by Wand et al. [22], and Wand et al. use a VB factorization different from ours and do not consider time-series inference.

The rest of this paper is structured as follows. In Section II, the filtering and smoothing problem involving the univariate skew t-distribution is posed. In Section III a solution based on VB for the formulated problem is proposed. The proposed solution is evaluated using simulated data as well as real-world data in Sections IV and V, respectively. The essential expressions to extend the proposed filtering and smoothing algorithms to problems involving multivariate skew-t (MVST) distribution are given in Section VI. A performance bound for time series data with MVST–distributed measurement noise is derived and evaluated in simulation in Section VII. The concluding remarks are given in Section VIII.

II. Inference Problem formulation

Consider the linear and Gaussian state evolution model

$$p(x_1) = \mathcal{N}(x_1; x_{1|0}, P_{1|0}), \tag{1a}$$

$$x_{k+1} = Ax_k + w_k, \qquad w_k \overset{\text{iid}}{\sim} \mathcal{N}(0, Q), \tag{1b}$$

where $\mathcal{N}(\cdot; \mu, \Sigma)$ denotes the probability density function (PDF) of the (multivariate) normal distribution with mean μ and covariance matrix Σ; $A \in \mathbb{R}^{n_x \times n_x}$ is the state transition matrix; $x_k \in \mathbb{R}^{n_x}$ indexed by $1 \leq k \leq K$ is the state to be estimated with initial prior distribution (1a), where the subscript "$a|b$" is read "at time a using measurements up to time b"; and $w_k \in \mathbb{R}^{n_x}$ is the process noise. Further, the measurements $y_k \in \mathbb{R}^{n_y}$ are assumed to be governed by the measurement equation

$$y_k = Cx_k + e_k, \tag{2}$$

where $C \in \mathbb{R}^{n_y \times n_x}$ is the measurement matrix, and the measurement noise vector e_k is independent of the process noise, and each component of e_k follows an independent univariate skew t-distribution

$$[e_k]_i \overset{\text{independent}}{\sim} \mathrm{ST}(0, R_{ii}, \Delta_{ii}, \nu_i), \tag{3}$$

where the operator $[\cdot]_i$ gives the ith entry of the argument vector, and $[\cdot]_{ij}$ gives the (i, j) entry of its argument matrix. The measurement noise process can also be nonstationary, but for the sake of lighter notation the k subscripts on A, Q, C, R, Δ, and ν are omitted. The univariate skew t-distribution $\mathrm{ST}(\mu, \sigma^2, \delta, \nu)$ is parametrized by its location parameter $\mu \in \mathbb{R}$, spread parameter $\sigma \in \mathbb{R}_+$, shape parameter $\delta \in \mathbb{R}$ and degrees of freedom $\nu \in \mathbb{R}_+$, and has the PDF

$$\mathrm{ST}(z; \mu, \sigma^2, \delta, \nu) = 2\,\mathrm{t}(z; \mu, \sigma^2 + \delta^2, \nu)\,\mathrm{T}(\widetilde{z}; 0, 1, \nu + 1), \tag{4}$$

where

$$\mathrm{t}(z; \mu, \sigma^2, \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{(z-\mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}} \tag{5}$$

is the PDF of Student's t-distribution, $\Gamma(\cdot)$ is the gamma function, and $\widetilde{z} = \frac{(z-\mu)\delta}{\sigma}\left(\frac{\nu+1}{\nu(\sigma^2+\delta^2)+(z-\mu)^2}\right)^{\frac{1}{2}}$. Also, $\mathrm{T}(\cdot; 0, 1, \nu)$ denotes the cumulative distribution function (CDF) of Student's t-distribution with degrees of freedom ν. Expressions for the first two moments of the univariate skew t-distribution can be found in [23].

The model (3) with independent univariate skew-t-distributed measurement noise components is justified when one-dimensional noises of different sensors can be assumed to be statistically independent [20]. Extension and comparison to multivariate skew-t-distributed noise will be discussed in Section VI.

The independent univariate skew-t noise model (3) induces the hierarchical representation of the measurement likelihood

$$y_k | x_k, u_k, \Lambda_k \sim \mathcal{N}(Cx_k + \Delta u_k, \Lambda_k^{-1} R), \tag{6a}$$

$$u_k | \Lambda_k \sim \mathcal{N}_+(0, \Lambda_k^{-1}), \tag{6b}$$

$$[\Lambda_k]_{ii} \sim \mathcal{G}(\tfrac{\nu_i}{2}, \tfrac{\nu_i}{2}), \tag{6c}$$

where $R \in \mathbb{R}^{n_y \times n_y}$ is a diagonal matrix whose diagonal elements' square roots $\sqrt{R_{ii}}$ are the spread parameters of the skew t-distribution in (3); $\Delta \in \mathbb{R}^{n_y \times n_y}$ is a diagonal matrix whose diagonal elements Δ_{ii} are the shape parameters; $\nu \in \mathbb{R}_+^{n_y}$ is a vector whose elements ν_i are the degrees of freedom. Λ_k is a diagonal matrix with a priori independent random diagonal elements $[\Lambda_k]_{ii}$. Also, $\mathcal{N}_+(\mu, \Sigma)$ is the TMND with closed positive orthant as support, location parameter μ, and squared-scale matrix Σ. Furthermore, $\mathcal{G}(\alpha, \beta)$ is the gamma distribution with shape parameter α and rate parameter β.

Bayesian smoothing means finding the smoothing posterior $p(x_{1:K}, u_{1:K}, \Lambda_{1:K}|y_{1:K})$. In [20], the smoothing posterior is approximated by a factorized distribution of the form $q_{\,[20]} \triangleq q_x(x_{1:K})\, q_u(u_{1:K})\, q_\Lambda(\Lambda_{1:K})$. Subsequently, the approximate posterior distributions are computed using the VB approach. The VB approach minimizes the Kullback–Leibler divergence (KLD) $D_{\mathrm{KL}}(q||p) \triangleq \int q(x) \log \frac{q(x)}{p(x)} \mathrm{d}x$ [24] of the true posterior from the factorized approximation. That is, $D_{\mathrm{KL}}(q_{\,[20]}||p(x_{1:K}, u_{1:K}, \Lambda_{1:K}|y_{1:K}))$ is minimized in [20]. An approximate Bayesian filter update, i.e. an approximation the filtering posterior $p(x_k, u_k, \Lambda_k|y_{1:k})$ given a normal filtering prior for x_k, is then a smoother update with $K=1$.

The numerical simulations in [20] manifest covariance matrix underestimation, which is a known weakness of the VB approach [21, Chapter 10]. One of the contributions of this paper is to reduce the covariance underestimation of the filter and smoother proposed in [20] by removing independence approximations from the VB factorization. The proposed filter and smoother are presented in Section III.

III. Proposed Filter and Smoother

A. VB factorization

Using Bayes' theorem, the state evolution model (1), and the likelihood (6), the joint smoothing posterior PDF is

$$\begin{aligned} &p(x_{1:K}, u_{1:K}, \Lambda_{1:K}|y_{1:K}) \\ &\propto \mathcal{N}(x_1; x_{1|0}, P_{1|0}) \prod_{l=1}^{K-1} \mathcal{N}(x_{l+1}; Ax_l, Q) \\ &\quad \times \prod_{k=1}^{K} \mathcal{N}(y_k; Cx_k + \Delta u_k, \Lambda_k^{-1} R)\, \mathcal{N}_+(u_k; 0, \Lambda_k^{-1}) \\ &\quad \times \prod_{k=1}^{K} \prod_{i=1}^{n_y} \mathcal{G}\left([\Lambda_k]_{ii}; \frac{\nu_i}{2}, \frac{\nu_i}{2}\right). \end{aligned} \tag{7}$$

The posterior is not analytically tractable. We propose to seek an approximation in the form

$$p(x_{1:K}, u_{1:K}, \Lambda_{1:K}|y_{1:K}) \approx \hat{q}_{xu}(x_{1:K}, u_{1:K})\, \hat{q}_\Lambda(\Lambda_{1:K}), \tag{8}$$

where the factors in (8) are specified by

$$\hat{q}_{xu}, \hat{q}_\Lambda = \underset{q_{xu}, q_\Lambda}{\operatorname{argmin}} D_{\mathrm{KL}}(q_{\mathrm{N}}||p(x_{1:K}, u_{1:K}, \Lambda_{1:K}|y_{1:K})), \tag{9}$$

where $q_{\mathrm{N}} \triangleq q_{xu}(x_{1:K}, u_{1:K})\, q_\Lambda(\Lambda_{1:K})$. Hence, $x_{1:K}$ and $u_{1:K}$ are not approximated as independent as in [20] because they can be highly correlated *a posteriori* [20]. The analytical solutions for $\hat{q}_{xu}$ and $\hat{q}_\Lambda$ are obtained by cyclic iteration of

$$\log q_{xu}(x_{1:K}, u_{1:K}) \leftarrow \mathop{\mathbb{E}}_{q_\Lambda}[\log p(y_{1:K}, x_{1:K}, u_{1:K}, \Lambda_{1:K})] + c_{xu} \tag{10a}$$

$$\log q_\Lambda(\Lambda_{1:K}) \leftarrow \mathop{\mathbb{E}}_{q_{xu}}[\log p(y_{1:K}, x_{1:K}, u_{1:K}, \Lambda_{1:K})] + c_\Lambda \tag{10b}$$

where $\leftarrow$ is the assignment operator, and the expected values on the right hand sides are taken with respect to the current q_{xu} and q_Λ [21, Chapter 10] [25], [26]. Also, c_{xu} and c_Λ are constants with respect to the variables $(x_{1:K}, u_{1:K})$ and $\Lambda_{1:K}$, respectively.

The detailed derivation of the proposed smoother is given in Appendix A. The distribution q_{xu} is a $K \times (n_x + n_y)$-dimensional TMND, whose marginals for each x_k can be obtained using the Rauch–Tung–Striebel smoother (RTSS) [27]. However, the first two moments of the marginals of each x_k are required in computation of the expectation in (10b). A TMND's moments cannot be computed in closed form.

An approximative filtering update step can be derived as the smoother for a state-space model with just one time-instant, and the moments of a relatively low-dimensional TMND can be approximated using a computationally light algorithm discussed in the next subsection. By approximating the x_k-marginal of the final VB iteration's TMND $\int q_{xu}(x_k, u_k)\mathrm{d}u_k$ with a normal distribution, we obtain a recursive filtering algorithm, the skew-t filter (STF) of Algorithm 1. While the marginal $\int q_{xu}(x_k, u_k)\mathrm{d}u_k$ is not exactly normal, it is unimodal and has the whole $\mathbb{R}^{n_x}$ as support, so the normal distribution with the matching first and second moments is a standard approximation. The filter's VB iterations do not use the normality assumption, but there is no convergence proof for the VB when the moments of the TMND are approximated. However, the approximative VB iterations show better accuracy and convergence speed in the numerical simulations presented in Sections IV than the exact VB iterations with the factorization $q_{[20]}$.

However, for a smoothing problem with large K, handling a $K \times (n_x + n_y)$-dimensional TMND is impractical. Therefore, we replace the RTSS's forward filtering step with the assumed-normal filter, where each filtering distribution is approximated by a normal distribution with the approximative mean and covariance matrix of the exact TMND filtering distribution. The result of this assumed-normal filter can be fed into the standard RTSS's backward smoothing step. The obtained skew-t smoother (STS) algorithm is given in Algorithm 2.

We propose three convergence criteria for the VB iterations of the filter and smoother: small enough change in the estimate, small enough increase in the variational lower bound (practical only for the filter), and a fixed number of iterations. The computation of the variational lower bound is explained in Subsection III-C. In our tests we fix the number of VB iterations into five, since the estimation accuracy does not improve after five iterations. Fixing the number of VB iterations is the most practical option in terms of predictability of the computation times, but the required number of iterations has to be verified for each model specifically.

