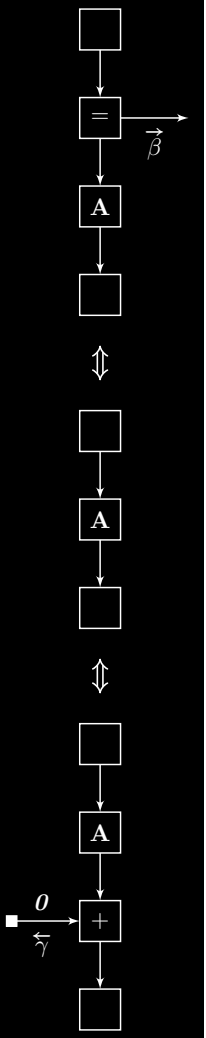
Series in Signal and Information Processing

Volume 23



Hartung Gorre Konstanz

Diss. ETH No. 20584

State-Space Methods in Statistical Signal Processing: New Ideas and Applications

A dissertation submitted to
ETH Zurich
for the degree of
Doctor of Sciences

presented by

Christoph Reller

M.Sc. ETH
born on September 4, 1973
citizen of Gsteig, BE

accepted on the recommendation of
Prof. Dr. Hans-Andrea Loeliger, examiner
Prof. Dr. Justin Dauwels, co-examiner

2012

Series in Signal and Information Processing **Vol. 23**

Editor: Hans-Andrea Loeliger

Bibliographic Information published by Die Deutsche Nationalbibliothek

Die Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the internet at http://dnb.d-nb.de.

First Edition 2013

HARTUNG-GORRE VERLAG KONSTANZ

ISSN 1616-671X
ISBN-10: 3-86628-447-0
ISBN-13: 978-3-86628-447-0

To Junia

Acknowledgments

First and foremost I express my sincere gratitude to my supervisor Hans-Andrea Loeliger. I highly appreciate his never ending support, wise guidance, and motivating style. Without him, this thesis would hardly have reached the present depth and richness. I am grateful for all the creative, highly valuable, and often surprising discussions, which have opened up a scope of research topics too wide to be described in a single text.

I deeply thank Justin Dauwels for taking the (hopefully enjoyable) burden of travelling a long way to Switzerland and co-examining my thesis. In his PhD thesis he has laid the foundations for many topics that I elaborate on in mine. Furthermore, he gave me valuable feedback, showed me new connections, and made important propositions.

As a researcher I rarely work on my own. Often, important ideas, rectifications, fundamental principles, and ways out of blind alleys emerge in a dialogue. In this sense, many of my fellow PhD students and some of my Master's thesis students have contributed to this thesis. Murthy Devarakonda has been a very supportive and open-minded office mate with whom I shared innumerable discussions, usually starting with: "Do you have some minutes?" and ending with a well-scribbled whiteboard. Additionally, he has given me valuable feedback for this thesis. I have had important and inspiring technical discussions with Ivo Trajković, Lukas Bolliger, Stefano Maranò, Lukas Bruderer, and many others, and my warm thanks go to all. Apart from my fellow researchers, I also thank the technical staff at ISI (Signal and Information Processing Laboratory) for their help and support.

Last but not least I thank my wife Karin and my kids Junia and Simon for all the love, patience, and support. Especially without Karin, I would

not have been able to put even half the power into this thesis; actually, I may not have started studying at all. Junia's life has started around the same time as the very beginnings of this thesis, but her sunshine is incomparable to any written text. Throughout this period, my parents and my brothers have always been at my side and I am grateful for this.

Abstract

This thesis is about several extensions of a general framework and about the application of these extensions to various problems arising in signal processing. The general framework is a graphical modeling technique, more precisely factor graphs, which provides the basis for the development of message passing algorithms. Such algorithms can be used to solve many statistical inference problems, most notably, estimation and detection problems.

Most of the problems addressed in this thesis are in some way linked to one or several discrete-time state-space models. While state-space representations of systems are widely used in control theory and somewhat less in statistics, a factor graph approach to such models seems to be neglected. Indeed, this thesis shows how the interaction between these two topics leads to a powerful framework for devising novel algorithms in a systematic yet uncomplicated manner.

This thesis is partitioned into two parts. The first part focuses on Gaussian message passing in linear models, and parameter estimation for such models. The second part is concerned with message-passing based computation of likelihoods or related quantities. We start with the first part.

The factor graph representation of a linear model leads to Gaussian message passing in the case of known model parameters. In a first extension we consider several variants and enhancements of recursive-least-squares-type algorithms that incorporate a forgetting factor. An application to outlier detection in a noisy quasi-periodic signal with known period is shown.

Infinite impulse response systems are treated in depth with a focus on the

real-valued Jordan canonical form. For the autonomous second-order case, analytic solutions for Gaussian messages across the whole state-space model are derived and the relation between a forgetting factor and state noise is shown. Continuous time signals and two interpolation models are touched upon.

In a further extension, the local factor graph view of three approximate inference principles (cyclic maximization, expectation maximization, and local Taylor approximation) is shown. These principles are applied to the estimation of a state-transition matrix that is given in real-valued Jordan canonical form. The same principles are used to estimate covariance matrices of a state-space model. The resulting algorithms are iterative in nature and the resulting messages are members of the exponential family. We show an application to the estimation of the time-varying fundamental frequency of a quasi-periodic signal.

The second part of this thesis starts with exposing connections between model likelihood and scale factors of sum-product messages in a factor graph. A main result is the derivation of message passing update rules for two types of such scale factors that arise in sum-product message passing. First, different types of general factors and general messages are considered. Then the setup is narrowed down to linear factors and Gaussian messages.

Since sum-product message passing is intimately connected with the computation of likelihoods, likelihood functions, and log-likelihood ratios, such quantities can be neatly expressed in terms of messages or message scale factors. The latter need, however, not in all cases be computed, and this case distinction is made precise.

Next, we consider a factor graph representation of linear state-space models augmented with an additional factor – the "glue factor" – connecting state variables of several models. This leads to the notion of a family of factor graphs parametrized by the glue factor parameters and its position on the time axis. A surprising variety of problems such as array processing and pulse modeling can be treated in this framework.

The glue factor view of likelihood computation by means of sum-product message passing leads to the novel concept of likelihood filtering. In essence, this is a message passing algorithm for computing efficiently likelihood-related quantities for each member in the family under consideration. This procedure can be considered as traditional sum-product message passing on several graphs, but without neglecting scale factors,

followed by a likelihood computation. Both offline (block based) and online algorithms are thus formulated for estimation and detection of model changes and for locating pulses-like events. Finally, we propose a hierarchical likelihood filter architecture for general signal analysis.

Keywords: State-space model, factor graph, sum-product message passing, parameter estimation, detection, recursive least squares, cyclo-stationary signal, quasi-periodic signal, frequency estimation, expectation maximization, cyclic maximization, real Jordan canonical form, variance estimation, parameter selection, hypothesis testing, glue factor, likelihood filtering, change-point estimation, hierarchical likelihood filtering.

Kurzfassung

Diese Dissertation behandelt mehrere Erweiterungen eines Ansatzes zur modellbasierten Signalverarbeitung und deren Anwendung in verschiedenen Bereichen. Der Ansatz basiert auf Faktorgrafen, einer Technik zur Modellierung von Funktionen mittels Grafen, welche die Basis für die Entwicklung von Message-Passing-Algorithmen schaffen. Solche Algorithmen können unter anderem zur Lösung zahlreicher statistischer Schätz- und Detektionsprobleme verwendet werden.

Die meisten hier betrachteten Problemstellungen sind mit einem oder mehreren zeitdiskreten linearen Zustandsraummodellen verknüpft. In der Regelungstechnik, und teilweise auch in der Statistik, sind Zustandsraum-Parametrisierungen weit verbreitet. In der Anwendung von Faktorgraftechniken auf Zustandsraummodelle scheint aber noch viel unausgeschöpftes Potenzial zu liegen. Diese Dissertation zeigt auf, wie das Verknüpfen dieser beiden Bereiche zu einem leistungsfähigen Rahmenwerk führt, mit dessen Hilfe auf einfache aber systematische Weise neuartige Algorithmen entwickelt werden können.

Die Dissertation ist in zwei Teile unterteilt. Der erste Teil beschäftigt sich mit gaussschem Message-Passing in linearen Modellen und mit der Schätzung von Parametern für ebensolche. Der zweite Teil behandelt Message-Passing-basierte Berechnungsmethoden für Likelihood-bezogene Grössen.

Faktorgrafen von linearen Zustandsraummodellen führen zu gaussschem Message-Passing, falls die Modellparameter bekannt sind. In einer ersten Erweiterung werden mehrere Varianten der rekursiven Methode der kleinsten Quadrate vorgestellt. Als Anwendungsbeispiel dient die Detektion von Ausreissern in einem verrauschten quasi-periodischen Signal mit bekannter Grundfrequenz.

Filter mit unendlicher Impulsantwort werden erläutert mit einem Fokus auf die reellwertige, jordansche Normalform. Für den Fall eines autonomen Systems zweiter Ordnung werden analytische Lösungen für gausssche Messages hergeleitet und die Beziehung zwischen einem Forgetting-Faktor und additivem Zustandsrauschen wird aufgezeigt. Zeitkontinuierliche Systeme und zwei Interpolationsmodelle werden kurz gestreift.

In einer weiteren Erweiterung wird die lokale Faktorgrafensichtweise dreier Prinzipien zur Approximation vorgestellt: zyklische Maximierung, Expectation-Maximization, und lokale Taylorreihenentwicklung. Diese Prinzipien werden auf die Schätzung der Zustandsübergangsmatrix in reellwertiger, jordanscher Normalform in einem linearen Zustandsraummodell angewendet und zur Schätzung von Kovarianzmatrizen. Die resultierenden Algorithmen sind iterative Message-Passing-Algorithmen, und die Messages gehören zur Exponentialfamilie. Eine Anwendung auf die Schätzung der zeitvarianten Grundfrequenz eines quasiperiodischen Signals wird gezeigt.

In der zweiten Hälfte dieser Dissertation liegt der Schwerpunkt auf Skalierungsfaktoren von Summenprodukt-Messages und deren Verbindung zu Methoden der Likelihood-Berechnung. Eines der Hauptresultate dieser Dissertation ist die Herleitung von Message-Passing-Aufdatierungsregeln für zwei Varianten von solchen Skalierungsfaktoren. Dazu gehört die Betrachtung allgemeiner Faktoren und Messages sowie linearer Faktoren und gaussscher Messages.

Dank der Verbindung zwischen Summenprodukt-Messages mit Likelihoods, Likelihood-Funktionen und Log-Likelihood-Quotienten können diese Grössen durch Messages oder Skalierungsfaktoren von Messages dargestellt werden. Die Berechnung von solchen Skalierungsfaktoren ist jedoch nicht immer notwendig. Es wird aufgezeigt, in welchen Situationen man sie nicht benötigt.

Des Weiteren werden Faktorgrafdarstellungen von linearen Zustandsraummodellen gezeigt, welche durch einen zusätzlichen Faktor – den "Glue-Faktor" – verbunden sind. Dies führt zum Begriff einer Familie von Faktorgrafen, welche durch den Glue-Faktor parametrisiert ist, genauer gesagt durch die Parameter des Glue-Faktors und durch seine Position auf der Zeitachse. Mit diesem Ansatz lässt sich eine erstaunliche Vielfalt von Signalen modellieren, z.B. Vektorsignale oder pulsartige Signale.

Likelihood-Berechnungen mittels Summenprodukt-Message-Passing in einer solchen Familie von Faktorgrafen führen zum neuartigen Konzept

des Likelihood-Filters. Dabei können auf effiziente Weise von Likelihoods abhängige Grössen für alle Elemente in dieser Familie berechnet werden. Die so entstandenen Algorithmen können als Summenprodukt-Algorithmen auf mehreren Grafen betrachtet werden, allerdings ohne die Vernachlässigung von Skalierungsfaktoren und mit anschliessender Likelihood-Berechnung. Sowohl Offline- als auch Onlinealgorithmen werden formuliert für die Schätzung und Detektion von abrupten Modelländerungen und für die Lokalisierung von pulsartigen Signalereignissen. Schliesslich wird ein hierarchisches Likelihood-Filter vorgeschlagen, welches für unterschiedliche Arten der Signalanalyse geeignet scheint.

Stichworte: Zustandsraummodell, Faktorgraf, Summenprodukt-Message-Passing, Schätzung, Detektion, rekursive Methode der kleinsten Quadrate, Zyklostationäre Signale, quasiperiodische Signale, Frequenzschätzung, Expectation-Maximization, zyklische Maximierung, reellwertige jordansche Normalform, Varianzschätzung, Parameterauswahl, Hypothesentest, Glue-Faktor, Likelihood-Filter, Change-Point-Schätzung, hierarchisches Likelihood-Filter

Contents

List of Symbols

Functional Notation

x	scalar
$\boldsymbol{x}$	vector or tuple
X	random variable
$\boldsymbol{X}$	random vector
$\mathbf{X}$	matrix
$\boldsymbol{0}$	zero vector
$\boldsymbol{1}$	vector with each element = 1
$\mathbf{0}$	zero matrix
$\mathbf{I}$	matrix identity
$\mathcal{X}$	a set, a hypothesis, or a model

General Mathematical Notation

$\square$	end of proof
$\lozenge$	end of example
$\setminus$	set difference
$\triangleq$	equal by definition
$\approx$	approximate equality
$\propto$	proportional
$\sim$	distributed according to the given distribution
$\overset{\text{iid}}{\sim}$	identically and independently distributed
$\otimes$	Kronecker matrix product
$\odot$	Hadamard matrix product (element-wise product)
$\mathbf{A}^{\#}$	pseudo inverse of a matrix

$\lvert\cdot\rvert$	absolute value
$\lVert\cdot\rVert$	ℓ_2 norm
$\lVert\cdot\rVert_1$	ℓ_1 norm
argmax	argument of the maximum
argmin	argument of the minimum
$\mathbb{C}$	complex numbers
cvect $\mathbf{A}$	matrix-to-vector operation by stacking matrix columns
$\delta(\cdot)$	Dirac delta, potentially multivariate
$\delta[\cdot]$	Kronecker delta, potentially multivariate
det $\mathbf{A}$	matrix determinant
diag$(\cdot)$	diagonal operator (on matrix or vector)
$\mathsf{E}[X]$	expectation
i	imaginary unit, $i = \sqrt{-1}$
lim	limit
log	logarithm
ln	natural logarithm
max	maximum
min	minimum
$\mathbb{R}$	real numbers
$\mathbb{R}_{>0}$	positive real numbers
$\mathbb{R}_{\geq 0}$	non-negative real numbers
rotm α	rotation matrix $\left[\begin{smallmatrix}\cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha\end{smallmatrix}\right]$ from scalar α
rotm $\boldsymbol{x}$	scaled rotation matrix $\left[\begin{smallmatrix}x_1 & -x_2 \\ x_2 & x_1\end{smallmatrix}\right]$ from vector $\boldsymbol{x} \triangleq \left[\begin{smallmatrix}x_1 \\ x_2\end{smallmatrix}\right]$
rvect $\mathbf{A}$	matrix-to-vector operation by concatenating matrix rows
$\mathbb{S}^n_{\succ 0}$	positive definite matrices of size $n \times n$
$\mathbb{S}^n_{\succcurlyeq 0}$	positive semi-definite matrices of size $n \times n$
$\mathbf{A}^\mathsf{T}$	matrix transpose
tr$(\cdot)$	matrix trace

Distributions

$\mathcal{N}\big(m, \sigma^2\big)$	scalar Gaussian distribution with mean m and variance σ^2
$\mathcal{N}\big(x \big\vert m, \sigma^2\big)$	scalar Gaussian PDF with mean m and variance σ^2
$\mathcal{N}(\boldsymbol{m}, \mathbf{V})$	multivariate Gaussian distribution with mean vector $\boldsymbol{m}$ and covariance matrix $\mathbf{V}$
$\mathcal{N}(\boldsymbol{x} \vert \boldsymbol{m}, \mathbf{V})$	multivariate Gaussian PDF with mean vector $\boldsymbol{m}$ and covariance matrix $\mathbf{V}$

$\mathcal{W}^{-1}(\mathbf{s}\|\nu, \mathbf{\Psi})$	inverse-Wishart PDF with ν degrees of freedom and scale matrix Ψ
$\mathcal{G}^{-1}(s\|\alpha, \beta)$	inverse gamma PDF with shape α and rate β

Decorations

$\overline{x}$	complex conjugate of a complex number x
$\hat{x}$	estimate of a random variable X or a parameter x
$\tilde{x}$	fixed value (in contrast to a variable or an unknown parameter)
$\dot{x}(t)$	time-derivative of $x(t)$
$\overset{(n)}{x}(t)$	n-th time-derivative of $x(t)$
$\mathring{\mathbf{A}}$	quantity related to a continuous-time system as opposed to a discrete-time system
$\vec{x}$	quantity related to a forward message in a factor graph
$\overleftarrow{x}$	quantity related to a backward message in a factor graph
x°	quantity related to a message that is computed in a factor graph without plugged-in observations

Problem-Specific Notation

β_X	scale factor of a pseudo-marginal
$\vec{\beta}_X$	scale factor of a forward message
$\overleftarrow{\beta}_X$	scale factor of a backward message
$f(\cdot)$	factor in a factor graph or global function of a factor graph
$g(\cdot)$	glue factor in a family of factor graphs
γ	forgetting factor
γ_X	scale factor of a pseudo-marginal
$\vec{\gamma}_X$	scale factor of a forward message
$\overleftarrow{\gamma}_X$	scale factor of a backward message
k	discrete time index
$\mu(\cdot)$	pseudo-marginal
$\vec{\mu}_X(\cdot)$	forward message
$\overleftarrow{\mu}_X(\cdot)$	backward message
$\vec{\nu}_X(\cdot)$	scaled forward message

$\overleftarrow{\nu}_X(\cdot)$	scaled backward message
$\nu_X(\cdot)$	scaled pseudo-marginal
$p(\cdot)$	a PDF or PMF
$p(\cdot\|\cdot)$	conditional PDF or PMF
$\overrightarrow{p}_X(\cdot)$	scaled forward message
$\overleftarrow{p}_X(\cdot)$	scaled backward message
t	continuous time variable

Acronyms

ARMA	autoregressive moving average
CM	cyclic maximization
DFT	discrete Fourier transform
EM	expectation maximization
FIR	finite impulse response
GLLR	generalized log-likelihood ratio
IIR	infinite impulse response
LLR	log-likelihood ratio
LTI	linear time-invariant
LTV	linear time-varying
MAP	maximum a posteriori
ML	maximum likelihood
PDF	probability density function
PMF	probability mass function
RLS	recursive least squares
SSM	state-space model

Chapter 1

Introduction

In simple terms, a signal is "data arranged along a time axis", and signal processing is concerned with extracting useful information from this data. Over the last decades signal processing has become increasingly important and nowadays finds a vast field of applications. Also, with the availability of increasingly powerful computers, signal processing systems and algorithms have become more and more complex. This heightens the need for structured methodical approaches.

Clearly, every signal processing algorithm makes assumptions about the structure of the signal to be analyzed. In many approaches, these assumptions stay hidden or unclear. In statistical signal processing, however, the signal structure is captured in the form of a probabilistic model. One of the great successes of probability theory and statistics is the thorough formulation of estimation and detection problems.

An important class of signal models are state-space models (SSMs). Such models are by definition structured along a time axis and hence they are especially well suited for signal processing. A SSM usually assumes an evolution of a hidden state and a process that connects this state with the observed signal.

One of the most powerful approaches to SSMs is based on factor graphs, one of several variants of graphical modeling techniques. Indeed, a whole toolbox for developing Gaussian message passing algorithms for linear SSMs has emerged from this interaction [65]. Yet, this alliance still bears great potential, which until now seems to be unexploited.

In this thesis, the toolbox is extended into several directions. A common starting point for many algorithms developed in this thesis is Gaussian message passing in linear models. On one line of attack we consider

situations in which Gaussianity is not preserved, most notably in the case of unknown model parameters. On this line we strive for iterative algorithms that solve an approximation of the original problem but still work in the Gaussian domain, or at least in the domain of the exponential family.

On another line we study scale factors of sum-product messages. A message in a factor graph is a non-negative function which can be represented by its shape and a pre-factor – the scale factor. We ask ourselves the question what happens to such scale factors as we do sum-product message passing. In particular, scale factors are intimately linked to likelihoods, likelihood functions, and log-likelihood ratios (LLRs), which in turn are important quantities for estimation and detection problems.

Furthermore, the factor graph view allows us to formulate the powerful concept of a glue factor that expresses some addition (initial, final, or otherwise "local") condition in one or several SSMs. This concept is used to formulate a generic algorithmic framework for the computation of likelihood-related quantities via sum-product message passing for a whole family of models at once.

The factor graph toolbox, now augmented by our extensions, can be used in a whole variety of statistical inference problems, most notably estimation and detection problems. In particular, we have the opportunity of deriving efficient algorithms for problems that, without the toolbox, seem to be very complicated if not intractable.

1.1 Background

The use of graphical models in topics of statistical modeling has by now a considerable history. First and foremost, factor graphs have emerged in coding theory. The essential ideas of the sum-product algorithm as applied to low-density parity-check codes date back to the 1960s [37]. It was however not until the 1980s that first interpretations of this algorithm in the form of Tanner graphs were considered [101], and only in the 1990s with the advent of turbo codes [6], these styles of iterative decoding became mainstream. Soon it became evident that a generalization of Tanner graphs encompasses many algorithms previously known under different names [114], such as the Viterbi algorithm [34] and the BCJR algorithm [4].

In machine learning, Bayesian networks and Markov random fields have a long tradition (cf. references in [8, 57]). The latter originally have emerged in statistical physics [98]. Also, the signal processing community has achieved many basic results in the domain of hidden Markov models, most notably the forward-backward algorithm and its expectation maximization (EM) interpretation [27]. In quantum mechanics, graphical notations have appeared in the form of Feynman diagrams and later as trace diagrams (or tensor diagrams) [83]. In control theory, Kalman filtering and recursive least squares (RLS)-type algorithms have been developed without any graphical interpretation [51, 94].

In this thesis we use a style of factor graphs due to Forney [35], which subsumes all the graphical models mentioned above as special cases. Forney graphs have been used to paint a unified picture of many algorithms as instances of message passing algorithms [60, 63, 66]. Most notably, this thesis stands on the foundations of Gaussian factor graphs as reviewed in [65].

Recently, algorithms that are not of a sum-product type or a max-product type have been described from a graphical modeling perspective, most notably EM-type algorithms [19, 26], algorithms based on variational approximations [25, 49], generalized belief propagation [116], and expectation propagation [75]. For this thesis, the motivation to use such approaches is the possibility of formulating algorithms that solve approximations to difficult statistical problems without leaving the domain of messages that belong to the exponential family.

Graphical models have been used in ways that differ largely from the Gaussian setting assumed for many parts of this thesis. Most notably, sampling based methods [102] and particle methods [21, 22] have shown a great potential. In this thesis these methods are not considered further.

In the field of classical signal processing, short-time transforms and various time-frequency representations are common [74]. More generally, wavelets have been applied in many signal processing areas [69]. Traditionally, these short-time representations are block-based methods or they have filter-bank like structures. In this thesis we will consider a general (nonlinear) filtering view that does away with block-based processing. Also, we will be particularly interested in a short-time Fourier transform of exponentially weighted signals that goes back to [103].

Statistical signal processing stands on the traditional foundation of estimation theory and detection theory [53, 54, 84, 107]. While SSMs do

feature in this field [31, 50, 93], the application of graphical models seems to be underused. In this thesis we try to contribute towards mending this deficit, at least for linear Gaussian models.

1.2 Motivation

There are several ways in which we would like to motivate this thesis. First and foremost we believe that the unifying view of a factor graph approach bears great potential for gaining insight, identifying fundamental mathematical structures, devising meaningful approximations, and designing fast algorithm implementations for a whole range of problems.

This is the main motivation to engage in the question: What happens to scale factors of the messages in sum-product message passing? As we will see, this question is intimately interlinked with likelihood computation.

The exponential family of probability density functions (PDFs), and most notably the Gaussian distribution, has many special properties [55, 111]. This is our main motivation to propose approximations to settings that would otherwise lead to more complicated messages in factor graphs. All the approximations used in this thesis result in messages that are members of the exponential family. For many practical applications these approximations are sufficiently accurate.

While SSMs are in general quite common in signal processing, the potential of the factor graph view remains largely unexploited. We believe that state-space related algorithms are cleaner and more appealing when viewed from a graphical modeling point of view. Furthermore, as we will see, this view allows us to formulate approximations and extensions that otherwise are not easily conceived.

In this thesis we focus on infinite impulse response (IIR) systems as opposed to finite impulse response (FIR) systems. We believe that many signals can be modeled more effectively using IIR systems for two reasons. First, IIR systems can have a continuous-time equivalent and thus facilitate the modeling of (potentially non-uniformg) samples of continuous-time phenomena. Second, in IIR systems, the time scale of signal features does not depend strongly on the system order, i.e., low-frequency features can be modeled with low system order as opposed to FIR systems, for which the order may increase dramatically.

Last but not least, we highlight that forward-processing algorithms can

have several advantages over block-based computation. Not only is the coupling across time inherently exploited, but also one can envisage implementations of these algorithms with analog electronics. In this thesis we formulate online (forward-only as well as smoothing) algorithms along with block-based algorithms.

1.3 Contributions

In this section we summarize the points in which this thesis provides new insights, models, and algorithms. In the domain of linear SSMs these are the following:

- We provide an RLS model, in which a regularization factor has been distributed over all time steps, and we show how to choose the parameters in order to obtain the same regularization effect as in the non-distributed case. While for classical RLS (forward message passing only) the resulting algorithm is known [44], the forward-backward algorithm formulated in this thesis is novel.

- An application of Fourier series in a model for slowly time-varying periodic signals with known fundamental frequency is presented. This model is used in an RLS-type algorithm for de-noising and detecting outliers. In a non-uniform sampling scenario, this approach overcomes the deficiencies of a cyclic autoregressive moving average (ARMA) model.

- The exact connections between models that incorporate a forgetting factor [64] and autonomous models (models without state noise) are shown. The steady state for these systems is characterized and derived. While this connection apparently has not been discovered before, steady-state solutions of such systems have been known for a long time [50].

- In the case of a second-order autonomous system with a complex pole pair, the Gaussian messages through the whole SSM are derived in analytic form. Evident connections with Fourier transforms are pointed out.

- We show an equivalence between spline signal processing and a continuous-time system with noisy discrete-time observations. (Cf.

[10–12] for a factor-graph representation of such systems.) In this case, the equivalent system happens to be a noisy variant of a continuous-time SSM for polynomials. We make an equivalent formulation for a continuous-time SSM for sinusoids.

- We show a way to split a high-dimensional SSM into smaller second-order SSMs. A general approach is derived for transiting from a completely decoupled treatment of these subsystems to a completely coupled treatment in an iterative algorithm.

We contribute the following to the topic of parameter estimation in linear SSMs:

- In this thesis, IIR systems are parametrized in the real Jordan canonical state-space form [45]. We apply cyclic maximization (CM), EM, and linear approximation to the problem of estimating the systems state-transition matrix in Jordan canonical form. A complete derivation is given and the results are presented in Gaussian message passing form. Interesting connections between the three approaches mentioned show up in these results. We use these findings to estimate quasi-periodic signals with unknown (time-varying) fundamental frequency.

- We review the application of EM and local Taylor approximation to the problem of variance (and covariance matrix) estimation in a factor graph setting.

In Part II of this thesis we make the following contributions to the topics of message scale factors and likelihood-related computation:

- We propose to define two types of scale factors for general real-valued sum-product messages based on the normalization factor for scaled PDFs and on the characteristic function of a random variable. While both notions on their own have been extensively studied [81], this seems to be the first joint treatment in a factor graph setting.

- For the proposed scale factors of sum-product messages, update rules are derived for propagating scale factors through the following factor graph nodes:

  - General node in the form of a conditional PDF for general real-valued messages.
  - Linear constraint nodes for general real-valued messages.
  - Linear constraint nodes for Gaussian messages.
  - Linear constraint composite blocks for Gaussian messages.

- We formulate the resulting connection between scale factors of sum-product messages and the likelihood of a given observation for a statistical model. While this in itself is not new [33], the specific scale factor viewpoint at hand seems to be unknown. We review some maximum likelihood (ML) estimation problems and detection problems from this viewpoint.

- We highlight a connection between a Bayesian approach and a detection-based approach to estimating input noise variance or the regularization parameter in any linear SSM that incorporates distributed regularization. While the basic connection is certainly not novel [7], the application to SSMs is.

- The notion of a glue factor has been proposed in [64] and some applications have been studied in [28, 29] in the setting of forward-only processing. In this thesis we elaborate more in-depth on this concept. We show how the removal of a prior bias on the glue factor position can be achieved by re-normalization or by re-parameterization of the glue factor. Furthermore, a framework for learning glue factor parameters is proposed.

- The glue factor approach is used to obtain LLRs for array processing [70–72, 88] in a more elegant way than with traditional methods. As a second application, we show how to construct a glue factor based model for sinusoidal pulses decaying both towards the past and the future.

- We define a general likelihood filtering algorithm and formulate a multitude of general ML estimation and detection tasks from this angle. The findings are used to formulate algorithms that find sparse model changes, both for block-based and online processing.

- Finally, we propose a hierarchical likelihood filtering system that bears some similarity with recurrent neural network, but can be understood as a generalization of the previously formulated likelihood filter.

1.4 Outline

This thesis is divided into three main parts:

- Part I concerns RLS-type problems, linear SSM related topics, and parameter estimation of such models.
- Part II exposes fundamental connections between likelihood and scale factors of sum-product messages in factor graphs. In the domain of SSMs, the related notions of a glue factor, likelihood filtering are treated.
- The third part contains some of the proofs for Parts I and II and a sketch of the linear-algebra interpretation of factor graphs.

A more detailed outline of Parts I and II is given below. Notational conventions and preliminaries are supplied after this outline.

Part I – Linear State-Space Models

This Part starts with Chapter 2 by reviewing least squares problems from a Gaussian factor graph perspective. We give the connection to statistical models and ML estimation, and present some examples. Classical RLS algorithms are extended to incorporate two-sided regularization and an equivalent formulation using distributed regularization is derived. Potentially, RLS finds application in a plethora of areas. We showcase an example application to detecting outliers in a periodic signal of unknown and slowly time-varying shape but with known fundamental frequency.

In Chapter 3, discrete-time linear SSMs with Gaussian input and Gaussian additive observation noise are brought into picture. The main focus lies on IIR models in real Jordan canonical form, autonomous models, and models with a forgetting factor. For such systems, the filtering view of forward Gaussian message passing and the steady state are studied. We discuss second-order systems and the connection to Fourier transforms. Next we elaborate on continuous-time systems with discrete-time noisy observations, for which we treat the case of noisy polynomials and noisy sinusoids. Finally, the notion of state-space splitting is touched upon.

Chapter 4 contains topics in parameter estimation for linear Gaussian SSMs. Among the many possible general principles, three are picked out:

CM, EM, and local Taylor approximation. The local factor graph view of these principles is presented. They are applied to the estimation of the system poles in real Jordan canonical form. One advantage of this form is, that it closely relates to a model for quasi-periodic signals with slowly time-varying shape and fundamental frequency. We show an example application to estimation of this time-varying fundamental frequency. Finally, EM and Taylor approximation are applied to the estimation of variances and covariance matrices.

Part I of this thesis is concluded in Chapter 5.

Part II – Glue Factors and Likelihoods

We start our view on the connection between likelihood computation and sum-product message passing in Chapter 6. In this chapter, the fundamentals are laid out for computing scale factors of sum-product messages in factor graphs. Our local view of this computation is presented by introducing two types of such scale factors, for which we tabulate update rules for certain factors. We start with update rules that are valid for general factors and general real-valued messages and continue with linear factors and Gaussian messages.

In the same chapter, we describe the connection between sum-product message scale factors and the notion of likelihood in a statistical model by introducing a general computation method. We give a view on how to approximate this method locally in a factor graph, and we exemplify situations in which scale-factor computation is not needed. This chapter ends with drawing a connection between a Bayesian view and a detection-based view on model selection in a SSM setting.

In Chapter 7 we delve into the notion of a glue factor, a factor in the factor graph that connects two (or more) SSMs. We give a formal definition of the induced model family, which is parameterized by the position of the glue factor on the time axis and by the glue factor parameters. Based on this model family we explain our notion of likelihood filtering, a message-passing view of the computation of likelihood-related quantities.

First, we elaborate on parameter estimation for fixed glue factor positions and treat in depth an application to array processing. As a further example, we present a way of modeling pulses by two-sided exponentially decaying sinusoids. Second, the estimation of the glue factor position on the time axis is studied, and applications to estimating model-change

positions and pulse-like events are given. In the last part of this chapter, the above is extended to a detection scenario, in which the presence of a glue factor is treated in a statistical decision setting. An extension to detecting the presence of several glue factors and online estimation is outlined.

Chapter 8 gives a first sketch on a proposed extension of likelihood filtering. In this extension, a population of second-order linear SSMs is used in conjunction with a population of glue factors all of which potentially have access to all the model states. For each glue factor, a posterior probability is computed in an forward-only fashion. Reusing these posterior probabilities as observations in the SSMs makes this system hierarchical.

In Chapter 9 we conclude Part II of this thesis and provide an outlook.

Finally, in the last appendix, we present the seemingly unrelated, but all the more beautiful, topic of representing linear-algebra expressions by means of factor graphs. Indeed, we use this framework to derive expressions which are used in the proofs.

1.5 Preliminaries and Notation

1.5.1 Notation

We write matrices in boldface ($\mathbf{X}$, $\mathbf{\Lambda}$) and vectors in italic boldface ($\boldsymbol{x}$, $\boldsymbol{\lambda}$). We loosely stick to the rule of using uppercase for matrices and random variables.

For ease of notation, we define that all integrals lacking lower and upper limits are assumed to go over the whole range of the variables. For example for some vector $\boldsymbol{x} \in \mathbb{R}^n$ and some function $f(\boldsymbol{x})\colon \mathbb{R}^n \to \mathbb{R}$, by

$$\int f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}$$

we mean

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(\boldsymbol{x}) \,\mathrm{d}x_1 \cdots \mathrm{d}x_n \,.$$

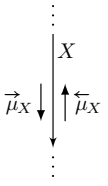
In case we are dealing with indefinite integrals, it will be clear from the context or it will be indicated explicitly in the text.

We use the symbol p to denote a both PDF or a probability mass function (PMF). We use the symbol f for global functions and local factors in factor graphs.

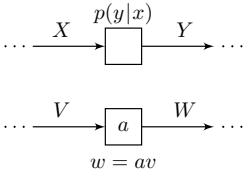
1.5.2 Factor Graph Notation

We use Forney factor graphs in the style of [65] with several added notational conventions and extensions.

- Dots in a factor graph represent parts not shown explicitly, either because they are irrelevant for the discussion or because they are evident continuations of the graph.
- We say that a local factor f represents a (conditional) PDF p if f is proportional to p with a finite constant of proportionality.
- We always draw edges with arrows although factor graphs are undirected graphs by definition. The sole reason for doing this is for notational ease. The direction of an edge (e.g. X) allows us to write the forward message (in the same direction as the edge) as $\vec{\mu}_X$ and the backward message (in the opposite direction) as $\overleftarrow{\mu}_X$.



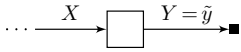
- Often, we choose the direction of an edge according to the following convention. If a factor represents a conditional PDF (or a conditional PMF), then the edges of all conditioning variables point towards the factor. All remaining edges point away from the factor. In other words, we choose the direction in the same way as in Bayesian networks, if we can. Moreover, linear constraint nodes such as addition nodes and multiplication nodes are not defined unless the node ports (the points at which edges can attach) are designated properly.



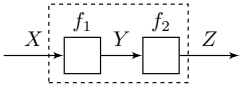
The above convention cannot be followed strictly. Specifically, we will violate it in at least two places: equality nodes in glue factors and equality nodes in the representation of linear algebra

expressions. For the latter, we generally believe that directed edges are actually not appropriate. Instead, one might envisage a different way of labeling the node ports.

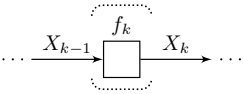
- For better readability we draw small filled boxes for edges that are fixed to a specific value. In the example shown, Y is fixed at $Y = \tilde{y}$ while X is not.



- Dashed boxes are assumed to be "closed" using the sum-product (or the max-product) rule, i.e., all internal variables are integrated (maximized) over. The example factor graph shown thus represents the function $\int f_1(x, y)\, f_2(y, z)\, \mathrm{d}y$.



- We often have to deal with factor graphs that consist of multiple identical, or analogous time slices. For the sake of definiteness we draw densely dotted and rounded braces to indicate such slices. In this context, the triple dots (ellipses) represent repetitions of the shown slice. In fact, this is a variant of the so-called "plate notation" [57].



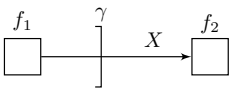
- We define the action of a forgetting factor $\gamma \in [0, 1]$ in the same way as in [64]. Consider the factorization

$$f(x) = f_1(x)\, f_2(x)\,,$$

where each factor might be a marginal of a more detailed graph. Based on this graph we draw the graph for the factorization

$$f(x) = f_1(x)^\gamma\, f_2(x)$$

as shown on the right. In other words we use the symbols $\overset{\gamma}{]}$ and $\overset{\gamma}{[}$ to indicate that every factor behind the bracket (i.e. to the left and to the right of the bracket respectively) is taken to the power of γ.



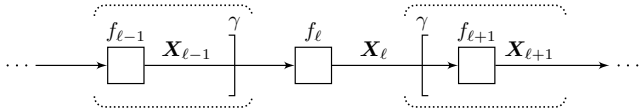


Figure 1.1: Example factor graph.

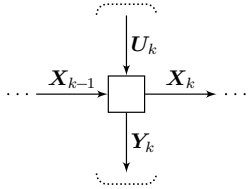


Figure 1.2: Factor graph representations of a general state-space model (SSM) (1.3).

We illustrate some of the introduced notation in the following example. Let X_k for $k \in \mathbb{Z}$ be the state of a SSM and let the PDF of this model factor as

$$p(\dots, x_{-1}, x_0, x_1, \dots) \propto \prod_{k\in\mathbb{Z}} f_k(x_{k-1}, x_k)\,. \tag{1.1}$$

Starting from this model we define for some $\ell \in \mathbb{Z}$ a new model

$$p_\ell(\dots, x_{-1}, x_0, x_1, \dots) \propto \prod_{k\in\mathbb{Z}} f_k(x_{k-1}, x_k)^{\left(\gamma^{|k-\ell|}\right)}\,. \tag{1.2}$$

With the use of the conventions mentioned, we can draw the factor graph of (1.2) as shown in Figure 1.1.

1.5.3 State-Space Models

We define a discrete-time SSM with state $\boldsymbol{x}_k$, input $\boldsymbol{u}_k$, and output $\boldsymbol{y}_k$ by the following equations:

$$\begin{aligned} \boldsymbol{x}_k &= f_{A_k}(\boldsymbol{x}_{k-1}) + f_{B_k}(\boldsymbol{u}_k) \\ \boldsymbol{y}_k &= f_{C_k}(\boldsymbol{x}_k)\,. \end{aligned} \tag{1.3}$$

The model is *time-invariant* if $f_{A_k} = f_A$, $f_{B_k} = f_B$, and $f_{C_k} = f_C$ for all $k \in \mathbb{Z}$. A SSM is *linear* if there exist $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$ such that we

can write

$$\begin{aligned} \boldsymbol{x}_k &= \mathbf{A}_k\, \boldsymbol{x}_{k-1} + \mathbf{B}_k\, \boldsymbol{u}_k\,, \\ \boldsymbol{y}_k &= \mathbf{C}_k\, \boldsymbol{x}_k\,. \end{aligned} \tag{1.4}$$

Often, we will deal with Gaussian statistical models

$$\begin{aligned} \boldsymbol{X}_k &= \mathbf{A}_k\, \boldsymbol{X}_{k-1} + \mathbf{B}_k\, \boldsymbol{U}_k \\ \boldsymbol{Y}_k &= \mathbf{C}_k\, \boldsymbol{X}_k + \boldsymbol{Z}_k\,, \end{aligned} \tag{1.5}$$

where $\boldsymbol{U}_k \sim \mathcal{N}(\boldsymbol{m}_{U_k}, \mathbf{V}_{U_k})$, and $\boldsymbol{Z}_k \sim \mathcal{N}(\boldsymbol{0}, \mathbf{V}_{Z_k})$. Such a SSM is *linear time-invariant* (LTI) if $\mathbf{A}_k = \mathbf{A}$, $\mathbf{B}_k = \mathbf{B}$, $\mathbf{C}_k = \mathbf{C}$, $\mathbf{V}_{U_k} = \mathbf{V}_U$, and $\mathbf{V}_{Z_k} = \mathbf{V}_Z$ for all $k \in \mathbb{Z}$. We call the SSM *autonomous* if the input is absent ($\mathbf{B}_k = 0$ for all $k \in \mathbb{Z}$). The factor graph representation of a general SSM is depicted in Figure 1.2.

Part I

Linear State-Space Models

"Euer Leben bildet nur menschlich, so habt ihr genug getan: aber die Höhe der Kunst und die Tiefe der Wissenschaft werdet ihr nie erreichen ohne ein Göttliches."

"Form your life humanly, and you have done enough: but you will never reach the height of art and the depth of science without something divine."

Friedrich von Schlegel (1772–1829)

Chapter 2

Regularized Recursive Least Squares

2.1 Introduction

In this chapter we lay out our view of some classical least-squares problems and extensions thereof. The aim is not to give an exhaustive overview of this vast topic, but to introduce notions that will be used later and to highlight examples of interest on the way.

We start by making some general comments on the connection between cost functions and statistical models, and we expose the factor graph view on recursive least squares (RLS). Next, the topic of regularization, both in a centralized and a distributed manner, is treated. This chapter is ended with an application of the presented methods in which a slowly changing periodic signal with known fundamental frequency is modeled. An implementation of this algorithm is used to estimate rain rates from microwave link gain measurements.

2.2 Cost Functions and Statistical Models

We define a cost function to be a mapping

$$\kappa(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) \colon \mathbb{R}^{n_X \times n_Y \times n_\theta} \to \mathbb{R}_{\geq 0}\,, \tag{2.1}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_X}$ is a variable vector, $\boldsymbol{y} \in \mathbb{R}^{n_Y}$ is a vector of observable values, and $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is a parameter vector. Many optimization problems

treated in this chapter have the form

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^{n_X}}{\operatorname{argmin}}\, \kappa(\boldsymbol{x}, \tilde{\boldsymbol{y}}, \boldsymbol{\theta})\,, \tag{2.2}$$

for some cost function κ, fixed observed values $\boldsymbol{y} = \tilde{\boldsymbol{y}}$, and chosen parameter (vector) $\boldsymbol{\theta}$. Often, $\tilde{\boldsymbol{y}}$ and $\boldsymbol{\theta}$ will be omitted in the notation.

One way to formulate a corresponding statistical model is by defining the exponentiated cost function

$$f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}') \triangleq e^{-\kappa(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta})/(2\sigma_Z^2)} \tag{2.3}$$

where $\boldsymbol{\theta}' \triangleq (\boldsymbol{\theta}, \sigma_Z)$ is an augmented parameter vector and $\sigma_Z > 0$ can be chosen arbitrarily. We additionally assume that either

$$\zeta^{\boldsymbol{\theta}} \triangleq \iint f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} \tag{2.4}$$

or

$$\zeta^{\boldsymbol{\theta},\boldsymbol{x}} \triangleq \int f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{y} \tag{2.5}$$

is positive and constant. Notationally, we use the construct $\zeta^{\boldsymbol{\theta},\boldsymbol{x}}$ to indicate that this value is a constant although no integration over $\boldsymbol{\theta}$ and $\boldsymbol{X}$ has been done. We can formulate two corresponding statistical models as either

$$p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) = f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})/\zeta^{\boldsymbol{\theta}} \tag{2.6}$$

or

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) = f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})/\zeta^{\boldsymbol{\theta},\boldsymbol{x}} \tag{2.7}$$

respectively. In this chapter, all cost functions will be integrable, in the sense of (2.4) or (2.5). With these statistical models, the optimization (2.2) can equivalently be formulated as a maximum a posteriori (MAP) estimation problem

$$\hat{\boldsymbol{x}}_{\mathrm{MAP}} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, p(\boldsymbol{x}, \tilde{\boldsymbol{y}}|\boldsymbol{\theta})\,, \tag{2.8}$$

or as an ML problem

$$\hat{\boldsymbol{x}}_{\mathrm{ML}} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, p(\tilde{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{\theta})\,. \tag{2.9}$$

In addition, for each of these two models, an ML estimate of $\boldsymbol{\theta}$ can be formulated as

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \int p(\boldsymbol{x}, \tilde{\boldsymbol{y}}|\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{x} \tag{2.10}$$

or

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \max_{\boldsymbol{x}} p(\tilde{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{\theta}) \tag{2.11}$$

$$= \operatorname*{argmin}_{\boldsymbol{\theta}} \min_{\boldsymbol{x}} \kappa(\boldsymbol{x}, \tilde{\boldsymbol{y}}, \boldsymbol{\theta}) \tag{2.12}$$

respectively.

2.3 Least Squares and Maximum Likelihood Estimation

For the classical least squares problem, the cost function is

$$\kappa(\boldsymbol{x}) \triangleq (\boldsymbol{y} - \mathbf{C}\boldsymbol{x})^{\mathsf{T}}(\boldsymbol{y} - \mathbf{C}\boldsymbol{x}) = \sum_{k=1}^{K} (y_k - \boldsymbol{c}_k\boldsymbol{x})^2\,, \tag{2.13}$$

where the given data consists of a vector

$$\boldsymbol{y} \triangleq \big[y_1, \ldots, y_K\big]^{\mathsf{T}} \in \mathbb{R}^K \tag{2.14}$$

and a matrix

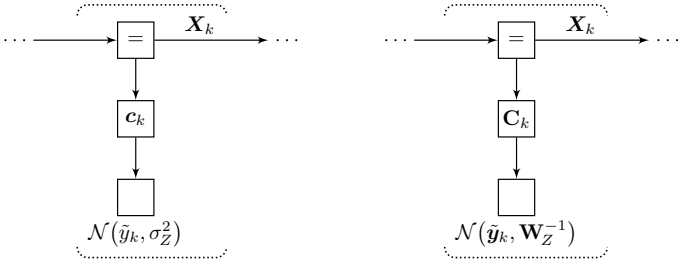
$$\mathbf{C} \triangleq \big[\boldsymbol{c}_1^{\mathsf{T}}, \ldots, \boldsymbol{c}_K^{\mathsf{T}}\big]^{\mathsf{T}} \in \mathbb{R}^{n\times K} \tag{2.15}$$

containing row vectors $\boldsymbol{c}_k$.

The equivalent statistical model can be derived from (2.3) and (2.7) as

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{k=1}^{K} \mathcal{N}\big(y_k\,\big|\,\boldsymbol{c}_k\boldsymbol{x}, \sigma_Z^2\big) \propto \prod_{k=1}^{K} e^{-(y_k - \boldsymbol{c}_k\boldsymbol{x})^2/(2\sigma_Z^2)}\,. \tag{2.16}$$

Given fixed observations $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$, the factorization (2.16) can be represented by the factor graph shown in Figure 2.1a by introducing additional random variables. ML estimation of $\boldsymbol{X}$ is done by forward message passing in this factor graph [65].