B. TMND's moments

The mean and covariance matrix of a TMND can be computed using the formulas presented in [28]. They require evaluating the CDF of general multivariate normal distributions. The MATLAB function `mvncdf` implements the numerical

Algorithm 1 Filtering for skew-t measurement noise

Inputs: A, C, Q, R, Δ, ν, $x_{1|0}$, $P_{1|0}$, $y_{1:K}$, APPROX_TMND
$\Lambda \leftarrow I_{n_y}$, $C_z \leftarrow [\, C \;\; \Delta \,]$
for k = 1 to K **do**
  $[a_{k|k}]_i \leftarrow \frac{\nu_i+2}{2}$, $[b_{k|k}]_i \leftarrow \frac{\nu_i+2}{2}$ for $i = 1, \cdots, n_y$
  repeat
    $[\Lambda_{k|k}]_{ii} \leftarrow \frac{[a_{k|k}]_i}{[b_{k|k}]_i}$ for $i = 1, \cdots, n_y$
  update $q_{xu}(x_k, u_k)$
    $Z_{k|k-1} \leftarrow \text{blockdiagonal}(P_{k|k-1}, \Lambda^{-1}_{k|k})$
    $K_z \leftarrow Z_{k|k-1} C_z^{\mathrm{T}} (C P_{k|k-1} C^{\mathrm{T}} + \Delta \Lambda^{-1}_{k|k} \Delta^{\mathrm{T}} + \Lambda^{-1}_{k|k} R)^{-1}$
    $\widetilde{z}_{k|k} \leftarrow \left[\begin{smallmatrix} x_{k|k-1} \\ 0 \end{smallmatrix}\right] + K_z (y_k - C x_{k|k-1})$
    $\widetilde{Z}_{k|k} \leftarrow (I - K_z C_z) P_{k|k-1}$
    $[z_{k|k}, Z_{k|k}] \leftarrow \texttt{APPROX\_TMND}(\widetilde{z}_{k|k}, \widetilde{Z}_{k|k}, \{n_x+1 \cdots n_x+n_y\})$
    $x_{k|k} \leftarrow [z_{k|k}]_{1:n_x}$, $P_{k|k} \leftarrow [Z_{k|k}]_{1:n_x, 1:n_x}$
    $u_{k|k} \leftarrow [z_{k|k}]_{n_x+(1:n_y)}$, $U_{k|k} \leftarrow [Z_{k|k}]_{n_x+(1:n_y), n_x+(1:n_y)}$
  update $q_\Lambda(\Lambda_k)$
    $\Psi \leftarrow (y_k - C_z z_{k|k})(y_k - C_z z_{k|k})^{\mathrm{T}} R^{-1}$
      $+ C_z Z_{k|k} C_z^{\mathrm{T}} R^{-1} + u_{k|k} u_{k|k}^{\mathrm{T}} + U_{k|k}$
    for i = 1 to n_y **do** $[b_{k|k}]_i \leftarrow \frac{\nu_i + \Psi_{ii}}{2}$ **end for**
  until converged
  $x_{k+1|k} \leftarrow A x_{k|k}$
  $P_{k+1|k} \leftarrow A P_{k|k} A^{\mathrm{T}} + Q$
end for
Outputs: $x_{k|k}$ **and** $P_{k|k}$ **for** $k = 1 \cdots K$

Algorithm 2 Smoothing for skew-t measurement noise

Inputs: A, C, Q, R, Δ, ν, $x_{1|0}$, $P_{1|0}$, $y_{1:K}$, APPROX_TMND
$\Lambda_{k|K} \leftarrow I_{n_y}$ for $k = 1 \cdots K$, $A_z \leftarrow [\begin{smallmatrix} A & 0 \\ 0 & 0 \end{smallmatrix}]$, $C_z \leftarrow [\, C \;\; \Delta \,]$
repeat
  update $q_{xu}(x_{1:K}, u_{1:K})$
  for k = 1 to K **do**
    $Z_{k|k-1} \leftarrow \text{blockdiagonal}(P_{k|k-1}, \Lambda^{-1}_{k|K})$
    $K_z \leftarrow Z_{k|k-1} C_z^{\mathrm{T}} (C P_{k|k-1} C^{\mathrm{T}} + \Delta \Lambda^{-1}_{k|K} \Delta^{\mathrm{T}} + \Lambda^{-1}_{k|K} R)^{-1}$
    $\widetilde{z}_{k|k} \leftarrow \left[\begin{smallmatrix} x_{k|k-1} \\ 0 \end{smallmatrix}\right] + K_z (y_k - C x_{k|k-1})$
    $\widetilde{Z}_{k|k} \leftarrow (I - K_z C_z) P_{k|k-1}$
    $[z_{k|k}, Z_{k|k}] \leftarrow \texttt{APPROX\_TMND}(\widetilde{z}_{k|k}, \widetilde{Z}_{k|k}, \{n_x+1 \cdots n_x+n_y\})$
    $x_{k|k} \leftarrow [z_{k|k}]_{1:n_x}$, $P_{k|k} \leftarrow [Z_{k|k}]_{1:n_x, 1:n_x}$
    $x_{k+1|k} \leftarrow A x_{k|k}$
    $P_{k+1|k} \leftarrow A P_{k|k} A^{\mathrm{T}} + Q$
  end for
  for $k = K-1$ down to 1 **do**
    $G_k \leftarrow Z_{k|k} A_z Z^{-1}_{k+1|k}$
    $z_{k|K} \leftarrow z_{k|k} + G_k (z_{k+1|K} - A_z z_{k|k})$
    $Z_{k|K} \leftarrow Z_{k|k} + G_k (Z_{k+1|K} - Z_{k+1|k}) G_k^{\mathrm{T}}$
    $x_{k|K} \leftarrow [z_{k|K}]_{1:n_x}$, $P_{k|K} \leftarrow [Z_{k|K}]_{1:n_x, 1:n_x}$
    $u_{k|K} \leftarrow [z_{k|K}]_{n_x+(1:n_y)}$, $U_{k|K} \leftarrow [Z_{k|K}]_{n_x+(1:n_y), n_x+(1:n_y)}$
  end for
  update $q_\Lambda(\Lambda_{1:K})$
  for k = 1 to K **do**
    $\Psi \leftarrow (y_k - C_z z_{k|K})(y_k - C_z z_{k|K})^{\mathrm{T}} R^{-1}$
      $+ C_z Z_{k|K} C_z^{\mathrm{T}} R^{-1} + u_{k|K} u_{k|K}^{\mathrm{T}} + U_{k|K}$
    for i = 1 to n_y **do** $[\Lambda_{k|K}]_{ii} \leftarrow \frac{\nu_i+2}{\nu_i+\Psi_{ii}}$ **end for**
  end for
until converged
Outputs: $x_{k|K}$ **and** $P_{k|K}$ **for** $k = 1 \cdots K$

quadrature of [29] in 2 and 3 dimensional cases and the quasi-Monte Carlo method of [30] for the dimensionalities 4–25. However, these methods can be prohibitively slow. Therefore, we approximate the TMND's moments using a fast sequential algorithm that is based on the Expectation Propagation (EP) algorithm [31]. An EP algorithm for computing the mean, covariance matrix, and the truncated probability of a TMND is derived in [32]. The method is initialized with the original normal density whose parameters are then updated by applying one linear constraint at a time. For each constraint, the mean and covariance matrix of the once-truncated normal distribution are computed analytically, and the once-truncated distribution is approximated by a non-truncated normal with the updated moments. The EP is an iterative algorithm, so each truncation can be re-made when, roughly speaking, the effect of the previous iteration of the considered truncation is removed from the normal distribution's moments. One iteration of this method is illustrated in Fig. 3, where a bivariate normal distribution truncated into the positive quadrant is approximated with a non-truncated normal distribution.

The result of the EP algorithm depends on the order in which the constraints are applied. Finding the optimal order of applying the truncations is a problem that has combinatorial complexity. Hence, we adopt a greedy approach, whereby the constraint to be applied is chosen from among the remaining constraints such that the resulting once-truncated normal distribution is closest to the true TMND. By Lemma 1, the optimal constraint to select is the one that truncates the most probability. The optimality is with respect to a KLD as the measure. For example, in Fig. 3 the vertical constraint truncates more probability, so it is applied first.

Lemma 1. *Let $p(\mathbf{z})$ be a TMND with the support $\{\mathbf{z} \geq 0\}$ and $q(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mu, \Sigma)$. Then,*

$$\underset{i}{\operatorname{argmin}}\, D_{\mathrm{KL}}\Big(p(\mathbf{z}) \,\Big|\Big|\, \tfrac{1}{c_i} q(\mathbf{z}) [\![\mathbf{z}_i \geq 0]\!]\Big) = \underset{i}{\operatorname{argmin}}\, \tfrac{\mu_i}{\sqrt{\Sigma_{ii}}}, \quad (11)$$

where μ_i is the ith element of μ, Σ_{ii} is the ith diagonal element of Σ, $[\![\cdot]\!]$ is the Iverson bracket, and $c_i = \int q(\mathbf{z}) [\![\mathbf{z}_i \geq 0]\!] \, \mathrm{d}\mathbf{z}$.

Proof:

$$D_{\mathrm{KL}}\Big(p(\mathbf{z}) \,\Big|\Big|\, \tfrac{1}{c_i} q(\mathbf{z}) [\![\mathbf{z}_i \geq 0]\!]\Big)$$
$$\stackrel{+}{=} -\int p(\mathbf{z}) \log(\tfrac{1}{c_i} q(\mathbf{z}) [\![\mathbf{z}_i \geq 0]\!]) \, \mathrm{d}\mathbf{z} \quad (12)$$
$$= \log c_i - \int p(\mathbf{z}) \log q(\mathbf{z}) \, \mathrm{d}\mathbf{z} \stackrel{+}{=} \log c_i, \quad (13)$$

where $\stackrel{+}{=}$ means equality up to an additive constant. Since c_i is an increasing function of $\frac{\mu_i}{\sqrt{\Sigma_{ii}}}$, the proof follows. ∎

The obtained EP algorithm with the greedy processing sequence for computing the mean and covariance matrix of a given multivariate normal distribution truncated to the positive orthant is given in Algorithm 3. The algorithm can also give the logarithm of the positive orthant's probability α, which is required in computing the variational lower bound. In many programming languages a numerically robust method to implement the line 12 of the algorithm in Algorithm 3 is using the scaled complementary error function erfcx through

$$\frac{\phi(\xi)}{\Phi(\xi)} = \frac{\sqrt{2/\pi}}{\mathrm{erfcx}(-\xi/\sqrt{2})}. \quad (14)$$

C. Variational lower bound

When the PDF $p(x|y)$ is approximated with the PDF $q(x)$, the variational lower bound is

$$\mathcal{L}(q) = \int q(x) \log \frac{p(y, x)}{q(x)} \mathrm{d}x. \quad (15)$$

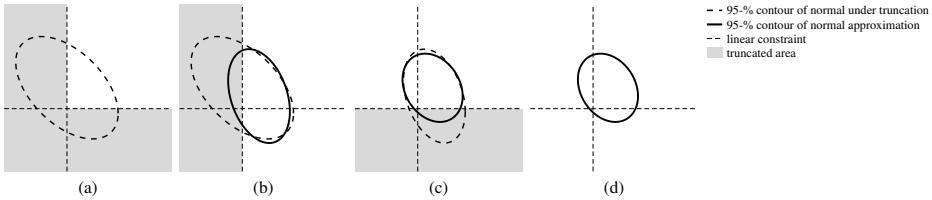


Fig. 3. An iteration of the EP algorithm for approximating a truncated normal distribution with a normal distribution: (a) the original normal distribution's contour ellipse that contains 95 % of the probability, and the truncated area in gray, (b) the first applied truncation in gray, and the 95-% contour of the resulting normal approximation, (c) the second applied truncation in gray, and the 95-% contour of the normal approximation, (d) the final normal approximation.