(a) Scalar observations (b) Vector observations

Figure 2.1: Factor graph representations of least squares (2.16).

More generally, the cost function for vector observations

$$\boldsymbol{y} \triangleq \left[\boldsymbol{y}_1^\mathsf{T}, \ldots, \boldsymbol{y}_K^\mathsf{T}\right]^\mathsf{T} \in \mathbb{R}^{K\,n_Y} \tag{2.17}$$

and some positive definite matrix $\mathbf{W}_Z \in \mathbb{S}_{\succ 0}^{n_Y}$ is

$$\kappa(\boldsymbol{x}) = \sum_{k=1}^{K} (\boldsymbol{y}_k - \mathbf{C}_k\,\boldsymbol{x})^\mathsf{T} \mathbf{W}_Z (\boldsymbol{y}_k - \mathbf{C}_k\,\boldsymbol{x})\,, \tag{2.18}$$

where

$$\mathbf{C} \triangleq \left[\mathbf{C}_1^\mathsf{T}, \ldots, \mathbf{C}_K^\mathsf{T}\right]^\mathsf{T} \in \mathbb{R}^{n_X \times K n_Y}\,. \tag{2.19}$$

The equivalent statistical model then is

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{k=1}^{K} \mathcal{N}\big(\boldsymbol{y}_k \big| \mathbf{C}_k\,\boldsymbol{x}, \mathbf{W}_Z^{-1}\big) \tag{2.20}$$

$$\propto \prod_{k=1}^{K} e^{-(\boldsymbol{y}_k - \mathbf{C}_k\,\boldsymbol{x})^\mathsf{T} \mathbf{W}_Z (\boldsymbol{y}_k - \mathbf{C}_k\,\boldsymbol{x})/2} \tag{2.21}$$

and the corresponding factor graph for given observations $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ is depicted in Figure 2.1b. All models in this chapter can be generalized to vector observations in this way. We refrain from doing so for ease of exposition.

Example 2.1: ARMA Filter Identification [58]
An autoregressive moving average (ARMA) process $Y_1, Y_2, \ldots$ is defined by

$$Y_k = a_0 + \sum_{\ell=1}^{L} a_\ell Y_{k-\ell} + \sum_{m=0}^{M} b_m u_{k-m} + Z_k \,, \tag{2.22}$$

where u_k is a known input signal, $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$ is additive white Gaussian noise, and $Y_{-L}, \ldots, Y_0$ are distributed such that the process is stationary. Given observations $Y_k = \tilde{y}_k$ for $k = 1, \ldots, K$ we want to make ML estimates $\hat{\boldsymbol{x}}_{\text{ML}}^{\mathsf{T}} \triangleq \left[\hat{\boldsymbol{a}}^{\mathsf{T}}, \hat{\boldsymbol{b}}^{\mathsf{T}}\right]$ of the filter coefficients $\boldsymbol{a}^{\mathsf{T}} \triangleq \left[a_0, \ldots, a_L\right]^{\mathsf{T}}$, and $\boldsymbol{b}^{\mathsf{T}} \triangleq \left[b_0, \ldots, b_M\right]^{\mathsf{T}}$. This can be done with forward message passing in the graph of Figure 2.1a by defining

$$\boldsymbol{c}_k \triangleq \left[1, \tilde{y}_{k-1}, \ldots, \tilde{y}_{k-L}, u_k, \ldots, u_{k-M}\right] . \tag{2.23}$$

The final estimate is $\hat{\boldsymbol{x}} = \overrightarrow{\boldsymbol{m}}_{X_K}$, i.e., the mean of the forward message on the last edge X_K. ◊

Example 2.2: Periodic ARMA Filter Identification
Example 2.1 can be adapted to account for periodically time-varying coefficients a_ℓ and b_m. Specifically, an N-periodic ARMA process $Y_1, Y_2, \ldots$ is defined as

$$Y_k = a_{0,n} + \sum_{\ell=1}^{L} a_{\ell,n} Y_{k-\ell} + \sum_{m=0}^{M} b_{m,n} u_{k-m} + Z_k \,, \tag{2.24}$$

where u_k is the known input, $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$, and $n \triangleq k \bmod N$ is the remainder of the division k/N. Given observations $Y_k = \tilde{y}_k$, we want to make ML estimates $\hat{\boldsymbol{x}}_{\text{ML}}^{\mathsf{T}} \triangleq \left[\hat{\boldsymbol{a}}_0^{\mathsf{T}}, \hat{\boldsymbol{b}}_0^{\mathsf{T}}, \ldots, \hat{\boldsymbol{a}}_{N-1}^{\mathsf{T}}, \hat{\boldsymbol{b}}_{N-1}^{\mathsf{T}}\right]$ of the periodically time-varying filter coefficients $\boldsymbol{a}_n^{\mathsf{T}} \triangleq \left[a_{0,n}, \ldots, a_{L,n}\right]$ and $\boldsymbol{b}_n^{\mathsf{T}} \triangleq \left[b_{0,n}, \ldots, b_{M,n}\right]$ for $n = 0, \ldots, N-1$. Again, this can be done by forward message passing in the graph of Figure 2.1a by defining

$$\boldsymbol{c}_k \triangleq \left[\boldsymbol{0}_{k \bmod N}, 1, \tilde{y}_{k-1}, \ldots, \tilde{y}_{k-L}, u_k, \ldots, u_{k-M}, 0, \ldots, 0\right] , \tag{2.25}$$

where $\boldsymbol{0}_{k \bmod N}$ is a zero row vector of length $k \bmod N$ and the trailing zeros are used to make the vector $\boldsymbol{c}_k$ have a length of $N(L+M)$. The disadvantage of this model is that the number of parameters grows linearly with the length of the period N. ◊

Example 2.3: Sinusoidal Parameter Estimation
A discrete-time noisy sinusoid can be written as

$$Y_k = \mathrm{Re}\big(\xi\, e^{ik\Omega}\big) + Z_k\,, \tag{2.26}$$

where Ω is the discrete-time frequency, $\xi \triangleq \alpha\, e^{i\phi} \in \mathbb{C}$ is a complex coefficient, α is the amplitude, ϕ is the phase, and $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$ is the noise. We consider estimation of α and ϕ given the frequency Ω and observations $Y_k = \tilde{y}_k$. We can write (2.26) as

$$Y_k = \boldsymbol{c}_k\, \boldsymbol{x} + Z_k\,, \tag{2.27}$$

where

$$\boldsymbol{c}_k \triangleq \big[\cos(k\Omega), -\sin(k\Omega)\big]\,, \text{ and } \boldsymbol{x} \triangleq \begin{bmatrix}\mathrm{Re}\,\xi \\ \mathrm{Im}\,\xi\end{bmatrix} = \alpha \begin{bmatrix}\cos\phi \\ \sin\phi\end{bmatrix}. \tag{2.28}$$

The ML estimate $\hat{\boldsymbol{x}}_{\mathrm{ML}}$ can be found using forward message passing in the factor graph in Figure 2.1a with the definitions (2.28). Note that the mapping $(\alpha, \phi) \mapsto \boldsymbol{x}$ is one-to-one and hence ML estimation of (α, ϕ) is equivalent to ML estimation of $\boldsymbol{x}$ followed by inverting the mapping. ♢

Example 2.4: Periodic Signal Estimation
This is a straightforward generalization of Example 2.3. Using a Fourier series, a discrete-time noisy periodic signal can be written as

$$Y_k = \mathrm{Re} \sum_{m=1}^{M} \xi_m\, e^{imk\Omega} + Z_k\,, \tag{2.29}$$

where Ω is the discrete-time fundamental frequency, $\xi_m \in \mathbb{C}$ are the coefficients, and $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$ is the noise. We consider estimation of ξ_m given the fundamental frequency Ω and observations $Y_k = \tilde{y}_k$. We can write (2.29) as

$$Y_k = \boldsymbol{c}_k\, \boldsymbol{x} + Z_k\,, \tag{2.30}$$

where

$$\boldsymbol{c}_k \triangleq \big[\cos(k\Omega), -\sin(k\Omega), \ldots, \cos(Mk\Omega), -\sin(Mk\Omega)\big]\,, \tag{2.31}$$

$$\boldsymbol{x} \triangleq \big[\mathrm{Re}\,\xi_1, \mathrm{Im}\,\xi_1, \ldots, \mathrm{Re}\,\xi_M, \mathrm{Im}\,\xi_M\big]^{\mathsf{T}}. \tag{2.32}$$

The ML estimate $\hat{\boldsymbol{x}}_{\mathrm{ML}}$ can be found using forward message passing in the factor graph of Figure 2.1a with the definitions (2.31) and (2.32). ♢

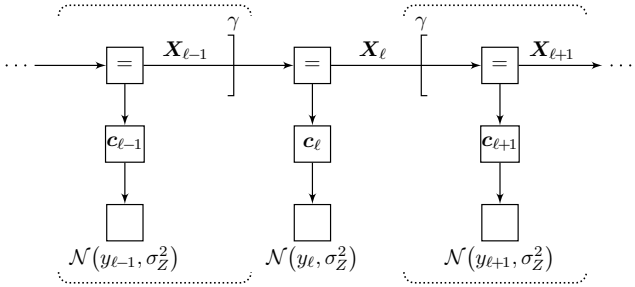


Figure 2.2: Factor graph representations of two-sided recursive least squares (RLS) (2.34).

2.4 Recursive Least Squares

The traditional RLS algorithm is equivalent to forward message passing in the graph of Figure 2.1 with the additional application of a forgetting factor [65]. Here, this traditional view is generalized by incorporating a forgetting factor for both forward and backward messages.

More specifically we define K cost functions for two-sided RLS as

$$\kappa_\ell(\boldsymbol{x}) \triangleq (\boldsymbol{y} - \mathbf{C}\boldsymbol{x})^{\mathsf{T}} \mathbf{W}_\ell (\boldsymbol{y} - \mathbf{C}\boldsymbol{x}) = \sum_{k=1}^{K} \gamma^{|k-\ell|} \left(y_k - \boldsymbol{c}_k \boldsymbol{x}\right)^2, \qquad (2.33)$$

for $\ell = 1, \ldots, K$, where γ is the forgetting factor, $\mathbf{W}_\ell \triangleq \operatorname{diag}\big(\boldsymbol{w}^{(\ell)}\big)$, and $w_k^{(\ell)} \triangleq \gamma^{|k-\ell|}$. We assume that $0 \leq \gamma \leq 1$, $\gamma \approx 1$. The ℓ-th equivalent statistical model can be derived from (2.3) and (2.7) as

$$p(\boldsymbol{y}|\boldsymbol{x}) \propto \prod_{k=1}^{K} e^{-\gamma^{|k-\ell|}(y_k - \boldsymbol{c}_k \boldsymbol{x})^2/(2\sigma_Z^2)} \qquad (2.34)$$

In this model we clearly see, how the effective variance is increased from σ_Z^2 to $\gamma^{-|k-\ell|}\sigma_Z^2$ with increasing distance $|k-\ell|$. For $\ell = K$ the statistical model for traditional RLS is recovered. For $\gamma = 1$ all models coincide and are identical with the least squares model (2.16). The factorization (2.34) can be represented by the factor graph in Figure 2.2 in which we use our notation for the forgetting factor (cf. Section 1.5.2).

Example 2.5: Linear Transfer-Function Estimation by Sinusoidal Excitation
Example 2.3 is extended to the estimation of time-varying sinusoidal parameters. Assume that a linear time-invariant (LTI) system with transfer-function $H\big(e^{i\Omega}\big)$ is excited with a swept sinusoid

$$u_k = \mathrm{Re}\big(e^{ik\Omega_k}\big)\,, \tag{2.35}$$

where the frequency Ω_k changes, e.g., linearly or logarithmically with k. We measure the output of this system under additive white Gaussian noise. At each frequency, the system will induce an amplification and a phase shift such that the measured signal can be written as

$$Y_k = \mathrm{Re}\big(\xi_k\, e^{ik\Omega_k}\big) + Z_k\,, \tag{2.36}$$

where $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$ is the noise and $\xi_k \triangleq \alpha_k\, e^{i\phi_k}$ is the linear distortion induced by the system. If we assume that $\Omega_k \approx \Omega_{k-1}$ then, for systems of low enough order, we also can assume

$$\xi_k \approx \xi_{k-1}\,, \tag{2.37}$$

and we can write our signal model as

$$Y_k = \boldsymbol{c}_k\, \boldsymbol{x}_k + Z_k\,, \tag{2.38}$$

$$\boldsymbol{x}_k \approx \boldsymbol{x}_{k-1}\,, \tag{2.39}$$

where

$$\boldsymbol{c}_k \triangleq \Big[\cos\textstyle\sum_{j=1}^{k}\Omega_j\,,\ \sin\sum_{j=1}^{k}\Omega_j\Big]\,, \tag{2.40}$$

$$\boldsymbol{x}_k \triangleq \begin{bmatrix}\mathrm{Re}\,\xi_k\\ \mathrm{Im}\,\xi_k\end{bmatrix} = \alpha_k\begin{bmatrix}\cos\phi_k\\ \sin\phi_k\end{bmatrix}. \tag{2.41}$$

The coefficients ξ_k can be estimated by forward and backward message passing in the RLS graph of Figure 2.2 in which the forgetting factor γ models the approximate equality of (2.39). Finally we can formulate an estimate of the system transfer-function at frequencies Ω_k as

$$\hat{H}(e^{i\Omega_k}) = [\hat{\boldsymbol{x}}_k]_1 + i[\hat{\boldsymbol{x}}_k]_2\,. \qquad \Diamond$$

Example 2.6: Slowly Changing Periodic Signal
Recall the estimation of a periodic signal in Example 2.4. We generalize the signal model by allowing the coefficients to change slowly over time. Our signal model now is

$$Y_k = \mathrm{Re} \sum_{m=1}^{M} \xi_k^{(m)} \, e^{imk\Omega} + Z_k \tag{2.42}$$

$$\xi_k^{(m)} \approx \xi_{k-1}^{(m)} \,, \quad m = 1, \ldots, M \,. \tag{2.43}$$

We implement the approximate equality using a forgetting factor γ in the factor graph of the corresponding RLS formulation (2.30). ◊

Example 2.7: Estimation of a Weakly Nonlinear Transfer-Function
We extend Example 2.5 to estimating a weekly nonlinear system by sinusoidal excitation. For a sinusoidal input, a nonlinear system produces an output that contains harmonics, each with its own amplitude and phase shift. We excite the system by a slowly swept sinusoid

$$u_k = \mathrm{Re}\big(e^{ik\Omega}\big) \,, \tag{2.44}$$

and we assume that the output signal can be modeled as

$$Y_k = \mathrm{Re} \sum_{m=1}^{M} \xi_k^{(m)} \, e^{imk\Omega_k} + Z_k \tag{2.45}$$

where $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$ is the noise and the coefficient $\xi_k^{(m)} \triangleq \alpha_k^{(m)} \, e^{i\phi_k^{(m)}}$ contains the amplitude $\alpha_k^{(m)}$ and the phase $\phi_k^{(m)}$ of the m-th harmonic. Since $\Omega_k \approx \Omega_{k-1}$ we can assume for weekly nonlinear systems of low enough order

$$\xi_k^{(m)} \approx \xi_{k-1}^{(m)} \tag{2.46}$$

for $m = 1, \ldots, M$. The signal model can now be written as

$$Y_k = \boldsymbol{c}_k \, \boldsymbol{x}_k + Z_k \,, \tag{2.47}$$

$$\boldsymbol{x}_k \approx \boldsymbol{x}_{k-1} \,, \tag{2.48}$$

where

$$\boldsymbol{c}_k \triangleq \Big[\cos \sum_{j=1}^{k} \Omega_j, \sin \sum_{j=1}^{k} \Omega_j, \ldots, \cos \sum_{j=1}^{k} M\Omega_j, \sin \sum_{j=1}^{k} M\Omega_j\Big] \,, \tag{2.49}$$

$$\boldsymbol{x} \triangleq \Big[\mathrm{Re}\, \xi_k^{(1)}, \mathrm{Im}\, \xi_k^{(1)}, \ldots, \mathrm{Re}\, \xi_k^{(M)}, \mathrm{Im}\, \xi_k^{(M)}\Big]^{\mathsf{T}} \,. \tag{2.50}$$

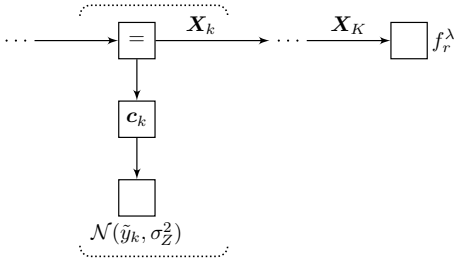


Figure 2.3: Factor graph representations of regularized least squares as in (2.52).

The coefficients $\xi_k^{(m)}$ can be estimated by forward and backward message passing in the RLS graph of Figure 2.2 in which the forgetting factor implements the approximate equality of (2.48). We thus can estimate the transfer-function $H_m(e^{im\Omega})$ from the input to the m-th harmonic at frequencies $m\Omega_k$ as

$$\hat{H}_m(e^{im\Omega_k}) = [\hat{\boldsymbol{x}}_k]_{2m-1} + i[\hat{\boldsymbol{x}}_k]_{2m}\,. \qquad \Diamond$$

2.5 Regularized Recursive Least Squares

The cost function for the regularized least squares problem can be stated as

$$\kappa(\boldsymbol{x}) = (\boldsymbol{y} - \mathbf{C}x)^{\mathsf{T}}(\boldsymbol{y} - \mathbf{C}x) + \lambda\, r(\boldsymbol{x}) \tag{2.51}$$

where $r(\boldsymbol{x})\colon \mathbb{R}^{nx} \mapsto \mathbb{R}_{\geq 0}$ is also a cost function, here named the regularizing function and λ is the regularization parameter. Usually, the regularizing function is chosen as the ℓ_2 norm $r(\boldsymbol{x}) = \|\boldsymbol{x}\|^2 = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{x}$, but also the ℓ_1 norm $r(\boldsymbol{x}) = \|\boldsymbol{x}\|_1 = \sum_{i=1}^{n}|x_i|$ or even functions violating the non-negativity such as $r(\boldsymbol{x}) = \sum_{i=1}^{n} x_i$ may be useful.

Again, the equivalent statistical model can be derived from (2.3) and (2.6) as

$$p(\boldsymbol{y}, \boldsymbol{x}) \propto e^{-\lambda\, r(\boldsymbol{x})/(2\sigma_Z^2)} \prod_{k=1}^{K} e^{-(y_k - \boldsymbol{c}_k\boldsymbol{x})^2/(2\sigma_Z^2)}\,. \tag{2.52}$$

By defining the exponentiated regularizing function

$$f_r(\boldsymbol{x}) \triangleq e^{-r(\boldsymbol{x})/(2\sigma_Z^2)} \tag{2.53}$$

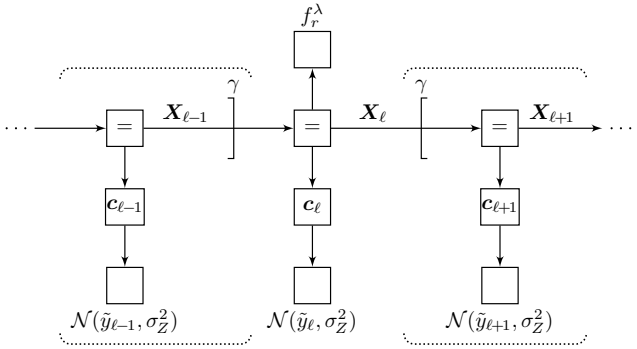


Figure 2.4: Factor graph representation of regularized recursive least squares (RLS) (2.56).

we obtain the factor graph representation of (2.52) in Figure 2.3 for fixed observations $Y_k = \tilde{y}_k$. The additional factor $f_r(\boldsymbol{x})^\lambda$ can be moved to any location $k = 1, \ldots, K$ in the graph without changing the global function.

We consider a regularized form of two-sided RLS by defining K (regularized) cost functions for $\ell = 1, \ldots, K$ as

$$\kappa_\ell(\boldsymbol{x}) = (\boldsymbol{y} - \mathbf{C}\boldsymbol{x})^\mathsf{T} \mathbf{W}_\ell (\boldsymbol{y} - \mathbf{C}\boldsymbol{x}) + \lambda\, r(\boldsymbol{x}) \tag{2.54}$$

$$= \sum_{k=1}^{K} \gamma^{|k-\ell|} (y_k - \boldsymbol{c}_k \boldsymbol{x})^2 + \lambda\, r(\boldsymbol{x})\,, \tag{2.55}$$

where γ is the forgetting factor, $r(\boldsymbol{x})$ is the regularizing function, and λ is the regularization parameter. The equivalent statistical model can be derived from (2.3) and (2.6) as

$$p(\boldsymbol{y}, \boldsymbol{x}) \propto e^{-\lambda\, r(\boldsymbol{x})/(2\sigma_Z^2)} \prod_{k=1}^{K} e^{-\gamma^{|k-\ell|}(y_k - \boldsymbol{c}_k \boldsymbol{x})^2/(2\sigma_Z^2)}\,. \tag{2.56}$$

The factorization (2.56) can be represented by the factor graph shown in Figure 2.4 for fixed observations $Y_k = \tilde{y}_k$.

Sometimes, we may want to distribute the regularization factor over all time slices as shown in Figure 2.5. The factorization represented by this

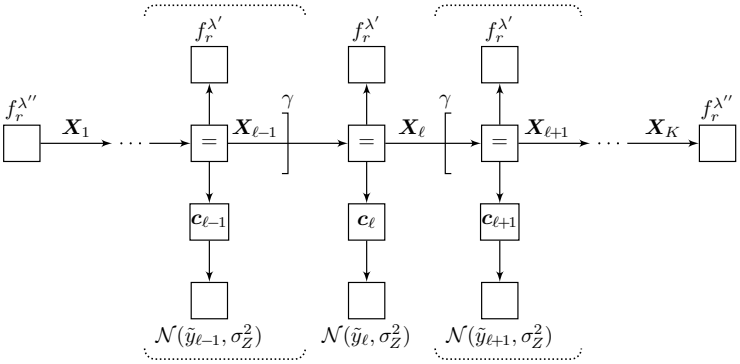


Figure 2.5: Factor graph representation of two-sided recursive least squares (RLS) with distributed regularization (2.57).

graph is

$$p(\boldsymbol{x}, \boldsymbol{y}) \propto e^{-\gamma^{\ell-1}\lambda'' \, r(\boldsymbol{x})/(2\sigma_Z^2)} \, e^{-\gamma^{K-\ell}\lambda'' \, r(\boldsymbol{x})/(2\sigma_Z^2)} \cdot \prod_{k=1}^{K} e^{-\gamma^{|k-\ell|}\lambda' \, r(\boldsymbol{x})/(2\sigma_Z^2)} \, e^{-\gamma^{|k-\ell|}(\tilde{y}_k - \boldsymbol{c}_k \boldsymbol{x})^2/(2\sigma_Z^2)} , \tag{2.57}$$

where we have introduced two regularization parameters λ' and λ''. At first sight, distributing the regularization is such a way seems to be uninteresting. We show, however, a useful extension of this model in Section 6.5.1.

Forward message passing in such a model is equivalent to an algorithm known under the name of "leaky" RLS [44].

Theorem 2.1: Distributed Regularization
The factorizations (2.56) *and* (2.57) *are equal up to a scale factor if*

$$\lambda' = \lambda \, \frac{1-\gamma}{1+\gamma} \tag{2.58}$$

and

$$\lambda'' = \lambda \, \frac{\gamma}{1+\gamma} \, . \tag{2.59}$$

Proof. When equating the logarithm of (2.56) and (2.57), the quadratic

terms involving y_k cancel and we are left with the regularization terms:

$$\frac{\lambda\, r(\boldsymbol{x})}{2\sigma_Z^2} = \frac{\lambda''\, r(\boldsymbol{x})\big(\gamma^{\ell-1} + \gamma^{K-\ell}\big)}{2\sigma_Z^2} + \frac{\lambda'\, r(\boldsymbol{x})}{2\sigma_Z^2} \sum_{k=1}^{K} \gamma^{|k-\ell|} \tag{2.60}$$

$$\lambda = \lambda''\big(\gamma^{\ell-1} + \gamma^{K-\ell}\big) + \lambda' \sum_{k=1}^{K} \gamma^{|k-\ell|} \tag{2.61}$$

$$= \lambda''\big(\gamma^{\ell-1} + \gamma^{K-\ell}\big) + \lambda'\, \frac{1+\gamma-\gamma^{K-\ell+1}-\gamma^{\ell}}{1-\gamma} \tag{2.62}$$

This is one equation with two unknowns and hence there may be several solutions λ' and λ'' in terms of λ. We additionally require that neither λ' nor λ'' depends on K. Hence, we first let the factor graph extend on both sides infinitely. Specifically, we let first $K \to \infty$ and then $\ell \to \infty$. Hence, $\gamma^{K-\ell} \to 0$ and $\gamma^{\ell} \to 0$, and we get

$$\lambda = \lambda'\, \frac{1+\gamma}{1-\gamma}\,. \tag{2.63}$$

Back in the truncated graph, the term involving λ'' must make up for the missing regularization. The latter is

$$\lambda'' = \lambda' \sum_{k=1}^{\infty} \gamma^k - \lambda'\, \frac{\gamma}{1-\gamma} = \lambda\, \frac{\gamma}{1+\gamma}\,. \tag{2.64}$$

The resulting values λ' and λ'' can be back substituted in (2.62) for confirmation. □

2.6 Connections with State-Space Models

Trivially, the Gaussian statistical model representation (2.16) of least squares can be viewed as a linear state-space model (SSM) (1.4) with $\mathbf{A}_k = \mathbf{I}$, $\mathbf{B}_k = \mathbf{0}$, and $\mathbf{C}_k = \boldsymbol{c}_k$ for $k = 1, \ldots, K$.

If the input $\boldsymbol{U}_k$ to a SSM is known, state estimation can be posed as a least squares problem. The SSM

$$\begin{aligned} \boldsymbol{X}_k &= \mathbf{A}_k \boldsymbol{X}_{k-1} + \mathbf{B}_k \boldsymbol{U}_k \\ Y_k &= \boldsymbol{c}_k \boldsymbol{X}_k + Z_k \end{aligned} \tag{2.65}$$

for $k = 1, \ldots, K$ can be reformulated as

$$Y_k = \boldsymbol{c}_k \prod_{k'=1}^{k} \mathbf{A}_{k'} \boldsymbol{X}_0 + \sum_{k'=1}^{k-1} \prod_{k''=k}^{k'+1} \mathbf{A}_{k''} \mathbf{B}_{k'} \boldsymbol{U}_{k'} + \mathbf{B}_k \boldsymbol{U}_k + Z_k \,. \tag{2.66}$$

Now assume that we have observed values $Y_k = \tilde{y}_k$ and $\boldsymbol{U}_k = \tilde{\boldsymbol{u}}_k$ for $k = 1, \ldots, K$, and that $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$. Then Equation (2.66) can be formulated as

$$\tilde{y}'_k = \mathbf{C}'_k \boldsymbol{X} + Z_k \,, \tag{2.67}$$

where

$$\tilde{y}'_k \triangleq \tilde{y}_k - \sum_{k'=1}^{k-1} \prod_{k''=k}^{k'+1} \mathbf{A}_{k''} \mathbf{B}_{k'} \tilde{\boldsymbol{u}}_{k'} + \mathbf{B}_k \tilde{\boldsymbol{u}}_k \,, \tag{2.68}$$

$$\boldsymbol{c}'_k \triangleq \boldsymbol{c}_k \prod_{k'=1}^{k} \mathbf{A}_{k'} \,, \text{ and } \boldsymbol{X} \triangleq \boldsymbol{X}_0 \,. \tag{2.69}$$

We see that (2.67) leads to the same statistical model $p(\boldsymbol{y}|\boldsymbol{x})$ as (2.16). The inclusion of a forgetting factor would, however, change the model.

2.7 Application to Slowly Changing Periodic Signals

Approximately periodic signals with known fundamental frequency occur in diverse areas such as financial time series, biological systems, and communication [31, 38]. In this section we elaborate on the view taken by Example 2.6 on such signals, and we show a real world example application similar to [89].

2.7.1 Microwave Link Gain Measurements and Rain

It is well known that outdoor microwave links commonly used in commercial telecommunication networks suffer from attenuation due to rain [79]. From this observation, it has been suggested to estimate rainfall rates based on available gain measurement data of microwave links [106]. Indeed, estimating rainfall rates in this way would be a welcome complement to rain gauges and rain radar measurements [62, 106].

However, estimating rainfall from gain data is challenging. The nature of the problem is illustrated by the gray line in the upper plot of Figure 2.9a, which shows the gain of a microwave link over 8 days. The deep jags in the plot are due to heavy rain. Without rain, the gain fluctuates smoothly within some fixed range exhibiting some degree of periodicity with a period of 1 day, which is due to the daily cycle of temperature, humidity, and air pressure.

Commonly, estimating the rainfall rate from such gain measurements involves three separate tasks [39, 95, 106]. The first task is to classify the data into segments with rain and segments without rain. The second task is to estimate the smoothed baseline within each rainy segment, which is subtracted from the measured attenuation; the result is a net attenuation due to rain only. The third task is to estimate the rainfall rate based on this net attenuation.

In this section we expand Example 2.4 to develop a model for solving the first two tasks. Note that the periodic ARMA model as defined in Example 2.2 is not well suited for this application for two reasons. First, the model cannot deal with the non-uniform time stamps for the data. Second, the model order would have to be chosen very high (approximately the number of data items per day).

2.7.2 Model-Based Rain Estimation

We extend (2.42) and (2.43) in Example 2.6 to non-uniform sampling and incorporate a DC drift. Let t_k for $k = 1, \ldots, K$ be the time stamps of the observed signal $Y_k = \tilde{y}_k$ for which we define the model

$$Y_k = \xi_k^{(0)} + \operatorname{Re} \sum_{m=1}^{M} \xi_k^{(m)} \, e^{imt_k\omega} + Z_k \,, \tag{2.70}$$

$$\xi_k^{(m)} \approx \xi_{k-1}^{(m)} \,, \quad m = 1, \ldots, M \,, \tag{2.71}$$

where ω is the continuous-time fundamental frequency, $\xi_k^{(m)} \in \mathbb{C}$ for $m = 1, \ldots, M$ are the Fourier coefficients, $\xi_k^{(0)} \in \mathbb{R}$ is the DC component, and Z_k is the noise.

First, we recall (cf. Example 2.4) that if we were to enforce strict equality in (2.71), then (2.70) would model noisy observations of a (strictly) periodic signal. The relaxation to an approximate equality in (2.71) is

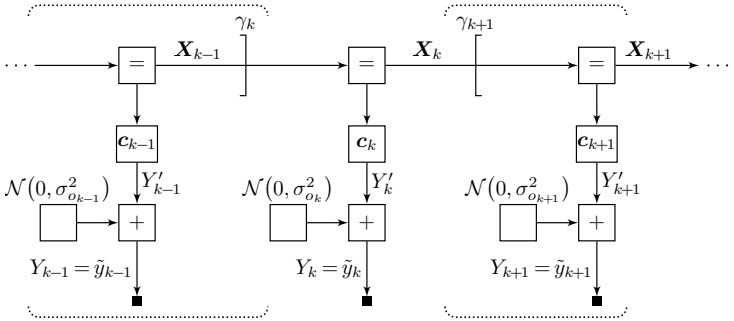


Figure 2.6: A recursive least squares (RLS) model with time-varying forgetting factor and time-varying noise variance.

implemented with a forgetting factor that depends on the time interval $\tau_k \triangleq t_k - t_{k-1}$ between the observations. Specifically we define

$$\gamma_k \triangleq \chi^{\tau_k} \tag{2.72}$$

to be the forgetting factor for both forward and backward message passing, where χ is the forgetting factor per unit time.

Second, we define the noise $Z_k \sim \mathcal{N}(0, \sigma^2_{o_k})$ to be time-varying too. This allows us to make a distinction between outliers and valid data items depending on the class label $o_k \in \{0, 1\}$, where $o_k = 1$ means that $\tilde{y}_k$ is an outlier. For the outliers we let $\sigma_1^{-2} = 0$, thus effectively removing this observation from the model. For valid data items, we let σ_0^2 be a fixed parameter. In the application of microwave link gain, we model data items corrupted with rain attenuation as outliers.

Our model is now defined by the factor graph in Figure 2.6 with the definition (2.72) and with

$$\boldsymbol{c}_k \triangleq \left[1, \cos(t_k\omega), -\sin(t_k\omega), \ldots, \cos(Mt_k\omega), -\sin(Mt_k\omega)\right]. \tag{2.73}$$

The equivalence with (2.70) and (2.71) is established by defining

$$\boldsymbol{x}_k \triangleq \left[\xi_k^{(0)}, \operatorname{Re}\xi_k^{(1)}, \ldots, , \operatorname{Im}\xi_k^{(1)}, \ldots, \operatorname{Re}\xi_k^{(M)}, \operatorname{Im}\xi_k^{(M)}\right]^{\mathsf{T}}. \tag{2.74}$$

Given the measurements $\tilde{y}_k$ for $k = 1, \ldots, K$ and some chosen values for the parameters ω, M, σ_0^2, and χ, we would like to estimate the class labels o_k and the baseline Y'_k. In our approach, we propose to initially assume the complete absence of outliers and then alternate between

a) estimating Y_k' for fixed class labels o_k and

b) updating the class labels o_k for $k = 1, \ldots, K$ based on the prediction probability density function (PDF) $p(y_k|\tilde{y}_1, \ldots, \tilde{y}_{k-1}, \tilde{y}_{k+1}, \ldots, \tilde{y}_K)$ and the observation $\tilde{y}_k$.

While the first step amounts to message passing in the graph of Figure 2.6, the second step is defined as follows. The prediction PDF is proportional to the prediction message $\vec{\mu}_{Y_k}$, i.e:

$$p(y_k|\tilde{y}_1, \ldots, \tilde{y}_{k-1}, \tilde{y}_{k+1}, \ldots, \tilde{y}_K) = \mathcal{N}\big(y_k \big| \vec{m}_{Y_k}, \vec{\sigma}^2_{Y_k}\big)\,. \tag{2.75}$$

We classify the data item $\tilde{y}_k$ as an outlier if the value $\tilde{y}_k$ lies outside of a confidence interval $[\vec{\mu}_{Y_k} - \theta\vec{\sigma}_{Y_k}, \infty)$, i.e:

$$o_k = \begin{cases} 1 & \text{if} \quad y_k < \vartheta_k \\ 0 & \text{else} \end{cases}, \vartheta_k \triangleq \vec{m}_{Y_k} - \theta\vec{\sigma}_{Y_k}\,, \tag{2.76}$$

where θ is a parameter of the algorithm determining the tail probability $Q(\vec{m}_{Y_k} - \theta\vec{\sigma}_{Y_k})$ we are willing to allow for outliers. This parameter implicitly defines a detection threshold ϑ_k.

Note that in this application only negative outliers occur. If outliers can occur in both directions then we would choose a confidence interval $[\vec{m}_{Y_k} - \theta\vec{\sigma}_{Y_k}, \vec{m}_{Y_k} + \theta\vec{\sigma}_{Y_k}]$. Also note that the threshold ϑ_k is time-varying and adapts automatically to non-uniform time stamps and hence to missing data.

Finally note that any chosen observation noise variance σ_0^2 can be absorbed into θ. This follows directly from the message update rules for Gaussian messages [65]. We are now ready to define the following offline and an online algorithm.

Offline Algorithm

In this algorithm we assume that a whole block of data is available. The message passing is done with reference to Figure 2.6.

a) Initialize $o_k = 0$ for $k = 1, \ldots, K$, $\overrightarrow{\mathbf{W}}_{X_0} = \mathbf{0}$, $\overleftarrow{\mathbf{W}}_{X_{K+1}} = \mathbf{0}$.

b) Do forward and backward message passing in the factor graph of Figure 2.6. For $k = 1, \ldots, K$, compute the estimate $\hat{y}_k = m_{Y_k'}$,

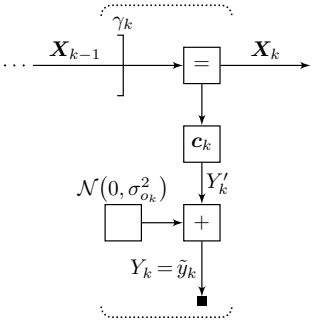


Figure 2.7: Approximately periodic signal estimation – Online algorithm.

where $m_{Y'_k}$ is the mean of the marginal on edge Y'_k, and compute the prediction message $\overrightarrow{\mu}_{Y_k}$.

c) Apply the classification rule (2.76) to update the class labels o_k for $k = 1, \ldots, K$.

d) Go to Step (b) unless the class labels are unchanged or the available time is over or the maximum number of iterations is reached.

Online Algorithm:

In this algorithm we assume that the data comes in a stream and we want to produce a new baseline estimate $\hat{y}_k$ and a new class label o_k as soon as the k-th data item has arrived. To do this, we consider forward-only message passing in the factor graph of Figure 2.7. This naturally inhibits iterations.

a) Initialize $\overrightarrow{\mathbf{W}}_{X_0} = \epsilon \mathbf{I}_{M+1}$, $\overrightarrow{\boldsymbol{m}}_{X_0} = \big[\tilde{y}_1, 0, \ldots, 0\big]$, for some small value $\epsilon > 0$. Set $k = 0$.

b) Increment k, get the data item $\tilde{y}_k$ and corresponding time stamp t_k.

c) Compute the prediction message $\overrightarrow{\mu}_{Y_k}$ in Figure 2.7 and apply the classification rule (2.76) to set the class label c_k.

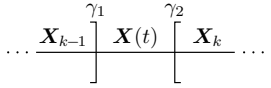


Figure 2.8: State estimation at an arbitrary time t.

d) Compute the estimate $\hat{y}_k = m_{Y'_k}$, where $m_{Y'_k}$ is the mean of the marginal on edge Y'_k, and compute the message $\vec{\mu}_{X_k}$.

e) Go to Step (b).

Results

Both algorithms have been applied to real-world data, of which we report the following example. In this example, the microwave link operates at a single frequency of 38 GHz and covers a distance of 2876 m. The link gain is provided in units dBm with a quantization of 0.1 dBm in time intervals of approximately 3 min. Part of the data was deleted on purpose to examine the behavior of the algorithms in the case of missing data.

The algorithm parameters were set as in Table 2.1. In the offline algorithm the class labels remained unchanged after the third iteration. Hence there was no need to specify an upper limit on the number of iterations.

Figure 2.9 shows results for both the offline and the online algorithm. Besides the measured gain data $\tilde{y}_k$, the upper plots in Figures 2.9a and 2.9b show the estimated baseline $\hat{y}_k$ and the detection threshold ϑ_k. The baseline can be estimated even in the region where the data has been deleted (around day 7).

To see how this is done consider Figure 2.8. In this figure we define the state $\boldsymbol{X}(t)$ at some time $t_{k-1} < t < t_k$ to lie between the edges $\boldsymbol{X}_{k-1}$ and $\boldsymbol{X}_k$ in the RLS factor graph. The targeted interpolation is done by two forgetting factors $\gamma_1 \triangleq \chi^{t-t_{k-1}}$ and $\gamma_2 \triangleq \chi^{t_k-t}$. For the online algorithm the message $\overleftarrow{\mu}_{X(t)}$ is assumed to be neutral.

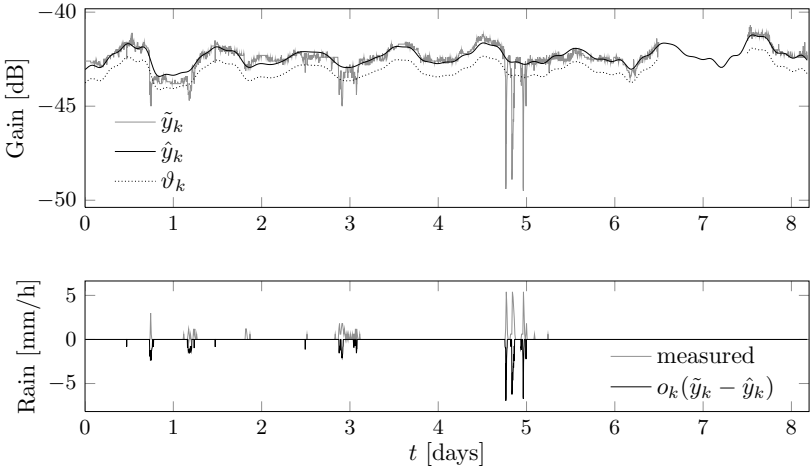
The lower plots in Figures 2.9a and 2.9b show rain rates measured by a rain gauge in the vicinity of the microwave link. To validate the algorithms qualitatively, the same plot shows a crude rain estimate $o_k(\hat{y}_k - \tilde{y}_k)$, i.e. an estimate of the attenuation due to rain. We recall that the class label o_k is 1 only if rain has been detected. Also note that the actually plotted

Parameter		Value
M	Number of harmonics	5
ω	Fundamental frequency	2π
χ	Forgetting factor per day	0.3
θ	Tail probability parameter	0.7
σ_0^2	Noise variance	1

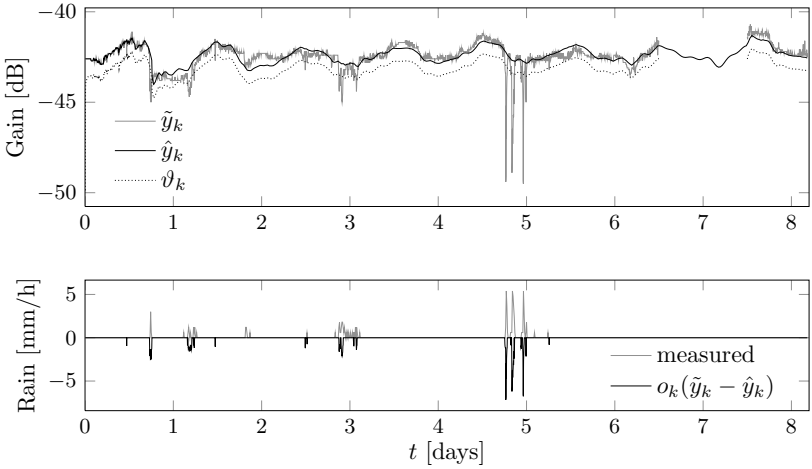
Table 2.1: Parameter settings used for Figure 2.9.

quantity is $o_k(\tilde{y}_k - \hat{y}_k)$ for better visibility. This estimate is certainly not accurate enough to give exact account on the rain rate: e.g., nonzero values are lower bounded by $\vartheta_k - \tilde{y}_k$.

Subjectively, however, this quantity captures the rain rate satisfactorily for both the online and the offline algorithm. Missing detections and false positives strongly depend on the chosen parameter θ which defines the threshold ϑ. Also note that the measured rain rate cannot be taken as ground truth since the measured rain is strongly localized as opposed to the rain that affects the microwave link gain.



(a) Offline algorithm.



(b) Online algorithm.

Figure 2.9: Rain rate estimation on example microwave link gain data.

Chapter 3

Infinite Impulse Response Filters

3.1 Introduction

In this chapter we treat exclusively linear systems that have an infinite impulse response (IIR). We will make this restriction exact in a formally stated assumption. Pure finite impulse response (FIR) models have been described in a Gaussian factor graph context, e.g., in [23, 65]. The use of IIR model allows us to consider continuous-time versions and model long time scales with low order.

In general we can use statistical models of linear systems for a variety of tasks, e.g. de-noising, input estimation, signal separation, and more. We will not describe concrete examples in this chapter though.

This chapter starts with some general consideration about IIR systems. We present the real Jordan canonical form and introduce the notion of a steady state. The latter is quite common in control theory. This notion allows us to bring into picture the linear filtering view of message passing in linear SSMs.

Next, we take a closer look at autonomous models – models lacking an input and thus running on their own. Connections between a forgetting factor, a scaled system matrix $\mathbf{A}$, and non-autonomous systems are made. The second-order case will be of special interest since in this case the means of the messages are related to the discrete-time Fourier transform of the observed signal.

Using the notion of a continuous-time stochastic system with noisy

discrete-time observations [11, 12] we formulate two models – one for polynomials and one for sinusoids. The former is equivalent with the model commonly assumed for spline smoothing.

We conclude this chapter by sketching a way in which a high dimensional SSM can be split up into second-order SSMs and we mention in what sense message passing in the split-up model approximates message passing in the joint model.

3.2 General Concepts for Linear State-Space Models

3.2.1 Some Basic Definitions and Assumptions

A linear discrete-time SSM with Gaussian input $\boldsymbol{U}_k \sim \mathcal{N}(\boldsymbol{0}, \mathbf{V}_{U_k})$ and additive Gaussian observation noise $\boldsymbol{Z}_k \sim \mathcal{N}(\boldsymbol{0}, \mathbf{V}_{Z_k})$ is defined as

$$\begin{aligned} \boldsymbol{X}_k &= \mathbf{A}_k \, \boldsymbol{X}_{k-1} + \mathbf{B}_k \, \boldsymbol{U}_k \,, \\ \boldsymbol{Y}_k &= \mathbf{C}_k \, \boldsymbol{X}_k + \boldsymbol{Z}_k \,, \end{aligned} \tag{3.1}$$

where $\boldsymbol{X}_k$ is the state vector and $\boldsymbol{Y}_k$ is the observable output. The parameters that define the system are the *system matrices* (or vectors) $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$ and the covariance matrices $\mathbf{V}_{U_k}$ and $\mathbf{V}_{Z_k}$. These parameters are possibly time-varying, such that both (!LTI) systems as well as linear time-varying (LTV) systems can be modeled. The factor graph representation of (3.1) is depicted in Figure 3.1.

General properties of such systems are treated in textbooks in a control theory context [115], in a dynamical systems context [82], or in a linear algebra context [45]. In general, systems of the form (3.1) can have FIRs or IIRs, and the time-varying version shown here even allows jumps between the two. In this thesis we restrict ourselves to an IIR case, which we formally define as follows.

Assumption 3.1: IIR systems

a) In each time step k, the system is controllable and observable.

b) In each time step k, the system transfer function induced by $\mathbf{A}_k$, $\mathbf{B}_k$ and $\mathbf{C}_k$ has at least as many poles as zeros.

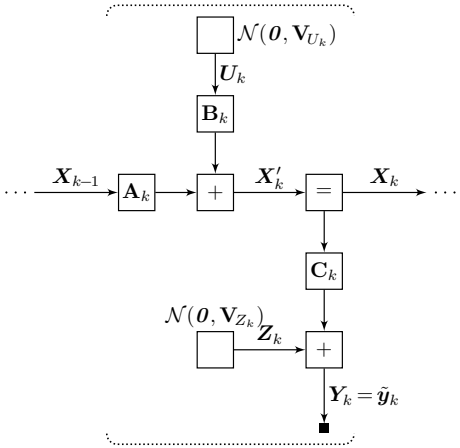


Figure 3.1: Factor graph representation of a linear discrete-time state-space model (SSM).

c) The system's input, state and output are real-valued.

Assumption 3.1a uses notions of control theory [97]. In our setup we usually have much liberty in choosing the system. Hence, for our purposes there is no need to treat systems that have uncontrollable or unobservable modes.

A consequence of Assumption 3.1b is that for each time step k, the discrete-time system (3.1) can be obtained by discretization of a continuous-time system either using a zero-order hold approach or using the bilinear transform. A second consequence of Assumption 3.1b is that $\mathbf{A}$ is non-singular [45].

Message passing on the graph of Figure 3.1 and the connection with Kalman filtering has been described in [65, 66, 110]. Indeed, for given observations $\boldsymbol{Y}_k = \tilde{\boldsymbol{y}}_k$, forward message passing in the factor graph in Figure 3.1 of a linear system (3.1) can be viewed as time-varying, linear filtering. The state of such a message-passing filter is the mean $\overrightarrow{\boldsymbol{m}}_{X_k}$ or a linear transformation thereof, the input is the observation $\tilde{\boldsymbol{y}}_k$, and the output depends on what needs to be computed.

The above claim is verified by noting that all the update rules for Gaussian messages of all the nodes in the factor graph are linear with respect to the mean vectors [65]. The coefficients of this filter depend implicitly on

$\vec{\mathbf{V}}_{X_{k-1}}$, $\mathbf{V}_{U_k}$, $\mathbf{V}_{Z_k}$, and the system parameters $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$.

3.2.2 The Real Jordan Canonical Form

For any given k, the transfer function of a system (3.1) from the input $\boldsymbol{U}_k$ to the output $\boldsymbol{Y}_k$ does not determine the system matrices $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$ uniquely. Among the infinitely many parameterizations, there exist a number of canonical forms. Earlier work [23, 59] focused on the observer and controller canonical form. In this thesis we focus on the real Jordan canonical form.

The real Jordan canonical form [45] is seldom used in applications. The reason is that for transforming an arbitrary linear system to the real Jordan canonical form, the corresponding similarity transformation matrix cannot in general be computed in a numerically stable way. Here, however, this transform matrix is usually not computed. Instead we start off directly with a Jordan form. Later, for parameter estimation, this implies that we assume complete knowledge of the number of complex and real poles.

Under the Assumptions 3.1, the real Jordan canonical form for $\mathbf{A}_k$ can be formulated as

$$\mathbf{A}_k = \begin{bmatrix} \mathbf{J}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{J}_M \end{bmatrix}, \tag{3.2}$$

where each Jordan block $\mathbf{J}_m$ for $m = 1, \ldots, M$ has either the form

$$\mathbf{J}_m = \begin{bmatrix} a_m & 1 & 0 & \cdots & 0 \\ 0 & a_m & 1 & \cdots & 0 \\ 0 & 0 & a_m & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_m \end{bmatrix} \tag{3.3}$$

or

$$\mathbf{J}_m = \begin{bmatrix} \operatorname{rotm} \boldsymbol{a}_m & \mathbf{I}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \operatorname{rotm} \boldsymbol{a}_m & \mathbf{I}_2 & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \operatorname{rotm} \boldsymbol{a}_m & \ddots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \operatorname{rotm} \boldsymbol{a}_m \end{bmatrix}, \tag{3.4}$$

where $\boldsymbol{a}_m \triangleq \left[\operatorname{Re} \xi_m, \operatorname{Im} \xi_m\right]^\mathsf{T}$, $\xi_m \triangleq \alpha_m e^{i\phi_m} \in \mathbb{C}$, and the (scaled) rotation matrix operator for a vector in $\mathbb{R}^2$ is defined as

$$\operatorname{rotm} \boldsymbol{a}_m \triangleq \begin{bmatrix} \operatorname{Re} \xi_m & -\operatorname{Im} \xi_m \\ \operatorname{Im} \xi_m & \operatorname{Re} \xi_m \end{bmatrix} = \alpha_m \operatorname{rotm} \phi_m \,, \tag{3.5}$$

where the rotation matrix operator for a scalar is defined as

$$\operatorname{rotm} \phi_m \triangleq \begin{bmatrix} \cos \phi_m & -\sin \phi_m \\ \sin \phi_m & \cos \phi_m \end{bmatrix}. \tag{3.6}$$

Cf. Appendix B.1 for properties of (scaled) rotation matrices used in this thesis. In general, the values a_m and $\boldsymbol{a}_m$ (or ξ_m) need not be distinct for all m.

A Jordan block of the form (3.3) has a repeated eigenvalue a_m, whose algebraic multiplicity equals the block size. A Jordan block of the form (3.4) has a repeated pair of complex conjugate eigenvalues ξ_m, $\overline{\xi}_m$ (or $\alpha_m e^{\pm i\phi_m}$), where the algebraic multiplicity of this pair equals by half the block size.

It can be shown that for any matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ there is a nonsingular matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ such that $\mathbf{A} = \mathbf{S}^{-1}\mathbf{M}\mathbf{S}$ is of the form (3.2). For any $\mathbf{M}$, the corresponding similar matrix $\mathbf{A}$ is unique up to permutation of the Jordan blocks.

In this thesis we will be interested in the two following special cases:

a) The matrix $\mathbf{A}_k$ consists of one Jordan block $\mathbf{J}$ as in (3.3) or (3.4).

b) The matrix $\mathbf{A}_k$ has no repeated eigenvalues.

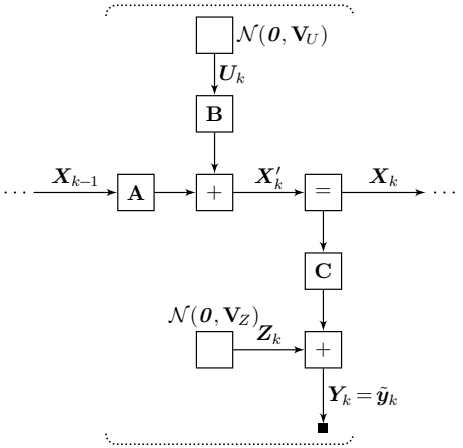


Figure 3.2: Factor graph for a linear time-invariant (LTI) system.

3.2.3 Linear Time-Invariant Systems and the Steady-State

A special case of the discrete-time linear system (3.1) is the time-invariant system

$$\begin{aligned} \boldsymbol{X}_k &= \mathbf{A}\boldsymbol{X}_{k-1} + \mathbf{B}\boldsymbol{U}_k \\ \boldsymbol{Y}_k &= \mathbf{C}\boldsymbol{X}_k + \boldsymbol{Z}_k\,, \end{aligned} \tag{3.7}$$

Note that the system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and the covariance matrices $\mathbf{V}_U$ and $\mathbf{V}_Z$ in this system remain constant across the time steps k. Figure 3.2 depicts the factor graph representation of (3.7).

The covariance matrices $\vec{\mathbf{V}}_{X'_{k+1}}$ and $\vec{\mathbf{V}}_{X'_k}$ on two consecutive edges $\mathbf{X}'_{k+1}$ and $\mathbf{X}'_k$ in this graph are related by the following recursion:

$$\vec{\mathbf{V}}_{X'_{k+1}} = \mathbf{A}\vec{\mathbf{V}}_{X'_k}\mathbf{A}^\mathsf{T} - \mathbf{A}\vec{\mathbf{V}}_{X'_k}\mathbf{C}^\mathsf{T}\big(\mathbf{V}_Z + \mathbf{C}\vec{\mathbf{V}}_{X'_k}\mathbf{C}^\mathsf{T}\big)^{-1}\mathbf{C}\vec{\mathbf{V}}_{X'_k}\mathbf{A}^\mathsf{T} + \mathbf{B}\mathbf{V}_U\mathbf{B}^\mathsf{T} \tag{3.8}$$

This equation follows directly from applying the message update rules in Gaussian factor graphs [65]. We define a *steady-state* solution to be a

covariance matrix $\vec{\mathbf{V}}_{X'}$ that satisfies the equation

$$\vec{\mathbf{V}}_{X'} = \mathbf{A}\vec{\mathbf{V}}_{X'}\mathbf{A}^{\mathsf{T}} - \mathbf{A}\vec{\mathbf{V}}_{X'}\mathbf{C}^{\mathsf{T}}\big(\mathbf{V}_Z + \mathbf{C}\vec{\mathbf{V}}_{X'}\mathbf{C}^{\mathsf{T}}\big)^{-1}\mathbf{C}\vec{\mathbf{V}}_{X'}\mathbf{A}^{\mathsf{T}} + \mathbf{B}\mathbf{V}_U\mathbf{B}^{\mathsf{T}}\,, \tag{3.9}$$

known as the *discrete-time algebraic Riccati equation*. Making an equivalent formulation to (3.8) for the matrices $\overrightarrow{\mathbf{W}}_{S_k}$, $\overleftarrow{\mathbf{W}}_{X'_k}$, and $\overleftarrow{\mathbf{V}}_{X_k}$ also leads to discrete-time Riccati algebraic equations. We extend the notion of steady-state covariance matrices to all edges in the factor graph in Figure 3.2 and we indicate this notationally by omitting the time index subscript.

Equation (3.9) is highly nonlinear [50, 97]. Existence and uniqueness of positive (semi-)definite solutions is, e.g., summarized in [50, Appendix E]. In our case, Assumption 3.1a suffices to guarantee the existence of unique positive definite steady-state solutions $\vec{\mathbf{V}}_{X'}$, $\overrightarrow{\mathbf{W}}_{S_k}$, $\overleftarrow{\mathbf{W}}_{X'_k}$, and $\overleftarrow{\mathbf{V}}_{X_k}$ as long as the system is stable, i.e. as long as all poles (eigenvalues of $\mathbf{A}$) lie in the interior of the unit disc on the complex plane. In general, however, no closed form for these solutions are known.

3.3 Autonomous Systems and Systems with Forgetting Factor

If the uncertainty of the input $\boldsymbol{U}_k$ in the LTI system (3.7) is removed, i.e. if $\boldsymbol{U}_k = \tilde{\boldsymbol{u}}_k$, then the model can be reformulated as an RLS graph (cf. Section 2.6). In this sections we take a closer look at such autonomous systems (Figure 3.3) without doing this conversion and we make connections with a non-autonomous system.

Example 3.1: Autonomous System for Superposed Sinusoids
We devise an autonomous SSM in Jordan canonical form that models a signal consisting of a sum of exponentially decaying (or increasing) sinusoids. Specifically, we want to formulate a factor graph whose global function is proportional to the PDF of the signal

$$Y_k = \sum_{m=1}^{M} \mathrm{Re}\big(\xi_m(t_k)\big) + Z_k\,, \tag{3.10}$$

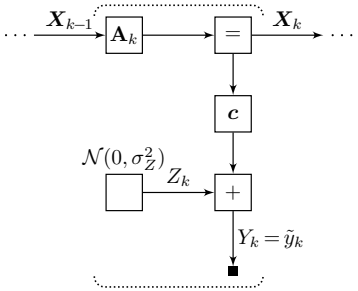


Figure 3.3: An autonomous linear state-space model (SSM).

with

$$\xi_m(t) \triangleq e^{\alpha_m t + i(\omega_m t + \phi_m)}\,, \tag{3.11}$$

$$\mathrm{Re}\big(\xi_m(t)\big) = e^{\alpha_m t}\cos(\omega_m t + \phi_m)\,, \tag{3.12}$$

where α_m, ω_m, and ϕ_m are the decay factor, the frequency and the phase respectively of the m-th sinusoid. In the above, $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0,\sigma_Z^2)$ and t_k are the time stamps at which we observe the signal.

Evidently, the corresponding autonomous SSM in Jordan canonical form in Figure 3.3 has

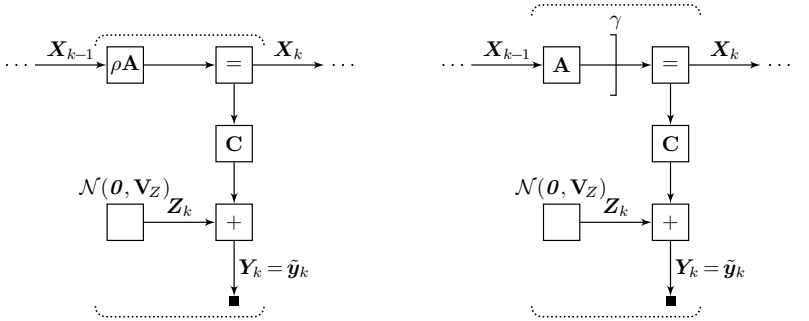
$$\mathbf{A}_k \triangleq \begin{bmatrix} \rho_k^{(1)}\,\mathrm{rotm}\,\Omega_k^{(1)} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \rho_k^{(M)}\,\mathrm{rotm}\,\Omega_k^{(M)} \end{bmatrix}, \tag{3.13}$$

$$\boldsymbol{c} \triangleq \big[1, 0, \cdots, 1, 0\big]\,, \tag{3.14}$$

where $\rho_k^{(m)} \triangleq e^{\alpha_m(t_k - t_{k-1})}$, $\Omega_k^{(m)} \triangleq \omega_m(t_k - t_{k-1})$ for $m = 1, \ldots, M$. In this system, the state

$$\boldsymbol{X}_k = \big[\mathrm{Re}\,\xi_1(t_k), \mathrm{Im}\,\xi_1(t_k), \cdots, \mathrm{Re}\,\xi_M(t_k), \mathrm{Im}\,\xi_M(t_k)\big]^{\mathsf{T}} \tag{3.15}$$

contains the real and imaginary parts of the signal components. Hence, ML estimation of the parameters α_m, ϕ_m based on observations $\tilde{y}_1, \ldots, \tilde{y}_K$ can be done by first computing the ML estimate $\hat{\boldsymbol{x}}_K = \overrightarrow{\boldsymbol{m}}_{X_K}$ followed by inverting the mapping to the parameter-space. ♢



(a) Autonomous system with scaled state-transition matrix.

(b) Autonomous system with forgetting factor γ.

Figure 3.4: Autonomous systems.

3.3.1 Induced Message Passing Filter

In this section we scrutinize the effects of scaling the state-transition matrix $\mathbf{A}$ of an autonomous system and of applying a forgetting factor to an autonomous system. Consider the two time-invariant SSMs that factor as shown in Figure 3.4:

(a) An autonomous system with scaled state-transition matrix $\rho\mathbf{A}$, where $\rho > 0$.

(b) An autonomous system with a forgetting factor γ where $0 < \gamma < 1$.

In the following we first make explicit the filter induced by message passing for given observed data $\boldsymbol{Y}_k = \tilde{\boldsymbol{y}}_k$ in these two SSMs. Second, we show in which sense these two SSMs can be made equivalent to the non-autonomous SSM in Figure 3.5. Finally we give an analytic solution to the steady-state covariance matrices for both SSMs.

Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of $\mathbf{A}$. We restrict ourselves to the steady-state case. The state of the message passing filter is chosen to be

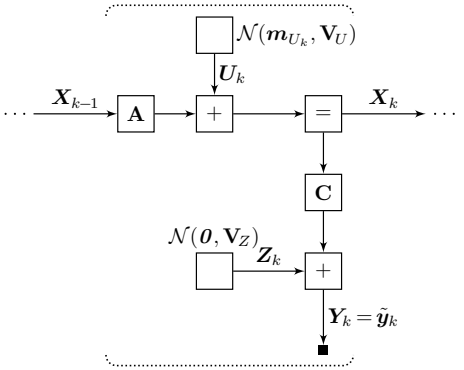


Figure 3.5: A linear system with nonzero input $\boldsymbol{m}_{U_k}$.

the weighted mean $\overrightarrow{\mathbf{W}}_{X_k}\overrightarrow{\boldsymbol{m}}_{X_k}$. For the SSM in Figure 3.4a we obtain

$$\overrightarrow{\mathbf{W}}_X = \rho^{-2}\mathbf{A}^{-\mathsf{T}}\overrightarrow{\mathbf{W}}_X\mathbf{A}^{-1} + \mathbf{C}^{\mathsf{T}}\mathbf{V}_Z^{-1}\mathbf{C}\,, \tag{3.16}$$

$$\overrightarrow{\mathbf{W}}_X\overrightarrow{\boldsymbol{m}}_{X_k} = \rho^{-1}\mathbf{A}^{-\mathsf{T}}\overrightarrow{\mathbf{W}}_X\overrightarrow{\boldsymbol{m}}_{X_{k-1}} + \mathbf{C}^{\mathsf{T}}\mathbf{V}_Z^{-1}\tilde{\boldsymbol{y}}_k\,, \tag{3.17}$$

$$\overrightarrow{\boldsymbol{m}}_{X_k} = \rho^{-1}\overrightarrow{\mathbf{W}}_X^{-1}\mathbf{A}^{-\mathsf{T}}\overrightarrow{\mathbf{W}}_X\overrightarrow{\boldsymbol{m}}_{X_{k-1}} + \overrightarrow{\mathbf{W}}_X^{-1}\mathbf{C}^{\mathsf{T}}\mathbf{V}_Z^{-1}\tilde{\boldsymbol{y}}_k\,. \tag{3.18}$$

In Equation (3.17) we identify the state-transition matrix for the message passing filter as $\rho^{-1}\mathbf{A}^{-\mathsf{T}}$. Hence, the message passing filter has poles at $\lambda_1/\rho, \ldots, \lambda_n/\rho$.

The equivalence with the non-autonomous SSM in Figure 3.5 is defined by specifying the missing parameters $\mathbf{V}_U$ and $\boldsymbol{m}_{U_k}$. (All the other parameters, i.e. $\mathbf{A}$, $\mathbf{C}$, and $\mathbf{V}_Z$ are assumed to be the same as in Figure 3.4.) It is straightforward to show from Equation (3.18) that the choice

$$\mathbf{V}_U = (\rho^2 - 1)\mathbf{A}\,\overrightarrow{\mathbf{W}}_X^{-1}\mathbf{A}^{\mathsf{T}} \tag{3.19}$$

$$\boldsymbol{m}_{U_k} = (\rho - 1)\mathbf{A}\,\overrightarrow{\boldsymbol{m}}_{X_{k-1}} \tag{3.20}$$

renders the two models (Figure 3.4a and Figure 3.5) equivalent up to a change in the scale factor with respect to the represented PDF.

Proof of Equations (3.19) *and* (3.20). Equation (3.19) is proved as

$$\overrightarrow{\mathbf{V}}_{X'} = \rho^2\mathbf{A}\,\overrightarrow{\mathbf{V}}_X\mathbf{A}^{\mathsf{T}} \tag{3.21}$$

$$= \mathbf{A}\,\overrightarrow{\mathbf{V}}_X\mathbf{A}^{\mathsf{T}} + \mathbf{V}_U\,, \tag{3.22}$$

where the first and the second equality follow from applying update rules in Figure 3.4a and Figure 3.5 respectively. Analogously for Equation (3.20) we have

$$\begin{aligned} \overrightarrow{\boldsymbol{m}}_{X'_k} &= \rho\, \mathbf{A}\, \overrightarrow{\boldsymbol{m}}_{X_{k-1}} && (3.23)\\ &= \mathbf{A}\, \overrightarrow{\boldsymbol{m}}_{X_{k-1}} + \boldsymbol{m}_{U_k} && (3.24) \end{aligned}$$

where again, the first and the second equality follow from applying update rules in Figure 3.4a and Figure 3.5 respectively. □

For the system in Figure 3.4b we obtain

$$\begin{aligned} \overrightarrow{\mathbf{W}}_X &= \gamma \mathbf{A}^{-\mathsf{T}} \overrightarrow{\mathbf{W}}_X \mathbf{A}^{-1} + \mathbf{C}^{\mathsf{T}} \mathbf{V}_Z^{-1} \mathbf{C}\,, && (3.25)\\ \overrightarrow{\mathbf{W}}_X \overrightarrow{\boldsymbol{m}}_{X_k} &= \gamma \mathbf{A}^{-\mathsf{T}} \overrightarrow{\mathbf{W}}_X \overrightarrow{\boldsymbol{m}}_{X_{k-1}} + \mathbf{C}^{\mathsf{T}} \mathbf{V}_Z^{-1} \tilde{\boldsymbol{y}}_k\,, && (3.26)\\ \overrightarrow{\boldsymbol{m}}_{X_k} &= \gamma \overrightarrow{\mathbf{W}}_X^{-1} \mathbf{A}^{-\mathsf{T}} \overrightarrow{\mathbf{W}}_X \overrightarrow{\boldsymbol{m}}_{X_{k-1}} + \overrightarrow{\mathbf{W}}_X^{-1} \mathbf{C}^{\mathsf{T}} \mathbf{V}_Z^{-1} \tilde{\boldsymbol{y}}_k\,. && (3.27) \end{aligned}$$

Equation (3.26) directly shows that the message passing filter has poles at $\gamma\lambda_1, \ldots, \gamma\lambda_n$. The equivalence with the non-autonomous system in Figure 3.5 is established by choosing

$$\begin{aligned} \mathbf{V}_U &= (\gamma^{-1} - 1)\mathbf{A}\, \overrightarrow{\mathbf{W}}_X^{-1} \mathbf{A}^{\mathsf{T}} && (3.28)\\ \boldsymbol{m}_{U_k} &= \boldsymbol{0}\,. && (3.29) \end{aligned}$$

Proof of Equations (3.28) *and* (3.29). Equation (3.28) is proved as

$$\begin{aligned} \overrightarrow{\mathbf{V}}_{X'} &= \gamma^{-1} \mathbf{A}\, \overrightarrow{\mathbf{V}}_X \mathbf{A}^{\mathsf{T}} && (3.30)\\ &= \mathbf{A}\, \overrightarrow{\mathbf{V}}_X \mathbf{A}^{\mathsf{T}} + \mathbf{V}_U\,, && (3.31) \end{aligned}$$

where the first and the second equality follow from applying update rules in Figure 3.4b and Figure 3.5 respectively. Analogously for Equation (3.29) we have

$$\overrightarrow{\boldsymbol{m}}_{X'_k} = \mathbf{A}\, \overrightarrow{\boldsymbol{m}}_{X_{k-1}} \tag{3.32}$$

in both Figures 3.4b and 3.5. □

We note that only in the version with a forgetting factor we can achieve the effect of an equivalent non-autonomous system with zero-mean input. The above findings can easily be extended to the time-varying case.

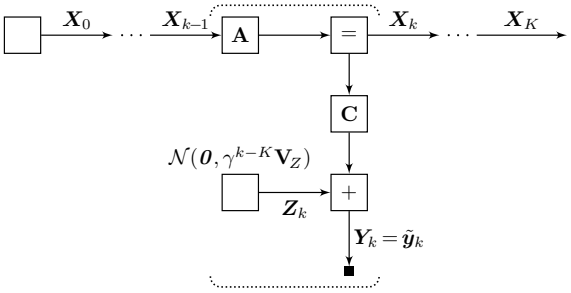


Figure 3.6: An equivalent system to Figure 3.4b.

We mention yet another SSM that is equivalent to the system in Figure 3.4b. For a given time period $k = 1, \ldots, K$, the definition of a forgetting factor allows us to convert the factor graph of in Figure 3.4b at time K into the equivalent graph shown in Figure 3.6. This latter factor graph represents a SSM with time-varying noise variance $\gamma^{k-K}\mathbf{V}_Z$. To achieve the steady state in this system, the initial message $\overrightarrow{\mu}_{X_0}$ should be chosen with $\overrightarrow{\boldsymbol{m}}_{X_0} = \boldsymbol{0}$, $\overrightarrow{\mathbf{W}}_{X_0} = \overrightarrow{\mathbf{W}}_X$. Note that the equivalence is only up to a change in the scale factor with respect to the represented PDF.

3.3.2 Steady-State Solution

For the systems of Figures 3.4a and 3.4b, the steady-state solution $\overrightarrow{\mathbf{W}}_X$ must fulfill (3.16) and (3.25) respectively. Equations of this form are known as Lyapunov equations. In contrast to the discrete-time algebraic Riccati equation (3.9), a Lyapunov equation is linear.

The solutions to Equation (3.16) for the system if Figure 3.4a can be summarized as follows [50]. Remember that $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of $\mathbf{A}$. If $\rho^2\lambda_i \neq 1$ for all $i = 1, \ldots, n$, then there exists a steady-state solution $\overrightarrow{\mathbf{W}}_X$ given as

$$\operatorname{cvect}\overrightarrow{\mathbf{W}}_X = \left(\mathbf{I}_n - \rho^{-2}\left(\mathbf{A}^{-\mathsf{T}} \otimes \mathbf{A}^{-\mathsf{T}}\right)\right)^{-1} \operatorname{cvect}\left(\mathbf{C}^{\mathsf{T}}\mathbf{V}_Z^{-1}\mathbf{C}\right). \tag{3.33}$$

Similarly, a steady-state solution $\overrightarrow{\mathbf{W}}_X$ to Equation (3.25) for the system (b)

exists if $\lambda_i/\gamma \neq 1$ for all $i = 1, \ldots, n$ and can be written as

$$\operatorname{cvect} \overrightarrow{\mathbf{W}}_X = \Big(\mathbf{I}_n - \gamma\big(\mathbf{A}^{-\mathsf{T}} \otimes \mathbf{A}^{-\mathsf{T}}\big)\Big)^{-1} \operatorname{cvect}\big(\mathbf{C}^{\mathsf{T}} \mathbf{V}_Z^{-1} \mathbf{C}\big)\,. \tag{3.34}$$

Proof of Equations (3.33) *and* (3.34). In Appendix E.3 we use a linear algebra interpretation of factor graphs to show that the Lyapunov equation (3.25) is equivalent to

$$\mathbf{A}' \operatorname{cvect} \overrightarrow{\mathbf{W}}_X = \operatorname{cvect}\big(\mathbf{C}^{\mathsf{T}} \mathbf{V}_Z^{-1} \mathbf{C}\big)\,, \tag{3.35}$$

where

$$\mathbf{A}' \triangleq \mathbf{I} - \rho^{-2}\big(\mathbf{A}^{-\mathsf{T}} \otimes \mathbf{A}^{-\mathsf{T}}\big)\,. \tag{3.36}$$

Thus, a solution exists if and only if $\mathbf{A}'$ is nonsingular. The $2n$ eigenvalues of $\big(\mathbf{A}^{-\mathsf{T}} \otimes \mathbf{A}^{-\mathsf{T}}\big)$ can be shown to be $\{\lambda_i^{-1}\, \lambda_j^{-1}\}$ for $i, j = 1, \ldots, n$. Hence the eigenvalues of $\mathbf{A}'$ are $\{1 - 1/(\rho^2\, \lambda_i\, \lambda_j)\}$ for $i, j = 1, \ldots, n$. The condition $\rho^2 \lambda_i \neq 1$ for all $i = 1, \ldots, n$ implies that none of the eigenvalues of $\mathbf{A}'$ vanish and hence this matrix is invertible. Moreover the steady-state solutions can be shown to be positive definite if the pair $\{\rho^{-2}\mathbf{A}, \mathbf{C}^{\mathsf{T}} \mathbf{V}_Z^{-1} \mathbf{C}\}$ is controllable.

An analogous argument is valid for Equation (3.34). □

Finally note that the steady-state message passing filter for both systems (a) and (b) can potentially be unstable: the former if $\rho^{-1} < |\lambda_1|$ and the latter if $\gamma < |\lambda_1|$, where λ_1 is the smallest eigenvalue of $\mathbf{A}$ in magnitude. This stands in contrast to the non-autonomous LTI case in which the message passing filter can be shown to be stable under the Assumptions 3.1 as long as the LTI system is stable.

As a closing remark let us note that analogous results apply for backward message passing. The main difference is that in all the above arguments $\lambda_1, \ldots, \lambda_n$ are the inverses of the eigenvalues of $\mathbf{A}$.

3.3.3 The Second-Order Case

Second-order systems with a complex pole pair are of special interest as they will appear in later parts of this thesis. In this subsection we will show how the message passing filter for such systems is connected with the discrete-time Fourier transform.

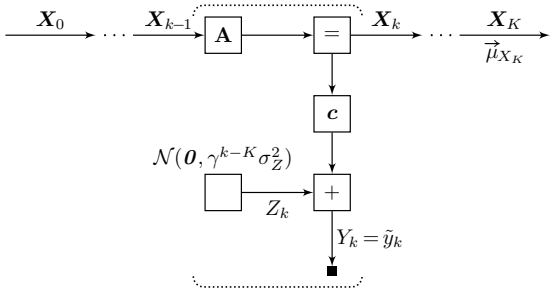


Figure 3.7: Second-order autonomous system with forgetting factor for forward message passing.

Forward Message Passing

Consider the factor graph in Figure 3.7, in which we have relocated the effect of a forgetting factor γ for forward message passing to the noise variance of Z_k. The system matrices are

$$\mathbf{A} \triangleq \rho \,\mathrm{rotm}\,\Omega\,, \;\; \boldsymbol{c} \triangleq \big[1, 0\big]\,. \tag{3.37}$$

Such a system models essentially an exponentially decaying (or increasing) sinusoid. Given the fixed data $Y_k = \tilde{y}_k$ the message $\vec{\mu}_{X_K}$ is

$$\vec{\mathbf{W}}_{X_K} = \frac{1}{2\sigma_Z^2}\left(\frac{\vec{v}^{\,K} - 1}{\vec{v} - 1}\mathbf{I}_2 + \frac{1}{d}\begin{bmatrix} a & b \\ b & -a \end{bmatrix}\right), \tag{3.38}$$

with

$$d \triangleq \vec{v}^{\,2} - 2\vec{v}\cos(2\Omega) + 1 \tag{3.39}$$

$$a \triangleq \vec{v}^{\,K+1}\cos(2K\Omega - 2\Omega) - \vec{v}^{\,K}\cos(2K\Omega) - \vec{v}\cos(2\Omega) + 1 \tag{3.40}$$

$$b \triangleq \vec{v}^{\,K+1}\sin(2K\Omega - 2\Omega) - \vec{v}^{\,K}\sin(2K\Omega) + \vec{v}\sin(2\Omega) \tag{3.41}$$

$$\vec{v} \triangleq \gamma/\rho^2\,, \tag{3.42}$$

and

$$\vec{\mathbf{W}}_{X_K}\vec{\boldsymbol{m}}_{X_K} = \frac{\mathrm{rotm}(K\Omega)}{\sigma_Z^2}\sum_{k=1}^{K}(\gamma/\rho)^{K-k}\tilde{y}_k\begin{bmatrix}\cos(k\Omega) \\ -\sin(k\Omega)\end{bmatrix}. \tag{3.43}$$

In (3.43) we immediately recognize the real and imaginary parts of the discrete-time Fourier transform of the windowed signal $(\gamma/\rho)^{K-k}\tilde{y}_k$. The concept of exponential windows for Fourier transforms goes back to [103].

Next we consider the case of no decay, i.e. $\gamma = 1$ and $\rho = 1$. In this case Equations (3.38) and (3.43) simplify to

$$\overrightarrow{\mathbf{W}}_{X_K} = \frac{K}{2\sigma_Z^2}\mathbf{I}_2 + \frac{\sin(K\Omega)}{2\sigma_Z^2 \sin\Omega}\,\mathrm{rotm}(K\Omega - \Omega)\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}, \tag{3.44}$$

$$\overrightarrow{\mathbf{W}}_{X_K}\overrightarrow{\boldsymbol{m}}_{X_K} = \frac{\mathrm{rotm}(K\Omega)}{\sigma_Z^2}\sum_{k=1}^{K}\tilde{y}_k\begin{bmatrix}\cos(k\Omega)\\ -\sin(k\Omega)\end{bmatrix}. \tag{3.45}$$

If we choose one of the discrete Fourier transform (DFT) frequencies $\Omega_n = 2\pi n/K$ then $\sin(K\Omega_n) = 0$ and $\cos(K\Omega_n) = 1$. Hence, the above simplifies to $\overrightarrow{\mathbf{W}}_{X_K} = \frac{K}{2\sigma_z^2}\mathbf{I}_2$ and

$$\overrightarrow{\mathbf{W}}_{X_K}\overrightarrow{\boldsymbol{m}}_{X_K} = \frac{1}{\sigma_Z^2}\sum_{k=1}^{K}\tilde{y}_k\begin{bmatrix}\cos(k\Omega_n)\\ -\sin(k\Omega_n)\end{bmatrix} \tag{3.46}$$

$$= \frac{\mathrm{rotm}(\Omega_n)^\mathsf{T}}{\sigma_Z^2}\sum_{k=0}^{K-1}\tilde{y}_{k+1}\begin{bmatrix}\cos(k\Omega_n)\\ -\sin(k\Omega_n)\end{bmatrix} \tag{3.47}$$

$$\overrightarrow{\boldsymbol{m}}_{X_K} = \frac{2\,\mathrm{rotm}(\Omega_n)^\mathsf{T}}{K}\begin{bmatrix}\mathrm{Re}\,\breve{y}_n\\ \mathrm{Im}\,\breve{y}_n\end{bmatrix}, \tag{3.48}$$

where

$$\breve{y}_n \triangleq \sum_{k=0}^{K-1}\tilde{y}_{k+1}\,e^{-i2\pi kn/K} \tag{3.49}$$

is the n-th component of the DFT of $\tilde{y}_1, \ldots, \tilde{y}_K$.

There are several consequences of this connection between the message $\overrightarrow{\mu}_{X_K}$ and Fourier theory:

- Orthogonality as occurring in Fourier theory carries over to the computation of messages. This will be treated in Section 3.5.
- In the DFT case, fast versions of the DFT can be used to efficiently compute messages of several SSMs in one go.

- Any recursive method of computing variants of Fourier transforms [47, 48] can be used as efficient method for computing messages in an online setting.

Finally, we give an analytic expression of the steady-state covariance matrix $\overrightarrow{\mathbf{W}}_X$ as

$$\overrightarrow{\mathbf{W}}_X = \frac{1}{2\sigma_Z^2}\left(\frac{1}{1-\overrightarrow{v}}\mathbf{I}_2 + \frac{1}{\overrightarrow{v}^2 - 2\overrightarrow{v}\cos(2\Omega)+1} \cdot \begin{bmatrix} 1-\overrightarrow{v}\cos(2\Omega) & \overrightarrow{v}\sin(2\Omega) \\ \overrightarrow{v}\sin(2\Omega) & \overrightarrow{v}\cos(2\Omega)-1 \end{bmatrix}\right), \tag{3.50}$$

where we recall the definition $\overrightarrow{v} \triangleq \gamma/\rho^2$. Equation (3.50) is valid only if $\overrightarrow{v} < 1$.

Details of the proof of all the results in this section are given in Appendix A.1.

Backward Message Passing

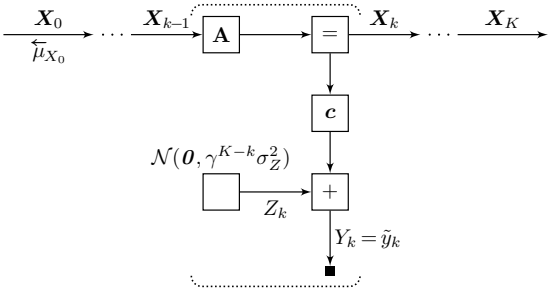


Figure 3.8: Second-order autonomous system with forgetting factor for backward message passing.

Consider the factor graph in Figure 3.8, in which we have relocated the effect of a forgetting factor γ for backward message passing to the noise variance of Z_k. The system matrices are again given in Equation (3.37). Given the fixed data $Y_k = \tilde{y}_k$ the message $\overleftarrow{\mu}_{X_0}$ is

$$\overleftarrow{\mathbf{W}}_{X_0} = \frac{\overleftarrow{v}}{2\sigma_Z^2}\left(\frac{\overleftarrow{v}^K - 1}{\overleftarrow{v} - 1}\mathbf{I}_2 - \frac{1}{d}\begin{bmatrix} -a & b \\ b & a \end{bmatrix}\right), \tag{3.51}$$

with d given in (3.39),

$$a \triangleq -\overleftarrow{v}^K \cos(2K\Omega + 2\Omega) + \overleftarrow{v}^{K+1} \cos(2K\Omega) + \cos(2\Omega) - \overleftarrow{v} \tag{3.52}$$

$$b \triangleq -\overleftarrow{v}^K \sin(2K\Omega + 2\Omega) + \overleftarrow{v}^{K+1} \sin(2K\Omega) + \sin(2\Omega) \tag{3.53}$$

$$\overleftarrow{v} \triangleq \gamma\rho^2\,, \tag{3.54}$$

and

$$\overleftarrow{\mathbf{W}}_{X_0} \overleftarrow{\boldsymbol{m}}_{X_0} = \frac{1}{\sigma_Z^2} \sum_{k=1}^{K} (\gamma\rho)^k \tilde{y}_k \begin{bmatrix} \cos(k\Omega) \\ -\sin(k\Omega) \end{bmatrix}. \tag{3.55}$$

In the case of no decay ($\gamma = 1$ and $\rho = 1$), Equations (3.51) and (3.55) simplify to

$$\overleftarrow{\mathbf{W}}_{X_0} = \frac{K}{2\sigma_Z^2}\mathbf{I}_2 - \frac{\sin(K\Omega)}{2\sigma_Z^2 \sin\Omega} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \operatorname{rotm}(K\Omega + \Omega)\,, \tag{3.56}$$

$$\overleftarrow{\mathbf{W}}_{X_0} \overleftarrow{\boldsymbol{m}}_{X_0} = \frac{1}{\sigma_Z^2} \sum_{k=1}^{K} \tilde{y}_k \begin{bmatrix} \cos(k\Omega) \\ -\sin(k\Omega) \end{bmatrix}. \tag{3.57}$$

If we choose one of the DFT frequencies $\Omega_n = 2\pi n/K$ then $\sin(K\Omega_n) = 0$. Hence, the above simplifies to $\overrightarrow{\mathbf{W}}_{X_K} = \frac{K}{2\sigma_z^2}\mathbf{I}_2$ and

$$\overleftarrow{\mathbf{W}}_{X_0} \overleftarrow{\boldsymbol{m}}_{X_0} = \frac{\operatorname{rotm}(\Omega_n)^\mathsf{T}}{\sigma_Z^2} \sum_{k=0}^{K-1} \tilde{y}_{k+1} \begin{bmatrix} \cos(k\Omega_n) \\ -\sin(k\Omega_n) \end{bmatrix} \tag{3.58}$$

$$\overleftarrow{\boldsymbol{m}}_{X_0} = \frac{2\operatorname{rotm}(\Omega_n)^\mathsf{T}}{K} \begin{bmatrix} \operatorname{Re} \breve{y}_n \\ \operatorname{Im} \breve{y}_n \end{bmatrix}, \tag{3.59}$$

where

$$\breve{y}_n \triangleq \sum_{k=0}^{K-1} \tilde{y}_{k+1}\, e^{-i2\pi kn/K} \tag{3.60}$$

is the n-th component of the DFT of $\tilde{y}_1, \ldots, \tilde{y}_K$.

Finally, we give an analytic expression of the steady-state covariance

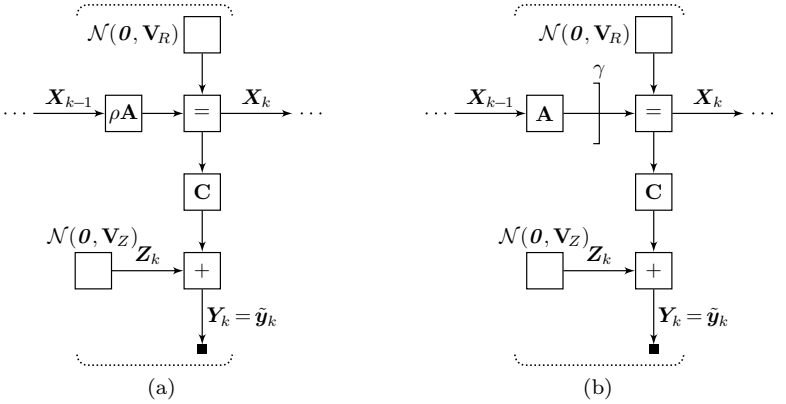


Figure 3.9: Autonomous systems including regularization.

matrix $\overleftarrow{\mathbf{W}}_X$ as

$$\overleftarrow{\mathbf{W}}_X = \frac{\overleftarrow{\upsilon}}{2\sigma_Z}\left(\frac{1}{1-\overleftarrow{\upsilon}} - \frac{1}{\overleftarrow{\upsilon}^2 - 2\overleftarrow{\upsilon}\cos(2\Omega) + 1} \cdot \begin{bmatrix} \overleftarrow{\upsilon} - \cos(2\Omega) & \sin(2\Omega) \\ \sin(2\Omega) & \cos(2\Omega) - \overleftarrow{\upsilon} \end{bmatrix}\right), \tag{3.61}$$

where we recall the definition $\overleftarrow{\upsilon} \triangleq \gamma\rho^2$. This equation is only valid if $\overleftarrow{\upsilon} < 1$.

Details of the proof of all the above results are given in Appendix A.2.

3.3.4 Generalization to Incorporating Distributed Regularization

The autonomous systems in Figure 3.4 can straightforwardly be generalized to include distributed regularization (cf. Section 2.5, Figure 2.5). The generalizations are depicted in Figure 3.9.

Conceptually, the regularization factor can be viewed as an additional noisy observation of the system state. Therefore, Equations (3.16)–(3.18) and Equations (3.25)–(3.27) are merely expanded by an additive term in

which $\mathbf{A}$ does not feature, and the poles of the corresponding message passing filters do not change.

The steady-state equations are still Lyapunov equations. Specifically, the solution in (3.33) changes to

$$\operatorname{cvect} \overrightarrow{\mathbf{W}}_X = \Big(\mathbf{I}_n - \rho^{-2}\big(\mathbf{A}^{-\mathsf{T}} \otimes \mathbf{A}^{-\mathsf{T}}\big)\Big)^{-1} \operatorname{cvect}\big(\mathbf{C}^{\mathsf{T}}\mathbf{V}_Z^{-1}\mathbf{C} + \mathbf{V}_R^{-1}\big)\,. \tag{3.62}$$

and the solution in (3.34) changes to

$$\operatorname{cvect} \overrightarrow{\mathbf{W}}_X = \Big(\mathbf{I}_n - \gamma\big(\mathbf{A}^{-\mathsf{T}} \otimes \mathbf{A}^{-\mathsf{T}}\big)\Big)^{-1} \operatorname{cvect}\big(\mathbf{C}^{\mathsf{T}}\mathbf{V}_Z^{-1}\mathbf{C} + \mathbf{V}_R^{-1}\big)\,. \tag{3.63}$$

The regularization factor considered here is a zero mean Gaussian factor with covariance matrix $\mathbf{V}_R$. The latter can be understood as a generalization of the regularization parameter λ introduced in Section 2.5, Equation (2.52). The Gaussian factor leads to a weighted ℓ_2 regularization $r(\boldsymbol{x}) = \boldsymbol{x}^{\mathsf{T}}\mathbf{V}_R^{-1}\boldsymbol{x}$ in the implied cost function. As mentioned in Section 2.5, in principle, other forms of distributed regularization may be useful, e.g., ℓ_1 regularization, thus leading to different factors, e.g., a Laplace PDF. The resulting messages, however, are not Gaussian anymore.

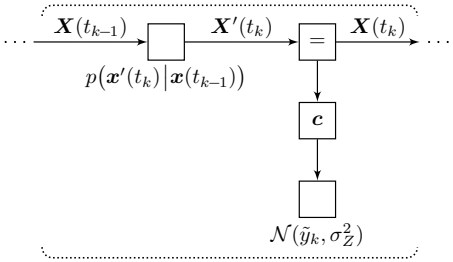
3.4 A Connection between Continuous-Time Systems and Discrete-Time Systems

Here, we draw on [10–12] to provide an equivalent discrete-time system for a given continuous-time system whose output is observed at discrete time instants.

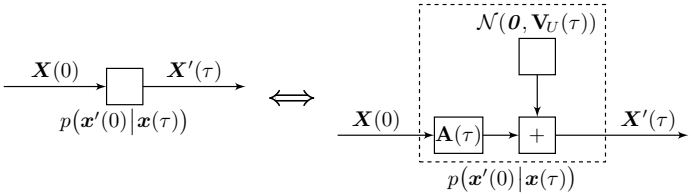
3.4.1 Continuous-Time Systems with Discrete-Time Observations

A stochastic continuous-time LTI SSM is defined by

$$\begin{aligned} \dot{\boldsymbol{X}}(t) &= \mathring{\mathbf{A}}\,\boldsymbol{X}(t) + \boldsymbol{b}\,U(t) \\ Y(t) &= \boldsymbol{c}\,\boldsymbol{X}(t) \end{aligned}\,, \tag{3.64}$$



(a) Continuous-time state-space model (SSM) (3.64) with discrete-time observations (3.65).



(b) Two equivalent factor graphs for the state evolution over a time period τ.

Figure 3.10: Factor graphs for continuous-time state-space models (SSMs) with discrete-time observations.

where $t \in \mathbb{R}$ is the time variable, $\dot{\boldsymbol{X}}(t)$ denotes the time-derivative of $\boldsymbol{X}(t)$, and $U(t) \in \mathbb{R}$ is a white Gaussian noise process with autocorrelation function $\mathsf{E}[U(t)\,U(t+\tau)] = \sigma_U^2\,\delta(\tau)$. We exclusively consider the case where we have discrete-time noisy observations $Y_k = \tilde{y}_k$ of

$$Y_k = Y(t_k) + Z_k\,, \tag{3.65}$$

where $k \in \mathbb{Z}$ are the indices of the sampling times t_k and $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$.

Figure 3.10a shows the factor graph of such a system, in which the factor $p\big(\boldsymbol{x}'(t_k)\big|\boldsymbol{x}(t_{k-1})\big)$ models the state evolution from time t_{k-1} to time t_k. In the following theorem an equivalent formulation of this state evolution node is given.

Theorem 3.1: Equivalence of Continuous-Time and Discrete–Time Systems [10–12]
The two factor graphs in Figure 3.10b are equivalent if

$$\mathbf{A}(\tau) = e^{\tau\mathring{\mathbf{A}}} \triangleq \sum_{m=0}^{\infty} \frac{\big(\tau\mathring{\mathbf{A}}\big)^m}{m!}\,, \tag{3.66}$$

and

$$\mathbf{V}_U(\tau) = \sigma_U^2 \int_0^\tau e^{t\mathring{\mathbf{A}}}\, \boldsymbol{b}\boldsymbol{b}^\mathsf{T}\, e^{t\mathring{\mathbf{A}}^\mathsf{T}}\, \mathrm{d}t\,. \tag{3.67}$$

In Equation (3.66) we recognize the zero-order hold discretization of $\mathring{\mathbf{A}}$ with sampling interval τ and Equation (3.67) is known as the controllability gramian [97]. As a consequence of Theorem 3.1, the factor graph in Figure 3.10a can be substituted by a factor graph that represents the following discrete-time LTI system

$$\begin{aligned} \boldsymbol{X}_k &= \mathbf{A}_k\, \boldsymbol{X}_{k-1} + \boldsymbol{U}_k \\ Y_k &= \boldsymbol{c}\, \boldsymbol{X}_k + Z_k \end{aligned}\,, \tag{3.68}$$

where $\mathbf{A}_k \triangleq \mathbf{A}(t_k - t_{k-1})$, $\boldsymbol{U}_k \sim \mathcal{N}(\boldsymbol{0}, \mathbf{V}_U(t_k - t_{k-1}))$ and $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$. Moreover, in the case of uniform sampling, this system is time-invariant.