Algorithm 3 Greedy expectation propagation for the moments of normal distribution truncated to positive orthant (function $[\mu, \Sigma, \alpha] \leftarrow \texttt{APPROX\_TMND}(\mu, \Sigma, \mathcal{T})$)

Inputs: μ, Σ, and set of the truncated components' indices $\mathcal{T}$
$\widetilde{\mu} \leftarrow \mu, \quad \widetilde{\Sigma} \leftarrow \Sigma$
if #output=3 **then** $\alpha \leftarrow -\frac{1}{2}\widetilde{\mu}^{\mathrm{T}}\widetilde{\Sigma}^{-1}\widetilde{\mu}, \quad M \leftarrow I_{n_\mu}$ **end if**
$\tau_k \leftarrow 0, \quad \eta_k \leftarrow 0$ for $k = 1, 2, \ldots, n_\mu$.
repeat
  $\mathcal{T}' \leftarrow \mathcal{T}$
  while $\mathcal{T}' \neq \emptyset$ **do**
    $k \leftarrow \operatorname{argmin}_i\{\mu_i/\sqrt{\Sigma_{ii}} \mid i \in \mathcal{T}'\}$
    $s^2 \leftarrow 1/(1/\Sigma_{kk} - \tau_k)$
    $m \leftarrow s^2(\mu_k/\Sigma_{kk} - \eta_k)$
    $\xi \leftarrow m/s$
    $\epsilon \leftarrow \phi(\xi)/\Phi(\xi)$ ▷ ϕ is the PDF of $\mathcal{N}(0,1)$, Φ its CDF
    $\overline{m} \leftarrow m + \epsilon s$
    $\overline{s}^2 \leftarrow (1 - \xi\epsilon - \epsilon^2)s^2$
    $\overline{\tau}_k \leftarrow 1/\overline{s}^2 - 1/s^2 - \tau_k, \quad \tau_k \leftarrow \tau_k + \overline{\tau}_k$
    $\overline{\eta}_k \leftarrow \overline{m}/\overline{s}^2 - m/s^2 - \eta_k, \quad \eta_k \leftarrow \eta_k + \overline{\eta}_k$
    $\mu \leftarrow \mu + \frac{\overline{\eta}_k - \overline{\tau}_k\mu_k}{1+\overline{\tau}_k\Sigma_{kk}} \cdot \Sigma_{:,k}$
    $\Sigma \leftarrow \Sigma - \frac{\overline{\tau}_k}{1+\overline{\tau}_k\Sigma_{jj}} \cdot \Sigma_{:,k}\Sigma_{k,:}$
    if #output=3 **then**
      $M \leftarrow M + \tau_k L_{k,:}^{\mathrm{T}} L_{k,:}$ ▷ $LL^{\mathrm{T}} = \widetilde{\Sigma}$
      $\alpha \leftarrow \alpha + \log\big(\Phi(\xi)\big) + \frac{1}{2}\log(1 + \tau_k s^2)$
        $+\frac{1}{2}\frac{m^2\tau_k - 2m\eta_k - s^2\eta_k^2}{1+\tau_k s^2} + \frac{1}{2}\tau_k\mu_k^2$
    end if
    $\mathcal{T}' \leftarrow \mathcal{T}' \backslash \{k\}$
  end while
  if #output=3 **then** $\alpha \leftarrow \alpha - \frac{1}{2}\log(\det(M)) + \frac{1}{2}\mu^{\mathrm{T}}\widetilde{\Sigma}^{-1}\mu$ **end if**
until converged
Outputs: moments μ, Σ, and the logarithm of the positive orthant's probability α

Minimizing the KLD is equivalent to maximizing the variational lower bound [33, Ch. 21]. Therefore, the variational lower bound can be used as a debugging means and convergence criterion for the VB iterations because the lower bound should increase at each iteration. Furthermore, because the logarithmic marginal likelihood $\log p(y)$ is the sum of the variational lower bound and the KLD, the maximal variational lower bound can be used as an approximation for $\log p(y)$. The model evidence in Bayesian model comparison can thus be approximated with $\exp(\mathcal{L}(q))$ [33, Ch. 21.5.1.6].

When evaluated immediately after the VB filter update of $q_{xu}(x_k, u_k)$, the variational lower bound for the skew-t filter is

$$\begin{aligned}\mathcal{L}_{\mathrm{f}}(q) = \log\mathcal{N}\big(y; Cx_{k|k-1}, CP_{k|k-1}C^{\mathrm{T}} + \Delta\Lambda_{k|k}^{-1}\Delta^{\mathrm{T}} + \Lambda_{k|k}^{-1}R\big) \\ + \sum_{j=1}^{n_y}\left[[a_{k|k}]_j\left(1 + \log\left(\tfrac{[a_{k|k}]_j - 1}{[b_{k|k}]_j}\right) - \tfrac{[a_{k|k}]_j - 1}{[b_{k|k}]_j}\right)\right. \\ \left. - \log\left(\tfrac{[a_{k|k}]_j}{[b_{k|k}]_j}\right)\right] + n_y\log(2) + \alpha_{k|k}, \end{aligned} \tag{16}$$

where the notations follow those in Algorithm 1, and $\alpha_{k|k}$ is the logarithm of the probability of the positive orthant given the distribution $\mathcal{N}([\widetilde{z}_{k|k}]_{n_x+(1:n_y)}, [\widetilde{Z}_{k|k}]_{n_x+(1:n_y),n_x+(1:n_y)})$. The probability $\alpha_{k|k}$ can be computed using the EP algorithm in Algorithm 3. The derivation of the lower bound (16) is straightforward but tedious and omitted here. Unfortunately, evaluation of the variational lower bound for the smoother is impractical because its expression includes a probability of the positive orthant given a high-dimensional normal distribution.

IV. Simulations

Our numerical simulations use satellite navigation pseudo-range measurements of the model

$$[y_k]_i = \|s_i - [x_k]_{1:3}\| + [x_k]_4 + [e_k]_i, [e_k]_i \overset{\text{iid}}{\sim} \mathrm{ST}(0, 1\,\mathrm{m}, \delta\,\mathrm{m}, 4) \tag{17}$$

where $s_i \in \mathbb{R}^3$ is the ith satellite's position, $[x_k]_4 \in \mathbb{R}$ is bias with prior $\mathcal{N}(0, (0.75\,\mathrm{m})^2)$, and $\delta \in \mathbb{R}$ is a parameter. The model is linearized using the first order Taylor polynomial approximation, and the linearization error is negligible because the satellites are far relative to the magnitude of uncertainty in the prior. The satellite constellation of the Global Positioning System (GPS) from the first second of the year 2015 provided by the International GNSS Service [34] is used with 8 visible satellites. The root-mean-square error (RMSE) is computed for the position $[x_k]_{1:3}$ as

$$\mathrm{RMSE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\big\|[x_{k|k}]_{1:3} - [x_k]_{1:3}\big\|^2}, \tag{18}$$

where $x_{k|k}$ is the filter estimate and x_k is the true state. The computations are made with MATLAB.

A. Computation of TMND statistics

In this subsection we study the computation of the moments of the untruncated components of a TMND. Per a Monte Carlo replication, one state value is generated from the prior $x \sim \mathcal{N}(0, \mathrm{diag}(20^2, 20^2, 0.22^2, 0.1^2)\,\mathrm{m}^2)$, and one measurement vector is generated from the model (17) with $\nu = \infty$ degrees of freedom (corresponding to skew-normal likelihood). 10 000

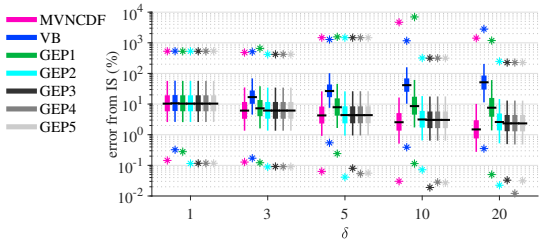


Fig. 4. Two EP iterations suffice. MVNCDF is slightly more accurate but computationally heavy.

Monte Carlo replications are used. The compared methods are expectation propagation (EP) with the greedy truncation order and one, two, three, four, and five EP iterations (GEP1, GEP2, GEP3, GEP4, GEP5), the variational Bayes (VB), and the analytical formulas of [28] using MATLAB function `mvncdf` (MVNCDF). VB is an update of the skew t VB filter (STVBF) [20] where the heavy-tailedness variable $\overline{\Lambda}_1$ is fixed to identity I_{n_y} and the VB iteration is terminated when the position estimate changes less than 0.005 m or at the 1000th iteration. The reference solution for the expectation value is an importance sampling (IS) update with 50 000 samples and the prior as the importance distribution.

Fig. 4 shows the distributions of the estimates' differences from the IS estimate. The errors are given per cent of the IS's estimation error. The box levels are 5 %, 25 %, 50 %, 75 %, and 95 % quantiles and the asterisks show minimum and maximum values. The results indicate that the accuracy of the EP approximation of the mean does not improve after two EP iterations. MVNCDF is slightly more accurate than GEP2 in the cases with high skewness, but MVNCDF's computational load can be roughly 40 000 times that of the GEP2. This justifies the use of the EP approximation.

The approximation of the posterior covariance matrix is tested by studying the normalized estimation error squared (NEES) values [35, Ch. 5.4.2]

$$\text{NEES}_k = (x_{k|k} - x_k)^{\mathrm{T}} P_{k|k}^{-1} (x_{k|k} - x_k), \tag{19}$$

where $x_{k|k}$ and $P_{k|k}$ are the filter's output mean and covariance matrix, and x_k is the true state. The algorithms' average NEES values are given in Table I. If the covariance matrix is correct, the NEES is χ^2-distributed with 3 degrees of freedom because the position is 3-dimensional, so the nominal expected value is 3 [35, Ch. 5.4.2]. VB gets large NEES values when δ is large, which indicates that VB underestimates the covariance matrix. Apart from MVNCDF, the GEP algorithms give NEES values closest to 3, so the EP provides a more accurate covariance matrix approximation than VB. Indicated by NEES being slightly smaller than 3, GEP1 in fact overestimates the covariance matrix when δ is large, but this issue is mostly fixed by the second EP iteration.

The order of the truncations in the EP algorithm affects the performance only when there are clear differences in the amounts of probability mass under each truncation. We compare GEP1 with the EP iteration with a random truncation order (REP1). In REP1 any of the non-optimal constraints is chosen randomly at each truncation. Fig. 5 presents an example where $\delta = 20$, and the measurement noise realization e has been generated from the skew normal distribution and then modified by

$$e_j = \min\{\min\{e_{1:n_y}\}, 0\} - c\sqrt{1 + 20^2}, \tag{20}$$

Table I
THE AVERAGE NEES VALUES. GEP1'S NEES IS CLOSER TO THE OPTIMAL VALUE 3 THAN THAT OF VB, SO EP GIVES A MORE ACCURATE POSTERIOR COVARIANCE MATRIX.

δ	1	3	5	10	20
MVNCDF	3.0	3.0	3.0	3.0	3.0
VB	3.8	9.1	19.1	65.6	229.2
GEP1	3.0	2.9	2.8	2.7	2.7
GEP2	3.0	3.0	3.0	2.9	2.9
GEP3	3.0	3.0	3.0	2.9	2.9
GEP4	3.0	3.0	3.0	2.9	2.9
GEP5	3.0	3.0	3.0	2.9	2.9

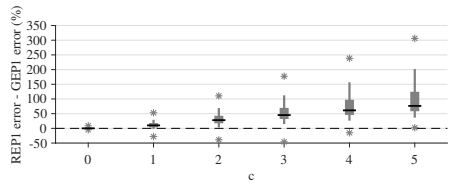


Fig. 5. GEP1 outperforms REP1 when one negative outlier is added to the measurement noise vector because there is one truncation that truncates much more probability than the rest.

where j is a random index, and c is a parameter. A large c generates one negative outlier to each measurement vector, which results in one truncation with significantly larger truncated probability mass than the rest of the truncations. Fig. 5 shows the percentual difference of REP1 error from GEP1 error; i.e. a positive difference means that GEP1 is more accurate. The errors here refer to distance from the IS estimate. The figure shows that with large c GEP1 is more accurate than REP1. Thus, the effect of the truncation ordering on the accuracy of the EP approximation is more pronounced when there is one truncation that truncates much more than the rest. This justifies our greedy approach and the result of Lemma 1 for ordering the truncations.