Note that, since Theorem 3.1 applies to arbitrary time intervals, the equivalence in Figure 3.10b can be used to model the system state $\boldsymbol{X}(t)$ at any time t. This is done, for instance for some $t_{k-1} < t < t_k$ by splitting the state evolution factor as

$$p\big(\boldsymbol{x}'(t_k)\big|\boldsymbol{x}(t_{k-1})\big) = p\big(\boldsymbol{x}'(t_k)\big|\boldsymbol{x}(t)\big)\, p\big(\boldsymbol{x}(t)\big|\boldsymbol{x}(t_{k-1})\big)\,. \tag{3.69}$$

3.4.2 A State-Space Model for Polynomials

We use the findings of Section 3.4.1 to represent polynomials. A polynomial or order N is defined as

$$y(t) = \sum_{n=0}^{N} d_n\, t^n\,, \tag{3.70}$$

where $d_n \in \mathbb{R}$ are the coefficients. In the following we describe an autonomous continuous-time SSM whose output is a polynomial (3.70). The coefficients d_n are hidden in the state vector of this system.

We start by defining the $(N+1)$-dimensional state as

$$\boldsymbol{x}(t) \triangleq \Big[y(t), \dot{y}(t), \ldots, \overset{\textcircled{N}}{y}(t)\Big]^\mathsf{T}\,, \tag{3.71}$$

where $\overset{(n)}{y}(t)$ denotes the n-th time-derivative of $y(t)$. Note that $\overset{(N)}{y}(t) = N!\, d_N$ is a constant. The autonomous continuous-time SSM thus is

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \mathring{\mathbf{A}}\, \boldsymbol{x}(t) \\ y(t) &= \boldsymbol{c}\, \boldsymbol{x}(t) \end{aligned}, \tag{3.72}$$

with

$$\mathring{\mathbf{A}} \triangleq \begin{bmatrix} \boldsymbol{0} & \mathbf{I}_N \\ 0 & \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{c} \triangleq \begin{bmatrix} 1, 0, \ldots, 0 \end{bmatrix}. \tag{3.73}$$

Using elementary rules of differentiation, the components of the state vector $\boldsymbol{x}(t)$ can be expressed as

$$[\boldsymbol{x}(t)]_n = \sum_{\ell=n}^{N-1} \frac{\ell!\, d_\ell\, t^{\ell-n}}{(\ell-n)!} \tag{3.74}$$

for $n = 0, \ldots, N$. Hence, for the model (3.72) to produce the polynomial (3.70) for $t \geq 0$, the initial state $\boldsymbol{x}(0)$ should be set to $[\boldsymbol{x}(0)]_n = \sum_{\ell=n}^{N-1} \frac{\ell!\, d_\ell}{(\ell-n)!}$ for $n = 0, \ldots, N$.

We extend the autonomous model (3.72) to the stochastic model (3.64) for which we define $\boldsymbol{b} \triangleq \begin{bmatrix} 0, \ldots, 0, 1 \end{bmatrix}^{\mathsf{T}}$. Also, we model discrete-time observations (3.65) at sampling times t_k. In the following, we invoke Theorem 3.1 to formulate an equivalent discrete-time system of the form (3.68).

Since $\mathring{\mathbf{A}}$ is nilpotent of degree $N-1$, the sum in (3.66) can be evaluated to

$$\mathbf{A}(\tau) = \begin{bmatrix} 1 & \tau & \frac{\tau^2}{2} & \frac{\tau^3}{3!} & \cdots & \frac{\tau^N}{N!} \\ 0 & 1 & \tau & \frac{\tau^2}{2} & \cdots & \frac{\tau^{N-1}}{(N-1)!} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{3.75}$$

$$[\mathbf{A}_k]_{i,j} = \begin{cases} \frac{\tau^{j-i}}{(j-i)!} & \text{if } j \geq i \\ 0 & \text{else} \end{cases} \quad \text{for } i, j \in \{0, \ldots, N\}, \tag{3.76}$$

Also, the integral (3.67) can be evaluated to

$$[\mathbf{V}_U(\tau)]_{i,j} = \frac{\sigma_U^2\, \tau^{2N-i-j+1}}{(2N-i-j+1)\,(N-i)!\,(N-j)!} \tag{3.77}$$

for $i, j \in \{0, \ldots, N\}$.

Example 3.2: Cubic Spline Smoothing
Consider the discrete-time linear time-varying SSM (3.68) with

$$\mathbf{A}_k \triangleq \begin{bmatrix} 1 & \tau_k \\ 0 & 1 \end{bmatrix}, \; \boldsymbol{b} \triangleq \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; \boldsymbol{c} \triangleq \begin{bmatrix} 1, 0 \end{bmatrix}, \; \mathbf{V}_{U_k} \triangleq \sigma_U^2 \begin{bmatrix} \tau_k^3/3 & \tau_k^2/2 \\ \tau_k^2/2 & \tau_k \end{bmatrix}, \quad (3.78)$$

where $\tau_k \triangleq t_k - t_{k-1}$ and t_k are the time stamps of the given data $Y_k = \tilde{y}_k$ for $k = 1, \ldots, K$. This is the discrete time equivalent of a noisy line model, i.e. a polynomial of degree $N = 1$ (cf. (3.73), (3.76) and (3.77)). With some given values σ_U^2 and σ_Z^2 Gaussian message passing can be used to compute

$$\hat{\boldsymbol{x}}(t) = \underset{\boldsymbol{x}(t)}{\operatorname{argmax}}\, p\big(\boldsymbol{x}(t) \big| \tilde{y}_1, \ldots, \tilde{y}_K\big) \quad (3.79)$$

for any t that satisfies $t_1 \leq t \leq t_K$. This happens to coincide with cubic spline smoothing [86]. This equivalence is established by comparing the cost functions (cf. [10–12]). In the special case where $\sigma_Z^2 = 0$ we have $[\hat{\boldsymbol{x}}(t_k)]_1 = \tilde{y}_k$ and we get spline interpolation. ♢

If $\tau < \sqrt{3}$, then $[\mathbf{V}_U(\tau)]_{N,N} = \sigma_U^2 \tau$ is the largest element in $\mathbf{V}_U(\tau)$ for any order N. In this case we may approximate $\mathbf{V}_U(\tau) \approx \sigma_U^2 \tau \boldsymbol{b}\boldsymbol{b}^{\mathsf{T}}$. Note that the resulting interpolation $\hat{\boldsymbol{x}}(t)$ for arbitrary $t_{k-1} < t < t_k$ is not a line between $\hat{\boldsymbol{x}}(t_{k-1})$ and $\hat{\boldsymbol{x}}(t_k)$.

Finally, we mention that the interpolation method introduced in Section 2.7.2, in which we have considered a forgetting factor per unit time (2.72), again has a different cost function. Specifically, it can be seen from (2.33) that the time-derivative of the cost function is discontinuous at points $t = t_k$, which results in interpolation estimates $\hat{\boldsymbol{x}}(t)$ with discontinuous time-derivative.

3.4.3 A State-Space Model for Sinusoidal Signals

As in the previous section, we use the findings of Section 3.4.1 to represent sinusoidal signals. We start with the continuous-time equivalent to Example 3.1 in the second-order case. A sinusoid is defined as

$$y(t) = a \cos(\omega t + \phi)\,, \quad (3.80)$$

where a is the amplitude, ω is the frequency, and ϕ is the phase. We define a second-order continuous-time SSM (3.72) whose state vector is

$$\boldsymbol{x}(t) \triangleq a \begin{bmatrix} \cos(\omega t + \phi) \\ \sin(\omega t + \phi) \end{bmatrix}, \tag{3.81}$$

by letting

$$\mathring{\mathbf{A}} \triangleq \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}, \quad \boldsymbol{c} \triangleq \begin{bmatrix} 1, 0 \end{bmatrix}. \tag{3.82}$$

We extend the autonomous system to the stochastic system (3.64) with $\boldsymbol{b} \triangleq \begin{bmatrix} 0, 1 \end{bmatrix}^{\mathsf{T}}$. Also, we consider fixed discrete-time observations $Y_k = \tilde{y}_k$ of (3.65) at sampling times t_k.

Theorem 3.1 allows us to formulate an equivalent discrete-time model of the form (3.68). By considering the eigenvalue decomposition $\mathring{\mathbf{A}} = \mathbf{Q}\,\mathring{\mathbf{\Lambda}}\,\mathbf{Q}^{-1}$ with $\mathbf{Q} \triangleq \frac{1}{\sqrt{2}} \left[\begin{smallmatrix} 1 & 1 \\ i & -i \end{smallmatrix} \right]$, and $\mathring{\mathbf{\Lambda}} \triangleq \operatorname{diag}(-\omega i, \omega i)$ the following results can be derived from Equations (3.66) and (3.67):

$$\mathbf{A}(\tau) = \operatorname{rotm}(\omega\tau) \tag{3.83}$$

$$\mathbf{V}_U(\tau) = \frac{\sigma_U^2}{4\omega} \begin{bmatrix} -\sin(2\omega\tau) + 2\omega\tau & \cos(2\omega\tau) - 1 \\ \cos(2\omega\tau) - 1 & \sin(2\omega\tau) + 2\omega\tau \end{bmatrix} \tag{3.84}$$

Generalizations to systems with repeated pole pairs can be done. Specifically, if $\mathring{\mathbf{A}} \in \mathbb{R}^{(N-1)\times(N-1)}$ is a Jordan block of the form (3.4) with diagonal blocks $\left[\begin{smallmatrix} 0 & -\omega \\ \omega & 0 \end{smallmatrix} \right]$, then

$$\mathbf{A}(\tau) = \begin{bmatrix} 1 & \tau & \frac{\tau^2}{2} & \frac{\tau^3}{3!} & \cdots & \frac{\tau^N}{N!} \\ 0 & 1 & \tau & \frac{\tau^2}{2} & \cdots & \frac{\tau^{N-1}}{(N-1)!} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \otimes \operatorname{rotm}(\omega\tau). \tag{3.85}$$

Analytic expressions of the matrix $\mathbf{V}_U(\tau)$ can be developed for specific values of N. It is conjectured that if $\omega\,\tau < \pi/2$ then $[\mathbf{V}_U(\tau)]_{N,N} = \sigma_U^2(\sin(2\omega\tau) + 2\omega\tau)/(4\omega)$ has the largest magnitude among all elements in $\mathbf{V}_U(\tau)$. In this case we may approximate $\mathbf{V}_U(\tau) \approx \sigma_U^2\,\tau\,\boldsymbol{b}\boldsymbol{b}^{\mathsf{T}}$.

All the remarks from the previous section on interpolation carry over to the case considered here.

3.5 State-Space Splitting and Loopy Graphs

In this section we consider several ways in which a high-dimensional linear SSM can be partitioned into smaller connected SSMs. We start with formulating a completely split model. We then consider the cases of a coupled input and a coupled output.

For simplicity of exposition, let us assume that a LTI 4-th order SSM

$$\begin{aligned} \boldsymbol{X}_k &= \mathbf{A}\,\boldsymbol{X}_{k-1} + \mathbf{B}\,\boldsymbol{U}_k \\ \boldsymbol{Y}_k &= \mathbf{C}\,\boldsymbol{X}_k + \boldsymbol{Z}_k \end{aligned} \tag{3.86}$$

with input noise $\boldsymbol{U}_k \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \mathbf{V}_U)$ and observation noise $\boldsymbol{Z}_k \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \mathbf{V}_Z)$ is parameterized in Jordan canonical form as

$$\boldsymbol{X}_k \triangleq \begin{bmatrix} \boldsymbol{X}_k^{(1)} \\ \boldsymbol{X}_k^{(2)} \end{bmatrix}, \; \mathbf{A} \triangleq \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix} \in \mathbb{R}^{2\times 2}. \tag{3.87}$$

In this section we work with a 4-th order system throughout. Generalizations to higher order SSMs in Jordan canonical form and to LTV systems are evident.

Depending on $\mathbf{B}$, $\mathbf{C}$, $\mathbf{V}_U$, and $\mathbf{V}_Z$ the system may completely decompose into second-order subsystems. Specifically, if

$$\mathbf{B} \triangleq \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{b}_2 \end{bmatrix} \in \mathbb{R}^{4\times 2}, \; \mathbf{V}_U \triangleq \operatorname{diag}\big(\sigma_U^2, \sigma_U^2\big), \tag{3.88}$$

$$\mathbf{C} \triangleq \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{c}_2 \end{bmatrix} \in \mathbb{R}^{2\times 4}, \; \mathbf{V}_Z \triangleq \operatorname{diag}\big(\sigma_Z^2, \sigma_Z^2\big), \tag{3.89}$$

then the two systems completely decouple and the factor graph representation of Figure 3.11 results. Generalizations of the above for vector inputs, vector outputs, and higher order systems in Jordan canonical form are straightforward. In such a situation we gain nothing by treating the 4-th order SSM jointly and Gaussian message passing can be done in each of the sub-graphs separately.

Let us now consider a case in which the input is coupled. Specifically, we let $\mathbf{C}$ and $\mathbf{V}_Z$ be defined as in (3.89), but we redefine

$$\mathbf{B} \triangleq \boldsymbol{b} \triangleq \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}, \; \mathbf{V}_U \triangleq \sigma_U^2, \tag{3.90}$$

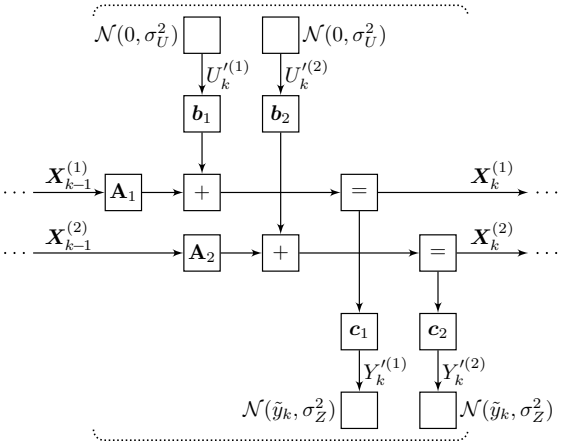


Figure 3.11: Completely split state-space model (SSM).

where $\boldsymbol{b}$ is a column vector. The system input $U_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_U^2)$ is now scalar has a direct influence on both parts of the state. Clearly, the second-order subsystems do not decouple and, in principle, we should treat the 4-th order SSM jointly by doing message passing on the factor graph of Figure 3.2.

We nevertheless can propose to do message passing in the split graph of Figure 3.11 even for this system. Any resulting estimates will only be approximations of actual ML or MAP estimates. In return, any Gaussian message passing algorithm will not have to compute full 4×4 covariance matrices (or precision matrices) but instead can operate with pairs of 2×2 covariance matrices (or precision matrices). While the gain for a 4-th order system may seem minor it increases as $O(n^2)$ for an n-dimensional system.

If the two subsystems are strongly coupled, a complete split as in Figure 3.11 is not attractive. We therefore propose to substitute the input part of the factor graph in Figure 3.11 by the input part shown in Figure 3.12a. Note that this modification results in a loopy factor graph. In this factor graph all covariance matrices (or precision matrices) to be computed still have dimensions 2×2 but we now have to consider iterative message passing algorithms because the factor graph has cycles.

As convergence of iterative message passing algorithms is not guaranteed

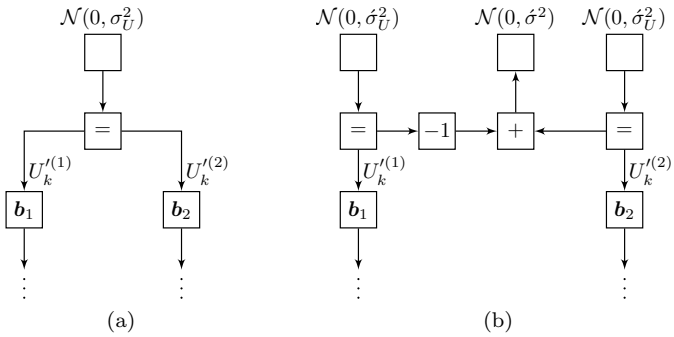


Figure 3.12: Modification at the input for complete coupling (a) and partial coupling (b).

even in the Gaussian case, a soft transition between the two regimes of Figure 3.11 (complete split) and Figure 3.12a (complete coupling) may be desirable. To this end we propose an algorithm based on the modification shown in Figure 3.12b to the input part of the factor graph in Figure 3.11. In this modification we have distributed the input noise over three factors. Clearly, we recover the completely split graph of Figure 3.11 if

$$\acute{\sigma}_U^2 = \sigma_U^2\,,\ \acute{\sigma}^2 \to \infty\,. \tag{3.91}$$

On the other hand, the coupled graph of Figure 3.12a can be recovered by setting

$$\acute{\sigma}_U^2 = 2\sigma_U^2\,,\ \acute{\sigma}^2 = 0\,. \tag{3.92}$$

We now can devise an iterative message passing algorithm by starting with the parameters set as in (3.91) and slowly adapting the parameters to the target values in (3.92). Generalizations to differing noise variances in each second-order subsystem, to vector inputs, and to higher order systems are feasible.

We now turn to a case in which the output is coupled. Specifically, we let $\mathbf{B}$ and $\mathbf{V}_U$ be defined as in (3.88), but we redefine

$$\mathbf{C} \triangleq \boldsymbol{c} \triangleq \left[\boldsymbol{c}_1, \boldsymbol{c}_2\right],\ \mathbf{V}_Z \triangleq \sigma_Z^2\,, \tag{3.93}$$

where $\boldsymbol{c}$ is a row vector. The system output Y_k is now scalar and is influenced by both parts of the state. Clearly, the subsystems do not

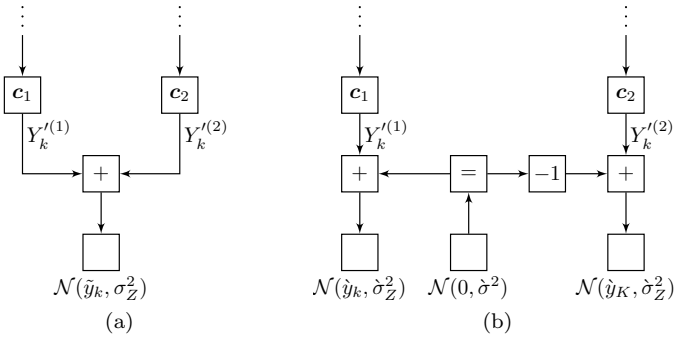


Figure 3.13: Modification at the output for complete coupling (a) and partial coupling (b).

decouple and, in principle, we should treat the 4-th order SSM jointly as in Figure 3.2.

In equivalence to the case of a coupled input we propose to substitute the output part of the factor graph in Figure 3.11 by the output part shown in Figure 3.13a. In the resulting factor graph we consider iterative message passing algorithms as opposed to the completely split graph of Figure 3.11. A soft transition between the two regimes can be achieved by considering the modification of Figure 3.13b in which we have distributed the observation noise over three factors. Clearly, we recover the completely split graph of Figure 3.11 if

$$\grave{y}_k = \tilde{y}_k\,,\ \grave{\sigma}_Z^2 = \sigma_Z^2\,,\ \grave{\sigma}^2 = 0\,. \tag{3.94}$$

On the other hand, the coupled graph of Figure 3.13a can be recovered by setting

$$\grave{y}_k = \tilde{y}_k/2\,,\ \grave{\sigma}_Z^2 = \sigma_Z^2/2\,,\ \grave{\sigma}^2 \to \infty\,. \tag{3.95}$$

We now can devise an iterative message passing algorithm by starting with the parameters set as in (3.94) and slowly adapting the parameters to the target values in (3.95). Generalizations to differing noise variances in each second-order subsystem, to vector inputs, and to higher order systems are feasible.

Of course, the two concepts of coupling at the input and the output of the factor graph can be used jointly and an iterative algorithm can be

devised which slowly transits from the completely decoupled setup in Figure 3.11 to a completely coupled setup of Figures 3.12a at the input and 3.13a at the output.

As has been shown in Section 3.3.3, an autonomous second-order SSM with complex conjugate poles models an exponentially decaying sinusoid, and the weighted means of the messages can be computed with variations of Fourier transforms. We recall the special case in which messages can be computed with the DFT for several models in one go.

Specifically, let us assume that we are given a block of data $\tilde{y}_1, \ldots, \tilde{y}_K$ and that $\mathbf{A}_m \triangleq \text{rotm}(2\pi m/K)$ for $m = 1, \ldots, K$. In this case, $\overrightarrow{\boldsymbol{m}}_{X_K^{(m)}}$ contains the real and imaginary part of the DFT of $\tilde{y}_1, \ldots, \tilde{y}_K$ (cf. Equation (3.48)). From the orthogonality of the components it immediately follows that, for the purpose of computing the means $\overrightarrow{\boldsymbol{m}}_{X_K^{(m)}}$, we can substitute the joint graph of Figure 3.2 by the completely decoupled graph of Figure 3.11. The above equivalence does not hold anymore as soon as the model is changed, e.g., by changing $\mathbf{A}$ or by including a forgetting factor or state noise. Depending on the severity of these changes the equivalence might, however, still hold approximately.

Chapter 4

Parameter Estimation in Linear State-Space Models

4.1 Introduction

In this chapter we consider situations in which a linear SSM is not given completely. Instead we assume that we have to estimate model parameters from knowing only the observations $\boldsymbol{Y}_k = \tilde{\boldsymbol{y}}_k$ for $k = 1, \ldots, K$. A linear SSM is determined by the following parameters:

a) The system order.

b) The system poles, i.e. the matrix $\mathbf{A}$.

c) The system zeros, i.e. the matrices $\mathbf{B}$ and $\mathbf{C}$.

d) The input noise variance or covariance matrix.

e) The observation noise variance or covariance matrix.

We will not address how to estimate the system order in this thesis. Furthermore, let us note that the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ can only be identified up to a similarity transformation.

All algorithms in this chapter are iterative in nature. They start with initial values for the parameters, and refine these in each iteration. We do not present any strategy for choosing the initial values.

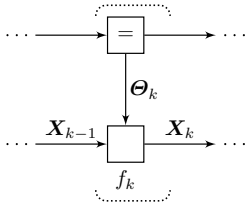


Figure 4.1: Overview factor graph for estimating a parameter (vector) $\boldsymbol{\Theta}$ in a state-space model (SSM) with state $\boldsymbol{X}_k$.

Parameter estimation for SSMs as presented here, can be cast into the general framework of the factor graph in Figure 4.1.

The node f_k contains the linear Gaussian SSM factor graph as detailed in Figure 3.1, with fixed observations $\boldsymbol{Y}_k = \tilde{\boldsymbol{y}}_k$. The model parameters are collected in a parameter vector $\boldsymbol{\theta}_k$. These parameters are assumed to be (at least approximately) equal over time, lest the number of parameters becomes too high. The equality constraints in Figure 3.1 may be softened by means of a forgetting factor, thus allowing us to model a slowly changing system. In the case of an LTI system the factor f_k is detailed in Figure 3.2.

The statistical estimation problem we want to solve is the ML estimation

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \int p(\tilde{\boldsymbol{y}}, \boldsymbol{x}|\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{x} \,, \tag{4.1}$$

where $\boldsymbol{X}^{\mathsf{T}} \triangleq \left[\boldsymbol{X}_0^{\mathsf{T}}, \ldots, \boldsymbol{X}_K^{\mathsf{T}}\right]$ contains all the state vectors and $\tilde{\boldsymbol{y}}^{\mathsf{T}} \triangleq \left[\tilde{\boldsymbol{y}}_1^{\mathsf{T}}, \ldots, \tilde{\boldsymbol{y}}_K^{\mathsf{T}}\right]$ contains all the observations. In general (4.1) is a non-trivial problem which cannot be solved by Gaussian message passing. Also, we might be interested in approximately solving (4.1).

This chapter is structured as follow. The next section exposes some details on the three underlying principles applied in this chapter:

- Cyclic maximization (CM)
- Expectation maximization (EM)
- Local Taylor approximation

These may generally be useful beyond the application to ML estimation in SSMs. CM has been described, e.g., in [99]. Our treatment of the

well-known EM algorithm is based heavily on [26]. Taylor approximations can be found in many places in the literature. We provide two versions, one for general positive factors and one for Dirac delta constraints. The former is known as Laplace approximation [3]. In the latter case the approach results in a linearization of the constraint. In the field of Kalman filtering this linearization is termed extended Kalman filter [18]. The linearization presented here is also used in [9].

In the remaining two sections, these principles are applied to the estimation of $\mathbf{A}$ and the estimation of noise variances.

4.2 General Principles

Here we establish principles that can be applied locally in a factor graph that contains factors for which exact sum-product message passing results in messages other than Gaussian.

The goal is to approximate factors such that Gaussian message passing (or at least messages from the exponential family) can be maintained and cycles in the graph do not prevent convergence. The latter can in general not be guaranteed for local Taylor approximations.

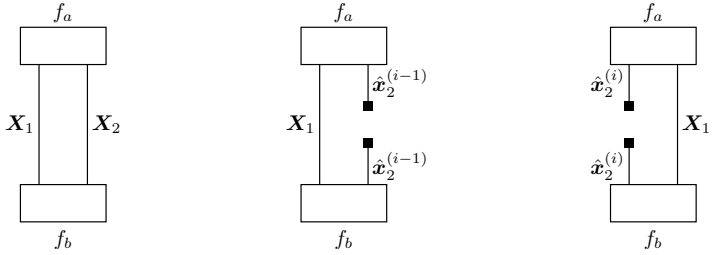
We only treat three possibilities here, while many more exist or would be candidates to consider. Others are variational message passing [8, 25], alternating direction method of multipliers [13], gradient ascent or gradient EM [24], and more.

4.2.1 Cyclic Maximization

In the literature [23, 99], the concept of CM goes under various names, e.g., "iterative conditional modes" (ICM), "alternating maximization", or "coordinate ascent". The idea behind CM is very simple. Consider the maximization problem

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, f(\boldsymbol{x})\,, \tag{4.2}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$. When we partition the vector as $\left[\boldsymbol{x}_1, \boldsymbol{x}_2\right] \triangleq \boldsymbol{x}$, then the following iterative algorithm can be proposed. First choose some



(a) Original graph. (b) Step (4.3). (c) Step (4.4).

Figure 4.2: Cyclic maximization (CM) and max-product message passing in graphs with cycles.

initial estimate $\hat{\boldsymbol{x}}_2^{(0)}$. Then repeat

$$\hat{\boldsymbol{x}}_1^{(i)} = \underset{\boldsymbol{x}_1}{\operatorname{argmax}}\, f\big(\boldsymbol{x}_1, \hat{\boldsymbol{x}}_2^{(i-1)}\big) \tag{4.3}$$

and

$$\hat{\boldsymbol{x}}_2^{(i)} = \underset{\boldsymbol{x}_2}{\operatorname{argmax}}\, f\big(\hat{\boldsymbol{x}}_1^{(i)}, \boldsymbol{x}_2\big)\,, \tag{4.4}$$

where i is the iteration number. The generalization of (4.3)–(4.4) to more than two partitions is straightforward.

Convergence of the CM algorithm is guaranteed as long as in every maximization step, there exists a unique maximum. This is easily seen, by noting that in every step, the function value cannot decrease. In practice, acCM converges to a local maximum for many functions.

If $f(\boldsymbol{x})$ is the global function of a factor graph, then CM may be used to cut cycles for max-product message passing. Consider for example the factorization

$$f(\boldsymbol{x}_1, \boldsymbol{x}_2) = f_a(\boldsymbol{x}_1, \boldsymbol{x}_2)\, f_b(\boldsymbol{x}_1, \boldsymbol{x}_2) \tag{4.5}$$

as depicted in Figure 4.2a. The iterations (4.3) and (4.4) are depicted in Figures 4.2b and 4.2c respectively, in which it is clearly seen that the cycles are broken up.

For SSMs we apply CM on a high level to the problem (4.1) (ML estimation of $\boldsymbol{\theta}$) by partitioning the vector $\boldsymbol{\theta}$ into sub-vectors. For example,

two such sub-vectors may correspond to the eigenvalues of the matrix $\mathbf{A}$ and the output noise variance σ_Z^2 respectively. We then take turns to maximize the sub-vectors.

On a lower level, we cannot in general apply CM directly to cut the cycles in the factor graph of Figure 4.1 for parameter estimation. The reason for this is that CM applies only to maximization problems, while we have to solve the mixed maximization integration problem (4.1).

We nevertheless can propose to solve the different problem

$$\hat{\boldsymbol{\theta}} \triangleq \operatorname*{argmax}_{\boldsymbol{\theta}} \max_{\boldsymbol{x}} p(\tilde{\boldsymbol{y}}, \boldsymbol{x}|\boldsymbol{\theta}) \,, \tag{4.6}$$

which in some cases can be a good approximation to (4.1). Indeed, if the joint PDF of $\boldsymbol{Y}$, $\boldsymbol{X}$, and $\boldsymbol{\theta}$ is Gaussian (i.e. the log-likelihood function is quadratic), then (4.6) is equivalent with (4.1). In our setup we are facing, however, a non-Gaussian problem.

To solve (4.6) by max-product message passing in the factor graph in Figure 4.1, the iteration (4.3)–(4.4) is formulated as

$$\hat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}} p\big(\tilde{\boldsymbol{y}}, \boldsymbol{x}\big|\hat{\boldsymbol{\theta}}\big) \tag{4.7}$$

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmax}_{\boldsymbol{\theta}} p\big(\tilde{\boldsymbol{y}}, \hat{\boldsymbol{x}}\big|\boldsymbol{\theta}\big) \,, \tag{4.8}$$

where we have omitted the iteration number in our notation.

The maximization in (4.7) is done by max-product message passing in the factor graph of the SSM with fixed parameters $\hat{\boldsymbol{\theta}}$, which for linear SSMs amounts to ordinary Gaussian message passing. The maximization in (4.8) is done by max-product message passing in the upper part of the graph of Figure 4.1, after having fixed $\boldsymbol{X}_k = \boldsymbol{m}_{X_k}$ for $k = 0, \ldots, K$.

4.2.2 Expectation Maximization

Expectation maximization (EM) has been described as a message passing algorithm for factor graphs in [23, 26, 59]. We refer the reader to [26] for a general exposition of this subject and for some important cases in the Gaussian setup.

For parameter estimation in linear SSMs as in Figure 4.1, the message passing view of EM can be described as follows.

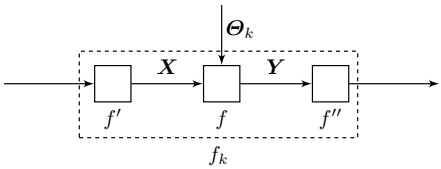


Figure 4.3: Expectation maximization (EM) message for a factor f_k as in Figure 4.1 with internal structure.

a) Choose some initial value $\hat{\boldsymbol{\theta}}$.

b) Perform sum-product message passing in the factor graph of the SSM with fixed $\hat{\boldsymbol{\theta}}$ in order to compute $\vec{\mu}_{X_k}$ and $\overleftarrow{\mu}_{X_k}$ for $k = 0, \ldots, K$.

c) Compute the EM messages

$$\overleftarrow{\mu}_{\Theta_k}(\boldsymbol{\theta}_k) \propto e^{\eta_k(\boldsymbol{\theta}_k)} \tag{4.9}$$

according to a rule detailed below in (4.10).

d) Use the messages $\overleftarrow{\mu}_{\Theta_k}$ to compute new estimates $\hat{\boldsymbol{\theta}}$ by max-product message passing in the upper part (the equality nodes) of the graph in Figure 4.1.

The EM message (4.9) is computed locally in the factor graph. On the level of detail given in Figure 4.1 the rule for computing the exponent in (4.9) is

$$\eta_k(\boldsymbol{\theta}_k) = \mathsf{E}_{p_{\text{local}}}[\log f_k(\boldsymbol{X}_{k-1}, \boldsymbol{X}_k, \boldsymbol{\theta}_k)]\,, \tag{4.10}$$

where the expectation is with respect to the local PDF

$$p_{\text{local}}\big(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k \big| \hat{\boldsymbol{\theta}}_k\big) \propto f_k(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, \hat{\boldsymbol{\theta}}_k)\, \vec{\mu}_{X_{k-1}}(\boldsymbol{x}_{k-1})\, \overleftarrow{\mu}_{X_k}(\boldsymbol{x}_k)\,. \tag{4.11}$$

More generally, if the factor f_k in Figure 4.1 has an internal structure as shown in Figure 4.3, then the exponent of the EM message (4.9) can be computed as

$$\eta(\boldsymbol{\theta}_k) = \mathsf{E}_{p_{\text{local}}}[\log f(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}_k)]\,, \tag{4.12}$$

where the expectation is with respect to the local PDF

$$p_{\text{local}}\big(\boldsymbol{x}, \boldsymbol{y}\big|\hat{\boldsymbol{\theta}}\big) \propto f(\boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{\theta}})\, \vec{\mu}_X(\boldsymbol{x})\, \overleftarrow{\mu}_Y(\boldsymbol{y})\,. \tag{4.13}$$

As detailed in [26, Section III.F] a problem arises if f is a Dirac delta constraint. In this case we must group the factor f with one of its (non-Dirac) neighbors, e.g. in Figure 4.3 this could be f' or f'' or any internal factors to those as long as they are not Dirac deltas.

In contrast to the CM approach of Section 4.2.1, the EM algorithm has a chance of finding the ML estimate of $\boldsymbol{\theta}$ as long as the initial value for $\hat{\boldsymbol{\theta}}$ is not too far from the true value.

4.2.3 Local Taylor Approximations

Here we describe, how non-Gaussian factors can be approximated by Gaussian factors, and nonlinear constraints by linear constraints. The resulting graph is suitable for Gaussian message passing. Although our view is different, there is a strong similarity to the factor graph approach to gradient ascent [20]. In contrast to the previous two sections, we do not address the problem of breaking cycles in the graph here.

General Node – The Laplace Approximation

Let the factor graph contain a factor $f(\boldsymbol{x})\colon \mathbb{R}^n \to \mathbb{R}_{>0}$ whose logarithm is assumed to be twice differentiable. We consider the substitution of this factor by a Gaussian factor by using a truncated Taylor series approximation, more precisely a Laplace approximation [3], of $\varphi(\boldsymbol{x}) \triangleq -\ln f(\boldsymbol{x})$ around an operating point $\hat{\boldsymbol{x}}$. This operating point can be chosen to be the means of the marginals or the means of messages of the edges connected to this factor.

In the following, we decorate the approximated factors with a tilde. The Taylor series expansion of $\varphi(\boldsymbol{x})$ around $\hat{\boldsymbol{x}}$ up until the second-order term is

$$\tilde{\varphi}(\boldsymbol{x}) \triangleq \varphi(\hat{\boldsymbol{x}}) + (\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathsf{T}}\, \nabla\varphi(\hat{\boldsymbol{x}}) + \frac{1}{2}(\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathsf{T}} \nabla^2\varphi(\hat{\boldsymbol{x}})\, (\boldsymbol{x}-\hat{\boldsymbol{x}})\,, \tag{4.14}$$

where $\nabla\varphi(\hat{\boldsymbol{x}}) = \nabla\varphi(\boldsymbol{x})\big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}}$ is the gradient vector evaluated at $\hat{\boldsymbol{x}}$ and $\nabla^2\varphi(\hat{\boldsymbol{x}}) = \nabla^2\varphi(\boldsymbol{x})\big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}}$ is the Hessian matrix evaluated at $\hat{\boldsymbol{x}}$. We define

the corresponding approximate factor as

$$\tilde{f}(\boldsymbol{x}) \triangleq e^{-\tilde{\varphi}(\boldsymbol{x})} \tag{4.15}$$

$$= f(\hat{\boldsymbol{x}})\, e^{-(\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathsf{T}} \nabla\varphi(\hat{\boldsymbol{x}}) - (\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathsf{T}} \nabla^2\varphi(\hat{\boldsymbol{x}})\,(\boldsymbol{x}-\hat{\boldsymbol{x}})/2} \tag{4.16}$$

$$\propto e^{\boldsymbol{x}^{\mathsf{T}}\mathbf{W}\boldsymbol{m} - \boldsymbol{x}^{\mathsf{T}}\mathbf{W}\boldsymbol{x}/2}\,, \tag{4.17}$$

where $\mathbf{W} \triangleq \nabla^2\varphi(\hat{\boldsymbol{x}})$, and $\mathbf{W}\boldsymbol{m} \triangleq \mathbf{W}\hat{\boldsymbol{x}} - \nabla\varphi(\hat{\boldsymbol{x}})$.

The approximation described above cannot be applied to Dirac delta constraints because the factor is assumed to be positive and finite. We propose a different kind of approximation for such constraints in the following.

Constraint Node – Linearization

Consider a factor graph in which the factor $f(\boldsymbol{x}) = \delta\big(h(\boldsymbol{x})\big)$ represents a (potentially multivariate) Dirac delta constraint, where $h\colon \mathbb{R}^{n_X} \to \mathbb{R}^{n_H}$ is a differentiable function. Let the variable vector $\boldsymbol{X}$ be split as $\boldsymbol{X}^{\mathsf{T}} \triangleq \big[\boldsymbol{X}_1^{\mathsf{T}}, \ldots, \boldsymbol{X}_n^{\mathsf{T}}\big]$ into sub-vectors each of which is represented by an edge in the factor graph, cf. Figure 4.4a.

In contrast to the previous section, we propose to linearize $h(\boldsymbol{x})$ by means of a Taylor series around an operating point $\hat{\boldsymbol{x}}$. This time, however, we truncate the series after the linear term as

$$\tilde{h}(\boldsymbol{x}) \triangleq h(\hat{\boldsymbol{x}}) + \mathbf{H}(\boldsymbol{x} - \hat{\boldsymbol{x}})\,, \tag{4.18}$$

where $\mathbf{H}$ is the Jacobian matrix evaluated at $\hat{\boldsymbol{x}}$ with elements

$$H_{i,j} \triangleq \left.\frac{\partial h_i(\boldsymbol{x})}{\partial x_j}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}}, \tag{4.19}$$

for $i = 1, \ldots, n_X$, $j = 1, \ldots, n_H$.

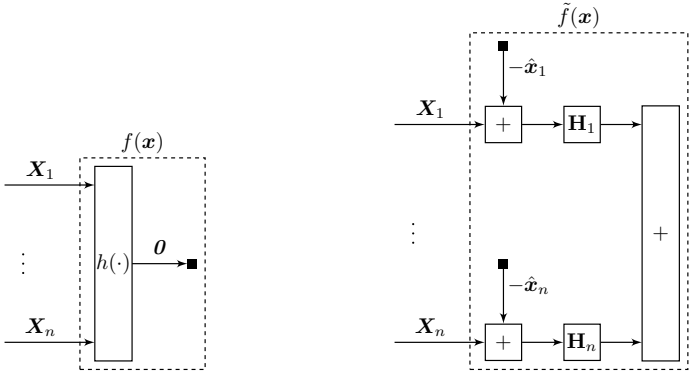
If $\hat{\boldsymbol{x}}$ is chosen such that it fulfils the constraint, then $h(\hat{\boldsymbol{x}}) = \boldsymbol{0}$ and

$$\tilde{h}(\boldsymbol{x}) = \mathbf{H}(\boldsymbol{x} - \hat{\boldsymbol{x}})\,, \tag{4.20}$$

$$\tilde{f}(\boldsymbol{x}) \triangleq \delta\big(\tilde{h}(\boldsymbol{x})\big) = \delta(\mathbf{H}(\boldsymbol{x} - \hat{\boldsymbol{x}}))\,, \tag{4.21}$$

where $\tilde{f}$ is the proposed approximation of f.

In a message passing algorithms it makes sense to choose the operating point $\hat{\boldsymbol{x}}$ in terms of locally available quantities, e.g., messages on the



(a) Original constraint (b) Approximate constraint

Figure 4.4: Local view of linearization a contraint $f(\boldsymbol{x}) \triangleq \delta\big(h(\boldsymbol{x})\big)$.

edges $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ from the last iteration. One possible choice for $\hat{\boldsymbol{x}}$ that fulfills the constraint h is

$$\hat{\boldsymbol{x}} = \big[\boldsymbol{m}_{X_1}^{\mathsf{T}}, \ldots, \boldsymbol{m}_{X_n}^{\mathsf{T}}\big]^{\mathsf{T}}, \tag{4.22}$$

where $\boldsymbol{m}_{X_1}, \ldots, \boldsymbol{m}_{X_n}$ are the means of the marginals. Alternative choices that fulfill the constraint h are

$$\hat{\boldsymbol{x}} = \Big[\overrightarrow{\boldsymbol{m}}_{X_1}^{\mathsf{T}}, \ldots, \overrightarrow{\boldsymbol{m}}_{X_{j-1}}^{\mathsf{T}}, \overleftarrow{\boldsymbol{m}}_{X_j}^{\mathsf{T}}, \overrightarrow{\boldsymbol{m}}_{X_{j+1}}^{\mathsf{T}}, \ldots, \overrightarrow{\boldsymbol{m}}_{X_n}^{\mathsf{T}}\Big]^{\mathsf{T}} \tag{4.23}$$

for any $j = 1, \ldots, n$. Note that the means $\overrightarrow{\boldsymbol{m}}_{X_1}, \ldots, \overrightarrow{\boldsymbol{m}}_{X_n}$ of the incoming messages in general do not fulfil the constraint h.

The approximate constraint in (4.21) can be interpreted in a local factor graph point-of-view as follows. We can rewrite (4.21) as

$$\tilde{f}(\boldsymbol{x}) = \delta\Big(\sum_{i=1}^{n} \mathbf{H}_i(\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i)\Big), \tag{4.24}$$

where $\mathbf{H}_i$ is the sub-matrix of $\mathbf{H}$ corresponding to $\boldsymbol{X}_i$. The corresponding factor graph is shown in Figure 4.4b.

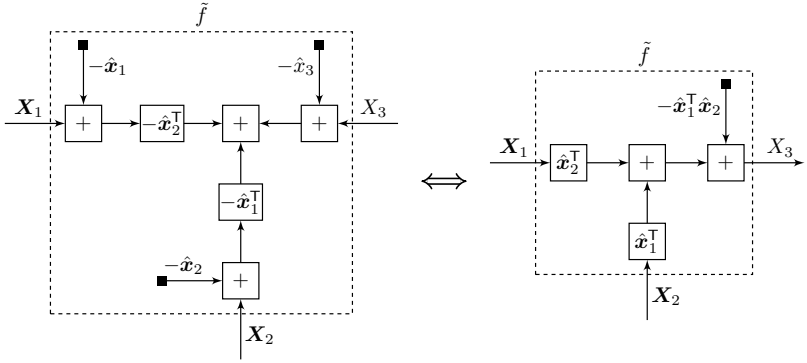


Figure 4.5: Example 4.1: Linearization of an inner product constraint $\delta(x_3 - \boldsymbol{x}_1^\mathsf{T}\boldsymbol{x}_2)$.

Example 4.1: Linearized Inner Product
Consider the constraint $f(\boldsymbol{x}) = \delta(x_3 - \boldsymbol{x}_1^\mathsf{T}\boldsymbol{x}_2)$, where $\boldsymbol{x}^\mathsf{T} \triangleq \big[\boldsymbol{x}_1^\mathsf{T}, \boldsymbol{x}_2^\mathsf{T}, x_3\big]$. In this case $h(\boldsymbol{x}_1, \boldsymbol{x}_2, x_3) = x_3 - \boldsymbol{x}_1\boldsymbol{x}_2^\mathsf{T}$ and $\mathbf{H} = \big[-\hat{\boldsymbol{x}}_2^\mathsf{T}, -\hat{\boldsymbol{x}}_1^\mathsf{T}, 1\big]$. The factor graph corresponding to (4.24) is shown in Figure 4.5 on the left. The graph on the right can be shown straightforwardly to be an equivalent version by writing out (4.21). ◊

Example 4.2: Rotation Matrix Multiplication
Consider the constraint $f(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y}) = \delta\big(h(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y})\big)$, where $h(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y}) \triangleq \boldsymbol{y} - \operatorname{rotm}(\boldsymbol{\theta})\,\boldsymbol{x}$ with $\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y} \in \mathbb{R}^2$. Assuming that the operating point $\big(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{y}}\big)$ has been chosen such that $h(\hat{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{y}}) = 0$ we write (4.20) as

$$\tilde{h}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y}) = \underbrace{\begin{bmatrix} -\operatorname{rotm}(\hat{\boldsymbol{\theta}}) & -\operatorname{rotm}(\hat{\boldsymbol{x}}) & \mathbf{I}_2 \end{bmatrix}}_{\triangleq \mathbf{H}} \begin{bmatrix} \boldsymbol{x} - \hat{\boldsymbol{x}} \\ \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \\ \boldsymbol{y} - \hat{\boldsymbol{y}} \end{bmatrix}. \tag{4.25}$$

Figure 4.6 shows the corresponding factor graph on the left. The equivalent factor graph on the right is derived from (4.25) as

$$\begin{aligned} \tilde{h}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y}) &= \boldsymbol{y} - \hat{\boldsymbol{y}} - \operatorname{rotm}(\hat{\boldsymbol{\theta}})(\boldsymbol{x} - \hat{\boldsymbol{x}}) - \operatorname{rotm}(\hat{\boldsymbol{x}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) && (4.26) \\ &= \boldsymbol{y} - \operatorname{rotm}(\hat{\boldsymbol{\theta}})\,\boldsymbol{x} - \operatorname{rotm}(\hat{\boldsymbol{x}})\,\boldsymbol{\theta} + \operatorname{rotm}(\hat{\boldsymbol{x}})\,\hat{\boldsymbol{\theta}} && (4.27) \\ &= \boldsymbol{y} - \operatorname{rotm}(\hat{\boldsymbol{\theta}})\,\boldsymbol{x} - \operatorname{rotm}(\hat{\boldsymbol{x}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\,, && (4.28) \end{aligned}$$

where we have used $\hat{\boldsymbol{y}} = \operatorname{rotm}(\hat{\boldsymbol{\theta}})\,\hat{\boldsymbol{x}}$. ◊

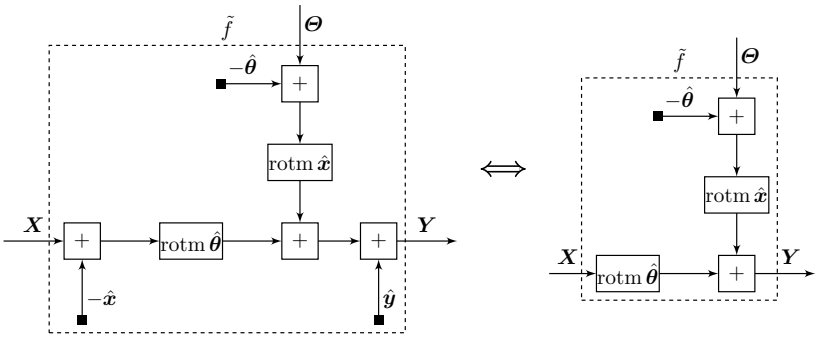


Figure 4.6: Example 4.2: Linearization of a rotation matrix product.

We close this section by conjecturing that the Gaussian approximation in [46] is in fact a linearization of a multiplication.

4.3 Estimation of Distinct System Poles in Jordan Form

4.3.1 The Setup

Consider the LTI system

$$\begin{aligned} \boldsymbol{X}_k &= \mathbf{A}(\boldsymbol{\theta})\, \boldsymbol{X}_{k-1} + \boldsymbol{U}_k\,, \\ \boldsymbol{Y}_k &= \mathbf{C}\boldsymbol{X}_k + \boldsymbol{Z}_k\,, \end{aligned} \tag{4.29}$$

where $\boldsymbol{X}_k$ is the state, $\boldsymbol{U}_k \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \mathbf{V}_U)$ is the state noise, $\boldsymbol{Y}_k$ is the observable output, and $\boldsymbol{Z}_k \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \mathbf{V}_Z)$ is the observation noise. The state-transition matrix $\mathbf{A}(\boldsymbol{\theta})$ depends on a parameter vector $\boldsymbol{\theta}$, which we want to estimate given observations $\boldsymbol{Y}_k = \tilde{\boldsymbol{y}}_k$ for $k = 1, \ldots, K$. The factor graph representation of (4.29) is shown in Figure 4.7. Note that this is a detailed view of Figure 4.1.