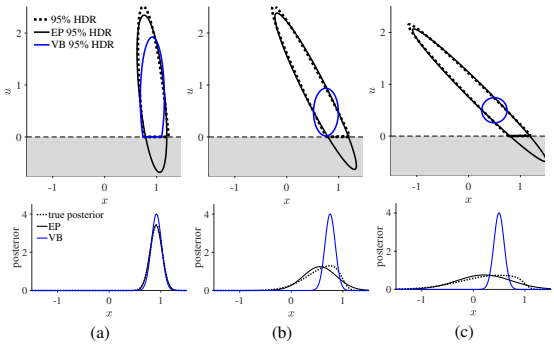
The skew-t measurement model essentially implies that given the scaling-related variable Λ, we are observing the sum $Cx + \Delta u$ plus normally distributed noise. Fig. 6 compares the EP approximation and the 30-iteration VB approximation of the posterior distribution for the model

$$p(x, u) = \mathcal{N}(x; 0, 1) \cdot \mathcal{N}_+(u; 0, 1) \tag{21a}$$
$$p(y|x, u) = \mathcal{N}(y; x + \delta u, 0.1^2) \tag{21b}$$

with the measurement value $y = 1$ and with δ values 0.1, 0.5, and 1. Fig. 6 illustrates that when δ is large, x and u are highly correlated. This makes VB seriously underestimate the covariance matrix, and EP provides a better approximation of the joint posterior and the marginal posterior of x.

B. Skew-t inference

In this section, the proposed skew-t filter (STF) is compared with state-of-the-art filters using numerical simulations of a 100-step trajectory. The tested STF uses two EP iterations. The state model is a random walk with process noise covariance $Q = \text{diag}(q^2, q^2, 0.2^2, 0)$ m, where q is a parameter. The compared methods are a bootstrap-type PF, STVBF [20], t variational Bayes filter (TVBF) [36], and Kalman filter (KF) with measurement validation gating [35, Ch. 5.7.2] that discards the measurement components whose normalized innovation squared is larger than the χ^2_1-distribution's 99 % quantile. The used KF parameters are the mean and variance



(a) (b) (c)

Fig. 6. EP gives a better approximation than VB for a bivariate normal distribution of (x, u), where u is truncated to be positive. The figures show the 95 % high-density regions (HDR) of the posteriors $p(x, u|y=1)$ (upper row) and the marginal posteriors $p(x|y=1)$ (lower row) of the model (21). (a) $\delta=0.1$, (b) $\delta=0.5$, (c) $\delta=1$.

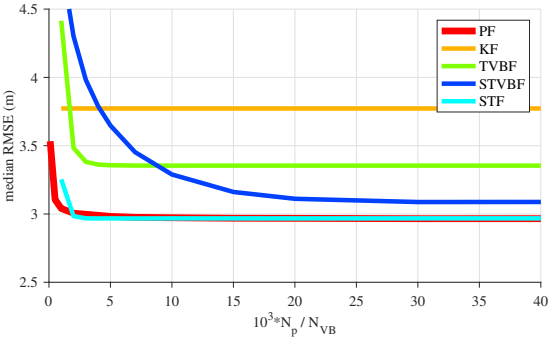


Fig. 7. STF's median RMSE does not improve after $N_{\mathrm{VB}}=5$ VB iterations per time instant. The required number of PF particles N_{p} can be more than 10 000. The x-axis is $10^3 \cdot N_{\mathrm{p}}$ for PF and N_{vb} for the rest of the filters. $q=5,\ \delta=5$.

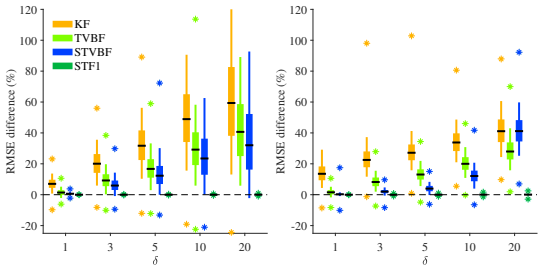


Fig. 8. STF outperforms the comparison methods with skew-t-distributed noise. RMSE differences from STF's RMSE per cent of the STF's RMSE. The differences increase as skewness increases. (upper) $q=0.5$, (lower) $q=5$.

of the used skew t-distribution, and the TVBF parameters are obtained by matching the degrees of freedom with that of the skew t-distribution and computing the maximum likelihood location and scale parameters for a set of pseudo-random numbers generated from the skew t-distribution. The results are based on 10 000 Monte Carlo replications.

Fig. 7 illustrates the filter iterations' convergence when the measurement noise components $[e_k]_i$ in (17) are generated independently from the univariate skew t-distribution. The figure shows that the proposed STF's median RMSE does not improve after five VB iterations, and STF outperforms the other filters in RMSE already with two VB iterations, except for PF that is the minimum-RMSE solution. Furthermore, Fig. 7 shows that STF's converged state is close to the PF's converged state in RMSE, and PF can require as many as 10 000 particles to outperform STF. In our implementation, the PF with 10 000 particles is computationally roughly 15 times heavier that the STF with five VB iterations. STF also converges faster than STVBF when the process noise variance parameter q is large.

Fig. 8 shows the distributions of the RMSE differences from the STF's RMSE as percentages of the STF's RMSE. STF1 is the skew-t filter with just one EP iteration per a VB iteration. STF, STF1, and TVBF use five VB iterations, and STVBF uses 30 VB iterations. STF clearly has the smallest RMSE when $\delta \geq 3$, i.e. when the skewness is high. STF1 and STF (with 2 EP iterations) have similar accuracies, so one EP iteration may be sufficient in practice. Unlike STVBF, the new STF improves accuracy even with small q and large δ, which can be explained by the improved covariance matrix approximation.

The proposed smoother is also tested with measurements generated from (17). The compared smoothers are the proposed skew-t smoother with two EP iterations (STS), skew-t variational Bayes smoother (STVBS) [20], t variational Bayes smoother (TVBS) [36], and the RTSS with 99 % measurement validation gating [27]. Fig. 9 shows that STS has lower RMSE than the smoothers based on symmetric distributions. Furthermore, STS's VB iteration converges in five iterations or less, so it is faster than STVBS.

V. Tests with real data

A. GNSS-based pseudorange positioning

Two GNSS (global navigation satellite system) positioning data sets were collected in central London (UK) to test the filters' performance in a challenging real-world satellite positioning scenario with numerous non-line-of-sight measurements. The data include time delay based pseudorange measurements from GPS satellites. Each set contains a trajectory that was collected by car using a u-blox M8 GNSS receiver. The lengths of the tracks are about 8 km and 10 km, the durations are about an hour for each, and measurements are received at about one-second intervals. The first track is used for fitting the filter parameters, while the second track is used for studying the filters' positioning errors. A ground truth was measured using an Applanix POS-LV220 system that improves the GNSS solution with tactical grade inertial measurement units. The GPS satellites' locations were obtained from the broadcast ephemerides provided by the International GNSS Service [34].

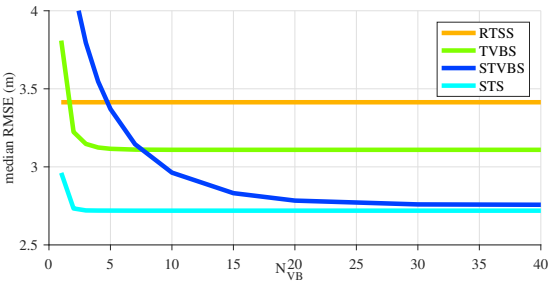


Fig. 9. Five STS iterations give the converged state's RMSE. $q=5,\ \delta=5$.

Table II
FILTER PARAMETERS FOR REAL GNSS DATA

Skew-t, $\nu = 4$			t, $\nu_\mathrm{t} = 4$		Normal	
μ (m)	σ (m)	δ (m)	μ_t (m)	σ_t (m)	μ_n (m)	σ_n (m)
-2.5	0.8	16.8	0	11.1	0	28.4

The algorithms are computed with MATLAB.

The used state evolution model is the almost-constant velocity model for both the user position $l_k \in \mathbb{R}^3$ and the receiver clock error $b_k \in \mathbb{R}$ used in [37, Section IV]. Thus, the filter state is $x_k = \begin{bmatrix} l_k^\mathrm{T} & \dot{l}_k^\mathrm{T} & b_k & \dot{b}_k \end{bmatrix}^\mathrm{T} \in \mathbb{R}^8$, and the state evolution model is

$$x_{k+1} = \begin{bmatrix} I_3 & d_k I_3 & \mathrm{O}_{3\times 2} \\ \mathrm{O}_3 & I_3 & \mathrm{O}_{3\times 2} \\ \mathrm{O}_{2\times 3} & \mathrm{O}_{2\times 3} & \begin{bmatrix} 1 & d_k \\ 0 & 1 \end{bmatrix} \end{bmatrix} x_k + w_k, \qquad (22)$$

where

$$w_k \sim \mathcal{N}\left(0, \begin{bmatrix} \frac{q^2 d_k^3}{3} I_3 & \frac{q^2 d_k^2}{2} I_3 & \mathrm{O}_{3\times 2} \\ \frac{q^2 d_k^2}{2} I_3 & q^2 d_k I_3 & \mathrm{O}_{3\times 2} \\ \mathrm{O}_{2\times 3} & \mathrm{O}_{2\times 3} & \begin{bmatrix} s_b d_k + \frac{s_f d_k^3}{3} & \frac{s_f d_k^2}{2} \\ \frac{s_f d_k^2}{2} & s_f d_k \end{bmatrix} \end{bmatrix}\right).$$

and d_k is the time difference of the measurements in seconds. The used parameter values are $q = 0.5\,\mathrm{m/s^{\frac{3}{2}}}$, $s_b = 70\,\frac{\mathrm{m}^2}{\mathrm{s}}$, and $s_f = 0.6\,\frac{\mathrm{m}^2}{\mathrm{s}^3}$. The initial prior is a normal distribution with mean given by the Gauss–Newton method with the first measurement and a large covariance matrix.

The measurement model is the same pseudorange model that is used in the simulations of Section IV, i.e.

$$[y_k]_i = \|s_{i,k} - [x_k]_{1:3}\| + [x_k]_7 + [e_k]_i, \qquad (23)$$

where $s_{i,k}$ is the 3-dimensional position of the ith satellite at the time of transmission. The measurement model is linearized with respect to x_k at each prior mean using the first order Taylor series approximation. The compared filters are based on three different models for the measurement noise e_k where

$$[e_k]_i \sim \mathrm{ST}(\mu, \sigma^2, \delta, \nu); \qquad (24)$$
$$[e_k]_i \sim \mathcal{T}(\mu_\mathrm{t}, \sigma_\mathrm{t}^2, \nu_\mathrm{t}); \qquad (25)$$
$$[e_k]_i \sim \mathcal{N}(\mu_\mathrm{n}, \sigma_\mathrm{n}^2). \qquad (26)$$

The skew-t model (24) is the basis for STF and STVBF, the t model (25) is the basis for TVBF, and the normal model (26) is the basis for the extended KF (EKF) with 99 % measurement validation gating. The pseudoranges are unbiased in the line-of-sight case, so the location parameters are fixed to $\mu_\mathrm{n} = \mu_\mathrm{t} = 0$. Furthermore, the degrees of freedom are fixed to $\nu = \nu_\mathrm{t} = 4$, which according to our experience is in general a good compromise between outlier robustness and performance based on inlier measurements, provides infinite kurtosis but finite skewness and variance, and is recommended in [38]. The deviation parameter σ_n of the normal model was then fitted to the data using the expectation–maximization algorithm [39, Ch. 12.3.3] and the parameter σ_t of the t model as well as the parameters σ and δ of the skew-t model were fitted with the particle–Metropolis algorithm [39, Ch. 12.3.4]. The location parameter μ was obtained by numerically finding the point that sets the mode of the skew-t noise distribution to zero. These three error distributions' parameters are given in Table II, and the PDFs are plotted in Fig. 10.

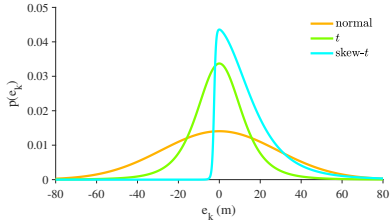


Fig. 10. Measurement error distributions fitted to the real GNSS data for normal, t, and skew-t error models. The modes are fixed to zero.