Here, we restrict ourselves to the case where $\mathbf{A}$ is in Jordan canonical form with distinct eigenvalues. Furthermore, we assume that the number $n \in \mathbb{N}$ of real eigenvalues and the number $m \in \mathbb{N}$ of complex eigenvalue pairs of $\mathbf{A} \in \mathbb{R}^{(n+2m)\times(n+2m)}$ is known in advance.

The real Jordan canonical form (3.2) inspires us to define the operator

$\mathrm{rotm}_{n,m}(\cdot)$ as the mapping

$$\mathrm{rotm}_{n,m}\colon \mathbb{R}^{n+2m} \to \mathbb{R}^{(n+2m)\times(n+2m)} \tag{4.30}$$

$$:\boldsymbol{\theta} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_m \end{bmatrix} \to \begin{bmatrix} a_1 & \cdots & 0 & \boldsymbol{0}^\mathsf{T} & \cdots & \boldsymbol{0}^\mathsf{T} \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & a_n & \boldsymbol{0}^\mathsf{T} & \cdots & \boldsymbol{0}^\mathsf{T} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \mathrm{rotm}(\boldsymbol{a}_1) & \cdots & \mathbf{0} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \mathbf{0} & \cdots & \mathrm{rotm}(\boldsymbol{a}_m) \end{bmatrix},$$

where $a_i = \theta_i$ for $i = 1, \ldots, n$ and $\boldsymbol{a}_j \triangleq \left[\theta_{2j+n-1}, \theta_{2j+n}\right]^\mathsf{T}$. The matrix $\mathrm{rotm}_{n,m}(\boldsymbol{\theta})$ has distinct real eigenvalues $a_1, \ldots, a_n$ and distinct complex conjugate eigenvalue pairs $[\boldsymbol{a}_1]_1 \pm i[\boldsymbol{a}_1]_2, \ldots, [\boldsymbol{a}_m]_1 \pm [\boldsymbol{a}_m]_2$.

We now define the matrix $\mathbf{A}(\boldsymbol{\theta})$ in our system (4.29) as

$$\mathbf{A}(\boldsymbol{\theta}) \triangleq \mathrm{rotm}_{n,m}(\boldsymbol{\theta})\,. \tag{4.31}$$

In the following we formulate iterative message-passing algorithms based on the principles exposed in Section 4.2 to estimate $\boldsymbol{\theta}$ given observations $\boldsymbol{Y}_k = \tilde{\boldsymbol{y}}_k$ for $k = 1, \ldots, K$.

4.3.2 Gaussian Messages for Rotation Matrix Product

The application of the three principles in Section 4.2 yields three different Gaussian messages $\overleftarrow{\mu}_{\Theta_k}$ in the factor graph of Figure 4.7, which are summarized in the following theorem.

Theorem 4.1: Estimation of A in Real Jordan Form
In an LTI system (4.29)*, consider the estimation of* $\mathbf{A}(\boldsymbol{\theta}) \triangleq \mathrm{rotm}_{n,m}(\boldsymbol{\theta})$ *given some* $n, m \in \mathbb{N}$ *and observations* $\boldsymbol{Y}_k = \tilde{\boldsymbol{y}}_k$ *for* $k = 1, \ldots, K$ *as described in Section 4.3.1. The three principles in Section 4.2 applied to this problem result in three different Gaussian messages* $\overleftarrow{\mu}_{\Theta_k}$ *in the factor graph of Figure 4.7.*

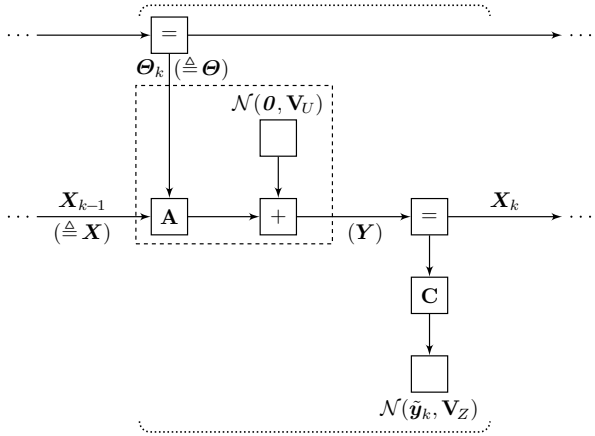


Figure 4.7: A linear time-invariant (LTI) system with state-transition matrix $\mathbf{A}$ parameterized by $\boldsymbol{\theta}$. The edge labels in brackets are used in Theorem 4.1 and Appendix B.2.

a) Applying CM (cf. Section 4.2.1) yields

$$\overleftarrow{\mathbf{W}}_{\Theta} = \mathbf{R}^{\mathsf{T}}\,\mathbf{W}_U\,\mathbf{R}\,, \tag{4.32}$$

$$\overleftarrow{\mathbf{W}}_{\Theta}\overleftarrow{\boldsymbol{m}}_{\Theta} = \mathbf{R}^{\mathsf{T}}\,\mathbf{W}_U\,\boldsymbol{m}_Y\,, \tag{4.33}$$

$$\overleftarrow{\boldsymbol{m}}_{\Theta} = \mathbf{R}^{-1}\,\boldsymbol{m}_Y\,, \tag{4.34}$$

where $\mathbf{R} \triangleq \mathrm{rotm}_{n,m}(\boldsymbol{m}_X)$.

b) Applying EM (cf. Section 4.2.2) yields

$$\begin{aligned}\overleftarrow{\mathbf{W}}_{\Theta} = {} & \mathbf{R}^{\mathsf{T}}\,\mathbf{W}_U\,\mathbf{R} + \mathbf{W}_U \odot \left(\bar{\mathbf{I}}\,\mathbf{V}_X\,\bar{\mathbf{I}}\right) \\ & + 2\left(\underline{\mathbf{I}}_1^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_3\right) \odot \left(\underline{\mathbf{I}}_4\,\mathbf{V}_X\,\underline{\mathbf{I}}_2^{\mathsf{T}}\right) \\ & + \left(\underline{\mathbf{I}}_1^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_1\right) \odot \left(\underline{\mathbf{I}}_2\,\mathbf{V}_X\,\underline{\mathbf{I}}_2^{\mathsf{T}}\right) \\ & + \left(\underline{\mathbf{I}}_3^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_3\right) \odot \left(\underline{\mathbf{I}}_4\,\mathbf{V}_X\,\underline{\mathbf{I}}_4^{\mathsf{T}}\right),\end{aligned} \tag{4.35}$$

$$\begin{aligned}\overleftarrow{\mathbf{W}}_{\Theta}\overleftarrow{\boldsymbol{m}}_{\Theta} = {} & \mathbf{R}^{\mathsf{T}}\,\mathbf{W}_U\,\boldsymbol{m}_Y + \mathrm{diag}\big(\mathbf{W}_U\,\mathbf{V}_{XY^{\intercal}}\,\bar{\mathbf{I}} \\ & + \underline{\mathbf{I}}_2\,\mathbf{W}_U\,\mathbf{V}_{XY^{\intercal}}\,\underline{\mathbf{I}}_1 + \underline{\mathbf{I}}_4\,\mathbf{W}_U\,\mathbf{V}_{XY^{\intercal}}\,\underline{\mathbf{I}}_3\big)\,,\end{aligned} \tag{4.36}$$

where $\mathbf{R} \triangleq \mathrm{rotm}_{n,m}(\boldsymbol{m}_X)$, $\bar{\mathbf{I}} \triangleq \left[\begin{smallmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{2m} \end{smallmatrix}\right]$,

$$\underline{\mathbf{I}}_1 \triangleq \begin{bmatrix} \mathbf{0}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \otimes \left[\begin{smallmatrix} 1 & 1 \\ 0 & 0 \end{smallmatrix}\right] \end{bmatrix}, \quad \underline{\mathbf{I}}_2 \triangleq \begin{bmatrix} \mathbf{0}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \otimes \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right] \end{bmatrix}, \tag{4.37}$$

$$\underline{\mathbf{I}}_3 \triangleq \begin{bmatrix} \mathbf{0}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \otimes \left[\begin{smallmatrix} 0 & 0 \\ 1 & 1 \end{smallmatrix}\right] \end{bmatrix}, \quad \underline{\mathbf{I}}_4 \triangleq \begin{bmatrix} \mathbf{0}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \otimes \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right] \end{bmatrix}. \tag{4.38}$$

c) Linearizing the constraint (cf. Section 4.2.3 and Example 4.2) around an operating point $\hat{\boldsymbol{x}} = \vec{\boldsymbol{m}}_X$, $\hat{\boldsymbol{\theta}}$ *arbitrary,* $\hat{\boldsymbol{y}} = \mathbf{A}(\hat{\boldsymbol{\theta}})\,\vec{\boldsymbol{m}}_X$ *yields*

$$\overleftarrow{\mathbf{V}}_\Theta = \mathbf{R}^{-1}\Big(\mathbf{A}(\hat{\boldsymbol{\theta}})\,\vec{\mathbf{V}}_X\,\mathbf{A}(\hat{\boldsymbol{\theta}})^\mathsf{T} + \overleftarrow{\mathbf{V}}_Y + \mathbf{V}_U\Big)\mathbf{R}^{-\mathsf{T}}\,, \tag{4.39}$$

$$\overleftarrow{\mathbf{W}}_\Theta\,\overleftarrow{\boldsymbol{m}}_\Theta = \mathbf{R}^\mathsf{T}\Big(\mathbf{A}(\hat{\boldsymbol{\theta}})\,\vec{\mathbf{V}}_X\,\mathbf{A}(\hat{\boldsymbol{\theta}})^\mathsf{T} + \overleftarrow{\mathbf{V}}_Y + \mathbf{V}_U\Big)^{-1}\overleftarrow{\boldsymbol{m}}_Y\,, \tag{4.40}$$

$$\overleftarrow{\boldsymbol{m}}_\Theta = \mathbf{R}^{-1}\,\overleftarrow{\boldsymbol{m}}_Y\,, \tag{4.41}$$

where $\mathbf{R} \triangleq \mathrm{rotm}_{n,m}(\vec{\boldsymbol{m}}_X)$.

The proof for Theorem 4.1 is given in Appendix B.2. It is worthwhile to highlight the similarity of the three different messages in Parts (a)–(c). In particular, (4.33) and (4.36) are equivalent if $\mathbf{V}_{XY^\mathsf{T}} = \mathbf{0}$, and (4.34) uses the mean $\boldsymbol{m}_Y$ of the marginal where (4.41) uses the mean $\overleftarrow{\boldsymbol{m}}_Y$ of the message.

4.4 Application to Quasi-Periodic Signals

Many signals occurring in nature are almost periodic. Examples can be found in biological signals [105], financial time series [31], and natural phenomena. Such signals often bear a certain amount of self similarity, but the period as well as the signals shape tends to drift over time.

The Gaussian messages presented in Theorem 4.1 can be used to estimate the slowly changing fundamental frequency of quasi-periodic signals. The actual shape of the signal is found by Gaussian message passing in the factor graph of a linear system as in Figure 3.1 with plugged-in fundamental frequency.

More precisely, we consider the following SSM:

$$\begin{aligned} \boldsymbol{X}_k &= \mathbf{A}(\Omega_k)\, \boldsymbol{X}_{k-1} + \boldsymbol{U}_k \\ Y_k &= \boldsymbol{c}\, \boldsymbol{X}_k + Z_k \,, \end{aligned} \tag{4.42}$$

where

$$\mathbf{A}(\Omega_k) \triangleq \begin{bmatrix} \mathrm{rotm}(\Omega_k) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathrm{rotm}(M\Omega_k) \end{bmatrix} \in \mathbb{R}^{2M \times 2M} \,, \tag{4.43}$$

$$\boldsymbol{c} \triangleq \big[1, 0, \ldots, 1, 0\big] \,, \tag{4.44}$$

$\boldsymbol{U}_k \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \sigma_U^2 \mathbf{I}_{2M})$, and $Z_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$.

In the case $\Omega_k = \Omega_1$ for all $k = 1, \ldots, K$ and $\sigma_U^2 = 0$, the SSM in (4.42) models a noisy periodic signal with fundamental frequency Ω_1 (and without a DC component). If we keep the restriction $\Omega_k = \Omega_1$ for all $k = 1, \ldots, K$ but allow $\sigma_U^2 > 0$, then we have a slowly changing periodic signal with constant fundamental frequency Ω_1. This case was treated in Section 2.7.2.

For the quasi-periodic signal model that we envisage, Ω_k should be allowed to change slowly over time as

$$\Omega_k \approx \Omega_{k-1} \,. \tag{4.45}$$

We make this informal constraint precise by defining the factor graph of a quasi-periodic signal in Figure 4.8, in which the soft constraint (4.45) is modeled by means of a forgetting factor γ.

We realize that the SSM (4.42) used for a quasi-periodic signal can be formulated in real Jordan canonical form (4.29) and (4.31) if we let $n = 0$, $m = M$, $\mathbf{V}_U \triangleq \sigma_U^2 \mathbf{I}_{2M}$, $\mathbf{C} \triangleq \boldsymbol{c}$, and

$$\boldsymbol{\Theta}(\Omega_k) \triangleq \big[\cos(\Omega_k), \sin(\Omega_k), \ldots, \cos(M\Omega_k), \sin(M\Omega_k)\big]^{\mathsf{T}} . \tag{4.46}$$

Now we can consider estimation of Ω_k via message passing using any of the approximation methods of Theorem 4.1. All the considered approximations yield a message that is Gaussian with respect to $\boldsymbol{\Theta}$. The

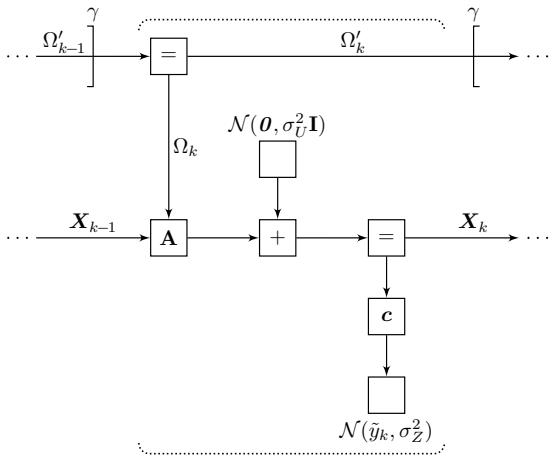


Figure 4.8: A linear slowly time-varying system for quasi-periodic signals.

corresponding message with respect to Ω_k is

$$\ln \overleftarrow{\mu}_{\Omega_k}(\omega) = -\boldsymbol{\theta}(\omega)^{\mathsf{T}} \overleftarrow{\mathbf{W}}_{\Theta_k} \boldsymbol{\theta}(\omega)/2 + \boldsymbol{\theta}(\omega)^{\mathsf{T}} \overleftarrow{\mathbf{W}}_{\Theta_k} \overleftarrow{\boldsymbol{m}}_{\Theta_k} + \text{const} \tag{4.47}$$

$$= -\operatorname{tr}\Big(\boldsymbol{\theta}(\omega)\, \boldsymbol{\theta}(\omega)^{\mathsf{T}} \overleftarrow{\mathbf{W}}_{\Theta_k}\Big)/2 + \boldsymbol{\theta}(\omega)^{\mathsf{T}} \overleftarrow{\mathbf{W}}_{\Theta_k} \overleftarrow{\boldsymbol{m}}_{\Theta_k} + \text{const}\,. \tag{4.48}$$

Clearly, $\ln \overleftarrow{\mu}_{\Omega_k}$ is a periodic function of the form

$$\ln \overleftarrow{\mu}_{\Omega_k}(\omega) = \operatorname{Re}\Big(\overleftarrow{\boldsymbol{\xi}}_{\Omega_k}^{\mathsf{T}}\, \boldsymbol{\varrho}(\omega)\Big) + \text{const}\,, \tag{4.49}$$

with $\boldsymbol{\varrho}(\omega) \triangleq \big[e^{i\omega}, e^{i2\omega}, \ldots, e^{iM\omega}\big]^{\mathsf{T}}$, and where the Fourier series coefficients $\overleftarrow{\boldsymbol{\xi}}_{\Omega_k} \in \mathbb{C}^M$ depend on $\overleftarrow{\mathbf{W}}_{\Theta_k}$ and $\overleftarrow{\mathbf{W}}_{\Theta_k} \overleftarrow{\boldsymbol{m}}_{\Theta_k}$. The mapping

$$\Gamma\colon \big(\overleftarrow{\mathbf{W}}_{\Theta_k}, \overleftarrow{\mathbf{W}}_{\Theta_k} \overleftarrow{\boldsymbol{m}}_{\Theta_k}\big) \to \overleftarrow{\boldsymbol{\xi}}_{\Omega_k} \tag{4.50}$$

can in general be formulated for all the three approximation methods.

While this mapping turns out to be quite intricate for EM and linearization, it can easily be formulated for CM. In this case, by using (4.32)

and (4.33) of Theorem 4.1a the mapping Γ evaluates to

$$\mathrm{Re}\Big[\overleftarrow{\boldsymbol{\xi}}_{\Omega_k}\Big]_m = [\boldsymbol{m}_{X_k}]_{2m-1}\big[\boldsymbol{m}_{X_{k-1}}\big]_{2m-1} + [\boldsymbol{m}_{X_k}]_{2m}\big[\boldsymbol{m}_{X_{k-1}}\big]_{2m}\,, \quad (4.51)$$

$$\mathrm{Im}\Big[\overleftarrow{\boldsymbol{\xi}}_{\Omega_k}\Big]_m = [\boldsymbol{m}_{X_k}]_{2m}\big[\boldsymbol{m}_{X_{k-1}}\big]_{2m-1} - [\boldsymbol{m}_{X_k}]_{2m-1}\big[\boldsymbol{m}_{X_{k-1}}\big]_{2m}\,. \quad (4.52)$$

Equations (4.51) and (4.52) are proved in Appendix B.4.

Once the mapping Γ has been found for any of the three approximation methods, the messages $\overleftarrow{\mu}_{\Omega_k}$ for all k can be formulated as in (4.49), and each such message is parameterized by its Fourier coefficient vector $\overleftarrow{\boldsymbol{\xi}}_{\Omega_k}$.

In the course of performing max-product message passing in the upper part of the factor graph in Figure 4.8, a forgetting factor γ is used. The forward message on edge Ω'_k in Figure 4.8 thus is formulated as

$$\overrightarrow{\mu}_{\Omega'_k}(\omega) = \Big(\overrightarrow{\mu}_{\Omega'_{k-1}}(\omega)\Big)^{\gamma}\,\overleftarrow{\mu}_{\Omega_k}(\omega)\,, \quad (4.53)$$

such that

$$\overrightarrow{\boldsymbol{\xi}}_{\Omega'_k} = \gamma\,\overrightarrow{\boldsymbol{\xi}}_{\Omega'_{k-1}} + \overleftarrow{\boldsymbol{\xi}}_{\Omega_k}\,. \quad (4.54)$$

Similarly, for the backward pass we have

$$\overleftarrow{\boldsymbol{\xi}}_{\Omega'_{k-1}} = \gamma\,\overleftarrow{\boldsymbol{\xi}}_{\Omega'_k} + \overleftarrow{\boldsymbol{\xi}}_{\Omega_k}\,. \quad (4.55)$$

We now have all the ingredients to devise estimation algorithms for Ω_k from a given signal $Y_k = \tilde{y}_k$ for $k = 1, \ldots, K$.

Offline Estimation of a Quasi-Periodic Signal:

a) Let $\hat{\omega}_k$ be an initial estimate of Ω_k for all $k = 1, \ldots, K$.

b) Do forward and backward Gaussian message passing in the factor graph of Figure 4.8 with plugged-in estimates $\Omega_k = \hat{\omega}_k$ for $k = 1, \ldots, K$.

c) From $\overrightarrow{\mu}_{X_k}$ and $\overleftarrow{\mu}_{X_k}$ for all $k = 0, \ldots, K$, compute $\overleftarrow{\mu}_{\Omega_k}$ for all $k = 1, \ldots, K$ using the mapping Γ.

d) Do forward and backward message passing in the upper part of the factor graph of Figure 4.8 using (4.54) and (4.55). Compute new estimates $\hat{\omega}_k = \mathrm{argmax}_\omega\, \mu_{\Omega_k}(\omega)$ for $k = 1, \ldots, K$.

e) Go to Step (b) or stop the algorithm.

In this thesis, we do not elaborate on how to obtain an initial estimate for Step (a). In Step (c), either of the three methods of Theorem 4.1 can be employed. The maximization of the pseudo-marginal μ_{Ω_k} in Step (d) is, in general, not analytically solvable. We propose to use an iterative Newton method, started at the estimate $\hat{\omega}_k$ known from the previous iteration.

In an online scenario, the given signal $\tilde{y}_1, \tilde{y}_2, \ldots$ arrives sequentially in a stream and we would like to produce the estimates $\hat{\omega}_1, \hat{\omega}_2, \ldots$ likewise. In the following algorithm, let $K \in \mathbb{N}$ be the time index of the current data item and let $D \in \mathbb{N}$ be a delay parameter. The algorithm is initialized by assuming an initial estimate $\hat{\omega}_k$ for $k = 1, \ldots, D$. As an additional initialization step, we have to fetch the first D data items $\tilde{y}_1, \ldots, \tilde{y}_D$ and compute $\overrightarrow{\mu}_{X_k}$ for $k = 1, \ldots, D$ in the factor graph of Figure 4.8 with plugged in estimates $\hat{\omega}_k$. We start by setting $K = D$.

Online Estimation of a Quasi-Periodic Signal:

a) Increment K and fetch the next data item $\tilde{y}_K$.

b) Compute $\overrightarrow{\mu}_{X_{K-D}}$ and $\overleftarrow{\mu}_{X_k}$ for $k = K, \ldots, K-D-1$ in the factor graph of Figure 4.8 with plugged-in estimates $\Omega_k = \hat{\omega}_{K-D-1}$ for $k = K-D, \ldots, K$.

c) From $\overrightarrow{\mu}_{X_{K-D-1}}$, $\overleftarrow{\mu}_{X_{K-D-1}}$, $\overrightarrow{\mu}_{X_{K-D}}$, and $\overleftarrow{\mu}_{X_{K-D}}$ compute $\overleftarrow{\mu}_{\Omega_{K-D}}$ using the mapping Γ.

d) Compute $\overrightarrow{\mu}_{\Omega'_{K-D}}$ in the upper part of the factor graph of Figure 4.8 using (4.54). Compute the new current estimate of the fundamental frequency as $\hat{\omega}_{K-D} = \operatorname{argmax}_\omega \overrightarrow{\mu}_{\Omega'_{K-D}}(\omega)$.

e) Go to Step (a).

Note that it is possible to refine the algorithm by devising an additional inner loop in which we use the estimate $\hat{\omega}_{K-D}$ obtained in Step (d) to redo the backward message passing in Step (b).

A variation of both algorithms described above arises if we consider state-space splitting as defined in Section 3.5. While this is in principle possible it may in general degrade the performance of the algorithms.

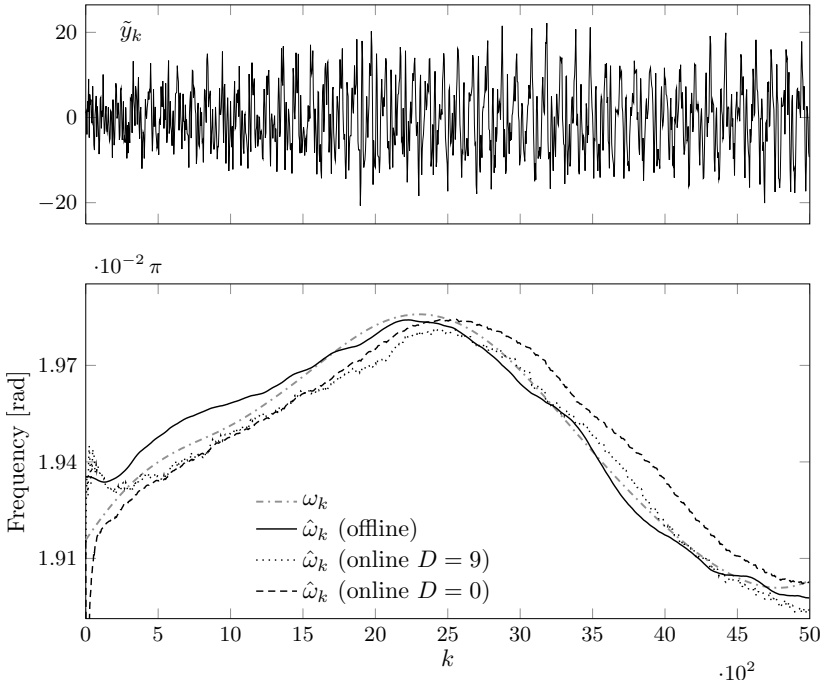


Figure 4.9: Pseudo-periodic signal estimation.

Parameter		Value
M	Number of harmonics	6
σ_U^2	State noise variance	0.006
σ_Z^2	Observation noise variance	8
Offline algorithm		
$\hat{\omega}_1, \ldots, \hat{\omega}_K$	Initial estimates	$0.02\,\pi$
γ	Forgetting factor	0.995
R	Number of iterations	10
Online algorithm		
$\hat{\omega}_1$	Initial estimate	$0.02\,\pi$
γ	Forgetting factor	0.996
D	Delay parameter	9 or 0

Table 4.1: Parameter settings used for Figure 4.9.

In the following we report an example for estimation of an artificially generated quasi-periodic signal. The signal (shown in the upper plot of Figure 4.9) was generated by first choosing values ω_k for $k = 1, \ldots, K$ and then drawing a sample from the inferred SSM. Note that the values ω_k need coincide with the ML estimates $\hat{\omega}_{k,\mathrm{ML}} = \operatorname{argmax}_\omega p(\tilde{\boldsymbol{y}}|\omega_k)$.

Both the online and the offline algorithm have been implemented using the CM approach to parameter estimation as in Theorem 4.1a. The corresponding mapping Γ is detailed in Equations (4.51) and (4.52).

Table 4.1 lists all the parameters used for signal generation and for the offline and the online algorithm. For the online algorithm the forgetting factor is chosen slightly higher because less information is aggregated in the marginal ξ_{Ω_k} than in the offline algorithm.

The original values ω_k and the estimates resulting from the algorithms are shown in the lower plot of Figure 4.9. Note that the obtained estimates from the offline algorithm are most probably not ML estimates, cf. Section 4.2.1. Both algorithms find reasonable estimates for Ω_k. In the online case, the algorithm takes some time until the estimates approach a meaningful value. Also, the version with $D = 0$ (zero delay) seems to lag slightly.

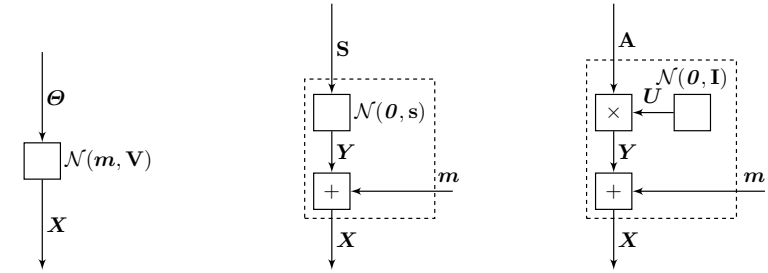
4.5 Variance Estimation

Variance estimation is a broad topic with many applications [53]. We do not intend to give a complete overview here. Instead, we highlight some of the results that can be obtained by applying the principles of Section 4.2.

4.5.1 The Setup

Consider a factor graph that contains a Gaussian factor $\mathcal{N}(\boldsymbol{m}, \mathbf{V})$. We are interested in estimating the parameters of this factor, namely the mean $\boldsymbol{m}$ and the covariance matrix $\boldsymbol{V}$ (or the precision matrix $\mathbf{W} = \mathbf{V}^{-1}$). The factor graph for this general local view is depicted in Figure 4.10a, in which $\boldsymbol{\theta}$ represents a parameter vector that defines $\boldsymbol{m}$ and $\mathbf{V}$ uniquely.

Figures 4.10b and 4.10c show two alternative possibilities for the choice of the parameter vector $\boldsymbol{\theta}$. In both alternatives the mean $\boldsymbol{m}$ is taken care



(a) General view. (b) Covariance matrix $\mathbf{s}$. (c) Multiplication by $\mathbf{A}$.

Figure 4.10: Local views for variance estimation.

of by means of an addition constraint, which is the natural choice for Gaussian message passing. We henceforth only treat the estimation of the variance or the covariance matrix, or some parameterization thereof.

In Figure 4.10b we directly take the covariance matrix as our parameter. In order to avoid a clash in the notation we use the symbol $\mathbf{s}$ to denote the covariance matrix when it takes on the role of a parameter. The edge in the factor graph is labeled with a capital $\mathbf{S}$ as usual.

Figure 4.10c in contrast shows a parametrization in which the parameter is a matrix $\mathbf{A}$, which is multiplied with a zero-mean unit-variance variable $\boldsymbol{U}$. The resulting covariance matrix is $\mathbf{A}\,\mathbf{A}^\mathsf{T}$. Note that for a given covariance matrix, a corresponding matrix $\mathbf{A}$ is not unique. More precisely, $\mathbf{A}$ can be substituted by $\mathbf{B}\,\mathbf{A}$ for any $\mathbf{B}$ that satisfies $\mathbf{B}\,\mathbf{B}^\mathsf{T} = \mathbf{I}$. One way to enforce uniqueness is to constrain $\mathbf{A}$ to be a lower (or upper) triangular matrix.

The parameterizations in Figures 4.10b and 4.10c are not the only options. A further parameterization is the precision matrix $\mathbf{W}$. This parameterization is specially attractive in situations in which many of the elements of $\boldsymbol{X}$ are independent because independence of X_i and X_j implies $[\mathbf{W}]_{i,j} = 0$. We refrain from considering this parameterization any further in this thesis.

In order to see the applicability of variance estimation in our setting, we recall the linear SSM

$$\begin{aligned} \boldsymbol{X}_k &= \mathbf{A}\,\boldsymbol{X}_{k-1} + \boldsymbol{U}_k \\ \boldsymbol{Y}_k &= \mathbf{C}\,\boldsymbol{X}_k + \boldsymbol{Z}_k\,, \end{aligned} \tag{4.56}$$

where $\boldsymbol{U}_k \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \mathbf{V}_U)$, and $\boldsymbol{Z}_k \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \mathbf{V}_Z)$. Here the assumption that $\mathbf{V}_U$ and $\mathbf{V}_Z$ are (at least approximately) constant for all time indices k is important. On the other hand, the system parameters $\mathbf{A}$ and $\mathbf{C}$ are allowed to change freely over time, i.e., we allow LTV systems. We refrain from reflecting this in our notation. The matrix $\mathbf{B}$ is absent in this system because we can absorb it into the state noise covariance matrix $\mathbf{V}_U$, which is to be estimated.

4.5.2 Direct Variance Estimation by Expectation Maximization

In this section, we apply EM as in Section 4.2.2 to estimate the covariance matrices. The parametrization of Figure 4.10c cannot directly be treated in the framework of [26] because the Gaussian node appears at the input of the multiplication. Reversing the multiplication does not solve this problem because we introduce a factor $\left|\det \mathbf{A}^{-1}\right|$ as detailed in Appendix C.1. It may be feasible nevertheless to formulate an EM algorithm for this parametrization. Here, we drop this option and choose directly the covariance matrix as our parameter, cf. Figure 4.10b.

We intend to use the EM algorithm to estimate the covariance matrix of the state noise $\boldsymbol{U}_k \in \mathbb{R}^n$. Figure 4.11 shows the factor graph for the local view envisaged. Note that an analogous graph can be devised for the estimation of the observation noise covariance matrix. Indeed, the EM procedure described here is not restricted to SSMs but can be used in any cycle-free factor graph to estimate covariance matrices.

In the factor graph of Figure 4.11, the EM message (4.9)–(4.11) is not a Gaussian message but a (scaled) inverse-Wishart PDF. (See Appendix B.3.1 for details on the inverse-Wishart distribution.) Specifically, we formulate the EM message on an edge representing a matrix $\mathbf{s}_k \in \mathbb{S}^n_{\succeq 0}$ as

$$\overleftarrow{\mu}_{S_k}(\mathbf{s}_k) \propto (\det \mathbf{s}_k)^{-(\overleftarrow{\nu}_{S_k}+n+1)/2}\, e^{-\operatorname{tr}\left(\overleftarrow{\mathbf{\Psi}}_{S_k}\, \mathbf{s}_k^{-1}\right)/2} \tag{4.57}$$

$$\propto \mathcal{W}^{-1}\left(\mathbf{s}_k \,\middle|\, \overleftarrow{\nu}_{S_k}, \overleftarrow{\mathbf{\Psi}}_{S_k}\right), \tag{4.58}$$

with

$$\overleftarrow{\nu}_{S_k} = -n\,, \tag{4.59}$$

$$\overleftarrow{\mathbf{\Psi}}_{S_k} = \mathbf{V}_{U_k} + \boldsymbol{m}_{U_k} \boldsymbol{m}_{U_k}^{\mathsf{T}}\,, \tag{4.60}$$

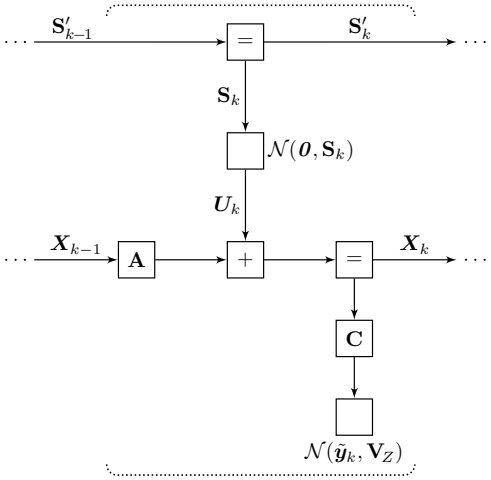


Figure 4.11: A linear system with input noise variance $\boldsymbol{S}$.

where $\mathbf{V}_{U_k}$ and $\boldsymbol{m}_{U_k}$ are the covariance matrix and the mean vector respectively of the marginal $\overrightarrow{\mu}_{U_k}(\boldsymbol{u}_k)\,\overleftarrow{\mu}_{U_k}(\boldsymbol{u}_k)$ with

$$\mathbf{V}_{U_k} = \left(\overrightarrow{\mathbf{V}}_{U_k}^{-1} + \overleftarrow{\mathbf{W}}_{U_k}\right)^{-1}, \tag{4.61}$$

$$\boldsymbol{m}_{U_k} = \mathbf{V}_{U_k}\overleftarrow{\mathbf{W}}_{U_k}\overleftarrow{\boldsymbol{m}}_{U_k}. \tag{4.62}$$

In $\overrightarrow{\mathbf{V}}_{U_k}$ we recognize the estimate $\hat{\mathbf{s}}$ from the previous iteration. The proof for Equations (4.57)–(4.60) is in Appendix B.3.2. Note that Equations (4.61) and (4.62) follow immediately from Gaussian message update rules and from $\overrightarrow{\boldsymbol{m}}_{U_k} = \boldsymbol{0}$.

In the maximization step, max-product message passing is applied to the upper part of the graph in Figure 4.11. Indeed, inverse-Wishart messages can be passed easily as

$$\overrightarrow{\mu}_{S'_k}(\mathbf{s}'_k) \propto (\det \mathbf{s}'_k)^{-(\overrightarrow{\nu}_{S'_k}+n+1)/2}\, e^{-\operatorname{tr}\left(\overrightarrow{\boldsymbol{\Psi}}_{S'_k}\,\mathbf{s}'^{-1}_k\right)/2}, \tag{4.63}$$

where

$$\overrightarrow{\nu}_{S'_k} = \overrightarrow{\nu}_{S'_{k-1}} + \overleftarrow{\nu}_{S_k} + n + 1 \tag{4.64}$$

$$\overrightarrow{\boldsymbol{\Psi}}_{S'_k} = \overrightarrow{\boldsymbol{\Psi}}_{S'_{k-1}} + \overleftarrow{\boldsymbol{\Psi}}_{S_k}. \tag{4.65}$$

From the above update rules it is evident, that the inclusion of forgetting factor is simple.

For the final maximization we consider the product of all the messages $\overleftarrow{\mu}_{S_k}$ for $k = 1, \ldots, K$, here denoted by μ_S:

$$\mu_S(\mathbf{s}) \propto (\det \mathbf{s})^{-(\nu_S+n+1)/2} \, e^{-\operatorname{tr}\left(\boldsymbol{\Psi}_S \, \mathbf{s}^{-1}\right)/2} \tag{4.66}$$

with

$$\nu_S = K - n - 1 \,, \tag{4.67}$$

$$\boldsymbol{\Psi}_S = \sum_{k=1}^{K} \overleftarrow{\boldsymbol{\Psi}}_{S_k} \,. \tag{4.68}$$

Finally the maximum, and hence our new estimate $\hat{\mathbf{s}}$, of μ_S is the mode of this inverse-Wishart PDF which is (cf. Appendix B.3.1)

$$\hat{\mathbf{s}} = \frac{1}{K} \sum_{k=1}^{K} \overleftarrow{\boldsymbol{\Psi}}_{S_k} \tag{4.69}$$

$$= \frac{1}{K} \sum_{k=1}^{K} \mathbf{V}_{U_k} + \boldsymbol{m}_{U_k} \boldsymbol{m}_{U_k}^{\mathsf{T}} \,. \tag{4.70}$$

In the following we consider the scalar case. Note that this case has already been worked out in [23, 59]. In the scalar case, i.e. if $U_k \in \mathbb{R}$, the inverse-Wishart EM message degenerates to an inverse-Gamma message. (See Appendix B.3.1 for details on the inverse-Gamma distribution.) Specifically,

$$\overleftarrow{\mu}_{S_k}(s_k) \propto s_k^{-(\overleftarrow{\alpha}_{S_k}+1)} \, e^{-\overleftarrow{\beta}_{S_k}/s_k} \tag{4.71}$$

$$\propto \mathcal{G}^{-1}\Big(s_k \Big| \overleftarrow{\alpha}_{S_k}, \overleftarrow{\beta}_{S_k}\Big) \,, \tag{4.72}$$

with

$$\overleftarrow{\alpha}_{S_k} = -1/2 \tag{4.73}$$

$$\overleftarrow{\beta}_{S_k} = \frac{\overrightarrow{\sigma}_U^2 \, \overleftarrow{\sigma}_{U_k}^4 + \overrightarrow{\sigma}_U^4 \big(\overleftarrow{\sigma}_{U_k}^2 + \overleftarrow{m}_{U_k}^2\big)}{2\big(\overrightarrow{\sigma}_U^2 + \overleftarrow{\sigma}_{U_k}^2\big)^2} \,, \tag{4.74}$$

where $\vec{\sigma}_U^2$ is the estimate of s from the previous iteration.

The maximizing value of the product over $k = 1, \ldots, K$ of all EM messages $\overleftarrow{\mu}_{S_k}$ is

$$\hat{s} = \frac{2}{K} \sum_{k=1}^{K} \overleftarrow{\beta}_{S_k} \tag{4.75}$$

$$= \frac{\hat{s}_{\text{old}}}{K} \sum_{k=1}^{K} \frac{\overleftarrow{\sigma}_{U_k}^4 + \hat{s}_{\text{old}} \overleftarrow{\sigma}_{U_k}^2 + \hat{s}_{\text{old}} \overleftarrow{m}_{U_k}^2}{\overleftarrow{\sigma}_{U_k}^4 + \hat{s}_{\text{old}} \overleftarrow{\sigma}_{U_k}^2 + \hat{s}_{\text{old}} (\hat{s}_{\text{old}} + \overleftarrow{\sigma}_{U_k}^2)} \,, \tag{4.76}$$

where $\hat{s}_{\text{old}} \triangleq \vec{\sigma}_U^2$ is the estimate from the previous iteration.

4.5.3 Variance Estimation by Local Approximation

The local view of Figure 4.10c allows us to treat the problem of variance estimation by linearization of a multiplication node in the style of (4.21). Let the multiplication be defined as in Figure 4.10c and let

$$\boldsymbol{x}^{\mathsf{T}} \triangleq \left[\text{cvect}(\mathbf{A})^{\mathsf{T}}, \boldsymbol{u}^{\mathsf{T}}, \boldsymbol{y}^{\mathsf{T}}\right] . \tag{4.77}$$

In this case the Jacobian (4.19) is

$$\mathbf{H} = \left[-\hat{\boldsymbol{u}}^{\mathsf{T}} \otimes \mathbf{I}, -\hat{\mathbf{A}}, \mathbf{I}\right] , \tag{4.78}$$

and the approximate constraint is

$$\tilde{H}(\boldsymbol{x}) = \mathbf{H}(\boldsymbol{x} - \hat{\boldsymbol{x}}) \tag{4.79}$$

$$= -\hat{\boldsymbol{u}}^{\mathsf{T}} \otimes \mathbf{I}\Big(\text{cvect}(\mathbf{A}) - \text{cvect}(\hat{\mathbf{A}})\Big) - \hat{\mathbf{A}}(\boldsymbol{u} - \hat{\boldsymbol{u}}) + \boldsymbol{y} - \hat{\boldsymbol{y}} \tag{4.80}$$

$$= -\Big(\mathbf{A} - \hat{\mathbf{A}}\Big)\hat{\boldsymbol{u}} - \hat{\mathbf{A}}(\boldsymbol{u} - \hat{\boldsymbol{u}}) + \boldsymbol{y} - \hat{\boldsymbol{y}} \tag{4.81}$$

$$= -\mathbf{A}\hat{\boldsymbol{u}} - \hat{\mathbf{A}}\boldsymbol{u} + \boldsymbol{y} - \hat{\boldsymbol{y}} \,. \tag{4.82}$$

This approximate constraint results in the factor graph depicted in Figure 4.12. In this case, the freedom in choosing the operating point $\hat{\boldsymbol{u}}$, $\hat{\mathbf{A}}$, and $\hat{\boldsymbol{y}}$ is slightly restricted. Specifically, $\vec{\boldsymbol{m}}_U$ is not a sensible choice for $\hat{\boldsymbol{u}}$ because we always have $\vec{\boldsymbol{m}}_U = \boldsymbol{0}$.

In principle, the alternative parametrization in Figure 4.10b, using directly the covariance matrix as a parameter, can be approximated in the

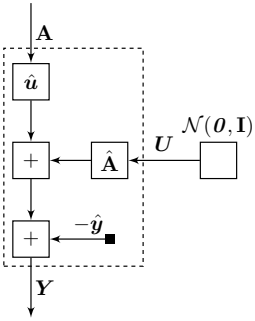


Figure 4.12: Variance estimation by linearization.

style of (4.17) by a Taylor series truncated to the second order. The case of a general covariance matrix, however, turns out to be unwieldy. Furthermore, there is a need for taking measure to ensure positive (semi-) definiteness of the solution.

Chapter 5

Conclusion and Outlook

In this first part of this thesis we have taken a look at various topics, such as RLS, Kalman filtering and EM, that traditionally are not treated in a factor graph framework. We have shown, how we can gain new insight into these topics and develop new algorithms by taking the structured approach of a SSM represented by a factor graph.

Also, we have put forward a modeling view of discrete-time signals that draws from continuous-time models. This view allows low-complexity online processing of a stream of data with low latency (in contrast to long FIR systems) and with strong coupling across time (in contrast to block-based processing).

The main achievements in this part are:

- The formulation of distributed regularization.
- Formal connections between LTI SSMs with and without input and autonomous systems with a forgetting factor.
- A connection between autonomous second-order systems and Fourier transforms of an exponentially weighted signal.
- A model for splitting state spaces with a smooth transition between completely decoupled and completely coupled treatment.
- The review of three relevant principles (CM, EM, and local Taylor approximation) for approximating non-Gaussian or nonlinear factor graph nodes and for devising iterative algorithms.
- The local factor graph view of parameter estimation for the system state-transition matrix in Jordan canonical form.

- The application of this view to the estimation of quasi-periodic signals.
- The local factor graph view of certain instances of variance and covariance matrix estimation.

We have not treated any further principles for dealing with non-Gaussian factors and nonlinear constraints. This will remain a topic of future work. Especially, it seems that different approaches yield algorithms with different convergence properties. These need to be studied. Also, we did not treat the question of initialization of these iterative algorithms.

No comparison has been done with other methods for estimating the (time-varying) fundamental frequency of a (quasi-)periodic signal. In the signal processing community is common, to first apply the Hilbert transform [17, 41, 67] to a given real valued signal in order to obtain the corresponding complex valued analytic signal. A similar approach could be envisaged in the context of factor graphs and EM. This approach might lead to potentially simpler and yet accurate algorithms.

A further topic that will be relevant in the future is model selection. There exist intriguing approaches [5, 96, 100] that until now have not been formulated in a way suitable for forward-only processing.

Part II

Likelihoods and Glue Factors

"How can we explain the fact that, of the thousand products of our unconscious activity, some are invited to cross the threshold, while others remain outside? Is it mere chance that gives them this privilege? Evidently not."

Henri Poincaré (1854–1912)

Chapter 6

On Scale Factors and Likelihoods

6.1 Introduction

In this chapter we first take a step back and make some general considerations about sum-product message passing in factor graphs. Specifically, we address the fact that messages are *scaled* probability density functions (PDFs) (or probability mass functions (PMFs)). Thus, a message is completely described by (the parameters of) a PDF (or a PMF) and a *scale factor*.

In the previous chapters, these scale factors were never needed because we were mostly considering problems of the type

$$\hat{x} = \underset{x}{\operatorname{argmax}}\, \beta\, f(x) = \underset{x}{\operatorname{argmax}}\, f(x)\,, \tag{6.1}$$

where any constant factor β is irrelevant. In this chapter, many problems still are of the type (6.1), but for others we cannot neglect the scale factors anymore.

This thesis is not the first to put forward the use of scale factors in factor graphs. For example, in [2, 22] a method for computing information rates of channels with memory is reported. This method is formulated in terms of scale factors of sum-product messages, thereby making use of the prediction message rule (I.3) in Table 6.1.

Here we intend to systematically lay the foundations for understanding how scale factors can be computed along with other parameters of a message in sum-product message passing. Although fairly general, our

view is biased towards signal processing based on state-space models (SSMs).

We start in Section 6.2 we define two different types of scale factors. For both types, various update rules applicable to sum-product message passing in factor graphs are derived. We follow [65] and tabulate these rules for general factors and for linear constraints both for general real-valued variables and in the Gaussian case.

In Section 6.3 the connection between such scale factors and the likelihood $p(\tilde{\boldsymbol{y}})$ of a particular observation $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ is presented. If the underlying statistical model is represented by a factor graph, this connection becomes particularly appealing. We extend this view to likelihood functions and certain log-likelihood ratios (LLRs), thereby pointing out cases in which the computation of scale factors is not necessary. Both, the general probabilistic setting for real-valued random variables and the Gaussian setting are treated.

Since actual likelihoods can take any values in the whole range of $\mathbb{R}$, the computation of such values is in general numerically not stable. In Section 6.4 we show a local approximation that brings these values closer to 0 but does not destroy too much information from a local perspective. The principle idea of this approximation has been introduced in [64].

In the last section of this chapter we elaborate on two similar but in general distinct approaches to a parameter selection problem: Should a parameter u in a model be used or not? We do not treat the general case but restrict ourselves to a Gaussian linear setting. In the first approach, a Bayesian view is adopted, in which we consider a prior distribution of u. In the second approach, we elaborate on the detection problem with hypotheses $u \neq 0$ versus $u = 0$.

6.2 Definitions and Message Update Rules

6.2.1 General Factor Graphs

Throughout this thesis the symbol μ refers to a sum-product message in a factor graph, as opposed to any scaled version of a sum-product message. Rarely, μ will stand for a max-product message. We will however always indicate if this is the case.

In the following, we assume that a sum-product message is a multivariate

non-negative but finite function $\mu\colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, or a multivariate Dirac delta. Moreover, we allow functions whose integral with respect to any subset of the n dimensions tends to infinity. These assumptions are motivated by the fact that even in simple Gaussian factor graphs, Dirac messages and messages with unbounded integral can easily occur. We name the latter *improper* messages as opposed to *proper*, i.e. integrable, messages.

Most of the considerations made here carry over, or even simplify, in the case of discrete messages (cf. Appendix C.1). We do not treat any other types of ranges here, e.g. mixed discrete and continuous ranges or subsets of $\mathbb{R}^n$, but such generalizations may be possible.

The above assumptions inspire us to define two types of scale factors, only one of which needs to be finite and nonzero for a given sum-product message at a time. We define the scale factor $\vec{\beta}_X$ of the forward sum-product message $\vec{\mu}_X$ on an edge $\boldsymbol{X}$ in a factor graph as

$$\vec{\beta}_X \triangleq \int \vec{\mu}_X(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \tag{6.2}$$

if the integral is finite and non-zero. Otherwise $\vec{\beta}_X$ remains undefined. The second type of a scale factor is defined by

$$\vec{\gamma}_X \triangleq \vec{\mu}_X(\boldsymbol{0}) \tag{6.3}$$

if $\vec{\mu}_X$ has a finite non-zero value at $\boldsymbol{0}$. Otherwise $\vec{\gamma}_X$ remains undefined. In the following, any statement about a β-type or a γ-type scale factor has the implicit assumption that β or γ is well defined respectively.

If $\vec{\beta}_X = 1$, then $\vec{\mu}_X$ is a properly scaled PDF. In the case of a neutral message $\vec{\mu}_X(\boldsymbol{x}) = 1$, the scale factor $\vec{\beta}_X$ tends to infinity while the scale factor $\vec{\gamma}_X = 1$. If on the other hand $\vec{\mu}_X(\boldsymbol{x}) = \delta(\boldsymbol{x} - \tilde{\boldsymbol{x}})$ for some fixed $\tilde{\boldsymbol{x}}$, then $\vec{\beta}_X = 1$ while the scale factor $\vec{\gamma}_X$ tends to infinity if $\tilde{\boldsymbol{x}} = \boldsymbol{0}$ or to 0 otherwise.

For backward messages, $\overleftarrow{\beta}_X$ and $\overleftarrow{\gamma}_X$ are defined analogously to (6.2) and (6.3). We define the pseudo-marginal

$$\mu_X(\boldsymbol{x}) \triangleq \vec{\mu}_X(\boldsymbol{x})\, \overleftarrow{\mu}_X(\boldsymbol{x})\,. \tag{6.4}$$

The computation of this pseudo-marginal can be viewed as message passing through an equality node inserted on the edge $\boldsymbol{X}$. (See Appendix C.1

Figure C.2 for more details on such a modification.) We denote the scale factor of a pseudo-marginal by β_X or γ_X.

Notationally, we sometimes use the symbol

$$\overrightarrow{p}_X(\boldsymbol{x}) \triangleq \overrightarrow{\mu}_X(\boldsymbol{x}) / \overrightarrow{\beta}_X \,, \tag{6.5}$$

to denote the properly scaled PDF induced by the message $\overrightarrow{\mu}_X$. Likewise we use

$$\overrightarrow{\nu}_X(\boldsymbol{x}) \triangleq \overrightarrow{\mu}_X(\boldsymbol{x}) / \overrightarrow{\gamma}_X \tag{6.6}$$

to denote a different kind of scaled message. With these definitions we are able to write a message in at least either of the following two ways:

$$\overrightarrow{\mu}_X(\boldsymbol{x}) = \overrightarrow{\beta}_X \, \overrightarrow{p}_X(\boldsymbol{x}) = \overrightarrow{\gamma}_X \, \overrightarrow{\nu}_X(\boldsymbol{x}) \,. \tag{6.7}$$

We use the symbols p and ν to denote the respective quantities relating to the pseudo-marginal (6.4).

The choice of the value $\boldsymbol{0}$ in the definition (6.3) of $\overrightarrow{\gamma}_X$ may seem arbitrary, and indeed, other definitions of the form $\overrightarrow{\gamma}_X \triangleq \overrightarrow{\mu}_X(\tilde{\boldsymbol{x}})$ for a fixed value $\tilde{\boldsymbol{x}}$ are possible, e.g., the expected value $\tilde{\boldsymbol{x}} = \mathsf{E}_{\overrightarrow{p}_X}[X]$, the mode $\tilde{\boldsymbol{x}} = \operatorname{argmax}_{\boldsymbol{x}} \overrightarrow{\mu}_X(\boldsymbol{x})$, or the median $\int_{-\infty}^{\tilde{\boldsymbol{x}}} \overrightarrow{\mu}_X(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = 1/2$.

The choice $\tilde{\boldsymbol{x}} = \boldsymbol{0}$ in the definition (6.3) has, besides being always well defined, the advantage that the two scale factor types β and γ are connected via a Fourier transform. Specifically, the γ-type scale factor can alternatively be written as

$$\overrightarrow{\gamma}_X = \frac{1}{(2\pi)^n} \int \overrightarrow{\phi}_X(\boldsymbol{\omega}) \, \mathrm{d}\boldsymbol{\omega} \,, \tag{6.8}$$

where

$$\overrightarrow{\phi}_X(\boldsymbol{\omega}) \triangleq \int \overrightarrow{\mu}_X(\boldsymbol{x}) \, e^{-i\boldsymbol{\omega}^\mathsf{T}\boldsymbol{x}} \, \mathrm{d}\boldsymbol{x} \tag{6.9}$$

is the Fourier-transformed message[1] and n is the dimensionality of the vector $\boldsymbol{X}$. Also, for the β-type scale factor, we have

$$\overrightarrow{\beta}_X = \overrightarrow{\phi}_X(\boldsymbol{0}) \,. \tag{6.10}$$

[1] In most texts on probability theory the characteristic function (the Fourier transformed PDF) is usually defined slightly different, lacking the minus sign in the exponent.

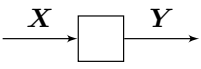
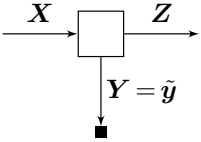
Node	Update rule
 $p(\boldsymbol{y}\|\boldsymbol{x})$	$$\vec{\beta}_Y = \vec{\beta}_X \quad \text{(I.1)}$$ If $p(\boldsymbol{y}\|\boldsymbol{0}) = \delta(\boldsymbol{y})$, then $$\overleftarrow{\gamma}_X = \overleftarrow{\gamma}_Y \quad \text{(I.2)}$$
 $p_{Y,Z\|X}(\tilde{\boldsymbol{y}}, \boldsymbol{z}\|\boldsymbol{x})$	If $\int \vec{\mu}_{\mathsf{pY}}(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} < \infty$, then $$\vec{\beta}_Z = \vec{\beta}_X\, \vec{p}_{\mathsf{pY}}(\tilde{\boldsymbol{y}})\,, \quad \text{(I.3)}$$ where $\vec{p}_{\mathsf{pY}}(\boldsymbol{y}) \propto \vec{\mu}_{\mathsf{pY}}(\boldsymbol{y})$ is the prediction PDF, and $\vec{\mu}_{\mathsf{pY}}(\boldsymbol{y})$ is the prediction message computed using the sum-product rule from $\vec{\mu}_X(\boldsymbol{x})$ and a neutral message $\overleftarrow{\mu}_Z(\boldsymbol{z}) = 1$.

Table 6.1: Scale factor update rules for general conditional probability density function (PDF) nodes.

Hence, we can translate between $\vec{\beta}_X$ and $\vec{\gamma}_X$ using

$$\frac{\vec{\beta}_X}{\vec{\gamma}_X} = \frac{\int \vec{\mu}_X(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\vec{\mu}_X(\boldsymbol{0})} = \frac{(2\pi)^n\, \vec{\phi}_X(\boldsymbol{0})}{\int \vec{\phi}_X(\omega)\,\mathrm{d}\omega} = \int \vec{\nu}_X(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\,, \tag{6.11}$$

provided that the respective values are finite and nonzero.

Table 6.1 lists three general update rules for scale factors when passing sum-product messages in a factor graph. These rules are valid for any messages conforming with the stated assumptions. The local factors considered here are conditional PDFs exclusively. Note that it is straightforward to generalize these rules to factors that are scaled conditional PDFs. We do not treat more general factors in this thesis.

The update rules (I.1) and (I.2) turn out beautifully simple. The update rule (I.3) involves the prediction PDF $\vec{p}_{\mathsf{pY}}$, i.e., our knowledge about $\boldsymbol{Y}$ based on knowing $\vec{\mu}_X$ but with complete ignorance about $\boldsymbol{Z}$. Indeed, the prediction PDF plays an important role in the literature [50]. The proofs for the update rules in Table 6.1 are in Appendix C.2.

We next turn to linear constraint nodes as in Table 6.2, all of which are

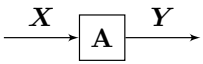
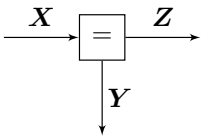
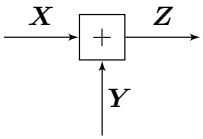
Node	Update rule
	For any $\mathbf{A} \in \mathbb{R}^{n\times m}$: $\vec{\beta}_Y = \vec{\beta}_X$ (II.1) $\overleftarrow{\gamma}_X = \overleftarrow{\gamma}_Y$ (II.2) If $\mathbf{A} \in \mathbb{R}^{n\times n}$ is nonsingular: $\vec{\gamma}_Y = \vec{\gamma}_X \left\lvert \det \mathbf{A}^{-1} \right\rvert$ (II.3) $\overleftarrow{\beta}_X = \overleftarrow{\beta}_Y \left\lvert \det \mathbf{A}^{-1} \right\rvert$ (II.4)
	$\vec{\gamma}_Z = \vec{\gamma}_X \overleftarrow{\gamma}_Y$ (II.5)
	$\vec{\beta}_Z = \vec{\beta}_X \vec{\beta}_Y$ (II.6) $\overleftarrow{\beta}_X = \vec{\beta}_Y \overleftarrow{\beta}_Z$ (II.7)

Table 6.2: Scale factor update rules for linear nodes.

special cases of the first row in Table 6.1. The listed update rules are still valid for any messages that comply with the stated assumptions. Note that in general it is not possible to pass both the β-type and the γ-type scale factor through all the nodes in every direction. In such cases, it may be possible to use (6.11) to translate between the two types. The computation of γ_X (the scale factor of a pseudo-marginal (6.4)) can be done using (II.5). All update rules in Table 6.2 are proved in Appendix C.3.

6.2.2 Gaussian Messages

In this thesis we are particularly interested in Gaussian messages, because they arise naturally in linear SSMs. For a Gaussian message, the

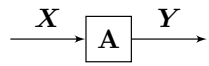
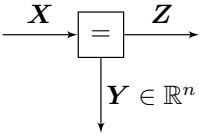
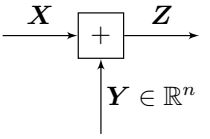
Node	Update rule
 $\mathbf{A} \in \mathbb{R}^{n\times m}$ $\operatorname{rank}\mathbf{A} = \min\{n,m\}$	If $m \geq n$ and if $\overrightarrow{\mathbf{V}}_X$ is nonsingular: $$\overrightarrow{\gamma}_Y = \overrightarrow{\gamma}_X \sqrt{\frac{(2\pi)^m \det \overrightarrow{\mathbf{W}}_Y}{(2\pi)^n \det \overrightarrow{\mathbf{W}}_X}}\, e^{\overrightarrow{\boldsymbol{m}}_X^{\mathsf{T}} \mathbf{W} \overrightarrow{\boldsymbol{m}}_X/2}, \quad \text{(III.1)}$$ where $\mathbf{W} \triangleq \overrightarrow{\mathbf{W}}_X - \mathbf{A}^{\mathsf{T}}\overrightarrow{\mathbf{W}}_Y\mathbf{A}$. If $m \leq n$ and if $\overleftarrow{\mathbf{V}}_Y$ is nonsingular: $$\overleftarrow{\beta}_X = \overleftarrow{\beta}_Y \sqrt{\frac{(2\pi)^m \det \overleftarrow{\mathbf{V}}_X}{(2\pi)^n \det \overleftarrow{\mathbf{V}}_Y}}\, e^{-\overleftarrow{\boldsymbol{m}}_Y^{\mathsf{T}} \overleftarrow{\mathbf{W}}_Y \mathbf{V} \overleftarrow{\mathbf{W}}_Y \overleftarrow{\boldsymbol{m}}_Y/2}, \quad \text{(III.2)}$$ where $\mathbf{V} \triangleq \overleftarrow{\mathbf{V}}_Y - \mathbf{A}\overleftarrow{\mathbf{V}}_X\mathbf{A}^{\mathsf{T}}$.
	If $\mathbf{V} \triangleq \overrightarrow{\mathbf{V}}_X + \overleftarrow{\mathbf{V}}_Y$ is nonsingular: $$\overrightarrow{\beta}_Z = \overrightarrow{\beta}_X \overleftarrow{\beta}_Y \sqrt{\frac{\det \mathbf{V}^{-1}}{(2\pi)^n}}\, e^{-\boldsymbol{m}^{\mathsf{T}}\mathbf{V}^{-1}\boldsymbol{m}/2}, \quad \text{(III.3)}$$ where $\boldsymbol{m} \triangleq \overrightarrow{\boldsymbol{m}}_X - \overleftarrow{\boldsymbol{m}}_Y$.
	If $\mathbf{W} \triangleq \overrightarrow{\mathbf{W}}_X + \overrightarrow{\mathbf{W}}_Y$ is nonsingular: $$\overrightarrow{\gamma}_Z = \overrightarrow{\gamma}_X\, \overrightarrow{\gamma}_Y \sqrt{\frac{(2\pi)^n}{\det \mathbf{W}}}\, e^{\boldsymbol{m}^{\mathsf{T}}\mathbf{W}\boldsymbol{m}/2}, \quad \text{(III.4)}$$ where $\boldsymbol{m} \triangleq \overrightarrow{\mathbf{W}}_X\overrightarrow{\boldsymbol{m}}_X - \overrightarrow{\mathbf{W}}_Y\overrightarrow{\boldsymbol{m}}_Y$. If $\mathbf{W} \triangleq \overrightarrow{\mathbf{W}}_Y + \overleftarrow{\mathbf{W}}_Z$ is nonsingular: $$\overleftarrow{\gamma}_X = \overrightarrow{\gamma}_Y\, \overleftarrow{\gamma}_Z \sqrt{\frac{(2\pi)^n}{\det \mathbf{W}}}\, e^{\boldsymbol{m}^{\mathsf{T}}\mathbf{W}\boldsymbol{m}/2}, \quad \text{(III.5)}$$ where $\boldsymbol{m} \triangleq \overrightarrow{\mathbf{W}}_Y\overrightarrow{\boldsymbol{m}}_Y + \overleftarrow{\mathbf{W}}_Z\overleftarrow{\boldsymbol{m}}_Z$.

Table 6.3: Additional scale factor update rules for Gaussian messages.

quantities (6.5), (6.6), and (6.9) can be written as

$$\begin{aligned}\overrightarrow{p}_X(\boldsymbol{x}) &= \mathcal{N}\big(\boldsymbol{x}\,\big|\,\overrightarrow{\boldsymbol{m}}_X, \overrightarrow{\mathbf{V}}_X\big) \\ &= \frac{1}{\sqrt{(2\pi)^n \det \overrightarrow{\mathbf{V}}_X}}\, e^{-(\boldsymbol{x}-\overrightarrow{\boldsymbol{m}}_X)^{\mathsf{T}}\overrightarrow{\mathbf{V}}_X^{-1}(\boldsymbol{x}-\overrightarrow{\boldsymbol{m}}_X)/2}\,, \end{aligned} \tag{6.12}$$

$$\overrightarrow{\nu}_X(\boldsymbol{x}) = e^{-\boldsymbol{x}^{\mathsf{T}}\overrightarrow{\mathbf{W}}_X\boldsymbol{x}/2+\boldsymbol{x}^{\mathsf{T}}\overrightarrow{\mathbf{W}}_X\overrightarrow{\boldsymbol{m}}_X}\,, \tag{6.13}$$

$$\overrightarrow{\phi}_X(\boldsymbol{\omega}) = \overrightarrow{\beta}_X\, e^{-\boldsymbol{\omega}^{\mathsf{T}}\overrightarrow{\mathbf{V}}_X\boldsymbol{\omega}/2-i\boldsymbol{\omega}^{\mathsf{T}}\overrightarrow{\boldsymbol{m}}_X}\,, \tag{6.14}$$

where $\overrightarrow{\mathbf{V}}_X = \overrightarrow{\mathbf{W}}_X^{-1}$ is the covariance matrix if $\overrightarrow{\mathbf{W}}_X$ is nonsingular, $\overrightarrow{\boldsymbol{m}}_X$ is the mean vector, and n is the dimensionality of $\boldsymbol{X}$. The translation (6.11) between the two scale factors types specializes to

$$\frac{\overrightarrow{\beta}_X}{\overrightarrow{\gamma}_X} = \int \overrightarrow{\nu}_X(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \tag{6.15}$$

$$= \frac{1}{\mathcal{N}\big(\boldsymbol{0}\,\big|\,\overrightarrow{\boldsymbol{m}}_X, \overrightarrow{\mathbf{W}}_X^{-1}\big)} = \sqrt{\frac{(2\pi)^n}{\det \overrightarrow{\mathbf{W}}_X}}\, e^{\overrightarrow{\boldsymbol{m}}_X^{\mathsf{T}}\overrightarrow{\mathbf{W}}_X\overrightarrow{\boldsymbol{m}}_X/2}\,. \tag{6.16}$$

Note that (6.16) is only valid if $\overrightarrow{\mathbf{W}}_X$ is nonsingular.

With the help of (6.16) we revisit uncovered cases in Table 6.2. Table 6.3 lists missing update rules for linear nodes. Note that these rules still do not cover all possible situations such as multiplication with a rank deficient matrix or cases in which covariance matrices of messages are singular.

Finally, in Table 6.4 we present update rules for the composite blocks used in Kalman filtering type algorithms [65]. Note that these rules apply even if $\overrightarrow{\mathbf{V}}_X$ (or $\overrightarrow{\mathbf{W}}_X$) is singular. This property enables us to augment the standard Kalman filter (or Kalman smoother) with the computation of scale factors, as we will see in Section 6.2.3 and later use in Chapter 7.

It is interesting to note that the auxiliary quantities $\mathbf{V}$ and $\mathbf{W}$ in this table are the inverse Kalman gain and the inverse of the equivalent quantity for the information filter. Here, these quantities arise very natural from the proofs without any application of the matrix inversion lemma.

The update rules in Tables 6.3 and 6.4 are proved in Appendices C.4. The proof idea used is not restricted to Gaussian messages but may be applied to other types of messages, provided that the conversion (6.11) between β and γ can be expressed.

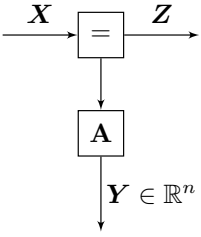
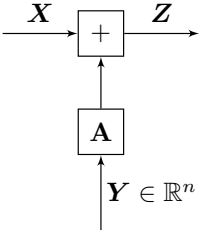
Node	Update rule
	If $\mathbf{V} \triangleq \overleftarrow{\mathbf{V}}_Y + \mathbf{A}\overrightarrow{\mathbf{V}}_X\mathbf{A}^{\mathsf{T}}$ is nonsingular: $\overrightarrow{\beta}_Z = \overrightarrow{\beta}_X \overleftarrow{\beta}_Y \sqrt{\frac{\det \mathbf{V}^{-1}}{(2\pi)^n}}\, e^{-\boldsymbol{m}^{\mathsf{T}}\mathbf{V}^{-1}\boldsymbol{m}/2}$, (IV.1) where $\boldsymbol{m} \triangleq \mathbf{A}\overrightarrow{\boldsymbol{m}}_X + \overleftarrow{\boldsymbol{m}}_Y$.
	If $\mathbf{W} \triangleq \overrightarrow{\mathbf{W}}_Y + \mathbf{A}^{\mathsf{T}}\overrightarrow{\mathbf{W}}_X\mathbf{A}$ is nonsingular: $\overrightarrow{\gamma}_Z = \overrightarrow{\gamma}_X \overrightarrow{\gamma}_Y \sqrt{\frac{(2\pi)^n}{\det \mathbf{W}}}\, e^{\boldsymbol{m}^{\mathsf{T}}\mathbf{W}\boldsymbol{m}/2}$, (IV.2) where $\mathbf{W}\boldsymbol{m} \triangleq \overrightarrow{\mathbf{W}}_Y\overrightarrow{\boldsymbol{m}}_Y - \mathbf{A}^{\mathsf{T}}\overrightarrow{\mathbf{W}}_X\overrightarrow{\boldsymbol{m}}_X$. If $\mathbf{W} \triangleq \overrightarrow{\mathbf{W}}_Y + \mathbf{A}^{\mathsf{T}}\overleftarrow{\mathbf{W}}_Z\mathbf{A}$ is nonsingular: $\overleftarrow{\gamma}_X = \overleftarrow{\gamma}_Z \overrightarrow{\gamma}_Y \sqrt{\frac{(2\pi)^n}{\det \mathbf{W}}}\, e^{\boldsymbol{m}^{\mathsf{T}}\mathbf{W}\boldsymbol{m}/2}$, (IV.3) where $\mathbf{W}\boldsymbol{m} \triangleq \overrightarrow{\mathbf{W}}_Y\overrightarrow{\boldsymbol{m}}_Y + \mathbf{A}^{\mathsf{T}}\overleftarrow{\mathbf{W}}_Z\overleftarrow{\boldsymbol{m}}_Z$.

Table 6.4: Additional scale factor update rules for composite blocks.

Example 6.1: Dual Scale Factor and Signal Energy in Autonomous Models

Consider an autonomous SSM as depicted in Figure 3.3. Assume that we have a observations $Y_k = \tilde{\boldsymbol{y}}_k$ for $k = 1, \ldots, K$. Then the logarithm of the dual (β-type) scale factor $\overleftarrow{\gamma}_{X_0}$ evaluates to

$$\ln \overleftarrow{\gamma}_{X_0} = -\frac{K}{2}\left(\ln(2\pi\sigma_Z^2) + \frac{\xi^2}{\sigma_Z^2}\right), \tag{6.17}$$

where σ_Z^2 is the observation noise variance and

$$\xi^2 \triangleq \frac{1}{K}\sum_{k=1}^{K} \tilde{\boldsymbol{y}}_k^2 \tag{6.18}$$

is the signal power. This can be shown easily by applying the rules (II.2) and (II.5). Equation (6.17) is valid for any autonomous model. If

additionally, the matrices $\mathbf{A}_k$ are nonsingular for $k = 2, \ldots, K$ then we can write

$$\ln \vec{\gamma}_{X_K} = -\frac{K}{2}\left(\ln(2\pi\sigma_Z^2) + \frac{\xi^2}{\sigma_Z^2}\right) - \sum_{k=2}^{K} \ln|\det A_k| \,. \tag{6.19}$$

where we have applied Rule (II.3). ◊

6.2.3 Kalman Smoothing with Scale Factor Computation

In [65], Kalman filtering type algorithms are described as Gaussian message passing algorithms in the factor graph of Figure 6.1 representing a linear SSM. The essential ingredient thereby is the usage of the update rules in [65, Table 4] for the composite equality and matrix multiplication block or for the composite addition and matrix multiplication block. These two composite blocks are labelled in Figure 6.1 by (b) and (a) respectively. (For the actual update rules we refer to [65].)

In the literature many versions of Kalman filtering and Kalman smoothing have been described [50]. In the following we specify our usage of the term *Kalman smoothing* by defining an algorithm with respect to Figure 6.1. We envisage an online scenario, in which the data $\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \ldots$ arrives sequentially in a stream. (Cf. Section 2.7.2 for an example of an offline and an online algorithm.) Let $K \in \mathbb{N}$ be the time index of the current data item and let $D \in \mathbb{N}$ be a delay parameter. We start the algorithm at $K = 1$ by defining

$$\vec{\mu}_{X_0}(\boldsymbol{x}_0) = \mathcal{N}\big(\boldsymbol{x}_0 \,\big|\, \boldsymbol{0}, \vec{\mathbf{V}}_X\big) \,, \tag{6.20}$$

where $\vec{\mathbf{V}}_X$ is the steady-state forward covariance matrix as defined in Section 3.2.3. As an additional initialization step, we have to compute $\vec{\mu}_{X_k}$ for $k = 1, \ldots, D$ and set $K = D$.

Kalman Smoothing:

a) Increase K and fetch the next data item $\tilde{\boldsymbol{y}}_K$.

b) Compute $\vec{\mu}_{X_{K-D}}$ in the factor graph of Figure 6.1.

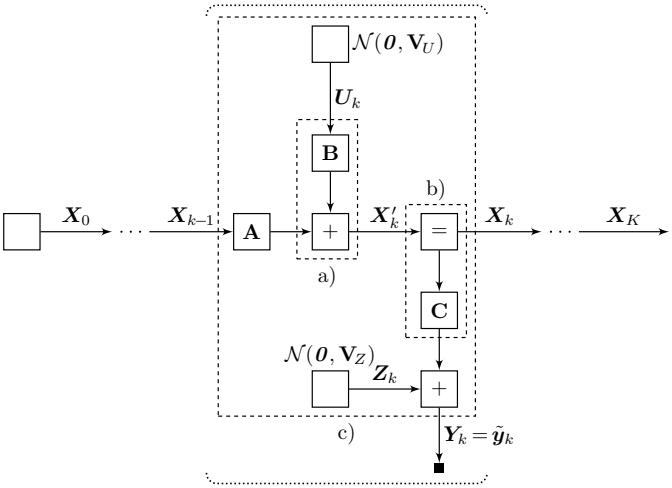


Figure 6.1: Linear state-space model (SSM) with composite blocks (a) and (b) used for Kalman filtering and smoothing. Block (c) can be interpreted as a conditional probability density function (PDF).

c) Compute $\overleftarrow{\mu}_{X_k}$ for $k = K, \ldots, K - D$ in the factor graph of Figure 6.1.

d) Compute any quantity of interest for time step $K - D$, e.g., a state estimate $\hat{\boldsymbol{x}}_{K-D}$, an input estimate $\hat{\boldsymbol{u}}_{K-D}$, or an estimate $\hat{\boldsymbol{y}}_{K-D}$ of the noise-free output.

In Steps (b) and (c), the message passing updates make use of either of the composite blocks (a) or (b) in Figure 6.1.

Now assume, that we want to compute scale factors along with the message passing algorithm outlined above. We will see in Section 6.3 why in principle it may be attractive to do this. The goal is to compute either of the two types, the β-type or the γ-type scale factor, for each message, but not necessarily both at the same time.

For forward message passing in Step (b), $\overrightarrow{\beta}_{X_{K-D}}$ can be computed using the prediction rule (I.3). This follows by noting that the block (c) in Figure 6.1 can be interpreted as a conditional PDF $p(\boldsymbol{x}_k, \boldsymbol{y}_k | \boldsymbol{x}_{k-1})$.

For backward message passing in Step (c), however, a direct application

of the prediction rule (I.3) is precluded because of two reasons. First, the block (c) in Figure 6.1 can be interpreted as a conditional PDF $p(\boldsymbol{x}_{k-1}, \boldsymbol{y}_k | \boldsymbol{x}_k)$ only if $\mathbf{A}$ is invertible. This is true for infinite impulse response (IIR) systems that comply with Assumptions 3.1 in Section 3.1, but not, e.g., for finite impulse response (FIR) systems. Second, the prediction message $\overrightarrow{\mu}_{\mathsf{p}Y_k}$ is not always proper, i.e. integrable, and hence a prediction PDF does not exist. The reason for this is, that we start off with a neutral message $\overleftarrow{\mu}_{X_K}(\boldsymbol{x}_K) = 1$. Depending on the system matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, it may take some steps in the backward pass until $\overleftarrow{\mu}_{X_k}$ becomes proper.

One way to overcome these difficulties would be to give up the factorization over the last few time steps, e.g., $K-1, K$ and collapse the graph for these time steps into one factor, e.g., $f(\boldsymbol{y}_{K-2}, \boldsymbol{y}_{K-1}, \boldsymbol{y}_K) = p(\boldsymbol{y}_{K-1}, \boldsymbol{y}_K | \boldsymbol{x}_{K-2})$.

A more attractive way is to compute the γ-type scale factor instead of the β-type scale factor. For the neutral message $\overleftarrow{\mu}_{X_K}$, we have $\overleftarrow{\gamma}_{X_K} = 1$. Backward message passing through the factor graph of Figure 6.1 is indeed possible if we use the update rule (IV.3) for the composite block (a).

We end this section with two remarks. First, note that it is always possible to convert a Kalman smoother with delay D to a Kalman filter (forward message passing only) by augmenting the state vector by a length of D. Since no backward message passing is done in this augmented SSM, we can always apply the prediction rule (I.3) to compute the β-type scale factor.

Second, we have to point out, that scale factor computation as described here is, in general not feasible for long signals, since the scale factors quickly tend to extreme values. We will elaborate on this in the following section. Nevertheless, in Chapter 7, we will encounter related algorithms based on glue factors, in which scale factor computation does make sense.

6.3 The Connection Between Likelihood and Scale Factors

Our primary usage of factor graphs is the representation of statistical models for an observable signal. We restrict ourselves to real-valued variables and real-valued parameters throughout. This is by no means the only possibility. See [68] for a way of constructing a PDF of real-valued

observable variables by means of complex-valued hidden variables.

In statistical models, likelihood computations play an important role, most notably for estimation and detection problems. In this section we present the connection between scale factors of sum-product messages and likelihoods.

6.3.1 About Normalization Factors and Normalization Functions

Consider a cycle-free factor graph with edges $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_m$ and global function $f(\boldsymbol{z}) \geq 0$, where $\boldsymbol{Z} \triangleq (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_m)$. We define the *normalization factor*

$$\zeta \triangleq \int f(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} \,. \tag{6.21}$$

A factor graph with global function $f(\boldsymbol{z})$ is said to *represent* a PDF $p(\boldsymbol{z})$ if the normalization factor (6.21) relates the two as

$$p(\boldsymbol{z}) = f(\boldsymbol{z})/\zeta \,. \tag{6.22}$$

In many contexts, ζ is termed partition function [33].

Consider a cycle-free factor graph with edges $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_m, \boldsymbol{\Theta}$ and global function $f(\boldsymbol{z}, \boldsymbol{\theta}) \geq 0$. We define the *normalization function*

$$\zeta(\boldsymbol{\theta}) \triangleq \int f(\boldsymbol{z}, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{z} \,. \tag{6.23}$$

A factor graph with global function $f(\boldsymbol{z}, \boldsymbol{\theta})$ is said to *represent* a conditional PDF $p(\boldsymbol{z}|\boldsymbol{\theta})$ if the normalization function (6.23) relates the two as

$$p(\boldsymbol{z}|\boldsymbol{\theta}) = f(\boldsymbol{z}, \boldsymbol{\theta})/\zeta(\boldsymbol{\theta}) \,. \tag{6.24}$$

Note that a factor graph can only represent a PDF if the normalization factor is finite and nonzero. Similarly, a factor graph can only represent a conditional PDF if for every value $\boldsymbol{\theta}$ the integral (6.23) converges to a finite and nonzero value. Always when we use the word *represent*, we implicitly assume a finite and nonzero normalization factor or normalization function.

It follows directly from these assumptions that any factor graph with global function $f(\boldsymbol{z}, \boldsymbol{\theta})$ that represents a joint PDF $p(\boldsymbol{z}, \boldsymbol{\theta})$ can also represent a conditional PDF $p(\boldsymbol{z}|\boldsymbol{\theta})$. In this case we can think of the normalization function $\zeta(\boldsymbol{\theta})$ as a scaled prior PDF of $\boldsymbol{\Theta}$, and (6.24) simply is a generalization of the definition of conditional PDFs. We refer to this case as the *re-normalized case*.

On the other hand, a factor graph with global function $f(\boldsymbol{z}, \boldsymbol{\theta})$ that is proportional to a conditional PDF $p(\boldsymbol{z}|\boldsymbol{\theta})$ cannot represent a joint PDF $p(\boldsymbol{z}, \boldsymbol{\theta})$, since in this case $\zeta(\boldsymbol{\theta})$ is a constant. We will refer to this case as the *strictly conditional* case. Notationally, we refer to this (constant) type of normalization function as $\zeta^{\not\boldsymbol{\theta}}$.

Theorem 6.1: Normalization Factor, Normalization Function

a) In any cycle-free factor graph with edges $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_m$ that represents a PDF $p(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m)$, the normalization factor (6.21) *can be computed as*

$$\zeta = \beta_{Z_k}\,, \tag{6.25}$$

for any $k = 1, \ldots, m$.

b) In any cycle-free factor graph with edges $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_m, \boldsymbol{\Theta}$ that represents a conditional PDF $p(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m|\boldsymbol{\theta})$, the normalization function (6.23) *can be computed as*

$$\zeta(\boldsymbol{\theta}) = \mu_{\Theta}(\boldsymbol{\theta})\,, \ \textit{or} \tag{6.26}$$
$$\zeta(\boldsymbol{\theta}) = \beta_{Z_k}(\boldsymbol{\theta}) \tag{6.27}$$

in the re-normalized case, and as

$$\zeta^{\not\boldsymbol{\theta}} = \gamma_{\Theta} \tag{6.28}$$

in the strictly conditional case, where $\boldsymbol{\Theta}$ is the edge that represents the conditioning variable in the factor graph and $\beta_{Z_k}(\boldsymbol{\theta})$ for any $k = 1, \ldots, m$ is regarded as a function of $\boldsymbol{\theta}$.

Note that the computation of $\beta_{Z_k}(\boldsymbol{\theta})$ is done without integrating over $\boldsymbol{\theta}$, i.e., in this case $\boldsymbol{\theta}$ is formally not an edge in the factor graph anymore but is treated as a parameter of some nodes. We will stick to this slightly abusive notation and somewhat misleading meaning throughout.

Proof. Part (a) follows from the definition (6.4) of a pseudo-marginal and the definition of sum-product messages as

$$\zeta = \int \vec{\mu}_{Z_k}(\boldsymbol{z}_k)\, \overleftarrow{\mu}_{Z_k}(\boldsymbol{z}_k)\, \mathrm{d}\boldsymbol{z}_k = \beta_{Z_k}\,. \tag{6.29}$$

In Part (b), Equations (6.26) and (6.27) follow from the definition of a pseudo-marginal and the definition of the β-type scale factor respectively as

$$\zeta(\boldsymbol{\theta}) = \int f(\boldsymbol{z}, \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{z} = \mu_{\Theta}(\boldsymbol{\theta})\,, \tag{6.30}$$

$$\zeta(\boldsymbol{\theta}) = \int f(\boldsymbol{z}, \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{z} = \beta_{Z_k}(\boldsymbol{\theta})\,. \tag{6.31}$$

In the strictly conditional case $\zeta(\boldsymbol{\theta})$ is constant and (6.30) simplifies to

$$\zeta^{\not\emptyset} = \mu_{\Theta}(\boldsymbol{\theta}) = \mu_{\Theta}(\boldsymbol{0}) = \gamma_{\Theta}\,. \qquad \square$$

Theorem 6.1 tells us how sum-product message passing can be used to compute normalization factors or normalization functions. Note that, in contrast to other applications of sum-product message passing, message scale factors must not be neglected. This puts to use the update rules for scale factors as presented in Section 6.2.

The strictly conditional case is, e.g., fulfilled if all the factor graph nodes are conditional PDFs, the directions of all arrows comply with the convention introduced in Section 1.5.2, and the only open edge directed towards a node is $\boldsymbol{\Theta}$. In such cases β_{Θ} tends to infinity and the factor graph cannot represent a PDF $p(\boldsymbol{z}, \boldsymbol{\theta})$.

In the strictly conditional case, the global function of a factor graph by definition is proportional to a conditional PDF. If, on the other hand, the factor graph is proportional to a joint PDF, we can formulate the new factor graph

$$f'(\boldsymbol{z}, \boldsymbol{\theta}) = \mu_{\Theta}(\boldsymbol{\theta})^{-1}\, f(\boldsymbol{z}, \boldsymbol{\theta}) \tag{6.32}$$

based on the original graph $f(\boldsymbol{z}, \boldsymbol{\theta})$. This new graph now has a global function that is a conditional PDF. Figure 6.2 shows the original version and the re-normalized version of such a graph. Note that in these graphs, the edges $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_m$ have been abstracted into a single edge $\boldsymbol{Z}$, and that the edge arrows in this case lack the meaning of conditional probabilities.

(a) Global function $f(\boldsymbol{z}, \boldsymbol{\theta}) \propto p(\boldsymbol{z}, \boldsymbol{\theta})$. (b) Global function $f'(\boldsymbol{z}, \boldsymbol{\theta}) \propto p(\boldsymbol{z}|\boldsymbol{\theta})$.

Figure 6.2: Conversion of a factor graph that represents a joint PDF to a factor graph that represents a conditional PDF by re-normalization.

It must be stressed that, in principle, by re-normalizing a factor graph as done in (6.32) and in Figure 6.2 we throw away any prior knowledge in the model about $\boldsymbol{\theta}$. In many cases, this might not be desirable.

Also, it has to be mentioned that there are other ways to compute the partition function in a factor graph, both exactly and approximately [108, 109]. These techniques, however, may not be easily adapted to a filtering or smoothing scenario.

Example 6.2: Conversion of a Factor Graph to a Bayesian Network

Assume that we are given a factor graph (i.e. a collection of factors and variables) and we would like to convert this graph into a Bayesian network [56, 57, 77]. Once we have decided on a directed acyclic structure of the target graph we need to scale every factor. In this process it may become clear that we have chosen an unsuitable structure, or that, indeed, the factor graph cannot be converted to a Bayesian network.

Assuming that this conversion is possible without changing the graph topology, we have to scale every factor that has both incoming and outgoing edges such that it represents a conditional PDF. (Factors with only outgoing edges need to be scaled to represent prior PDFs and factors with only incoming edges need to be scaled to represent PDFs evaluated at some specific values.)

Let $f(\boldsymbol{x}, \boldsymbol{y})$ be a factor with edges $(\boldsymbol{X}_1, , \ldots, \boldsymbol{X}_n) \triangleq \boldsymbol{X}$, $(\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m) \triangleq \boldsymbol{Y}$ in a factor graph and let $p(\boldsymbol{y}|\boldsymbol{x})$ be the target factor in the Bayesian network. Then, the missing normalization function is $\zeta(\boldsymbol{x}) = \int f(\boldsymbol{x}, \boldsymbol{y}) \, \mathrm{d}\boldsymbol{y}$.

We restrict our setting to the case in which this normalization function factors as $\zeta(\boldsymbol{x}) = \prod_{j=1}^{n} \zeta(\boldsymbol{x}_j)$. In this case we send $\zeta(\boldsymbol{x}_1), \ldots, \zeta(\boldsymbol{x}_n)$ as messages to the nodes that connect to f via the edges $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ respectively. We then are ready to construct the Bayesian factor as

$$p(\boldsymbol{y}|\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{y}) \frac{\prod_{j=1}^{m} \zeta(\boldsymbol{y}_j)}{\prod_{j=1}^{n} \zeta(\boldsymbol{x}_j)} \,. \tag{6.33}$$

Bayesian factors with only outgoing edges are constructed as

$$p(\boldsymbol{y}) = f(\boldsymbol{y}) \prod_{j=1}^{m} \zeta(\boldsymbol{y}_j) \,. \qquad \Diamond$$

6.3.2 Statistical Models and Likelihood

In most statistical models we make a distinction between hidden variables and observable variables. Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X}$ and $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y}$ be edges in a cycle-free factor graph with global function $f(\boldsymbol{x}, \boldsymbol{y}) \geq 0$, where $\boldsymbol{X} \triangleq (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X})$ are the hidden variables and $\boldsymbol{Y} \triangleq (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y})$ are the observable variables.

In the following we consider message passing in such a graph with and without plugged-in observations $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$. In our notation we use a circle $(\cdot)^\circ$ to indicate messages (and their parameters) that are computed without plugging in any observations. Usually, in the corresponding factor graph, the edges $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y}$ are then unconnected "open" edges. We use no special symbol for messages that are computed in the factor graph with plugged-in observations $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$.

The normalization factor of a statistical model with hidden and observable variables is computed according to Theorem 6.1a as $\zeta = \beta_Z^\circ$ on any edge $\boldsymbol{Z}$ among $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X}$ and $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y}$. The actual PDF represented by the factor graph is then

$$p(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}, \boldsymbol{y}) / \beta_Z^\circ \,. \tag{6.34}$$

Analogously, for a statistical model that additionally depends on a parameter $\boldsymbol{\theta}$, the normalization function is computed according to Theorem 6.1b as $\zeta(\boldsymbol{\theta}) = \mu_\Theta^\circ(\boldsymbol{\theta})$, and the conditional PDF represented by the factor graph is obtained by re-normalization as

$$p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) = f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) / \mu_\Theta^\circ(\boldsymbol{\theta}) \tag{6.35}$$

or in the strictly conditional case as

$$p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) = f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})/\gamma_\Theta^\circ \,. \tag{6.36}$$

In a statistical model $p(\boldsymbol{x}, \boldsymbol{y})$ with hidden variables $\boldsymbol{X}$ and observable variables $\boldsymbol{Y}$, the *likelihood* of a particular observation $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ is defined as

$$p(\tilde{\boldsymbol{y}}) \triangleq \int p(\boldsymbol{x}, \tilde{\boldsymbol{y}}) \, \mathrm{d}\boldsymbol{x} \,. \tag{6.37}$$

Analogously, in a statistical model $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$ with hidden variables $\boldsymbol{X}$, observable variables $\boldsymbol{Y}$, and parameter vector $\boldsymbol{\theta}$, the *likelihood function* given a particular observation $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ is defined equivalently to (6.37) as

$$p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) \triangleq \int p(\boldsymbol{x}, \tilde{\boldsymbol{y}}|\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{x} \,. \tag{6.38}$$

Theorem 6.2: Likelihood and Likelihood Function

a) Assume that a cycle-free factor graph with edges $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X}$ and $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y}$ represents a statistical model $p(\boldsymbol{x}, \boldsymbol{y})$ with hidden variables $\boldsymbol{X} \triangleq (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X})$ and observable variables $\boldsymbol{Y} \triangleq (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y})$. The likelihood of a particular observation $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ can be computed as

$$p(\tilde{\boldsymbol{y}}) = \beta_Z/\beta_Z^\circ \,, \tag{6.39}$$

for any edge $\boldsymbol{Z}$ among $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X}$ and $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y}$.

b) Assume that a cycle-free factor graph with edges $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X}$, $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y}$, and $\boldsymbol{\Theta}$ represents a statistical model $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$ with hidden variables $\boldsymbol{X} \triangleq (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X})$, observable variables $\boldsymbol{Y} \triangleq (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y})$ and parameter (vector) $\boldsymbol{\theta}$. The likelihood function given a particular observation $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ can be computed in the renormalized case as

$$p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \mu_\Theta(\boldsymbol{\theta})/\mu_\Theta^\circ(\boldsymbol{\theta}) \,, \tag{6.40}$$

$$p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \beta_Z(\boldsymbol{\theta})/\beta_Z^\circ(\boldsymbol{\theta}) \,, \tag{6.41}$$

for any edge $\boldsymbol{Z}$ among $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X}$ and $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y}$, where $\beta_Z(\boldsymbol{\theta})$ and $\beta_Z^\circ(\boldsymbol{\theta})$ are regarded as functions of $\boldsymbol{\theta}$. In the strictly conditional case, the likelihood function can be computed as

$$p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \mu_\Theta(\boldsymbol{\theta})/\gamma_\Theta^\circ \,, \tag{6.42}$$

$$p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \beta_Z(\boldsymbol{\theta})/\beta_Z^\circ \,. \tag{6.43}$$

We recall that for the computation of $\beta_Z(\boldsymbol{\theta})$, $\boldsymbol{\theta}$ is not treated as an edge but as a parameter of some nodes, and hence no integration or summation over $\boldsymbol{\theta}$ is done.

Proof. Part (a) follow directly from Theorem 6.1a and the definition (6.37) of the likelihood as

$$p(\tilde{\boldsymbol{y}}) = \int f(\boldsymbol{x}, \tilde{\boldsymbol{y}}) \, \mathrm{d}\boldsymbol{x} / \beta_Z^\circ \tag{6.44}$$

$$= \beta_Z / \beta_Z^\circ \, . \tag{6.45}$$

Part (b) follows directly from Theorem 6.1b and the definition (6.38) of the likelihood function as

$$p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \int f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{x} / \mu_\Theta^\circ(\boldsymbol{\theta}) \tag{6.46}$$

$$= \mu_\Theta(\boldsymbol{\theta}) / \mu_\Theta^\circ(\boldsymbol{\theta}) \, , \tag{6.47}$$

$$p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \int f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{x} / \beta_Z^\circ(\boldsymbol{\theta}) \tag{6.48}$$

$$= \beta_Z(\boldsymbol{\theta}) / \beta_Z^\circ(\boldsymbol{\theta}) \, , \tag{6.49}$$

and equivalently for the strictly conditional case. □

Theorem 6.2 gives us a general view on how to compute likelihoods and likelihood functions by sum-product message passing in a given factor graph. Note that, in general, the scale factors of the sum-product messages must not be neglected.

There are, however, many problems for which the computation of message scale factors can be neglected nevertheless. Some of these problems are treated in Section 6.3.3.

Again, we stress that in the re-normalized case of Equations (6.40) and (6.41), we are throwing away prior information about $\boldsymbol{\theta}$. In many cases, it may be more appropriate to consider $\boldsymbol{\theta}$ as a random variable and then let the factor graph represent the joint PDF $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$.

6.3.3 Problems that are Unaffected by Scale Factors

In general, the computation of the message scale factors is unavoidable when computing likelihoods or likelihood functions by message passing.

There are, however, important classes of problems for which the exact knowledge of the likelihood or the likelihood function is not important, and the computation of scale factors can be omitted. In the following we exemplify such situations.