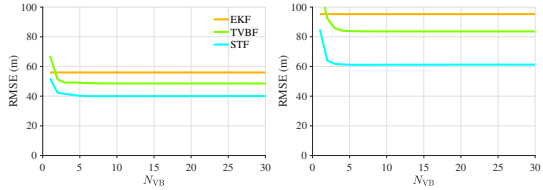


Fig. 11. RMSE of horizontal (left) and vertical (right) position for real GNSS data as a function of the number of VB iterations

Fig. 11 shows the filter RMSEs as a function of the number of VB iterations. Both STF and TVBF converge within five VB iterations. The empirical CDF graphs of the user position errors with five VB iterations are shown in Fig. 12, and the RMSEs as well as the relative running times are given in Table III. The results show that modelling the skewness improves the positioning accuracy and is important especially for the accuracy in vertical direction. This can be explained by the sensitivity of the vertical direction to large measurement errors; due to bad measurement geometry the accuracy in the vertical direction is low even with line-of-sight measurements, so correct downweighting of erroneous altitude information requires careful modelling of the noise distribution's tails. The computational burden of our STF implementation with five VB iterations is about three times that of TVBF, but Fig. 11 shows that two STF iterations would already be enough to match TVBF's average RMSE.

VI. EXTENSION TO MVST

The skew t-distribution has several multivariate versions. In [5]–[7] the PDF of the multivariate skew t-distribution (MVST) involves the CDF of a univariate t-distribution, while the definition of skew t-distribution given in [40] involves the CDF of a multivariate t-distribution. These versions of MVST are special cases of more general multivariate skew-t-type distribution families, which include the multivariate canonical fundamental skew t-distribution (CFUST) [41] and the multivariate unified skew t-distribution [42]. A comprehensive review on the different variants of the MVST is given in [23].

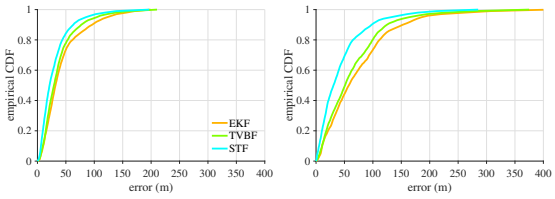


Fig. 12. Empirical error CDFs for the real GNSS data for the horizontal error (left) and the vertical error (right)

Table III
THE RMSES AND RELATIVE RUNNING TIMES FOR REAL GNSS DATA

	EKF	TVBF	STF
$\text{RMSE}_{\text{horizontal}}$ (m)	56	49	40
$\text{RMSE}_{\text{vertical}}$ (m)	95	84	61
Running time	1	1.3	4.1

The MVST variant used in this article is based on the CFUST discussed in [23], and it is the most general variant of the MVST. In this variant the parameter matrix $R \in \mathbb{R}^{n_z \times n_z}$ is a square positive-definite matrix, and $\Delta \in \mathbb{R}^{n_z \times n_z}$ is an arbitrary matrix. The PDF is

$$\text{MVST}(z; \mu, R, \Delta, \nu) = 2^{n_z} \text{t}(z; \mu, \Omega, \nu)\, \text{T}(\overline{z}; 0, L, \nu + n_z), \tag{27}$$

where $L = I_{n_z} - \Delta^{\text{T}} \Omega^{-1} \Delta$, $\Omega = R + \Delta\Delta^{\text{T}}$,

$$\begin{aligned} &\text{t}(z; \mu, \Sigma, \nu) \\ &= \frac{\Gamma\left(\frac{\nu+n_z}{2}\right)}{(\nu\pi)^{\frac{n_z}{2}} \det(\Sigma)^{\frac{1}{2}} \Gamma\left(\frac{\nu}{2}\right)} \left(1 + \tfrac{1}{\nu}(z-\mu)^{\text{T}} \Sigma^{-1} (z-\mu)\right)^{-\frac{\nu+n_z}{2}} \end{aligned} \tag{28}$$

is the PDF of the n_z-variate t-distribution and $\text{T}(z; \mu, \Sigma, \nu)$ its CDF, and

$$\overline{z} = \Delta^{\text{T}} \Omega^{-1} (z - \mu) \sqrt{\tfrac{\nu + n_z}{\nu + (z-\mu)^{\text{T}} \Omega^{-1} (z-\mu)}}. \tag{29}$$

The inference algorithms proposed in this paper can be extended to cover the case where the elements of the measurement noise vector are not statistically independent but jointly multivariate skew-t-distributed. When the measurement noise follows a MVST, i.e.

$$e_k \sim \text{MVST}(0, R, \Delta, \nu), \tag{30}$$

the filtering and smoothing algorithms presented in Tables 1 and 2 apply with slight modifications. At the core of this convenient extension is the fact that the MVST can be represented by a similar hierarchical model as in (6). However, the shape matrices Δ and R are not required to be diagonal, and the matrix Λ_k has the form $\lambda_k \cdot I_{n_y}$, where λ_k is a scalar with the prior

$$\lambda_k \sim \mathcal{G}(\tfrac{\nu}{2}, \tfrac{\nu}{2}). \tag{31}$$

Notice that when λ_k admits a small value, all the measurement components can potentially be outliers simultaneously with a higher probability than for the independent univariate skew-t components model. A univariate skew-t is also a MVST, but a vector of univariate independently skew-t distributed components is not a special case of MVST. This difference is illustrated by the PDF contour plots in Fig. 13. See also further discussion in [23].

The specific modification required by MVST measurement noise to the STS algorithm in Table 2 is replacing line 23 by

$$\Lambda_{k|K} \leftarrow \frac{\nu + 2n_y}{\nu + \text{tr}\{\Psi_k\}} \cdot I_{n_y} \tag{32}$$

Similarly, the specific modification required by MVST measurement noise to the STF algorithm in Table 1 is replacing line 15 by

$$\Lambda_{k|k} \leftarrow \frac{\nu + 2n_y}{\nu + \text{tr}\{\Psi_k\}} \cdot I_{n_y}. \tag{33}$$

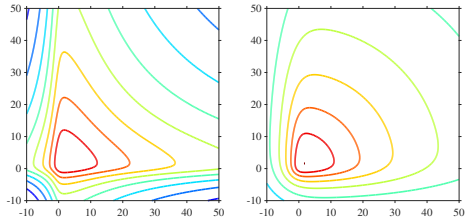


Fig. 13. PDF of bivariate measurement noise from (a) independent univariate skew-t components model (3) with $\Delta = 5I_2$, $R = I_2$, $\nu = \left[\begin{smallmatrix} 4 \\ 4 \end{smallmatrix}\right]$ and (b) MVST model (30) with $\Delta = 5I_2$, $R = I_2$, $\nu = 4$.

VII. PERFORMANCE BOUND

A. Cramér–Rao lower bound

The Bayesian Cramér–Rao lower bound (CRLB) B is a lower bound for the mean-square-error (MSE) matrix of the state estimator $\hat{x}$ of the random variable x using the observations y

$$M = \mathop{\mathbb{E}}_{p(x,y)} [(x - \hat{x})(x - \hat{x})^{\text{T}}] \tag{34}$$

in the sense that the matrix difference $M - B$ is positive semidefinite for any state estimator [43, Ch. 2.4]. The regularity conditions sufficient for the positive-semidefiniteness to hold [43, Ch. 2.4] are the integrability of the first two partial derivatives of the joint density $p(x_{1:k}, y_{1:k})$ for an asymptotically unbiased estimator. These conditions are satisfied by the skew-t likelihood and the normal prior distribution, even though they do not hold for $p(x_{1:k}, u_{1:k}, \Lambda_{1:k}, y_{1:k})$ of the hierarchical model used in the proposed variational estimator due to restriction of $u_{1:k}$ to the positive orthant. This is sufficient, since we only seek for the CRLB for the actual state x, not for the artificial variables u and Λ.

The filtering CRLB $B_{k|k}$ for the state-space model (1)–(2) follows the recursion [44]

$$B_{1|0} = P_{1|0} \tag{35a}$$

$$B_{k+1|k+1} = \left((AB_{k|k}A^{\text{T}} + Q)^{-1} + \mathop{\mathbb{E}}_{p(x_k|y_{1:k-1})} [\mathcal{I}(x_k)]\right)^{-1}, \tag{35b}$$

where $\mathcal{I}(e_k)$ is the Fisher information matrix of the measurement noise distribution. Furthermore, the smoothing CRLB for the state-space model (1)–(2) follows the recursion [44]

$$B_{k|K} = B_{k|k} + G_k (B_{k+1|K} - B_{k+1|k}) G_k^{\text{T}}, \tag{36}$$

where

$$G_k = B_{k|k} A^{\text{T}} B_{k+1|k}^{-1}, \tag{37}$$

$$B_{k+1|k} = AB_{k|k}A^{\text{T}} + Q. \tag{38}$$

This coincides with the covariance matrix update of Rauch–Tung–Striebel smoother's backward recursion [27].

The Fisher information matrix for the multivariate skew-t-distributed measurement noise of (30) is

$$\mathcal{I}(x) = C^{\text{T}} (R + \Delta\Delta^{\text{T}})^{-\frac{\text{T}}{2}} E (R + \Delta\Delta^{\text{T}})^{-\frac{1}{2}} C, \tag{39}$$

where

$$E = \mathop{\mathbb{E}}_{p(r)} \left[\tfrac{\nu+n_y}{\nu+r^{\text{T}}r} \left(I_{n_y} - \tfrac{2}{\nu+r^{\text{T}}r} rr^{\text{T}} + \widetilde{R}_r \widetilde{R}_r^{\text{T}}\right)\right] \tag{40}$$

with $r \sim \mathrm{MVST}(0, I_{n_y} - \Theta\Theta^{\mathrm{T}}, \Theta, \nu)$, $\Theta = (R + \Delta\Delta^{\mathrm{T}})^{-\frac{1}{2}}\Delta$, $A^{\frac{1}{2}}$ is a square-root matrix such that $A^{\frac{1}{2}}(A^{\frac{1}{2}})^{\mathrm{T}} = A$, $A^{-\frac{1}{2}} \triangleq (A^{\frac{1}{2}})^{-1}$, $A^{-\frac{\mathrm{T}}{2}} \triangleq ((A^{\frac{1}{2}})^{-1})^{\mathrm{T}}$, and

$$\begin{aligned}\widetilde{R}_r = &\big(\mathrm{T}\big(\Theta^{\mathrm{T}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, I_{n_y} - \Theta^{\mathrm{T}}\Theta, \nu + n_y\big)\big)^{-1} \\ &\times (I_{n_y} - \tfrac{1}{\nu+r^{\mathrm{T}}r} rr^{\mathrm{T}})\Theta \\ &\times \nabla_u \mathrm{T}(u; 0, I_{n_y} - \Theta^{\mathrm{T}}\Theta, \nu + n_y)\Big|_{u=\Theta^{\mathrm{T}} r\sqrt{\frac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}}, \end{aligned} \quad (41)$$

where ∇_u is the gradient with respect to u. The derivation is given in Appendix B. The evaluation of the expectation in (40) is challenging with high-dimensional measurements due to the requirement to evaluate the CDF of the multivariate t-distribution and its partial derivatives. By the Woodbury matrix identity, the recursion (35) is equivalent to the covariance matrix update of the Kalman filter with the measurement noise covariance $(R + \Delta\Delta^{\mathrm{T}})^{\frac{1}{2}} E^{-1} ((R + \Delta\Delta^{\mathrm{T}})^{\frac{1}{2}})^{\mathrm{T}}$.