Theorem 6.3: Scale Factors, Estimation, and Hypothesis Testing
Assume that a factor graph with edges $\boldsymbol{X}_1, \dots, \boldsymbol{X}_{m_X}$, $\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_{m_Y}$, *and* $\boldsymbol{\Theta}$ *represents a statistical model* $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$ *or* $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$, *where* $\boldsymbol{X} \triangleq (\boldsymbol{X}_1, \dots, \boldsymbol{X}_{m_X})$ *are hidden variables,* $\boldsymbol{Y} \triangleq (\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_{m_Y})$ *are observable variables, and* $\boldsymbol{\theta}$ *is a parameter (vector) or also a hidden variable. Let* $\tilde{\boldsymbol{y}}$ *be observations of* $\boldsymbol{Y}$.

a) *For the computation of the maximum likelihood (ML) estimate* $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$, *the scale factors of the messages can be neglected.*

b) *For the computation of the joint maximum a posteriori (MAP) estimate* $\hat{\boldsymbol{x}}$ *or the separate MAP estimate* $\hat{\boldsymbol{x}}_j$ *for any* $j = 1, \dots, m_X$ *given fixed a parameter* $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$, *the scale factors of the messages can be neglected.*

c) *Let* $p(\tilde{\boldsymbol{y}}|\mathcal{H}_\theta)$ *denote the likelihood under a hypothesis* $\mathcal{H}_\theta$ *that depends only on* $\boldsymbol{\theta}$. *For the computation of any LLR involving such hypotheses, the scale factors of the messages can be neglected.*

Proof. Part (a) follows from the definition of the ML estimate and Theorem 6.2b in the re-normalized case as

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} \triangleq \operatorname*{argmax}_{\boldsymbol{\theta}} p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) \tag{6.50}$$

$$= \operatorname*{argmax}_{\boldsymbol{\theta}} \mu_{\Theta}(\boldsymbol{\theta}) / \mu^{\circ}_{\Theta}(\boldsymbol{\theta}) \,, \tag{6.51}$$

and in the strictly conditional case

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \mu_{\Theta}(\boldsymbol{\theta}) / \gamma^{\circ}_{\Theta} \,. \tag{6.52}$$

Clearly, in neither (6.51) and (6.52), any constant factor is of no importance.

For Part (b) we first consider a point-wise MAP estimate defined by

$$\hat{\boldsymbol{x}}_{j,\mathrm{MAP}} \triangleq \operatorname*{argmax}_{\boldsymbol{x}_j} p\big(\boldsymbol{x}_j, \tilde{\boldsymbol{y}}\big|\tilde{\boldsymbol{\theta}}\big) \,, \tag{6.53}$$

for which, due to sum-product message passing, we can write

$$\hat{\boldsymbol{x}}_{j,\mathrm{MAP}} = \operatorname*{argmax}_{\boldsymbol{x}_j} \mu_{X_j}(\boldsymbol{x}_j)\,. \tag{6.54}$$

Second, we consider a joint MAP estimate $\hat{\boldsymbol{x}} \triangleq (\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{m_X})$ defined by

$$\hat{\boldsymbol{x}}_{\mathrm{MAP}} \triangleq \operatorname*{argmax}_{\boldsymbol{x}} p\big(\boldsymbol{x}, \tilde{\boldsymbol{y}}\,\big|\,\tilde{\boldsymbol{\theta}}\big)\,. \tag{6.55}$$

For this estimate we switch to max-product message passing and hence we can write

$$\big[\hat{\boldsymbol{x}}_{\mathrm{MAP}}\big]_j = \operatorname*{argmax}_{\boldsymbol{x}_j} \mu_{X_j}(\boldsymbol{x}_j)\,, \tag{6.56}$$

where μ stands now temporarily for the max-marginal. It is evident that neither in (6.54) nor in (6.56) any constant factor in μ_{X_j} does matter.

For Part (c) we consider two fixed values $\tilde{\boldsymbol{\theta}}_1$ and $\tilde{\boldsymbol{\theta}}_0$ corresponding to two hypotheses. If the factor graph represents a joint PDF $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$ we consider two options for constructing an LLR: Either we re-normalize the graph to a conditional PDFs in the sense of (6.35) or we take the prior information in $\boldsymbol{\theta}$ as a prior on the hypotheses. In the first case, the LLR can be written with the help of Theorem 6.2b as

$$\mathrm{LLR} \triangleq \log \frac{p\big(\tilde{\boldsymbol{y}}\,\big|\,\tilde{\boldsymbol{\theta}}_1\big)}{p\big(\tilde{\boldsymbol{y}}\,\big|\,\tilde{\boldsymbol{\theta}}_0\big)} \tag{6.57}$$

$$= \log \frac{\mu_\Theta(\tilde{\boldsymbol{\theta}}_1)\,\mu^\circ_\Theta(\tilde{\boldsymbol{\theta}}_0)}{\mu^\circ_\Theta(\tilde{\boldsymbol{\theta}}_1)\,\mu_\Theta(\tilde{\boldsymbol{\theta}}_0)}\,. \tag{6.58}$$

In the second case, the LLR follows from sum-product message passing as

$$\mathrm{LLR} \triangleq \log \frac{p(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\theta}}_1)}{p(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\theta}}_0)} \tag{6.59}$$

$$= \log \frac{\mu_\Theta(\tilde{\boldsymbol{\theta}}_1)}{\mu_\Theta(\tilde{\boldsymbol{\theta}}_0)}\,. \tag{6.60}$$

In the strictly conditional case the LLR can be written with the help of Theorem 6.2b as

$$\mathrm{LLR} = \log \frac{\mu_\Theta(\tilde{\boldsymbol{\theta}}_1)\,\gamma^\circ_\Theta}{\gamma^\circ_\Theta\,\mu_\Theta(\tilde{\boldsymbol{\theta}}_0)} = \log \frac{\mu_\Theta(\tilde{\boldsymbol{\theta}}_1)}{\mu_\Theta(\tilde{\boldsymbol{\theta}}_0)}\,. \tag{6.61}$$

The claim is proved by noting that, in Equations (6.58), (6.60) and (6.61), any constant factor in μ_Θ and μ°_Θ cancels. □

Note that if we re-normalize the function represented by the graph as suggested in (6.35), we implicitly throw away any prior information in the model about $\boldsymbol{\theta}$. In a likelihood ratio test, such prior information can, however, be reintroduced in the threshold [54, 107]. More precisely, we can formulate the following prior ratio about the hypotheses

$$\frac{p(\mathcal{H}_1)}{p(\mathcal{H}_0)} = \frac{p(\tilde{\boldsymbol{\theta}}_1)}{p(\tilde{\boldsymbol{\theta}}_0)} = \frac{\mu^\circ_\Theta(\tilde{\boldsymbol{\theta}}_1)}{\mu^\circ_\Theta(\tilde{\boldsymbol{\theta}}_0)}\,, \tag{6.62}$$

which is the missing factor between (6.60) and (6.58). Note that this ratio does not depend on the observation $\tilde{\boldsymbol{y}}$ and hence can be computed offline.

Now we turn to three special cases of Theorem 6.3c which will be of interest in later sections. For the following we recall the definition (6.6) for the scaled messages ν:

$$\nu_Z(\boldsymbol{z}) \triangleq \mu_Z(\boldsymbol{z})/\gamma_Z\,. \tag{6.63}$$

Theorem 6.4: Three Log-Likelihood Ratios (LLRs)
Assume that a factor graph with edges $\boldsymbol{X}_1, \dots, \boldsymbol{X}_{m_X}$, $\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_{m_Y}$, and $\boldsymbol{\Theta}$ represents a statistical model $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$ or $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$, where $\boldsymbol{X} \triangleq (\boldsymbol{X}_1, \dots, \boldsymbol{X}_{m_X})$ are hidden variables, $\boldsymbol{Y} \triangleq (\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_{m_Y})$ are observable variables, and $\boldsymbol{\theta}$ is a parameter vector or also a hidden variable. Given observations $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ we consider the following three binary hypothesis testing problems:

$$\text{a)} \qquad \mathcal{H}_1\colon \boldsymbol{\theta} \neq \boldsymbol{0} \qquad\qquad \mathcal{H}_0\colon \boldsymbol{\theta} = \boldsymbol{0}\,, \tag{6.64}$$

$$\text{b)} \qquad \mathcal{H}_1\colon \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} \qquad\qquad \mathcal{H}_0\colon \boldsymbol{\theta} = \boldsymbol{0}\,, \tag{6.65}$$

$$\text{c)} \qquad \mathcal{H}_1\colon \boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_{\mathrm{ML}}\,,\ \text{or}\ \boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_{\mathrm{MAP}} \qquad \mathcal{H}_0\colon \boldsymbol{\theta} = \boldsymbol{0}\,, \tag{6.66}$$

with corresponding LLRs $\log \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_1)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_0)}$. In all but the re-normalized cases, the LLRs can be written as

$$\text{a)} \qquad \mathrm{LLR} = \log \int \nu_\Theta(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}\,, \tag{6.67}$$

$$\text{b)} \qquad \mathrm{LLR} = \log \nu_\Theta(\tilde{\boldsymbol{\theta}})\,, \tag{6.68}$$

$$\text{c)} \qquad \mathrm{GLLR} = \log \max_{\boldsymbol{\theta}} \nu_\Theta(\boldsymbol{\theta})\,, \tag{6.69}$$

where, in the strictly conditional case, for (a) we have assumed a uninformative prior.

As a variation of the above theorem we state that, if the factor graph represents a conditional PDF $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$ then in both the re-normalized case and the strictly conditional case the LLRs can be written as

$$\text{a)} \qquad \text{LLR} = \log \int \frac{\nu_\Theta(\boldsymbol{\theta})}{\nu_\Theta^\circ(\boldsymbol{\theta})} \, \mathrm{d}\boldsymbol{\theta} \,, \tag{6.70}$$

$$\text{b)} \qquad \text{LLR} = \log \frac{\nu_\Theta(\tilde{\boldsymbol{\theta}})}{\nu_\Theta^\circ(\tilde{\boldsymbol{\theta}})} \,, \tag{6.71}$$

$$\text{c)} \qquad \text{GLLR} = \log \max_{\boldsymbol{\theta}} \frac{\nu_\Theta(\boldsymbol{\theta})}{\nu_\Theta^\circ(\boldsymbol{\theta})} \,, \tag{6.72}$$

where for (a) we have assumed a uninformative prior and where, in the strictly conditional case, $\nu_\Theta^\circ(\boldsymbol{\theta}) = 1$ is a constant.

In Case (a), the hypothesis $\mathcal{H}_1$ can alternatively be formulated as $\boldsymbol{\theta}$ being arbitrary. More precisely, the LLR assumes an "uninformative prior" PDF on $\boldsymbol{\theta}$ under $\mathcal{H}_1$ [53]. For our purposes we make a definition of such a prior that is only valid inside a LLR. In essence this is a prior PDF whose variance tends to infinity. Since $\boldsymbol{\theta}$ is a real-valued vector we take a Gaussian PDF $\mathcal{N}\big(\boldsymbol{\theta}\big|\,\boldsymbol{0}, \mathbf{W}_\Theta^{-1}\big)$ and define the LLR in the re-normalized and the strictly conditional case as

$$\text{a)} \qquad \text{LLR} \triangleq \log \lim_{\mathbf{W}_\Theta \to \mathbf{0}} \frac{\int p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) \, \mathcal{N}\big(\boldsymbol{\theta}\big|\,\boldsymbol{0}, \mathbf{W}_\Theta^{-1}\big) \, \mathrm{d}\boldsymbol{\theta}}{p(\tilde{\boldsymbol{y}}|\,\boldsymbol{0}) \, \mathcal{N}\big(\boldsymbol{0}\big|\,\boldsymbol{0}, \mathbf{W}_\Theta^{-1}\big)} \tag{6.73}$$

$$= \log \frac{\int p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}}{p(\tilde{\boldsymbol{y}}|\,\boldsymbol{0})} \,. \tag{6.74}$$

The LLR in Case (c) with plugged-in ML estimate is known as the generalized log-likelihood ratio (GLLR) [54].

Proof of Theorem 6.4 and Equations (6.70)–(6.72). We start with the case in which the factor graph represents a joint PDF $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$. In the following we give the definitions of the likelihoods and we use the

definition of sum-product messages and the definition (6.6) to derive

$$\text{a)–c)} \qquad p(\tilde{\boldsymbol{y}}|\mathcal{H}_0) \triangleq p(\tilde{\boldsymbol{y}}, \boldsymbol{\theta} = \boldsymbol{0}) = \gamma_\Theta \,, \tag{6.75}$$

$$\text{a)} \qquad p(\tilde{\boldsymbol{y}}|\mathcal{H}_1) \triangleq \int p(\tilde{\boldsymbol{y}}, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} = \gamma_\Theta \int \nu_\Theta(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \,, \tag{6.76}$$

$$\text{b)} \qquad p(\tilde{\boldsymbol{y}}|\mathcal{H}_1) \triangleq p(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{\theta}}) = \gamma_\Theta \, \nu_\Theta(\tilde{\boldsymbol{\theta}}) \,, \tag{6.77}$$

$$\text{c)} \qquad p(\tilde{\boldsymbol{y}}|\mathcal{H}_1) \triangleq \max_{\boldsymbol{\theta}} p(\tilde{\boldsymbol{y}}, \boldsymbol{\theta}) = \gamma_\Theta \max_{\boldsymbol{\theta}} \nu_\Theta(\boldsymbol{\theta}) \,. \tag{6.78}$$

Note that in Case (a) the LLR is defined by Equation (6.74), which involves the uninformative prior. The LLRs in Equations (6.67)–(6.69) follow straightforwardly.

Now we treat the case in which the factor graph represents a conditional PDF $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$. For this we use Theorem 6.2b to rewrite the likelihoods for the re-normalized case.

$$\text{b)–c)} \qquad p(\tilde{\boldsymbol{y}}|\boldsymbol{0}) \triangleq p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta} = \boldsymbol{0}) = \gamma_\Theta / \gamma_\Theta^\circ \,, \tag{6.79}$$

$$\text{a)} \qquad p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \frac{\gamma_\Theta \, \nu_\Theta(\boldsymbol{\theta})}{\gamma_\Theta^\circ \, \nu_\Theta^\circ(\boldsymbol{\theta})} \,, \tag{6.80}$$

$$\text{b)} \qquad p(\tilde{\boldsymbol{y}}|\mathcal{H}_1) \triangleq p\big(\tilde{\boldsymbol{y}}\big|\tilde{\boldsymbol{\theta}}\big) = \frac{\gamma_\Theta \, \nu_\Theta(\tilde{\boldsymbol{\theta}})}{\gamma_\Theta^\circ \, \nu_\Theta^\circ(\tilde{\boldsymbol{\theta}})} \,, \tag{6.81}$$

$$\text{c)} \qquad p(\tilde{\boldsymbol{y}}|\mathcal{H}_1) \triangleq \max_{\boldsymbol{\theta}} p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \frac{\gamma_\Theta \, \nu_\Theta(\boldsymbol{\theta})}{\gamma_\Theta^\circ \, \nu_\Theta^\circ(\boldsymbol{\theta})} \,. \tag{6.82}$$

The LLRs in (6.70)–(6.72) follow straightforwardly. The strictly conditional case of Equations (6.67)–(6.69) follows from noting that in this case $\nu_\Theta^\circ(\boldsymbol{\theta}) = 1$ for any $\boldsymbol{\theta}$. □

6.3.4 The Gaussian Setting

While the findings in Sections 6.3.1–6.3.3 apply to general PDFs on real-valued variables with infinite support, we now consider the special case of Gaussian PDFs. First we mention that in the Gaussian case, sum-product messages and max-product messages are proportional to each other, and MAP estimation and MMSE (minimum mean squared error) estimation coincide. Hence, any statements that have involved max-product instead of sum-product messages need no special treatment in the Gaussian case as long as the message scale factors do not matter.

In this section, we walk through the same topics as in Sections 6.3.1–6.3.3. We start with normalization factors and normalization functions, continue with likelihoods and likelihood functions and finally give expressions for certain LLRs.

Let $p(\boldsymbol{z})$ be a multivariate Gaussian PDF that is represented by a factor graph with edges $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_m$, where $\boldsymbol{Z} \triangleq (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_m)$. The normalization factor of Equation (6.25) can be formulated by means of Equation (6.16) and Rule (II.5) as

$$\zeta = \overrightarrow{\gamma}_{Z_k} \overleftarrow{\gamma}_{Z_k} / \mathcal{N}\big(\boldsymbol{0} \big| \boldsymbol{m}_{Z_k}, \mathbf{W}_{Z_k}^{-1}\big)\,, \tag{6.83}$$

where we recall that $\boldsymbol{m}_{Z_k}$ and $\mathbf{W}_{Z_k}$ are parameters of the marginal on edge $\boldsymbol{Z}_k$. Rule (III.3) allows us to alternatively formulate (6.25) in terms of the β-type scale factor as

$$\zeta = \overrightarrow{\beta}_{Z_k} \overleftarrow{\beta}_{Z_k} \mathcal{N}\big(\boldsymbol{0} \big| \overrightarrow{\boldsymbol{m}}_{Z_k} - \overleftarrow{\boldsymbol{m}}_{Z_k}, \overrightarrow{\mathbf{V}}_{Z_k} + \overleftarrow{\mathbf{V}}_{Z_k}\big)\,. \tag{6.84}$$

In the case where a factor graph represents the conditional Gaussian PDF $p(\boldsymbol{z}|\boldsymbol{\theta})$ the normalization function (6.26) can be written as

$$\zeta(\boldsymbol{\theta}) = \gamma_\Theta \, e^{-\boldsymbol{\theta}^\mathsf{T}\mathbf{W}_\Theta\boldsymbol{\theta}/2 + \boldsymbol{\theta}^\mathsf{T}\mathbf{W}_\Theta\boldsymbol{m}_\Theta}\,, \tag{6.85}$$

where we have made use of (6.13). Alternatively,

$$\zeta(\boldsymbol{\theta}) = \beta_\Theta \, \mathcal{N}\big(\boldsymbol{\theta} \big| \boldsymbol{m}_\Theta, \mathbf{V}_\Theta\big)\,. \tag{6.86}$$

For a Gaussian factor graph with edges $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X}, \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y}$ that represents a statistical model with hidden variables $\boldsymbol{X} \triangleq (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{m_X})$ and observable variables $\boldsymbol{Y} \triangleq (\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{m_Y})$, the likelihood in Equation (6.39) can be reformulated in the style of (6.83) or (6.84) as

$$p(\tilde{\boldsymbol{y}}) = \frac{\overrightarrow{\gamma}_Z \overleftarrow{\gamma}_Z \, \mathcal{N}(\boldsymbol{0}|\boldsymbol{m}_Z^\circ, \mathbf{V}_Z^\circ)}{\overrightarrow{\gamma}_Z^\circ \overleftarrow{\gamma}_Z^\circ \, \mathcal{N}(\boldsymbol{0}|\boldsymbol{m}_Z, \mathbf{V}_Z)} \tag{6.87}$$

$$= \frac{\overrightarrow{\beta}_Z \overleftarrow{\beta}_Z \, \mathcal{N}\big(\boldsymbol{0} \big| \overrightarrow{\boldsymbol{m}}_Z - \overleftarrow{\boldsymbol{m}}_Z, \overrightarrow{\mathbf{V}}_Z + \overleftarrow{\mathbf{V}}_Z\big)}{\overrightarrow{\beta}_Z^\circ \overleftarrow{\beta}_Z^\circ \, \mathcal{N}\big(\boldsymbol{0} \big| \overrightarrow{\boldsymbol{m}}_Z^\circ - \overleftarrow{\boldsymbol{m}}_Z^\circ, \overrightarrow{\mathbf{V}}_Z^\circ + \overleftarrow{\mathbf{V}}_Z^\circ\big)}\,, \tag{6.88}$$

for any $\boldsymbol{Z}$ among all the edges.

If a statistical Gaussian model depends on a parameter vector $\boldsymbol{\theta}$ then the likelihood function in the re-normalized case can be written in the style

of (6.85) or (6.86) as

$$p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \frac{\gamma_\Theta}{\gamma_\Theta^\circ}\, e^{-\boldsymbol{\theta}^\mathsf{T}(\mathbf{W}_\Theta - \mathbf{W}_\Theta^\circ)\boldsymbol{\theta}/2 + \boldsymbol{\theta}^\mathsf{T}(\mathbf{W}_\Theta \boldsymbol{m}_\Theta - \mathbf{W}_\Theta^\circ \boldsymbol{m}_\Theta^\circ)} \tag{6.89}$$

$$= \frac{\beta_\Theta\, \mathcal{N}\big(\boldsymbol{0}\big|\boldsymbol{m}_\Theta, \mathbf{V}_\Theta\big)\, \mathcal{N}\big(\boldsymbol{\theta}\big|\boldsymbol{m}, \mathbf{V}\big)}{\beta_\Theta^\circ\, \mathcal{N}\big(\boldsymbol{0}\big|\boldsymbol{m}_\Theta^\circ, \mathbf{V}_\Theta^\circ\big)\, \mathcal{N}\big(\boldsymbol{0}\big|\boldsymbol{m}, \mathbf{V}\big)}\,, \tag{6.90}$$

with

$$\mathbf{V} \triangleq \big(\mathbf{W}_\Theta - \mathbf{W}_\Theta^\circ\big)^{-1}, \tag{6.91}$$

$$\boldsymbol{m} \triangleq \mathbf{V}\big(\mathbf{W}_\Theta \boldsymbol{m}_\Theta - \mathbf{W}_\Theta^\circ \boldsymbol{m}_\Theta^\circ\big)\,. \tag{6.92}$$

In the strictly conditional case, i.e., if $\mu_\Theta^\circ(\boldsymbol{\theta}) = \gamma_\Theta^\circ$, then Equation (6.42) applies and the likelihood function can be written in the Gaussian case as

$$p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \frac{\gamma_\Theta}{\gamma_\Theta^\circ} e^{-\boldsymbol{\theta}^\mathsf{T}\mathbf{W}_\Theta\boldsymbol{\theta}/2 + \boldsymbol{\theta}^\mathsf{T}\mathbf{W}_\Theta \boldsymbol{m}_\Theta}\,, \tag{6.93}$$

$$= \frac{\beta_\Theta}{\gamma_\Theta^\circ}\, \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m}_\Theta, \mathbf{V}_\Theta)\,. \tag{6.94}$$

We recall that β_Θ° tends to infinity in this special case.

We have given all the results in this section as pairs of equations, one for each scale factor type. Each result is only valid if the corresponding scale factors are finite and in (the case of ratios) non-zero. Also, corresponding matrices may have to be nonsingular. At least one equation in each pair is, however, always valid, as long as the factor graph represents the PDF.

Finally, we give the Gaussian case of the LLRs in Theorem 6.4. We recall the hypotheses

a) $\mathcal{H}_1\colon \boldsymbol{\theta} \neq \boldsymbol{0}$ $\mathcal{H}_0\colon \boldsymbol{\theta} = \boldsymbol{0}\,,$ (6.95)

b) $\mathcal{H}_1\colon \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$ $\mathcal{H}_0\colon \boldsymbol{\theta} = \boldsymbol{0}\,,$ (6.96)

c) $\mathcal{H}_1\colon \boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_\mathrm{ML}\,,$ or $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_\mathrm{MAP}$ $\mathcal{H}_0\colon \boldsymbol{\theta} = \boldsymbol{0}\,.$ (6.97)

The corresponding LLRs $\ln \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_1)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_0)}$ can be written in all but the renormalized cases as

a) $$\mathrm{LLR} = \boldsymbol{m}_\Theta^\mathsf{T}\mathbf{W}_\Theta\boldsymbol{m}_\Theta/2 + \frac{1}{2}\ln\frac{(2\pi)^n}{\det \mathbf{W}_\Theta}\,, \tag{6.98}$$

b) $$\mathrm{LLR} = \tilde{\boldsymbol{\theta}}^\mathsf{T}\mathbf{W}_\Theta\boldsymbol{m}_\Theta - \tilde{\boldsymbol{\theta}}^\mathsf{T}\mathbf{W}_\Theta\tilde{\boldsymbol{\theta}}/2\,, \tag{6.99}$$

c) $$\mathrm{GLLR} = \boldsymbol{m}_\Theta^\mathsf{T}\mathbf{W}_\Theta\boldsymbol{m}_\Theta/2\,, \tag{6.100}$$

and in the re-normalized and the strictly conditional case as

$$\text{a)} \qquad \mathrm{LLR} = (\mathbf{W}_\Theta \boldsymbol{m}_\Theta - \mathbf{W}^\circ_\Theta \boldsymbol{m}^\circ_\Theta)^\mathsf{T} (\mathbf{W}_\Theta - \mathbf{W}^\circ_\Theta)^{-1} \cdot (\mathbf{W}_\Theta \boldsymbol{m}_\Theta - \mathbf{W}^\circ_\Theta \boldsymbol{m}^\circ_\Theta)/2 + \frac{1}{2} \ln \frac{(2\pi)^n}{\det\big(\mathbf{W}_\Theta - \mathbf{W}^\circ_\Theta\big)}\,, \tag{6.101}$$

$$\text{b)} \qquad \mathrm{LLR} = \tilde{\boldsymbol{\theta}}^\mathsf{T} \big(\mathbf{W}_\Theta \boldsymbol{m}_\Theta - \mathbf{W}^\circ_\Theta \boldsymbol{m}^\circ_\Theta\big) - \tilde{\boldsymbol{\theta}}^\mathsf{T} \big(\mathbf{W}_\Theta - \mathbf{W}^\circ_\Theta\big) \tilde{\boldsymbol{\theta}}/2\,, \tag{6.102}$$

$$\text{c)} \qquad \mathrm{GLLR} = (\mathbf{W}_\Theta \boldsymbol{m}_\Theta - \mathbf{W}^\circ_\Theta \boldsymbol{m}^\circ_\Theta)^\mathsf{T} (\mathbf{W}_\Theta - \mathbf{W}^\circ_\Theta)^{-1} \cdot (\mathbf{W}_\Theta \boldsymbol{m}_\Theta - \mathbf{W}^\circ_\Theta \boldsymbol{m}^\circ_\Theta)/2\,, \tag{6.103}$$

where n is the dimensionality of $\boldsymbol{\theta}$. Note that in the strictly conditional case $\mathbf{W}^\circ_\Theta = \mathbf{0}$.

It is interesting to observe that the LLR of Case (a) and the GLLR of Case (c) are very similar. Since in Gaussian SSMs the only quantity that usually depends on the observations $\tilde{\boldsymbol{y}}$ is $\boldsymbol{m}_\Theta$, such that the only significant difference between Cases (a) and (c) is a constant offset.

The message parameters computed without plugging in observations appear only in the re-normalized case. For a large class of models, $\boldsymbol{m}^\circ_\Theta = \boldsymbol{0}$ such that the only distortion happens due to $\mathbf{W}^\circ_\Theta \neq \mathbf{0}$. A suitable chosen matrix norm may therefore yield a measure $\|\mathbf{W}^\circ_\Theta\|$ of the implied distortion.

Proof of (6.98)–(6.103). All the six LLRs follow from (6.67)–(6.72) the first three of which belong to Theorem 6.4, by applying Equations (6.13) and (6.16), which we repeat here for convenience:

$$\nu_\Theta(\boldsymbol{\theta}) = e^{-\boldsymbol{\theta}^\mathsf{T} \mathbf{W}_\Theta \boldsymbol{\theta}/2 + \boldsymbol{\theta}^\mathsf{T} \mathbf{W}_\Theta \boldsymbol{m}_\Theta}\,, \tag{6.104}$$

$$\int \nu_\Theta(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = \sqrt{\frac{(2\pi)^n}{\det \mathbf{W}_\Theta}}\, e^{\boldsymbol{m}_\Theta^\mathsf{T} \mathbf{W}_\Theta \boldsymbol{m}_\Theta/2}\,. \tag{6.105}$$

For Case (c) we additionally note that

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \nu_\Theta(\boldsymbol{\theta}) = \boldsymbol{m}_\Theta\,, \tag{6.106}$$

and

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \frac{\nu_\Theta(\boldsymbol{\theta})}{\nu^\circ_\Theta(\boldsymbol{\theta})} = (\mathbf{W}_\Theta - \mathbf{W}^\circ_\Theta)^{-1} (\mathbf{W}_\Theta \boldsymbol{m}_\Theta - \mathbf{W}^\circ_\Theta \boldsymbol{m}^\circ_\Theta)\,. \qquad \square$$

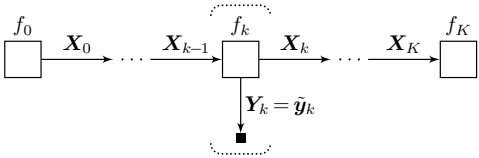


Figure 6.3: A general state-space model (SSM).

6.4 A Family of Local Approximations

Direct implementation of likelihood computation as a in (6.39) may be numerically unstable, because both the numerator as well as the denominator easily tend to extreme values for large factor graphs. For instance, the likelihood of any sequence $\tilde{y}_1, \ldots, \tilde{y}_K$ under the model $Y_k \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$ will decrease with increasing K as long as σ^2 is large enough.

In the following we formalize a method for computing a localized approximation to the likelihood, which may be especially attractive for SSMs of the general form as shown in Figure 6.3. We start by realizing that all the update rules in Tables 6.2–6.4 have a multiplicative general form. We conclude that in any SSM that is assembled from nodes appearing in these tables the following recursion can be formulated:

$$\log \overrightarrow{\beta}_{X_k} = \alpha_k + \log \overrightarrow{\beta}_{X_{k-1}}\,, \tag{6.107}$$

where α_k is the logarithm of the factor corresponding to all the update rules that have been applied to pass the scale factor from $\boldsymbol{X}_{k-1}$ to $\boldsymbol{X}_k$. For example, assume that

$$f_k(\boldsymbol{x}_k, \boldsymbol{y}_k, \boldsymbol{x}_{k-1}) = p(\boldsymbol{x}_k, \boldsymbol{y}_k | \boldsymbol{x}_{k-1}) \tag{6.108}$$

is a conditional PDF. Then the update rule (I.3) tells us that $\alpha_k = \log \overrightarrow{p}_{\mathsf{p}Y}(\tilde{\boldsymbol{y}}_k)$ where $\overrightarrow{p}_{\mathsf{p}Y}(\tilde{\boldsymbol{y}}_k)$ is the prediction PDF evaluated at $\tilde{\boldsymbol{y}}_k$. A similar recursion as in Equation (6.107) may be formulated for the γ-type scale factors.

If the values $\overrightarrow{\beta}_{X_k}$ in (6.107) tend to infinity or zero as k increases we might want to compute a better behaved localized approximation $\overrightarrow{\underset{\sim}{\beta}}_{X_k}$ thereof, which we define by the recursion

$$\log \overrightarrow{\underset{\sim}{\beta}}_{X_k} = \alpha_k + \gamma \log \overrightarrow{\underset{\sim}{\beta}}_{X_{k-1}}\,, \tag{6.109}$$

where γ is a forgetting factor with $0 \leq \gamma \leq 1$, $\gamma \approx 1$. Likewise an approximation $\underset{\sim}{\overleftarrow{\beta}}_{X_{k-1}}$ of $\overleftarrow{\beta}_{X_{k-1}}$ can be defined by the recursion

$$\log \underset{\sim}{\overleftarrow{\beta}}_{X_{k-1}} = \alpha_{k-1} + \gamma \log \underset{\sim}{\overleftarrow{\beta}}_{X_k} , \tag{6.110}$$

and similar approximations can be considered for the γ-type scale factors.

Note that in the example of Equation (6.108) we have $\zeta = 1$ and the proposed approximation (6.109) is exactly equivalent with the "signal class likelihood filter" in [64].

More generally, the approximations (6.109) and (6.110) can be applied in forward and backward message passing for both the factor graph with plugged-in observations and without. Then, in analogy to (6.39) the ratio

$$\underset{\sim}{p}_k(\tilde{\boldsymbol{y}}) \triangleq \underset{\sim}{\beta}_{X_k} / \underset{\sim}{\beta}^{\circ}_{X_k} \tag{6.111}$$

can be computed for every k.

When applying the procedure proposed by Equations (6.109)–(6.111), we obtain a different value for every k instead of one single value (the likelihood). Put differently, we implicitly have constructed a model family, parametrized by k. Note that this model family is not the same as the one proposed in Chapter 7.

Finally we remark that the concept of this proposed procedure is not confined to approximating likelihoods (6.37). It may as well be applied to likelihood functions, LLRs or other quantities related to scale factors of sum-product messages.

6.5 Views on Parameter Selection

Assume that we have a statistical model $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{u})$ or $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$, where $\boldsymbol{X}$ are the hidden variables, $\boldsymbol{Y}$ are the observable variables, and $\boldsymbol{u}$ is a parameter vector or another vector hidden variables. We are interested in estimating some of the hidden variables given the observations $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$. If we have no doubts about our model, then we may plug in the ML estimate

$$\hat{\boldsymbol{u}}_{\mathrm{ML}} = \underset{\boldsymbol{u}}{\operatorname{argmax}}\, p(\tilde{\boldsymbol{y}}|\boldsymbol{u}) , \tag{6.112}$$

or the MAP estimate

$$\hat{\boldsymbol{u}}_{\mathrm{MAP}} = \operatorname*{argmax}_{\boldsymbol{u}} p(\boldsymbol{u}|\tilde{\boldsymbol{y}})\,, \tag{6.113}$$

in order to make an estimate of any of the hidden variables based on $p(\boldsymbol{x}, \tilde{\boldsymbol{y}}|\hat{\boldsymbol{u}}_{\mathrm{ML}})$ or $p(\boldsymbol{x}, \tilde{\boldsymbol{y}}, \hat{\boldsymbol{u}}_{\mathrm{MAP}})$ respectively.

We may however be in a situation where potentially (parts of) $\boldsymbol{u}$ and possibly some of the connected hidden variables should be "switched off", i.e., set to a default value, e.g., to zero. Given the observation $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ we would like select whether to use or not to use certain hidden variables, thus potentially reducing the effective number of variables in our model. In other words, we are not sure about our model and would like to regularize it in a data-dependent way.

There exist many approaches to address such model selection problems in general [5, 96, 100]. In this section we propose two variants of attacking this problem locally in a factor graph. One variant is a Bayesian approach with the idea introduce an (additional) prior PDF on $\boldsymbol{u}$ and estimate the parameters of this prior from the data. In the literature this approach is known as the "empirical Bayes method" [90]. Variations of this principle can be taken into extremes (see, e.g., [80] and references therein), but message passing might not be feasible anymore. In the other variant we consider a binary hypothesis testing problem. Although in general different, both variants coincide in special cases. In these cases, the methods presented here are equivalent with a simple form of a "relevance vector machine" [7, 104].

6.5.1 A Bayesian View

In the following we discuss an approach to regularize the model by introducing (additional) prior information about $\boldsymbol{u}$. In case our model is $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{u})$, i.e. if $\boldsymbol{u}$ is a parameter vector, we introduce a prior PDF $p(\boldsymbol{u}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a parameter of the prior PDF. Out statistical model has now changed to

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}|\boldsymbol{\theta}) = p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{u})\, p(\boldsymbol{u}|\boldsymbol{\theta})\,. \tag{6.114}$$

In case our model is $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$, i.e. if $\boldsymbol{U}$ is a vector of hidden variables, we also introduce a prior PDF $p(\boldsymbol{u}|\boldsymbol{\theta})$ but we change our statistical model to

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{\theta}) \propto p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})\, p(\boldsymbol{u}|\boldsymbol{\theta})\,. \tag{6.115}$$

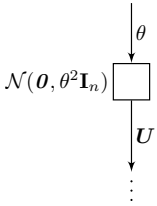


Figure 6.4: Augmenting a variable by a Gaussian prior.

Now we can envisage making an ML or a MAP estimate of $\boldsymbol{\theta}$ as

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \operatorname*{argmax}_{\boldsymbol{\theta}} p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta})\,, \tag{6.116}$$

or

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \operatorname*{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\tilde{\boldsymbol{y}})\,, \tag{6.117}$$

and a corresponding estimate of $\boldsymbol{U}$ as

$$\hat{\boldsymbol{u}} = \operatorname*{argmax}_{\boldsymbol{u}} p\big(\boldsymbol{u}\big|\tilde{\boldsymbol{y}}, \hat{\boldsymbol{\theta}}\big)\,. \tag{6.118}$$

Note that, if our statistical model is $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{\theta})$ then it might be more correct to integrate over $\boldsymbol{\theta}$ instead of plugging in the MAP estimate.

The approach sketched here can be viewed as a general principle: If our model has many parameters or variables and we are not sure whether over-fitting occurs, then we can augment the model by assuming an (additional) prior PDF on the parameters or variables. The changed model will lead to different (and maybe more appropriate) estimates. Interestingly, the changed model, despite having an increased number of variables, can lead to fewer variables actually being used.

We henceforth restrict the discussion to the Gaussian case. Specifically, we consider a Gaussian factor graph that represents some statistical model $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{u})$ or $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$. Furthermore, we assume that $\boldsymbol{U} \in \mathbb{R}^n$ is represented by a single edge. Let us now change this graph (and therewith the represented model) by introducing a factor

$$p(\boldsymbol{u}|\theta) \triangleq \mathcal{N}\big(\boldsymbol{u}\big|\,\boldsymbol{0}, \theta^2 \mathbf{I}_n\big) \tag{6.119}$$

connected to this edge as depicted in Figure 6.4. For the variable $\boldsymbol{U}$, this factor is a prior PDF, parameterized by θ.

First, note that in Section 2.5 such a factor has been named a regularization factor, and θ^2 is the regularization parameter. Concrete examples have been described in Section 2.5 (Figure 2.5) or Section 3.3.4 (Figure 3.9). The problem at hand can therefore be viewed as an approach to estimating the regularization parameter.

Second, note that estimation of θ^2 amounts to variance estimation as discussed in Section 4.5.1 with the sole difference that here we restrict the covariance matrix to be a scaled identity matrix. Therefore, in principle, the methods discussed there may be adapted to solve the problem at hand. In reverse, methods proposed here may, in principle, be used to estimate diagonal covariance matrices.

To characterize the solution to (6.116) and (6.117) in the setting of Figure 6.4, we consider the eigenvalue decomposition of $\overleftarrow{\mathbf{V}}_U$ as

$$\overleftarrow{\mathbf{V}}_U = \mathbf{Q}\,\operatorname{diag}(\boldsymbol{\lambda})\,\mathbf{Q}^\mathsf{T}\,, \quad \boldsymbol{\lambda} \triangleq (\lambda_1, \ldots, \lambda_n)\,. \tag{6.120}$$

In the above, $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of $\overleftarrow{\mathbf{V}}_U$ and $\mathbf{Q}$ contains the eigenvectors. Furthermore we define

$$\boldsymbol{m} \triangleq \mathbf{Q}^\mathsf{T}\overleftarrow{\boldsymbol{m}}_U\,. \tag{6.121}$$

With these definitions, in case our original model is $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{u})$, the log-likelihood function corresponding to (6.116) can be written as

$$\ln p(\tilde{\boldsymbol{y}}|\theta) \propto \sum_{i=1}^{n} \left(-\ln\bigl(\lambda_i + \theta^2\bigr) - \frac{m_i^2}{\lambda_i + \theta^2} \right) + \text{const}\,. \tag{6.122}$$

In case our original model is $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$, the log-marginal corresponding to (6.117) results in the same expression

$$\begin{aligned} \ln p(\theta|\tilde{\boldsymbol{y}}) &= \ln p(\tilde{\boldsymbol{y}}, \theta) + \text{const} && (6.123)\\ &\propto \sum_{i=1}^{n} \left(-\ln\bigl(\lambda_i + \theta^2\bigr) - \frac{m_i^2}{\lambda_i + \theta^2} \right) + \text{const}'\,. && (6.124) \end{aligned}$$

up to an additive constant.

Proof of (6.122) *and* (6.124). In case our model is $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}|\theta)$ we are in a strictly conditional setting as defined in Section 6.3.1. Thus we can use

Theorem 6.2b, Equation (6.43) to express the log-likelihood function as

$$\ln p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta}) = \ln \beta_U(\boldsymbol{\theta}) - \ln \beta_U^\circ \tag{6.125}$$

$$\propto -\ln\det\Big(\overleftarrow{\mathbf{V}}_U + \theta^2\mathbf{I}_n\Big) - \overleftarrow{\boldsymbol{m}}_U^\mathsf{T}\Big(\overleftarrow{\mathbf{V}}_U + \theta^2\mathbf{I}_n\Big)^{-1}\overleftarrow{\boldsymbol{m}}_U + \text{const}\,, \tag{6.126}$$

where, in the second equality, we have used the update rule (III.3). Next we note that using the eigenvalue decomposition (6.120) we can write

$$\overleftarrow{\mathbf{V}}_U + \theta^2\mathbf{I}_n = \mathbf{Q}\ \mathrm{diag}(\boldsymbol{\lambda})\,\mathbf{Q}^\mathsf{T} + \theta^2\mathbf{Q}\mathbf{Q}^\mathsf{T} \tag{6.127}$$

$$= \mathbf{Q}\,\mathrm{diag}\big(\boldsymbol{\lambda} + \theta^2\boldsymbol{1}\big)\,\mathbf{Q}^\mathsf{T}\,, \tag{6.128}$$

$$\Big(\overleftarrow{\mathbf{V}}_U + \theta^2\mathbf{I}_n\Big)^{-1} = \mathbf{Q}\,\mathrm{diag}\big(\boldsymbol{\lambda} + \theta^2\boldsymbol{1}\big)^{-1}\,\mathbf{Q}^\mathsf{T}\,, \tag{6.129}$$

$$\det\Big(\overleftarrow{\mathbf{V}}_U + \theta^2\mathbf{I}_n\Big) = \prod_{i=1}^{n}\big(\lambda_i + \theta^2\big)\,. \tag{6.130}$$

Plugging (6.129), (6.130), and the definition (6.121) into (6.125) the log-likelihood function of Equation (6.122) follows straightforwardly.

In case our model is $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \theta)$, we realize that we can write the marginal as

$$p(\tilde{\boldsymbol{y}}, \theta) = \beta_U(\theta)/\beta_U^\circ\,. \qquad \square$$

In the case in which our original model is $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$, i.e., the augmented model is $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \theta)$, we can consider re-normalizing the model as defined in Sectionsec:normFactAndFunc to a conditional PDF $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}|\theta)$. In this case, Equation (6.41) of Theorem 6.2b can be used in a similar way as in the above proof to show that the log-likelihood function in this case is

$$\ln p(\tilde{\boldsymbol{y}}|\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(\ln\big(\lambda_i^\circ + \theta^2\big) + \frac{m_i^{\circ 2}}{\lambda_i^\circ + \theta^2} - \ln\big(\lambda_i + \theta^2\big) - \frac{m_i^2}{\lambda_i + \theta^2}\right). \tag{6.131}$$

In general, maximizing the log-likelihood function (6.122) or (6.131) or the log-marginal (6.124) is not feasible analytically if $n > 1$. In principle, the iterative approaches detailed in Section 4.2 can be applied to approximately solve this problem. In addition, we provide three alternatives here, first for the general log-likelihood function of Equation (6.122):

a) We can try to find the maximum of (6.122) by solving

$$\sum_{i=1}^{n} \frac{\theta(\theta^2 + \lambda_i - m_i^2)}{\left(\lambda_i + \theta^2\right)^2} \stackrel{!}{=} 0 \tag{6.132}$$

numerically, e.g., by Newtons method. The solution $\theta = 0$ to (6.132) is a maximum if

$$\sum_{i=1}^{n} \frac{m_i^2 - \lambda_i}{\lambda_i^2} < 0\,. \tag{6.133}$$

b) We can plug in a fixed value $\tilde{\theta}$ into the denominator of (6.132). Using the definition

$$s_i \triangleq \left(\lambda_i + \tilde{\theta}^2\right)^{-2} \prod_{j=1}^{n} \left(\lambda_j + \tilde{\theta}^2\right)^2, \tag{6.134}$$

we can write the solution to (6.132) as

$$\hat{\theta}^2 = \frac{\sum_{i=1}^{n} s_i \left(m_n^2 - \lambda_n\right)}{\sum_{i=1}^{n} s_i} \tag{6.135}$$

if this value is positive. Otherwise we may choose $\hat{\theta} = 0$.

c) We can approximate the vector case ($n > 1$) by an average of scalar cases as

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \sum_{i=1}^{n} \left(-\ln\left(\lambda_i + \theta^2\right) - \frac{m_i^2}{\lambda_i + \theta^2} \right) \tag{6.136}$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \operatorname*{argmax}_{\theta_i} \left(-\ln\left(\lambda_i + \theta_i^2\right) - \frac{m_i^2}{\lambda_i + \theta_i^2} \right). \tag{6.137}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \hat{\theta}_i\,, \tag{6.138}$$

where

$$\hat{\theta}_i = \begin{cases} m_i^2 - \lambda_i & \text{if } m_i^2 > \lambda_i \\ 0 & \text{otherwise.} \end{cases} \tag{6.139}$$

The proof of the last two equations follows shortly.

In the re-normalized case, for which the log-likelihood function is given in (6.131), equivalent approaches can be followed resulting in similar equations as Equations (6.132)–(6.137). (Equations (6.138) and (6.139) have no equivalent in this case.)

In the scalar case ($n = 1$) there exists a simple analytic solution to finding the maximum $\hat{\theta}_{\mathrm{ML}}$ or $\hat{\theta}_{\mathrm{MAP}}$ of (6.124) or (6.124) respectively. The solution is given by

$$\hat{\theta}^2 = \begin{cases} \overleftarrow{m}_U^2 - \overleftarrow{\sigma}_U^2 & \text{if } \overleftarrow{m}_U^2 > \overleftarrow{\sigma}_U^2\,, \\ 0 & \text{else,} \end{cases} \tag{6.140}$$

$$\hat{u} = \begin{cases} \dfrac{\overleftarrow{m}_U^2 - \overleftarrow{\sigma}_U^2}{\overleftarrow{m}_U} & \text{if } \overleftarrow{m}_U^2 > \overleftarrow{\sigma}_U^2\,, \\ 0 & \text{else,} \end{cases} \tag{6.141}$$

where $\hat{\theta}$ is either an ML estimate or a MAP estimate.

Proof of (6.140) *and* (6.141). For the scalar case, Equation (6.132) translates to

$$\theta\big(\theta^2 + \overleftarrow{\sigma}_U^2 - \overleftarrow{m}_U^2\big) \stackrel{!}{=} 0\,. \tag{6.142}$$

We consider the case $\overleftarrow{\sigma}_U^2 > \overleftarrow{m}_U^2$. Assuming $\theta \neq 0$, Equation (6.142) results in $\theta^2 = \overleftarrow{m}_U^2 - \overleftarrow{\sigma}_U^2 < 0$, which is a contradiction. Therefore the only remaining solution is $\theta = 0$. In the alternative case $\overleftarrow{\sigma}_U^2 \leq \overleftarrow{m}_U^2$ the condition (6.133) for $\theta = 0$ to be a maximum of (6.132) translates to $\overleftarrow{\sigma}_U^2 > \overleftarrow{m}_U^2$, which contradicts our assumption. Hence in this case we must have $\theta^2 = \overleftarrow{m}_U^2 - \overleftarrow{\sigma}_U^2$. The estimate $\hat{u}$ is zero for $\overleftarrow{\sigma}_U^2 \geq \overleftarrow{m}_U^2$. For the case $\overleftarrow{\sigma}_U^2 < \overleftarrow{m}_U^2$ the first case in (6.141) results directly from message update rules by noting that $\overrightarrow{\sigma}_U^2 = \theta^2$ and $\overrightarrow{\mu}_U = 0$. □

Note the discontinuous behavior in (6.140) and (6.141) that tells us in which cases the variable U should be "switched off". We conjecture that this behavior carries over to the vector case ($n > 1$) for both log-likelihood functions (6.122) and (6.131), because in the corresponding derivative (6.132) and in the equivalent derivative in the re-normalized case, $\theta = 0$ is a tentative solution.