In the model (3) the measurement noise components are independently univariate skew-t-distributed. In this case the Fisher information is obtained by applying (39) to each conditionally independent measurement component and summing. The resulting formula matches with (39), the matrix E now being a diagonal matrix with the diagonal entries

$$\begin{aligned} E_{ii} = &\mathop{\mathbb{E}}_{p(r_i)}\Bigg[\tfrac{\nu_i - r_i^2}{(\nu_i + r_i^2)^2} \\ &+ \tfrac{\theta_i^2}{1-\theta_i^2}\tfrac{\nu_i^2}{(\nu_i+r_i^2)^3}\left(\tau_{\nu_i+1}\big(\tfrac{\theta_i}{\sqrt{1-\theta_i^2}} r_i \sqrt{\tfrac{\nu_i+1}{\nu_i+r_i^2}}\big)\right)^2\Bigg], \end{aligned} \quad (42)$$

where $r_i \sim \mathrm{ST}(0, 1-\theta_i^2, \theta_i, \nu_i)$ is a univariate skew-t-distributed random variable, $\theta_i = \Delta_{ii}/\sqrt{R_{ii} + \Delta_{ii}^2}$ and $\tau_\nu(x) = \mathrm{t}(x; 0, 1, \nu)/\mathrm{T}(x; 0, 1, \nu)$. Substituted into (39), this formula matches the Fisher information formula obtained for the univariate skew t-distribution in [45]. In this case only integrals with respect to one scalar variable are to be evaluated numerically.

B. Simulation

We study the CRLB in (35) of a linear state-space model with skew-t-distributed measurement noise by generating realizations of the model

$$x_{k+1} = \left[\begin{smallmatrix}1 & 1\\ 0 & 1\end{smallmatrix}\right] x_k + w_k, \;\; w_k \sim \mathcal{N}(0, Q) \quad (43a)$$

$$y_k = \left[\begin{smallmatrix}1 & 0\end{smallmatrix}\right] x_k + e_k, \;\; e_k \sim \mathrm{ST}(\mu, \sigma^2, \delta, \nu), \quad (43b)$$

where $x \in \mathbb{R}^2$ is the state, $Q = \left[\begin{smallmatrix}1/3 & 1/2\\ 1/2 & 1\end{smallmatrix}\right]$ is the process noise covariance matrix, $y_k \in \mathbb{R}$ is the measurement, and ν and δ_c are parameters that determine other parameters by the formulas

$$\mu = -\gamma\delta_c\sigma, \quad (44a)$$

$$\sigma^2 = \tfrac{\omega^2}{\frac{\nu}{\nu-2}(1+\delta_c^2) - \gamma^2\delta_c^2}, \quad (44b)$$

$$\delta = \delta_c\sigma, \quad (44c)$$

$$\gamma = \sqrt{\tfrac{\nu}{\pi}}\tfrac{\Gamma((\nu-1)/2)}{\Gamma(\nu/2)}. \quad (44d)$$

Thus, the measurement noise distribution is zero-mean and has the variance $\omega^2 = 5^2$. We generate 10 000 realizations of a 50-step process, and compute the CRLB and mean-square-errors (MSE) of the bootstrap PF with 2000 particles and the STF. The CRLB and the MSEs were computed for the first component of the state at the last time instant $[x_{50}]_1$.

Fig. 14 shows the CRLB of the model (43). The figure shows that increase in the skewness as well as heavy-tailedness can decrease the CRLB significantly, which suggests that a nonlinear filter can be significantly better than the KF, which gives MSE 11.8 for all δ_c and ν. Fig. 15 shows the MSEs of PF and STF. As expected, when $\nu \to \infty$ and $\delta_c \to 0$, the PF's MSE approaches the CRLB. STF is only slightly worse than PF. The figures also show that although the CRLB becomes looser when the distribution becomes more skewed and/or heavy-tailed, it correctly indicates that modeling the skewness still improves the filtering performance.

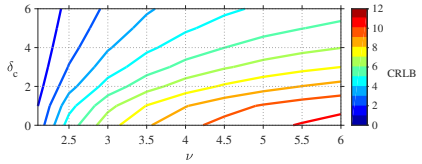


Fig. 14. The CRLB of the 50th time instant for the model (43) with a fixed measurement noise variance. Skewness and heavy-tailedness decreases the CRLB significantly.

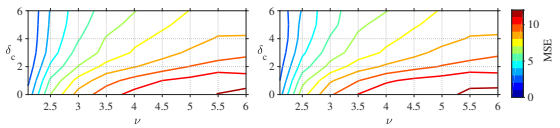


Fig. 15. The MSEs of PF (left) and STF's (right) are close to each other.

VIII. Conclusions

We have proposed a novel approximate filter and smoother for linear state-space models with heavy-tailed and skewed measurement noise distribution, and derived the Cramér–Rao lower bounds for the filtering and smoothing estimators. The algorithms are based on the variational Bayes approximation, where some posterior independence approximations are removed from the earlier versions of the algorithms to avoid significant underestimation of the posterior covariance matrix. Removal of independence approximations is enabled by the expectation propagation (EP) algorithm for approximating the mean and covariance matrix of truncated multivariate normal distribution. A greedy processing sequence is given for the EP. Simulations and real-data tests with GNSS positioning data show that the proposed algorithms outperform the state-of-the-art low-complexity methods.

References

[1] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, July 2005.

[2] B.-S. Chen, C.-Y. Yang, F.-K. Liao, and J.-F. Liao, "Mobile location estimator in a rough wireless environment using Extended Kalman-based IMM and data fusion," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1157–1169, March 2009.

[3] M. Kok, J. D. Hol, and T. B. Schön, "Indoor positioning using ultra-wideband and inertial measurements," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, 2015.

[4] K. Kaemarungsi and P. Krishnamurthy, "Analysis of WLAN's received signal strength indication for indoor location fingerprinting," *Pervasive and Mobile Computing*, vol. 8, no. 2, pp. 292–316, 2012, special Issue: Wide-Scale Vehicular Sensor Networks and Mobile Sensing.

[5] M. D. Branco and D. K. Dey, "A general class of multivariate skew-elliptical distributions," *Journal of Multivariate Analysis*, vol. 79, no. 1, pp. 99–113, October 2001.

[6] A. Azzalini and A. Capitanio, "Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t-distribution," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 65, no. 2, pp. 367–389, 2003.

[7] A. K. Gupta, "Multivariate skew t-distribution," *Statistics*, vol. 37, no. 4, pp. 359–363, 2003.

[8] H. Nurminen, T. Ardeshiri, R. Piché, and F. Gustafsson, "A NLOS-robust TOA positioning filter based on a skew-t measurement noise model," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, October 2015, pp. 1–7.

[9] S. Frühwirth-Schnatter and S. Pyne, "Bayesian inference for finite mixtures of univariate and multivariate skew-normal and skew-t distributions," *Biostatistics*, vol. 11, no. 2, pp. 317–336, 2010.

[10] N. Counsell, M. Cortina-Borja, A. Lehtonen, and A. Stein, "Modelling psychiatric measures using skew-normal distributions," *European Psychiatry*, vol. 26, no. 2, pp. 112–114, 2010.

[11] M. Eling, "Fitting insurance claims to skewed distributions: Are the skew-normal and skew-student good models?" *Insurance: Mathematics and Economics*, vol. 51, no. 2, pp. 239–248, 2012.

[12] Y. V. Marchenko, "Multivariate skew-t distributions in econometrics and environmetrics," Ph.D. dissertation, Texas A&M University, December 2010.

[13] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, July 2000.

[14] P. Naveau, M. G. Genton, and X. Shen, "A skewed Kalman filter," *Journal of Multivariate Analysis*, vol. 94, pp. 382–400, 2005.

[15] H.-M. Kim, D. Ryu, B. K. Mallick, and M. G. Genton, "Mixtures of skewed Kalman filters," *Journal of Multivariate Analysis*, vol. 123, pp. 228–251, 2014.

[16] J. Rezaie and J. Eidsvik, "Kalman filter variants in the closed skew normal setting," *Computational Statistics and Data Analysis*, vol. 75, pp. 1–14, 2014.

[17] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, Aug. 1972.

[18] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*, ser. Mathematics in Science and Engineering Series. Academic Press, 1988.

[19] J. L. Williams and P. S. Maybeck, "Cost-function-based hypothesis control techniques for multiple hypothesis tracking," *Mathematical and Computer Modelling*, vol. 43, no. 9–10, pp. 976–989, May 2006.

[20] H. Nurminen, T. Ardeshiri, R. Piche, and F. Gustafsson, "Robust inference for state-space models with skewed measurement noise," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1898–1902, 2015.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[22] M. P. Wand, J. T. Ormerod, S. A. Padoan, and R. Frühwirth, "Mean field variational Bayes for elaborate distributions," *Bayesian Analysis*, vol. 6, no. 4, pp. 847–900, 2011.

[23] S. X. Lee and G. J. McLachlan, "Finite mixtures of canonical fundamental skew t-distributions – the unification of the restricted and unrestricted skew t-mixture models," *Statistics and Computing*, no. 26, pp. 573–589, 2016.

[24] T. M. Cover and J. Thomas, *Elements of Information Theory*. John Wiley and Sons, 2006.

[25] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 131–146, Nov. 2008.

[26] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, Gatsby Computational Neuroscience Unit, University College London, 2003.

[27] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum Likelihood Estimates of Linear Dynamic Systems," *Journal of the American Institute of Aeronautics and Astronautics*, vol. 3, no. 8, pp. 1445–1450, 1965.

[28] G. Tallis, "The moment generating function of the truncated multinormal distribution," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 23, no. 1, pp. 223–119, 1961.

[29] A. Genz, "Numerical computation of rectangular bivariate and trivariate normal and t probabilities," *Statistics and Computing*, vol. 14, pp. 251–260, 2004.

[30] A. Genz and F. Bretz, "Comparison of methods for the computation of multivariate t probabilities," *Journal of Computational and Graphical Statistics*, vol. 11, no. 4, pp. 950–971, 2002.

[31] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *17th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001, pp. 362–369.

[32] J. P. Cunningham, P. Hennig, and S. Lacoste-Julien, "Gaussian probabilities and expectation propagation," Arxiv, November 2013. [Online]. Available: arxiv.org/abs/1111.6832

[33] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: The MIT Press, 2012.

[34] J. M. Dow, R. Neilan, and C. Rizos, "The international GNSS service in a changing landscape of global navigation satellite systems," *Journal of Geodesy*, vol. 83, no. 7, p. 689, February 2009.

[35] Y. Bar-Shalom, R. X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation, Theory Algorithms and Software*. John Wiley & Sons, 2001.

[36] R. Piché, S. Särkkä, and J. Hartikainen, "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate Student-t distribution," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, September 2012.

[37] P. Axelrad and R. Brown, "GPS navigation algorithms," in *Global Positioning System: Theory and Applications I*, B. W. Parkinson and J. J. Spilker Jr., Eds. Washington D.C.: AIAA, 1996, ch. 9.

[38] K. L. Lange, R. J. Little, and J. M. Taylor, "Robust statistical modeling using the t distribution," *Journal of the Americal Statistical Association*, vol. 84, no. 408, pp. 881–896, December 1989.

[39] S. Särkkä and J. Hartikainen, "On Gaussian optimal smoothing of nonlinear state space models," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1938–1941, August 2010.

[40] S. K. Sahu, D. K. Dey, and M. D. Branco, "A new class of multivariate skew distributions with applications to Bayesian regression models," *Canadian Journal of Statistics*, vol. 31, no. 2, pp. 129–150, 2003.

[41] R. B. Arellano-Valle and M. G. Genton, "On fundamental skew distributions," *Journal of Multivariate Analysis*, no. 96, pp. 93–116, 2005.

[42] ——, "Multivariate extended skew-t distributions and related families," *METRON - International Journal of Statistics*, vol. 68, no. 3, pp. 201–234, 2010.

[43] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Linear Modulation Theory*. New York: John Wiley & Sons, Inc., 1968.

[44] M. Šimandl, J. Královec, and P. Tichavský, "Filtering, predictive, and smoothing Cramér–Rao bounds for discrete-time nonlinear dynamic systems," *Automatica*, vol. 37, pp. 1703–1716, 2001.

[45] T. J. Di Ciccio and A. C. Monti, "Inferential aspects of the skew t-distribution," *Quaderni di Statistica*, vol. 13, pp. 1–21, 2011.

[46] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, UK: Cambridge University Press, 2013.

[47] R. Piché, "Cramér-Rao lower bound for linear filtering with t-distributed measurements," in *19th International Conference on Information Fusion (FUSION)*, July 2016, pp. 536–540.

APPENDIX A
DERIVATIONS FOR THE SKEW-t SMOOTHER

A. Derivations for q_{xu}

Eq. (10a) gives

$$\begin{aligned}
\log q_{xu}(x_{1:K}, u_{1:K}) &= \log \mathcal{N}(x_1; x_{1|0}, P_{1|0}) \\
&+ \sum_{l=1}^{K-1} \log \mathcal{N}(x_{l+1}; Ax_l, Q) \\
&+ \sum_{k=1}^{K} \mathbb{E}_{q_\Lambda}[\log \mathcal{N}(y_k; Cx_k + \Delta u_k, \Lambda_k^{-1} R) \\
&+ \log \mathcal{N}_+(u_k; 0, \Lambda_k^{-1})] + c \qquad (45) \\
&= \log \mathcal{N}(x_1; x_{1|0}, P_{1|0}) + \sum_{l=1}^{K-1} \log \mathcal{N}(x_{l+1}; Ax_l, Q) \\
&- \frac{1}{2} \sum_{k=1}^{K} \mathbb{E}_{q_\Lambda}[(y_k - Cx_k - \Delta u_k)^{\mathrm{T}} R^{-1} \Lambda_k (y_k - Cx_k - \Delta u_k) \\
&+ u_k^{\mathrm{T}} \Lambda_k u_k] + c \qquad (46) \\
&= \log \mathcal{N}(x_1; x_{1|0}, P_{1|0}) + \sum_{l=1}^{K-1} \log \mathcal{N}(x_{l+1}; Ax_l, Q) \\
&- \frac{1}{2} \sum_{k=1}^{K} \{(y_k - Cx_k - \Delta u_k)^{\mathrm{T}} R^{-1} \Lambda_{k|K} (y_k - Cx_k - \Delta u_k) \\
&+ u_k^{\mathrm{T}} \Lambda_{k|K} u_k\} + c \qquad (47) \\
&= \log \mathcal{N}(x_1; x_{1|0}, P_{1|0}) + \sum_{l=1}^{K-1} \log \mathcal{N}(x_{l+1}; Ax_l, Q) \\
&+ \sum_{k=1}^{K} \{\log \mathcal{N}(y_k; Ax_k + \Delta u_k, \Lambda_{k|K}^{-1} R)
\end{aligned}$$

$$+\log\mathcal{N}(u_k;0,\Lambda_{k|K}^{-1})\}+c \tag{48}$$

$$=\log\mathcal{N}\left(\left[\begin{smallmatrix}x_1\\u_1\end{smallmatrix}\right];\left[\begin{smallmatrix}x_{1|0}\\0\end{smallmatrix}\right],\left[\begin{smallmatrix}P_{1|0}&\mathrm{O}\\\mathrm{O}&\Lambda_{1|K}^{-1}\end{smallmatrix}\right]\right)$$
$$+\sum_{l=1}^{K-1}\log\mathcal{N}\left(\left[\begin{smallmatrix}x_{l+1}\\u_{l+1}\end{smallmatrix}\right];\left[\begin{smallmatrix}A&\mathrm{O}\\\mathrm{O}&\mathrm{O}\end{smallmatrix}\right]\left[\begin{smallmatrix}x_l\\u_l\end{smallmatrix}\right],\left[\begin{smallmatrix}Q&\mathrm{O}\\\mathrm{O}&\Lambda_{l+1|K}^{-1}\end{smallmatrix}\right]\right)$$
$$+\log\mathcal{N}\left(y_k;\left[\begin{smallmatrix}C&\Delta\end{smallmatrix}\right]\left[\begin{smallmatrix}x_k\\u_k\end{smallmatrix}\right],\Lambda_{k|K}^{-1}R\right)+c,u_{1:K}\geq 0, \tag{49}$$

where c is a term that is constant with respect to $(x_{1:K},u_{1:K})$ but admits different values in different equations, $\Lambda_{k|K}\triangleq\mathbb{E}_{q_\Lambda}[\Lambda_k]$ is derived in Appendix A, Subsection B, and $u_{1:K}\geq 0$ means that all the components of all u_k are required to be nonnegative for each $k=1\cdots K$. Up to the truncation of the u components, $q_{xu}(x_{1:K},u_{1:K})$ has thus the same form as the joint smoothing posterior of a linear state-space model with the state transition matrix $\widetilde{A}\triangleq\left[\begin{smallmatrix}A&\mathrm{O}\\\mathrm{O}&\mathrm{O}\end{smallmatrix}\right]$, process noise covariance matrix $\widetilde{Q_k}\triangleq\left[\begin{smallmatrix}Q&\mathrm{O}\\\mathrm{O}&\Lambda_{k+1|K}^{-1}\end{smallmatrix}\right]$, measurement model matrix $\widetilde{C}\triangleq\left[\begin{smallmatrix}C&\Delta\end{smallmatrix}\right]$, and measurement noise covariance matrix $\widetilde{R}\triangleq\Lambda_{k|K}^{-1}R$. We denote the PDFs related to this state-space model with $\widetilde{p}$.

It would be possible to compute the truncated multivariate normal posterior of the joint smoothing distribution $\widetilde{p}\left(\left[\begin{smallmatrix}x_{1:K}\\u_{1:K}\end{smallmatrix}\right]|y_{1:K}\right)$, and account for the truncation of $u_{1:K}$ to the positive orthant using the sequential truncation. However, this would be impractical with large K due to the large dimensionality $K\times(n_x+n_y)$. A feasible solution is to approximate each filtering distribution in the Rauch–Tung–Striebel smoother's (RTSS [27]) forward filtering step with a multivariate normal distribution by

$$\widetilde{p}(x_k,u_k|y_{1:k})=\tfrac{1}{C}\mathcal{N}\left(\left[\begin{smallmatrix}x_k\\u_k\end{smallmatrix}\right];z'_{k|k},Z'_{k|k}\right)\cdot[\![u_k\geq 0]\!] \tag{50}$$
$$\approx\mathcal{N}\left(\left[\begin{smallmatrix}x_k\\u_k\end{smallmatrix}\right];z_{k|k},Z_{k|k}\right) \tag{51}$$

for each $k=1\cdots K$, where $[\![u_k\geq 0]\!]$ is the Iverson bracket notation, C is the normalization factor, and $z_{k|k}\triangleq\mathbb{E}_{\widetilde{p}}\left[\left[\begin{smallmatrix}x_k\\u_k\end{smallmatrix}\right]|y_{1:k}\right]$ and $Z_{k|k}\triangleq\mathrm{Var}_{\widetilde{p}}\left[\left[\begin{smallmatrix}x_k\\u_k\end{smallmatrix}\right]|y_{1:k}\right]$ are approximated using the sequential truncation. Given the multivariate normal approximations of the filtering posteriors $\widetilde{p}(x_k,u_k|y_{1:k})$, by Lemma 2 the backward recursion of the RTSS gives multivariate normal approximations of the smoothing posteriors $\widetilde{p}(x_k,u_k|y_{1:K})$. The quantities required in the derivations of Subsection B are the expectations of the smoother posteriors $x_{k|K}\triangleq\mathbb{E}_{q_{xu}}[x_k]$, $u_{k|K}\triangleq\mathbb{E}_{q_{xu}}[u_k]$, and the covariance matrices $Z_{k|K}\triangleq\mathrm{Var}_{q_{xu}}\left[\begin{smallmatrix}x_k\\u_k\end{smallmatrix}\right]$ and $U_{k|K}\triangleq\mathrm{Var}_{q_{xu}}[u_k]$.

Lemma 2. *Let $\{z_k\}_{k=1}^K$ be a linear–Gaussian process, and $\{y_k\}_{k=1}^K$ a measurement process such that*

$$z_1\sim\mathcal{N}(z_{1|0},Z_{1|0}) \tag{52a}$$
$$z_k|z_{k-1}\sim\mathcal{N}(Az_{k-1},Q) \tag{52b}$$
$$y_k|z_k\sim p(y_k|z_k), \tag{52c}$$

where $p(y_k|z_k)$ is a known distribution, and the standard Markovianity assumptions hold. Then, if the filtering posterior $p(z_k|y_{1:k})$ is a multivariate normal distribution for each k, then for each $k<K$ holds $z_k|y_{1:K}\sim\mathcal{N}(z_{k|K},Z_{k|K})$, where

$$z_{k|K}=z_{k|k}+G_k(z_{k+1|K}-Az_{k|k}), \tag{53}$$
$$Z_{k|K}=Z_{k|k}+G_k(Z_{k+1|K}-AZ_{k|k}A^{\mathrm{T}}-Q)G_k^{\mathrm{T}}, \tag{54}$$
$$G_k=Z_{k|k}A^{\mathrm{T}}(AZ_{k|k}A^{\mathrm{T}}+Q)^{-1}, \tag{55}$$

and $z_{k|k}$ and $Z_{k|k}$ are the mean and covariance matrix of the filtering posterior $p(z_k|y_{1:k})$.

Proof: The details are omitted here because the proof is mostly similar to that of [46, Theorem 8.2]. ∎

B. Derivations for q_Λ

Eq. (10b) gives

$$\log q_\Lambda(\Lambda_{1:K})=\sum_{k=1}^K\Big\{\mathop{\mathbb{E}}_{q_{xu}}\left[\log p(y_k|x_k,u_k,\Lambda_k)+\log p(u_k|\Lambda_k)\right]$$
$$+\log p(\Lambda_k)\Big\}+c. \tag{56}$$

Thus, $q_\Lambda(\Lambda_{1:K})=\prod_{k=1}^K q_\Lambda(\Lambda_k)$.

In the model with independent univariate skew-t-distributed measurement noise components (3), the diagonal entries of Λ_k are separate random variables, as given in (6c). Therefore,

$$\log q_\Lambda(\Lambda_k)$$
$$=-\frac{1}{2}\mathop{\mathbb{E}}_{q_{xu}}\big[\,\mathrm{tr}\{(y_k-Cx_k-\Delta u_k)(y_k-Cx_k-\Delta u_k)^{\mathrm{T}}R^{-1}\Lambda_k\}$$
$$+\mathrm{tr}\{u_ku_k^{\mathrm{T}}\Lambda_k\}\big]+\sum_{i=1}^{n_y}\left(\tfrac{\nu_i}{2}\log[\Lambda_k]_{ii}-\tfrac{\nu_i}{2}[\Lambda_k]_{ii}\right)+c \tag{57}$$
$$=-\frac{1}{2}\mathrm{tr}\Big\{\big[((y_k-Cx_{k|K}-\Delta u_{k|K})(y_k-Cx_{k|K}-\Delta u_{k|K})^{\mathrm{T}}$$
$$+\left[\begin{smallmatrix}C&\Delta\end{smallmatrix}\right]Z_{k|K}\left[\begin{smallmatrix}C^{\mathrm{T}}\\\Delta^{\mathrm{T}}\end{smallmatrix}\right])R^{-1}+(u_{k|K}u_{k|K}^{\mathrm{T}}+U_{k|K})\big]\Lambda_k\Big\}$$
$$+\sum_{i=1}^{n_y}\left(\tfrac{\nu_i}{2}\log[\Lambda_k]_{ii}-\tfrac{\nu_i}{2}[\Lambda_k]_{ii}\right)+c \tag{58}$$
$$=\sum_{i=1}^{n_y}\left(\tfrac{\nu_i}{2}\log[\Lambda_k]_{ii}-\tfrac{\nu_i+\Psi_{ii}}{2}[\Lambda_k]_{ii}\right)+c, \tag{59}$$

where

$$\Psi=(y_k-Cx_{k|K}-\Delta u_{k|K})(y_k-Cx_{k|K}-\Delta u_{k|K})^{\mathrm{T}}R^{-1}$$
$$+\left[\begin{smallmatrix}C&\Delta\end{smallmatrix}\right]Z_{k|K}\left[\begin{smallmatrix}C^{\mathrm{T}}\\\Delta^{\mathrm{T}}\end{smallmatrix}\right]R^{-1}+u_{k|K}u_{k|K}^{\mathrm{T}}+U_{k|K}. \tag{60}$$

Therefore,

$$q_\Lambda(\Lambda_k)=\prod_{i=1}^{n_y}\mathcal{G}\left([\Lambda_k]_{ii};\tfrac{\nu_i}{2}+1,\tfrac{\nu_i+\Psi_{ii}}{2}\right). \tag{61}$$

In the derivations of Subection A, $\Lambda_{k|K}\triangleq\mathbb{E}_{q_\Lambda}[\Lambda_k]$ is required. $\Lambda_{k|K}$ is a diagonal matrix with the diagonal elements

$$[\Lambda_{k|K}]_{ii}=\tfrac{\nu_i+2}{\nu_i+\Psi_{ii}}. \tag{62}$$

In the model (30) with multivariate skew-t-distributed measurement noise Λ_k is of the form $\lambda_k\cdot I_{n_y}$. There, λ_k is a scalar random variable, and there is just one degrees-of-freedom parameter ν, as given in (31). Therefore,

$$\log q_\Lambda(\lambda_k)$$
$$=-\frac{1}{2}\mathop{\mathbb{E}}_{q_{xu}}[\mathrm{tr}\{(y_k-Cx_k-\Delta u_k)(y_k-Cx_k-\Delta u_k)^{\mathrm{T}}R^{-1}\lambda_k\}]$$
$$-\frac{1}{2}\mathop{\mathbb{E}}_{q_{xu}}[\mathrm{tr}\{u_ku_k^{\mathrm{T}}\lambda_k\}]+\tfrac{\nu+2n_y-1}{2}\log\lambda_k-\frac{\nu}{2}\lambda_k+c \tag{63}$$
$$=\tfrac{\nu+2n_y-1}{2}\log\lambda_k-\tfrac{\nu+\mathrm{tr}\{\Psi\}}{2}\lambda_k, \tag{64}$$

where Ψ is given in (60). Thus,

$$q_\Lambda(\lambda_k) = \mathcal{G}\left(\lambda_k; \tfrac{\nu+2n_y}{2}, \tfrac{\nu+\mathrm{tr}\{\Psi\}}{2}\right). \tag{65}$$

so the required expectation is

$$\Lambda_{k|K} = \tfrac{\nu+2n_y}{\nu+\mathrm{tr}\{\Psi\}} \cdot I_{n_y}. \tag{66}$$

Appendix B
Derivation for the Fisher information of MVST

Consider the multivariate skew-t measurement model $y|x \sim \mathrm{MVST}(Cx, R, \Delta, \nu)$, where $C \in \mathbb{R}^{n_y \times n_x}$, $R \in \mathbb{R}^{n_y \times n_y}$, $\Delta \in \mathbb{R}^{n_y \times n_y}$, and $\nu \in \mathbb{R}^+$. The logarithm of the PDF of $y|x$ is

$$\begin{aligned}\log p(y|x) = &\log(2^{n_y}/\det(\Omega)^{\frac{1}{2}}) + \log \mathrm{t}(r; 0, I_{n_y}, \nu) \\ &+ \log \mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y),\end{aligned} \tag{67}$$

where $r = \Omega^{-\frac{1}{2}}(y - Cx)$ is a function of x and y, $\Omega = R + \Delta\Delta^{\mathrm{T}}$, $L = I_{n_y} - \Delta^{\mathrm{T}}\Omega^{-1}\Delta$, and $\mathrm{t}(\cdot; \mu, \Sigma, \nu)$ and $\mathrm{T}(\cdot; \mu, \Sigma, \nu)$ denote the PDF and CDF of the scaled non-central multivariate t-distribution with ν degrees of freedom. $A^{\frac{1}{2}}$ is a square-root matrix such that $A^{\frac{1}{2}}(A^{\frac{1}{2}})^{\mathrm{T}} = A$, $A^{-\frac{1}{2}} \triangleq (A^{\frac{1}{2}})^{-1}$, and $A^{-\frac{\mathrm{T}}{2}} \triangleq ((A^{\frac{1}{2}})^{-1})^{\mathrm{T}}$.

The Hessian matrix of the term $\log \mathrm{t}(r; 0, I_{n_y}, \nu)$ is derived in [47], and it is

$$\begin{aligned}&\tfrac{\mathrm{d}^2}{\mathrm{d}x^2} \log \mathrm{t}(r; 0, I_{n_y}, \nu) \\ =&\tfrac{\nu+n_y}{\nu} C^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} \left(-\tfrac{1}{1+\frac{1}{\nu}r^{\mathrm{T}}r} I_{n_y} + \tfrac{2/\nu}{(1+\frac{1}{\nu}r^{\mathrm{T}}r)^2} rr^{\mathrm{T}}\right) \Omega^{-\frac{1}{2}} C \qquad (68)\\ =&\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r} C^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} \left(-I_{n_y} + \tfrac{2}{\nu+r^{\mathrm{T}}r} rr^{\mathrm{T}}\right) \Omega^{-\frac{1}{2}} C \qquad (69)\end{aligned}$$

The term $\log \mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\frac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)$ can be differentiated twice using the chain rule

$$\tfrac{\mathrm{d}^2 \log(f)}{\mathrm{d}x^2} = \tfrac{1}{f}\tfrac{\mathrm{d}^2 f}{\mathrm{d}x^2} - \tfrac{1}{f^2}\left(\tfrac{\mathrm{d}f}{\mathrm{d}x}\right)^{\mathrm{T}} \tfrac{\mathrm{d}f}{\mathrm{d}x}, \tag{70}$$

which gives

$$\begin{aligned}&\tfrac{\mathrm{d}^2}{\mathrm{d}x^2} \log \mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y) \\ =&\left(\mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)\right)^{-1} g(r) \\ &- \left(\mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)\right)^{-2} D_r^{\mathrm{T}} P_r^{\mathrm{T}} P_r D_r,\end{aligned} \tag{71}$$

where the function g is antisymmetric because it is the second derivative of a function that is antisymmetric up to an additive constant,

$$P_r = \tfrac{\mathrm{d}}{\mathrm{d}u}\mathrm{T}(u; 0, L, \nu+n_y)\Big|_{u=\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\frac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}}, \tag{72}$$

and

$$\begin{aligned}D_r =&\tfrac{\mathrm{d}}{\mathrm{d}x}\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}} \qquad (73)\\ =&\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}}(\tfrac{1}{\nu+r^{\mathrm{T}}r} rr^{\mathrm{T}} - I_{n_y})\Omega^{-\frac{1}{2}} C. \qquad (74)\end{aligned}$$

Because the function g is antisymmetric, $\int g(r)p(r)\,\mathrm{d}y = 0$ for any symmetric function p for which the integral exists.

We now outline the proof of integrability of certain functions to show that the CRLB exists and fulfils the regularity conditions given in [43, Ch. 2.4]. The integral $\int g(r)\,\mathrm{t}(r; 0, 1, \nu)\,\mathrm{d}y$ exists because the terms of g are products of positive powers of rational expressions where the denominator is of a higher degree than the nominator and derivatives of $\mathrm{T}(u; 0, 1, \nu + n_y)$ evaluated at $\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\frac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}$, which is a bounded continuous function of y. The integral

$$\begin{aligned}&\int \left(\mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)\right)^{-1} D_r^{\mathrm{T}} P_r^{\mathrm{T}} P_r D_r \\ &\quad\times \tfrac{2}{\det(\Omega)^{\frac{1}{2}}}\mathrm{t}(r; 0, I_{n_y}, \nu)\,\mathrm{d}y\end{aligned}$$

also exists because $\left(\mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\frac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)\right)^{-1}$ and P_r are bounded and continuous and D_r is a positive power of a rational expression where the denominator is of a higher degree than the nominator. Similar arguments show the integrability of the first and second derivative of the likelihood $p(y|x)$, which guarantees that the regularity conditions of the CRLB are satisfied.

Thus, the expectation of (71) is

$$\begin{aligned}&\mathop{\mathbb{E}}_{p(y|x)}\left[\tfrac{\mathrm{d}^2}{\mathrm{d}x^2} \log \mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)\right] \\ =&\int g(r)\,\tfrac{2}{\det(\Omega)^{\frac{1}{2}}}\mathrm{t}(r; 0, I_{n_y}, \nu)\,\mathrm{d}y - \int \tfrac{2}{\det(\Omega)^{\frac{1}{2}}}\mathrm{t}(r; 0, I_{n_y}, \nu) \\ &\times \left(\mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)\right)^{-1} D_r^{\mathrm{T}} P_r^{\mathrm{T}} P_r D_r \mathrm{d}y \qquad (75)\\ =&\int 2g(r)\mathrm{t}(r; 0, I_{n_y}, \nu)\,\mathrm{d}r - \int 2\,\mathrm{t}(r; 0, I_{n_y}, \nu) \\ &\times \left(\mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)\right)^{-1} D_r^{\mathrm{T}} P_r^{\mathrm{T}} P_r D_r\,\mathrm{d}r \qquad (76)\\ =&-\mathop{\mathbb{E}}_{p(r|x)}\left[\left(\mathrm{T}(\Theta^{\mathrm{T}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)\right)^{-2} D_r^{\mathrm{T}} P_r^{\mathrm{T}} P_r D_r\right], \qquad (77)\end{aligned}$$

where $\Theta = \Omega^{-\frac{1}{2}}\Delta$, and $r|x \sim \mathrm{MVST}(0, I_{n_y} - \Theta\Theta^{\mathrm{T}}, \Theta, \nu)$ because $z \sim \mathrm{MVST}(\mu, R, \Delta, \nu)$ implies $Az \sim \mathrm{MVST}(A\mu, ARA^{\mathrm{T}}, A\Delta, \nu)$. This gives

$$\begin{aligned}&\mathop{\mathbb{E}}_{p(y|x)}\left[\tfrac{\mathrm{d}^2}{\mathrm{d}x^2} \log \mathrm{T}(\Delta^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)\right] \\ &\quad= -C^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} \mathop{\mathbb{E}}_{p(r|x)}\left[\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}\widetilde{R}_r\widetilde{R}_r^{\mathrm{T}}\right]\Omega^{-\frac{1}{2}} C,\end{aligned} \tag{78}$$

where

$$\begin{aligned}\widetilde{R}_r =&\left(\mathrm{T}(\Theta^{\mathrm{T}} r\sqrt{\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}; 0, L, \nu+n_y)\right)^{-1}(I_{n_y} - \tfrac{1}{\nu+r^{\mathrm{T}}r}rr^{\mathrm{T}})\Theta \\ &\times \left(\tfrac{\mathrm{d}}{\mathrm{d}u}\mathrm{T}(u; 0, L, \nu+n_y)\Big|_{u=\Theta^{\mathrm{T}} r\sqrt{\frac{\nu+n_y}{\nu+r^{\mathrm{T}}r}}}\right)^{\mathrm{T}},\end{aligned} \tag{79}$$

where $L = I_{n_y} - \Theta^{\mathrm{T}}\Theta$. Thus, the Fisher information for the measurement model $y|x \sim \mathrm{MVST}(Cx, R, \Delta, \nu)$ is

$$\begin{aligned}\mathcal{I}(x) =&\mathop{\mathbb{E}}_{p(y|x)}\left[-\tfrac{\mathrm{d}^2}{\mathrm{d}x^2}\log p(y|x)\right] \qquad (80)\\ =&C^{\mathrm{T}}\Omega^{-\frac{\mathrm{T}}{2}} \mathop{\mathbb{E}}_{p(r|x)}\left[\tfrac{\nu+n_y}{\nu+r^{\mathrm{T}}r}(I_{n_y} - \tfrac{2}{(\nu+r^{\mathrm{T}}r)^2}rr^{\mathrm{T}} + \widetilde{R}_r\widetilde{R}_r^{\mathrm{T}})\right]\Omega^{-\frac{1}{2}} C, \qquad (81)\end{aligned}$$

where $r|x \sim \mathrm{MVST}(0, I_{n_y} - \Theta\Theta^{\mathrm{T}}, \Theta, \nu)$, $\Theta = \Omega^{-\frac{1}{2}}\Delta$, $\Omega = R + \Delta\Delta^{\mathrm{T}}$, and $\widetilde{R}_r$ is defined in (79).

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-4010-3
ISSN 1459-2045