The soft version of this discontinuous behavior of (6.140) can be directly observed in the expectation maximization (EM) update (4.76) for (scalar) variance estimation: By focusing on the last term in the numerator and in the denominator of (4.76), we see that $\hat{s}$ decreases if $\overleftarrow{m}_{U_k}^2 - \overleftarrow{\sigma}_{U_k}^2 < \hat{s}_{\mathrm{old}}$.

6.5.2 A Hypothesis Testing View

Recall that our general objective is to decide based on fixed observations $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$, whether to "switch off" the parameter $\boldsymbol{u}$. In this section we define this vague statement by adopting a binary hypothesis testing view in which the hypotheses of interest are $\mathcal{H}_1\colon \boldsymbol{u} \neq \boldsymbol{0}$ versus $\mathcal{H}_0\colon \boldsymbol{u} = \boldsymbol{0}$ of Theorem 6.4a, or $\mathcal{H}_1\colon \boldsymbol{u} = \hat{\boldsymbol{u}}_{\mathrm{ML}}$ versus $\mathcal{H}_0\colon \boldsymbol{u} = \boldsymbol{0}$ of Theorem 6.4c.

In contrast to the previous section we are not in a Bayesian setting now, and hence there is no need to introduce a new parameter. Likewise, we refrain from making an estimate (6.118) of $\boldsymbol{u}$. Instead we first perform a LLR test

$$\mathrm{LLR} \triangleq \ln \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_1)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_0)} \overset{?}{\gtrless} \vartheta \tag{6.143}$$

and accept $\mathcal{H}_1$ if the threshold ϑ is exceeded. Then we select

$$\hat{\boldsymbol{u}} = \begin{cases} \hat{\boldsymbol{u}}_{\mathrm{ML}} \text{ or } \hat{\boldsymbol{u}}_{\mathrm{MAP}} & \text{if } \mathcal{H}_1 \text{ is accepted,} \\ \boldsymbol{0} & \text{otherwise,} \end{cases} \tag{6.144}$$

where $\hat{\boldsymbol{u}}_{\mathrm{ML}}$ applies if our model is a conditional PDF $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{u})$ and $\hat{\boldsymbol{u}}_{\mathrm{MAP}}$ applies if our model is a joint PDF $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$.

We recall that in the Gaussian case, the LLR and the GLLR as given in (6.98) and (6.100) amount to

$$\mathrm{LLR} = \overleftarrow{\boldsymbol{m}}_U^{\mathsf{T}} \overleftarrow{\mathbf{W}}_U \overleftarrow{\boldsymbol{m}}_U / 2 + \mathrm{const}\,. \tag{6.145}$$

In the scalar case the corresponding LLR test is

$$\overleftarrow{m}_U^2 / \overleftarrow{\sigma}_U^2 \overset{?}{\gtrless} \vartheta' \tag{6.146}$$

where we have absorbed constant terms into the detection threshold ϑ'. Surprisingly, the case distinction in (6.140) in the Bayesian view of the previous section is the same as (6.146) with $\vartheta = 1$.

We clearly see now that in the scalar case, ML estimation of the parameter θ of the additional Gaussian prior (6.119) implicitly solves a detection problem with a fixed threshold. Note that a fixed threshold does not automatically imply a fixed false alarm probability.

In the vector case, however, the two approaches diverge. In contrast to the analytic expression for the LLR in Equation (6.145), the log-likelihood

function (6.122) or the log-marginal (6.124) of θ for the prior cannot be maximized in closed form.

By using the eigenvalue decomposition (6.120) of $\overleftarrow{\mathbf{V}}_U$ and the definition (6.121) as in the previous section, the LLR test (6.101) and the GLLR test (6.103) can be written as

$$\sum_{i=1}^{n} m_i^2 \lambda_i \overset{?}{\gtrless} \vartheta \,. \tag{6.147}$$

There does not seem to be any direct connection between the left-hand side of this equation and expressions in the previous section.

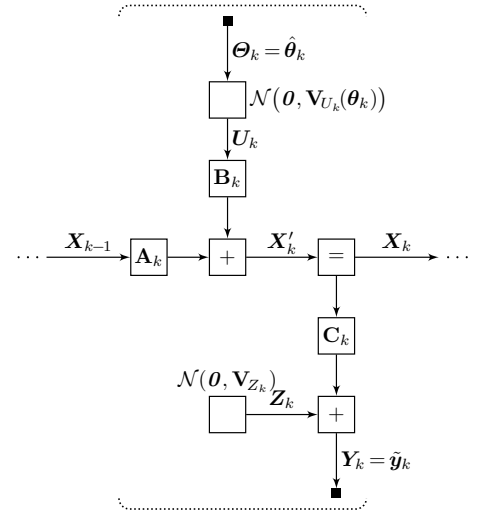
We summarize the findings in the following remarks:

- Both, the Bayesian and the detection-based viewpoint allow us to select which of the hidden variables to set to zero, dependent solely on the values of the observations.
- In the scalar case, both are equivalent as long as we refrain from re-normalizing the model. In the Bayesian model, however, the corresponding detection threshold is blindly set to 1.
- In the vector case and in case we re-normalize a joint PDF, the two methods are different. ML estimation in a Bayesian model is not anymore feasible analytically in the vector case. On the other hand, the test statistic for the detection problem can easily be computed.
- For an actual application of detection-based parameter selection we are lacking a general method to compute the threshold for a given false alarm probability. This issue will not be addressed in this thesis.

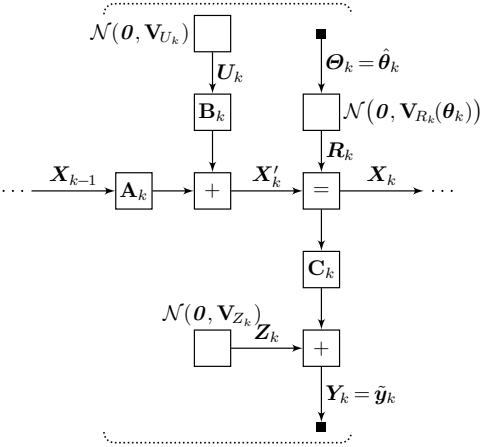
Finally, we point towards two different generic ways of applying the two parameter selection approaches to the general linear SSM of Equation (3.1), Figure 3.1:

a) Parameter selection can be applied to the input $\boldsymbol{U}_k$ of a SSM. Specifically, we propose the model of Figure 6.5a, in which the covariance matrix $\mathbf{V}_{U_k}(\boldsymbol{\theta}_k)$ is parameterized by a vector $\boldsymbol{\Theta}_k$.

b) Parameter selection can be applied to the state vector $\boldsymbol{X}_k$ of a SSM. Specifically, we propose the model of Figure 6.5b, in which we have



(a)



(b)

Figure 6.5: Two potential applications of parameter selection.

introduced a (distributed) regularization factor $\mathcal{N}\big(\boldsymbol{0}, \mathbf{V}_{R_k}(\boldsymbol{\theta}_k)\big)$ whose covariance matrix is parameterized by a vector $\boldsymbol{\Theta}_k$. (Cf. Section 2.5 for the notion of distributed regularization.)

In both applications an example parameterization of the covariance matrix in terms of the parameter vector $\boldsymbol{\Theta}_k$ could be block diagonal $\mathbf{V}_{U_k} = \operatorname{diag}(\boldsymbol{\theta}_k)$ or $\mathbf{V}_{R_k} = \operatorname{diag}(\boldsymbol{\theta}_k)$. (As an alternative to assuming structured covariance matrices, state-space splitting may be used.) In both applications, we envisage estimation of the parameter vector $\boldsymbol{\Theta}_k$ in a ML sense (the Bayesian view) or by solving a detection problem (the hypothesis testing view).

Clearly, for processing a whole block of observed data $\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_K$, the joint estimate $\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_K$ cannot be obtained easily and we would have to resort, e.g., to cyclic maximization (CM). For online processing, we can consider estimating only $\boldsymbol{\Theta}_K$ for the currently last arrived data item $\tilde{\boldsymbol{y}}_K$.

As a result we would expect sparse input estimates for Application (a) and sparse state estimates for Application (b), where sparsity is understood in both the dimension of the input or the state as well as in time. Apparently, the application of a relevance vector machine [7] to the input of a SSM also leads to sparse state estimates [52].

Application (a) is used in the third example in Section 6.5.3 in the simplistic setting of a scalar SSM. An interesting example of Application (b) would be an extension to the model for quasi-periodic signals (cf. Section 4.4). In this extension we define

$$\mathbf{V}_{R_k}(\boldsymbol{\theta}_k) = \begin{bmatrix} [\theta_k]_1 \mathbf{I}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & [\theta_k]_2 \mathbf{I}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & [\theta_k]_M \mathbf{I}_2 \end{bmatrix}, \tag{6.148}$$

where we recall that M is the number of harmonics. In every time step k we thus allow the model to "switch off" any of the harmonics. As an alternative to the structure proposed in (6.148) we can envisage state-space splitting as in Section 3.5.

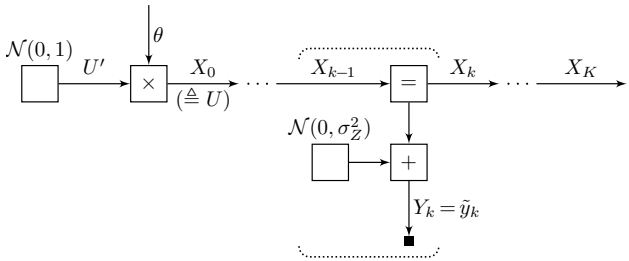


Figure 6.6: DC signal in noise.

6.5.3 Examples

In the following, We give three rather simple examples, all concerned with estimating scalar quantities. Recall that in this case the Bayesian approach has a direct connection with a binary hypothesis testing problem. In all examples the global function of the factor graph is a conditional PDF. Hence, the likelihood function of (6.122) applies.

DC (Direct Current) Signal in Noise

Consider the signal model

$$Y_k = \theta\, U' + Z_k \tag{6.149}$$

for $k = 1, \ldots, K$, where $U' \sim \mathcal{N}(0,1)$ and $Z_1, \ldots, Z_K \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_Z^2)$. The parameter θ is unknown. The factor graph of this model is shown in Figure 6.6. Given observations $Y_k = \tilde{y}_k$ the ML estimate of θ is

$$\hat{\theta}^2_{\text{ML}} = \begin{cases} \bar{y}^2 - \sigma_Z^2/K & \text{if } \bar{y}^2 > \sigma_Z^2/K, \\ 0 & \text{else,} \end{cases} \tag{6.150}$$

$$\hat{u}_{\text{MAP}} = \begin{cases} \bar{y}^2 - \sigma_Z^2/(K\bar{y}) & \text{if } \bar{y}^2 > \sigma_Z^2/K, \\ 0 & \text{else,} \end{cases} \tag{6.151}$$

where

$$\bar{y} \triangleq \frac{1}{K} \sum_{k=1}^{K} \tilde{y}_k \,. \tag{6.152}$$

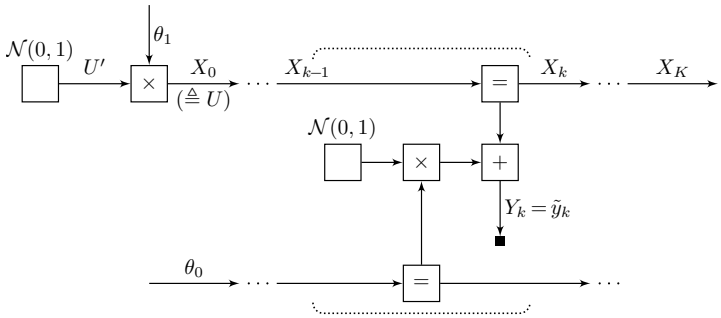


Figure 6.7: DC signal in noise with unknown variance.

Note that $\bar{y}^2$ is the standard test statistic for the detection problem with hypotheses $u \neq 0$ versus $u = 0$ [54]. Equations (6.150) and (6.151) follow directly from (6.140) and (6.141) by noting that $\overleftarrow{m}_U = \bar{y}$ and $\overleftarrow{\sigma}_U^2 = \sigma_Z^2/K$.

DC (Direct Current) Signal in Noise With Unknown Variance

Consider the signal

$$Y_k = \theta_1 U' + \theta_0 Z_k \tag{6.153}$$

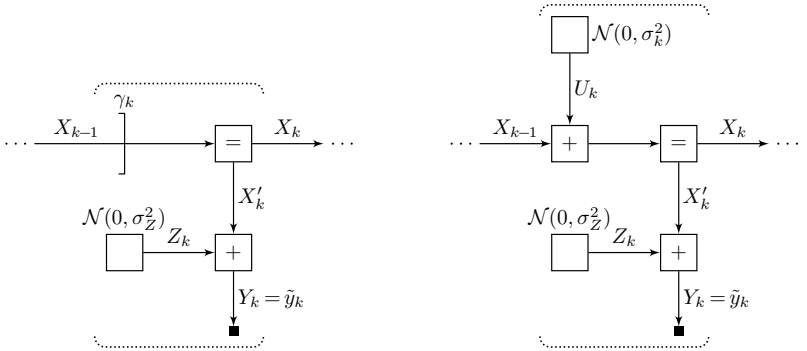
for $k = 1, \ldots, K$, where $U' \sim \mathcal{N}(0,1)$ and $Z_1, \ldots, Z_K \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$. The parameters $\boldsymbol{\theta} \triangleq (\theta_0, \theta_1)$ are unknown. The factor graph of this model is shown in Figure 6.7. Given observations $Y_k = \tilde{y}_k$ the ML estimate of $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = \begin{cases} \left(\sqrt{\dfrac{K(\check{y}^2 - \bar{y}^2)}{K-1}}, \sqrt{\dfrac{K\bar{y}^2 - \check{y}^2}{K-1}} \right) & \text{if } \dfrac{\bar{y}^2}{\check{y}^2} > \dfrac{1}{K}\,, \\ (\check{y}^2, 0) & \text{else,} \end{cases} \tag{6.154}$$

$$\hat{u}_{\text{MAP}} = \begin{cases} \dfrac{K\bar{y}^3 - \bar{y}\check{y}^2}{K\bar{y}^2} & \text{if } \dfrac{\bar{y}^2}{\check{y}^2} > \dfrac{1}{K}\,, \\ 0 & \text{else,} \end{cases} \tag{6.155}$$

where

$$\check{y}^2 \triangleq \frac{1}{K}\sum_{k=1}^{K} \tilde{y}_k^2\,,\ \bar{y} \triangleq \frac{1}{K}\sum_{k=1}^{K} \tilde{y}_k\,. \tag{6.156}$$



(a) Forgetting factor (b) Random walk

Figure 6.8: DC (direct current) value with relaxed constraints.

The term $\bar{y}^2/\check{y}^2$ is the standard test statistic for detecting a DC constant in noise [54]. Equations (6.154) and (6.155) can be proved by expressing the log-likelihood function analytically in terms of the messages on edge U, taking the derivative and setting to zero.

Estimation of a Forgetting Factor

In the following we formulate an online algorithm for estimating a forgetting factor in a scalar SSM. For simplicity we assume that the model is a noisily observed DC constant. We relax the constant by means of a time-varying forgetting factor γ_k. The resulting model is shown in Figure 6.8a.

For online estimation we assume that the observed data $\tilde{y}_1, \tilde{y}_2, \ldots$ arrives in a stream. Let K be the time index of our newly arrived data item $Y_K = \tilde{y}_K$. We recall that in the present online setting $\overleftarrow{\mu}_{X_K}(x_K) = 1$ is neutral. Hence, our model of Y_K is

$$Y_K = X'_K + Z_K \,, \tag{6.157}$$

where $X'_K \sim \overrightarrow{\mu}^{\gamma_K}_{X_{k-1}}$ is the message forward from the past with already applied forgetting factors. We now ask, which forgetting factor does best fit the new data item $\tilde{y}_K$ given all the past data.

In [32], the following adaptive forgetting factor estimate is proposed:

$$\hat{\gamma}'_K = \left(\frac{\vec{\mu}_{Y_K}(\tilde{y}_K)}{\vec{\mu}_{Y_K}(\vec{m}_{Y_K})} \right)^{(1/\rho)}, \tag{6.158}$$

where $\rho \in \mathbb{R}_{>0}$ is a parameter of the estimate and where $\vec{\mu}_{Y_K}$ is computed without forgetting factor. In the present Gaussian setting we get

$$\hat{\gamma}'_K = \exp\left(\frac{-(\tilde{y}_K - \vec{m}_{X_{K-1}})^2}{2\rho(\vec{\sigma}^2_{X_{K-1}} + \sigma_Z^2)} \right). \tag{6.159}$$

We contrast this method with a different approach. In this approach we replace the forgetting factor γ_k by an equivalent state noise $U_k \sim \mathcal{N}(0, \sigma_k^2)$, which results in the random walk model depicted in Figure 6.8b. For this online scenario, this equivalence is established as

$$\vec{\sigma}^2_{X'_k} = \vec{\sigma}^2_{X_{k-1}} + \sigma_k^2 = \vec{\sigma}^2_{X_{k-1}} / \gamma_k \,, \tag{6.160}$$

$$\gamma_k = \frac{\vec{\sigma}^2_{X_{k-1}}}{\vec{\sigma}^2_{X_{k-1}} + \sigma_k^2} \,. \tag{6.161}$$

At time step K we now do ML estimation of σ_K^2 and deduce the equivalent forgetting factor estimate. Using (6.140), we formulate $\overleftarrow{\mu}_{U_K}$ as

$$\overleftarrow{\sigma}^2_{U_K} = \sigma_Z^2 + \vec{\sigma}^2_{X_{K-1}} \,, \tag{6.162}$$

$$\overleftarrow{m}_{U_K} = \tilde{y}_K - \vec{m}_{X_{K-1}} \tag{6.163}$$

Now we can formulate the ML estimate $\hat{\sigma}_K^2$ as

$$\hat{\sigma}_K^2 = \begin{cases} \left(\tilde{y}_K - \vec{m}_{X_{K-1}}\right)^2 - \sigma_Z^2 - \vec{\sigma}^2_{X_{K-1}} & \text{if } \left(\tilde{y}_K - \vec{m}_{X_{K-1}}\right)^2 > \sigma_Z^2 + \vec{\sigma}^2_{X_{K-1}}, \\ 0 & \text{else.} \end{cases} \tag{6.164}$$

The corresponding ML estimate $\hat{\gamma}_k$ is

$$\hat{\gamma}_K = \begin{cases} \frac{\vec{\sigma}^2_{X_{K-1}}}{(y - \vec{m}_{X_{K-1}})^2 - \sigma_Z^2} & \text{if } (y - \vec{m}_{X_{K-1}})^2 - \sigma_Z^2 > \vec{\sigma}^2_{X_{K-1}}, \\ 1 & \text{else.} \end{cases} \tag{6.165}$$

Figure 6.9 shows an example comparison between the adaptive forgetting factor $\hat{\gamma}'_K$ of Equation (6.159) and forgetting factor $\hat{\gamma}_K$ as well as the

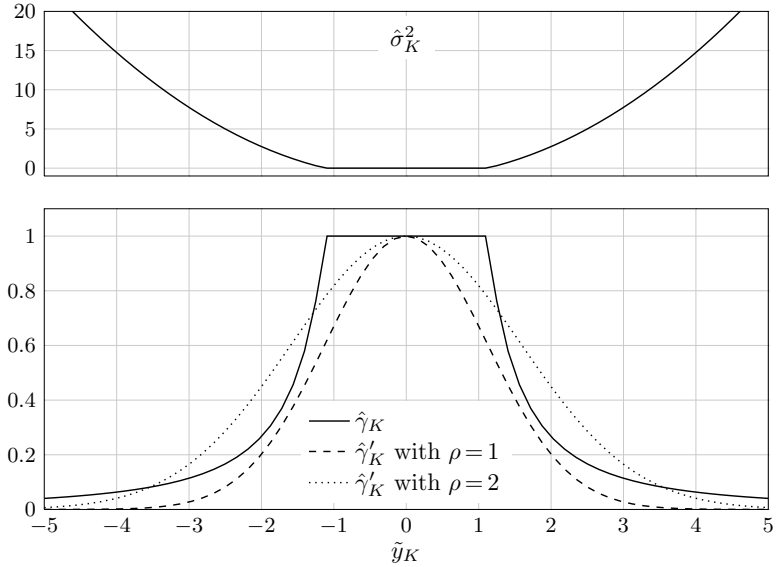


Figure 6.9: Example for noise variance estimation and equivalent forgetting factor estimation for $\vec{\sigma}^2_{X_{K-1}} = 1$, $\vec{m}_{X_{K-1}} = 0$, $\sigma_Z^2 = 0.25$.

equivalent noise variance estimate $\hat{\sigma}_K$ as proposed in Equation (6.165). Clearly in the likelihood-based estimates, there the case distinction between the regime in which the variable U_K is used ($\hat{\sigma}_K^2 > 0$, $\hat{\gamma}_K < 1$) and the regime in which the variable is not used ($\hat{\sigma}_K^2 = 0$, $\hat{\gamma}_K = 1$) is visible, while the simple estimate $\hat{\gamma}'_K$ does not exhibit such behavior.

Chapter 7

Glue Factor

7.1 Introduction

In this chapter we propose a new paradigm for state-space based signal processing, extending the ideas of [29, 64, 87]. Specifically, we do not assume a single factor graph anymore, but instead a whole family of graphs. Each member in this family takes the form of a factor graph in state-space form, i.e., with state variables and observable variables arranged along a time axis, but with one factor inserted at some position into the state space: the *glue factor*. For each member of the family, the glue factor is allowed to "sit" at a different position. Moreover, we allow the glue factor to have parameters.

Switching dynamical systems, a closely related concept to our model family, are a research topic with many applications [15, 36]. Usually, for such systems, a hyper-parameter indicating which dynamical system is active is assumed and Gaussianity is given up even for linear models. We do not take this viewpoint here. Instead we stick to Gaussian models but in return we consider likelihood computation for the purpose of ML estimation and detection.

The modeling of piecewise smooth signals has been studied in wavelet-based approaches, cf. e.g. [30]. In these approaches, however, there exists no apparent underlying statistical model, and depending on the particular choice of wavelets, the SSM view is lost or not possible.

We start this chapter in Section 7.2 by formally defining the model family envisaged. The findings from the previous chapter are used to show how the likelihood or the likelihood function given observations $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$

can be computed for each member in the model family. We name this procedure *likelihood filtering* and we work out the differences for offline (block-based) processing and online processing.

Next, we consider ML estimation of the parameters of the model family. In Section 7.3, we let the glue factor position be fixed and solely treat ML estimation of the glue factor parameters or the formulation of LLRs with respect to these parameters. We consider offline and online learning scenarios for such parameters in a general setting. In Section 7.4 we apply the glue factor view of parameter estimation in the context of array processing [70–72, 88]. Section 7.5 treats the important example of modeling two-sided pulses with a sum of decaying sinusoids. Specifically, we design a glue factor that ensures continuity and smoothness at the glueing position.

While for ML estimation of glue factor parameters the message passing scale factors can be neglected, this is in general not the case for ML estimation of the time position of the glue factor. We start with this topic in Section 7.6 by introducing a LLR based procedure. Subsequently, special cases are identified in which scale factors can be neglected even for this task. This section is concluded by touching upon the problem of estimating several glue factors, and potential algorithms to solve an approximation to this problem.

In Section 7.7 we take a look at a different scenario in which we do not know beforehand, if a glue factor is present at all. We show how we can solve this problem, at least in principle, by means of formulating a statistical detection problem. The scenario once again changes if we want to detect the presence of several glue factors. As before, we only solve an approximation to this problem.

The offline and online algorithms for estimating a glue factor position and for detecting the presence of a glue factor are illustrated by three examples in Section 7.9, one on locating an additional input, one on model change point estimation, and one on locating a sinusoidal pulse for known or unknown pulse shape.

Finally, in Section 7.8 we sketch an online algorithm for detecting sparse changes.

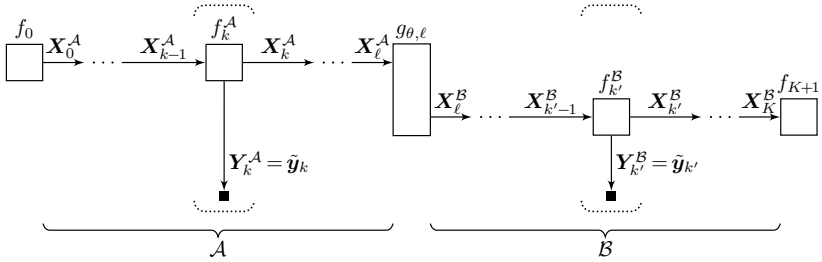


Figure 7.1: General SSM factor graph with a glue factor.

7.2 General Concepts

7.2.1 A Family of Factor Graphs

In a state-space approach, the construction of a statistical model for a given block of data $\tilde{\boldsymbol{y}} \triangleq \tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_K$ usually assumes a single model repeated for each time step $k = 1, \ldots, K$. We break this traditional view by introducing a whole family of models as depicted in Figure 7.1. In this graph, two potentially different SSMs $\mathcal{A}$ and $\mathcal{B}$ are connected at time ℓ by a factor labeled $g_{\theta,\ell}$ – the *glue factor*. This factor is allowed to depend on its position ℓ on the time axis and on an additional parameter or random variable (vector) $\boldsymbol{\theta}$. The slightly abusive notation $g_{\theta,\ell}$ is solely used as a label in the factor graph, the actual factor is $g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell)$.

First, for fixed $\boldsymbol{\theta}$, Figure 7.1 describes a family of $K+1$ factor graphs, one for each glue factor position $\ell = 0, \ldots, K$, all sharing the same observations $\tilde{\boldsymbol{y}}_k$ for $k = 1, \ldots, K$. Second, for a fixed glue factor position ℓ, Figure 7.1 describes a family of factor graphs, parametrized by the parameter $\boldsymbol{\theta}$. We combine the two views in the following definition of the global function

$$
\begin{aligned}
f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}, \ell) \triangleq f_0\big(\boldsymbol{x}_0^{\mathcal{A}}\big) \left(\prod_{k=1}^{\ell} f_k^{\mathcal{A}}\big(\boldsymbol{x}_k^{\mathcal{A}}, \boldsymbol{y}_k^{\mathcal{A}}, \boldsymbol{x}_{k-1}^{\mathcal{A}}\big) \right) g\big(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell, \big) \\
\cdot \left(\prod_{k'=\ell+1}^{K} f_{k'}^{\mathcal{B}}\big(\boldsymbol{x}_{k'}^{\mathcal{B}}, \boldsymbol{y}_{k'}^{\mathcal{B}}, \boldsymbol{x}_{k'-1}^{\mathcal{B}}\big) \right) f_{K+1}\big(\boldsymbol{x}_K^{\mathcal{B}}\big)\,,
\end{aligned}
\tag{7.1}
$$

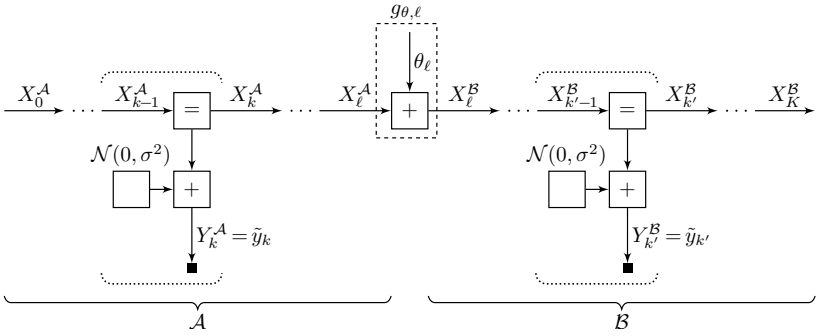


Figure 7.2: Factor graph for Example 7.1: Two DC (direct current) values are connected by a glue factor that models a jump.

where

$$\boldsymbol{X} \triangleq \left(\boldsymbol{X}^{\mathcal{A}}, \boldsymbol{X}^{\mathcal{B}}\right), \quad \boldsymbol{X}^{\mathcal{A}} \triangleq \left(\boldsymbol{X}_0^{\mathcal{A}}, \ldots, \boldsymbol{X}_\ell^{\mathcal{A}}\right), \quad \boldsymbol{X}^{\mathcal{B}} \triangleq \left(\boldsymbol{X}_\ell^{\mathcal{B}}, \ldots, \boldsymbol{X}_K^{\mathcal{B}}\right),$$
$$\boldsymbol{Y} \triangleq \left(\boldsymbol{Y}^{\mathcal{A}}, \boldsymbol{Y}^{\mathcal{B}}\right), \quad \boldsymbol{Y}^{\mathcal{A}} \triangleq \left(\boldsymbol{Y}_1^{\mathcal{A}}, \ldots, \boldsymbol{Y}_\ell^{\mathcal{A}}\right), \quad \boldsymbol{Y}^{\mathcal{B}} \triangleq \left(\boldsymbol{Y}_{\ell+1}^{\mathcal{B}}, \ldots, \boldsymbol{Y}_K^{\mathcal{B}}\right).$$

In general $\boldsymbol{\theta}$ can depend on ℓ, but we refrain from reflecting this in our notation.

Before considering any likelihood-related computation, let us realize that such a model family can yield useful message-passing algorithms. Indeed, estimation of the states $\boldsymbol{X}_\ell^{\mathcal{A}}$, $\boldsymbol{X}_\ell^{\mathcal{B}}$ or of any variable can be done by one forward and one backward pass. We give a simple example to clarify this.

Example 7.1: Estimation of a DC (Direct Current) Jump
Consider the factor graph of Figure 7.2, in which a constant value $X_k^{\mathcal{A}}$ jumps by an amount of θ_ℓ at time ℓ to a different constant value $X_{k'}^{\mathcal{B}}$. Given noisy observations $\tilde{y}_1, \ldots, \tilde{y}_K$ of these constant values, estimation of the difference θ_ℓ for any ℓ can be done by forward message passing in model $\mathcal{A}$, backward message passing in model $\mathcal{B}$, and computing $\hat{\theta}_\ell = \overleftarrow{m}_{\theta_\ell}$ in the glue factor for $\ell = 0, \ldots, K$. ◊

More advanced usage of glue factor models incorporate the computation of likelihood-related quantities. Before describing individual scenarios in which such computations are needed, we recall some concepts from Chapter 6 and apply these to the model family at hand. We start by noting that the factor graph in Figure 7.1 represents either a joint

PDF/PMF

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}, \ell) = f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}, \ell)/\zeta\,, \tag{7.2}$$

where the normalization factor ζ is

$$\zeta = \sum_{\ell=0}^{K} \iiint f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}, \ell)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}\,\mathrm{d}\boldsymbol{\theta}\,, \tag{7.3}$$

or the factor graph represents a conditional PDF.

In this latter case we recall the distinction between the *strictly conditional* case and the *re-normalized case*. By strictly conditional we mean that we can write

$$p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}, \ell) = f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}, \ell)/\zeta^{\boldsymbol{\theta},\ell}\,, \tag{7.4}$$

where

$$\zeta^{\boldsymbol{\theta},\ell} = \iint f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}, \ell)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} \tag{7.5}$$

is a constant normalization factor that depends neither on $\boldsymbol{\theta}$ nor on ℓ, i.e., in this case the function represented by the factor graph is proportional to a conditional PDF.

In the re-normalized case there is no such constant of proportionality and the factor graph actually represents a joint PDF/PMF as in (7.2). We nevertheless can construct a conditional PDF by re-normalization as

$$p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}, \ell) = f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}, \ell)/\zeta(\boldsymbol{\theta}, \ell)\,, \tag{7.6}$$

where

$$\zeta(\boldsymbol{\theta}, \ell) = \iint f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}, \ell)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} \tag{7.7}$$

is a normalization function. This normalization function can be understood as a scaled prior PDF/PMF on $\boldsymbol{\theta}$ and ℓ.

In analogy to the above we can consider the strictly conditional and re-normalized cases of a factor graph representing $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}|\ell)$ or $p(\boldsymbol{x}, \boldsymbol{y}, \ell|\boldsymbol{\theta})$.

In general, the strictly conditional case does not apply for neither $\boldsymbol{\theta}$ nor ℓ and one can be tempted to use re-normalized versions. We emphasize

that re-normalization discards any prior information in the model about $\boldsymbol{\theta}$ and/or ℓ. In many cases, however, it might be desirable to remove any unwanted bias, at least for ℓ.

We recall that Theorem 6.1 allows us to compute normalization factors and normalization functions in terms of messages in the factor graph. Similarly, Theorem 6.2 establishes connections between likelihoods or likelihood functions and messages. Based on these theorems, we can write the various conditional marginals and likelihood functions for given observations $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$.

In the following, $\boldsymbol{\Theta}_\ell$ refers to an edge that represents $\boldsymbol{\theta}$ within the glue factor, when the glue factor sits at position ℓ. Similarly, $\boldsymbol{U}_\ell$ refers to any edge in the factor graph of Figure 7.1, when the glue factor sits at position ℓ. Also, we recall the notation $(\cdot)^\circ$ relating to quantities that are computed without plugging in any observations. These quantities hence can be computed before any observations $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ are revealed.

In the joint case we can write

$$p(\boldsymbol{\theta}, \ell | \tilde{\boldsymbol{y}}) = \mu_{\Theta_\ell}(\boldsymbol{\theta}) / \sum_{j=0}^{K} \beta_{\Theta_j} = \beta_{U_\ell}(\boldsymbol{\theta}) / \sum_{j=0}^{K} \beta_{U_j} \,. \tag{7.8}$$

In all cases we can write

$$p(\tilde{\boldsymbol{y}} | \boldsymbol{\theta}, \ell) = \mu_{\Theta_\ell}(\boldsymbol{\theta}) / \mu^\circ_{\Theta_\ell}(\boldsymbol{\theta}) = \beta_{U_\ell}(\boldsymbol{\theta}) / \beta^\circ_{U_\ell}(\boldsymbol{\theta}) \,, \tag{7.9}$$

where $\mu^\circ_{\Theta_\ell}(\boldsymbol{\theta}) = \gamma^\circ_{\Theta_\ell}$ in those cases that are strictly conditional with respect to $\boldsymbol{\theta}$. Finally, for all cases that apply, we can write

$$p(\ell | \tilde{\boldsymbol{y}}, \boldsymbol{\theta}) = \mu_{\Theta_\ell}(\boldsymbol{\theta}) / \sum_{j=0}^{K} \mu_{\Theta_j}(\boldsymbol{\theta}) = \beta_{U_\ell}(\boldsymbol{\theta}) / \sum_{j=0}^{K} \beta_{U_j}(\boldsymbol{\theta}) \tag{7.10}$$

and

$$p(\boldsymbol{\theta} | \tilde{\boldsymbol{y}}, \ell) = \mu_{\Theta_\ell}(\boldsymbol{\theta}) / \beta_{\Theta_\ell} = \beta_{U_\ell}(\boldsymbol{\theta}) / \beta_{U_\ell} \,. \tag{7.11}$$

In the above $\beta_{U_\ell}(\boldsymbol{\theta})$ is the β-type scale factor on edge $\boldsymbol{U}_\ell$, when regarded as a function of $\boldsymbol{\theta}$. We recall from Theorem 6.1 that the difference between $\beta_{U_\ell}(\boldsymbol{\theta})$ and β_{U_ℓ} is that for the former, $\boldsymbol{\theta}$ is treated as a parameter of some nodes and no integration over $\boldsymbol{\theta}$ is done while for the latter, $\boldsymbol{\Theta}_\ell$ is treated as an edge in the factor graph. The proofs for Equations (7.8)–(7.11) are in Appendix D.1.

If our model family does not feature a parameter or random variable (vector) $\boldsymbol{\theta}$, then (7.8) and (7.9) simplify to

$$p(\ell|\tilde{\boldsymbol{y}}) = \beta_{U_\ell} / \sum_{j=0}^{K} \beta_{U_j} \,, \tag{7.12}$$

$$p(\tilde{\boldsymbol{y}}|\ell) = \beta_{U_\ell} / \beta^{\circ}_{U_\ell} \,, \tag{7.13}$$

where now, by definition, $\boldsymbol{\Theta}$ is no longer an edge in the graph.

Finally, from Theorem 6.4, we recall ways to compute LLRs. Specifically, we fix ℓ and consider the following binary hypothesis tests

$$\text{a)} \qquad \mathcal{H}_1 \colon \boldsymbol{\theta} \neq \boldsymbol{0} \qquad \mathcal{H}_0 \colon \boldsymbol{\theta} = \boldsymbol{0} \,, \tag{7.14}$$

$$\text{b)} \qquad \mathcal{H}_1 \colon \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} \qquad \mathcal{H}_0 \colon \boldsymbol{\theta} = \boldsymbol{0} \,, \tag{7.15}$$

$$\text{c)} \qquad \mathcal{H}_1 \colon \boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_{\mathrm{ML}} \text{ or } \boldsymbol{\theta} = \hat{\boldsymbol{\theta}}_{\mathrm{MAP}} \qquad \mathcal{H}_0 \colon \boldsymbol{\theta} = \boldsymbol{0} \,, \tag{7.16}$$

where $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$ applies in the strictly conditional and the re-normalized cases while $\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}$ applies in the joint case. The corresponding LLRs are

$$\text{a)} \qquad \mathrm{LLR}_\ell = \log \int \nu_{\Theta_\ell}(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \,, \tag{7.17}$$

$$\text{b)} \qquad \mathrm{LLR}_\ell = \log \nu_{\Theta_\ell}(\tilde{\boldsymbol{\theta}}) \,, \tag{7.18}$$

$$\text{c)} \qquad \mathrm{GLLR}_\ell = \log \max_{\boldsymbol{\theta}} \nu_{\Theta_\ell}(\boldsymbol{\theta}) \,, \tag{7.19}$$

where we recall the definition (6.6) of ν as $\nu_{\Theta_\ell}(\boldsymbol{\theta}) = \mu_{\Theta_\ell}(\boldsymbol{\theta}) / \gamma_{\Theta_\ell}$.

7.2.2 Likelihood Filtering

A natural choice for $\boldsymbol{U}_\ell$ in Equations (7.8)–(7.11) is $\boldsymbol{X}_\ell^{\mathcal{A}}$, $\boldsymbol{X}_\ell^{\mathcal{B}}$ in the factor graph of Figure 7.1, or some edge internal to the glue factor. This choice allows us to compute the conditional marginals or likelihood functions in (7.8)–(7.11), or the LLRs in (7.17)–(7.19) by message passing on the factor graph in Figure 7.1 as follows.

Offline Likelihood Computation:

a) Do forward (left-to-right) sum-product message passing in the factor graph of model $\mathcal{A}$, if necessary both with and without plugged in observations.

b) Do backward message (right-to-left) sum-product passing in the factor graph of model $\mathcal{B}$, if necessary both with and without plugged in observations.

c) From $\overrightarrow{\mu}_{\boldsymbol{X}_\ell^{\mathcal{A}}}$, $\overrightarrow{\mu}^{\circ}_{\boldsymbol{X}_\ell^{\mathcal{A}}}$, $\overleftarrow{\mu}_{\boldsymbol{X}_\ell^{\mathcal{B}}}$, and $\overleftarrow{\mu}^{\circ}_{\boldsymbol{X}_\ell^{\mathcal{B}}}$ compute any quantity related to the conditional marginals or likelihood functions (7.8)–(7.11) or the LLRs in (7.17)–(7.19).

Note that alternatively, the computation of $\overrightarrow{\mu}^{\circ}_{\boldsymbol{X}_\ell^{\mathcal{A}}}$ and $\overleftarrow{\mu}^{\circ}_{\boldsymbol{X}_\ell^{\mathcal{B}}}$ can be done offline, before any data is observed. Moreover, note that often these quantities are not needed. Also, the scale factors of the messages can often be neglected, specifically in situations that comply with the setting in Theorem 6.3. In general, however, scale factors must be computed.

At first it seems that the computation of message scale factors can be implemented by means of the prediction rule (I.3) for both forward and the backward message passing. This is, however, only true if (a) each factor $f_k^{\mathcal{A}}$ and $f_{k'}^{\mathcal{B}}$ in Figure 7.1 is proportional to a conditional PDF and if (b) all prediction messages $\overrightarrow{\mu}_{\mathsf{p}Y_k^{\mathcal{A}}}$ and $\overrightarrow{\mu}_{\mathsf{p}Y_{k'}^{\mathcal{B}}}$ are proper, i.e., integrable. In linear SSMs Condition (a) can be violated by including distributed regularization (cf. Section 3.3.4) and Condition (b) is usually not true if the factors f_0 or f_{K+1} are neutral ($\propto 1$). The latter has already been elaborated on in Section 6.2.3, and in this case, the update rules (IV.2)–(IV.3) for composite blocks provide a solution.

For linear SSMs as in Figure 3.1 for both $\mathcal{A}$ and $\mathcal{B}$, one might be tempted to view Steps (a) and (b) of the above algorithm as Kalman smoothing. In the traditional Kalman smoother, however, usually $\mathcal{B} = \mathcal{A}$ and the glue factor is a simple connection of the states. Moreover, we envisage not only state estimation but also likelihood computation.

The filtering view of the offline likelihood computation procedure outlined above is established by considering an online scenario, in which the data to be analyzed is arriving in a stream. The output of this *likelihood filter* is still any quantity related to the conditional marginals or likelihood functions (7.8)–(7.11) or the LLRs in (7.17)–(7.19). But now, for each newly arrived data item $\tilde{\boldsymbol{y}}_K$ (or packet of items), the computation is restricted to $\ell = K - D_1, \ldots, K - D_2$, where K is the time index of the currently last data item and the delay parameters $D_1, D_2 \in \mathbb{N}_0$, $D_1 \leq D_2$ define a slice of past time indices thus identifying glue factor positions ℓ of interest. The following filtering (or rather smoothing) message passing algorithm in the factor graph of Figure 7.1 results.

Online Likelihood Filtering:

a) Increment K and fetch the next data item $\tilde{\boldsymbol{y}}_K$.

b) Compute $\overrightarrow{\mu}_{X^{\mathcal{A}}_{K-D_2}}$ and possibly $\overrightarrow{\mu}^{\circ}_{X^{\mathcal{A}}_{K-D_2}}$ by doing forward (left-to-right) message passing in the factor graph of model $\mathcal{A}$.

c) Compute $\overleftarrow{\mu}_{X^{\mathcal{B}}_{k}}$ and possibly $\overleftarrow{\mu}^{\circ}_{X^{\mathcal{B}}_{k}}$ for $k = K, \ldots, (K - D_1)$ by doing backward (right-to-left) message passing in the factor graph of model $\mathcal{B}$.

d) From $\overrightarrow{\mu}_{\boldsymbol{X}^{\mathcal{A}}_{\ell}}$, $\overrightarrow{\mu}^{\circ}_{\boldsymbol{X}^{\mathcal{A}}_{\ell}}$, $\overleftarrow{\mu}_{\boldsymbol{X}^{\mathcal{B}}_{\ell}}$, and $\overleftarrow{\mu}^{\circ}_{\boldsymbol{X}^{\mathcal{B}}_{\ell}}$ compute any quantity related to the conditional marginals or likelihood functions (7.8)–(7.11) or the LLRs in (7.17)–(7.19) for $\ell = K - D_1, \ldots, K - D_2$.

Go to Step (a).

Note that for Step (d), the messages $\overrightarrow{\mu}_{X^{\mathcal{A}}_{k}}$ and $\overrightarrow{\mu}^{\circ}_{X^{\mathcal{A}}_{k}}$ for $k = (K - D_1), \ldots, (K - D_2 - 1)$ are known from previous filtering steps. The computation of $\overrightarrow{\mu}^{\circ}_{X^{\mathcal{A}}_{K-D_2}}$ and $\overleftarrow{\mu}^{\circ}_{X^{\mathcal{B}}_{k}}$ in Steps (b) and (c) can be done beforehand if the models $\mathcal{A}$ and $\mathcal{B}$ are completely known. Moreover, $\overrightarrow{\mu}^{\circ}_{X^{\mathcal{A}}_{K-D_2}}$ will converge to a steady-state message as K increases. Note that if the models $\mathcal{A}$ and $\mathcal{B}$ are learned from the data it is in general necessary to compute these quantities, since they depend on the models.

In contrast to the offline algorithm, likelihood filtering is formulated for indefinitely continuing signals $(\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \ldots)$. In such a situation, actual likelihoods, likelihood functions, or marginal PDFs are almost never directly of interest since the values typically become very small over time. The locally approximated likelihoods described in Section 6.4 may provide more appealing alternatives.

An important special case arises if ℓ is fixed at $\ell = K$, i.e., $D_1 = D_2 = 0$. The filtering algorithm then performs forward message passing without ever computing backward messages. This type of processing may be particularly well suited for continuous-time analog implementations. In [28, 29], this setting is used exclusively.

For most models $\mathcal{A}$ that factor into (potentially scaled) conditional PDFs, the scale factors of the messages $\overrightarrow{\mu}_{X^{\mathcal{A}}_{k}}$ can be computed using the prediction message rule (I.3). Improper prediction messages potentially appear only for the very first steps and only if f_0 is improper. In this

latter case it may be feasible to make f_0 proper without changing the long-term behavior of the algorithm. This applies to a broad class of models, and hence the prediction PDF plays an important role in the literature [2, 50].

The proposed procedures can, in principle, be applied to any kind of SSMs. In the linear Gaussian case, both forward and backward message passing effectively are linear filters (cf. Sections 3.2.3 and 3.3.1). Computations related to conditional marginals, likelihood functions, or LLRs are, however, usually nonlinear.

For both models $\mathcal{A}$ and $\mathcal{B}$ one can envisage the approximate decomposition of the state space as detailed in Section 3.5. Such a decomposition may bear significant practical relevance. Indeed, the communication systems in [28, 29] are based on this approximation.

7.2.3 Cases with Constant Normalization Factor

In the following we identify three special cases in which the normalization function $\beta^{\circ}_{U_\ell}(\boldsymbol{\theta})$ in (7.9) or $\beta^{\circ}_{U_\ell}$ in (7.13) is constant with respect to both $\boldsymbol{\theta}$ and ℓ, or ℓ alone. Earlier, we named these cases strictly conditional.

The three cases are defined by the factor graphs depicted in Figure 7.3. In all three sub-figures we assume that the factors f_0 and f_{K+1} are (potentially scaled) PDFs, and that the factors $f_k^{\mathcal{A}}$ and $f_{k'}^{\mathcal{B}}$ are (potentially scaled) conditional PDFs in the sense of the arrows as explained in Section 1.5.2. In particular, this is true if the factors $f_k^{\mathcal{A}}$ and $f_{k'}^{\mathcal{B}}$ are linear SSMs as in Figure 3.1, for some sub-figures of 7.3 time-reversed to obey the directions of the arrows. We now go through the three special cases of Figure 7.3.

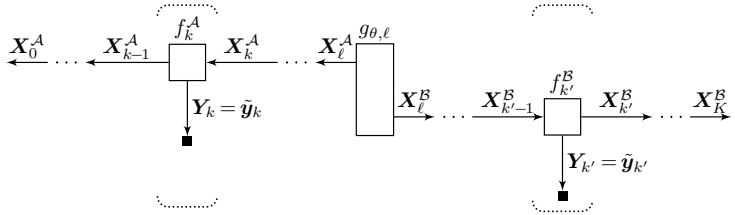
In Figure 7.3a, we define the glue factor to be proportional to a conditional PDF as

$$g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \stackrel{\triangle}{\propto} p\big(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}} \big| \boldsymbol{\theta}, \ell\big)\,, \tag{7.20}$$

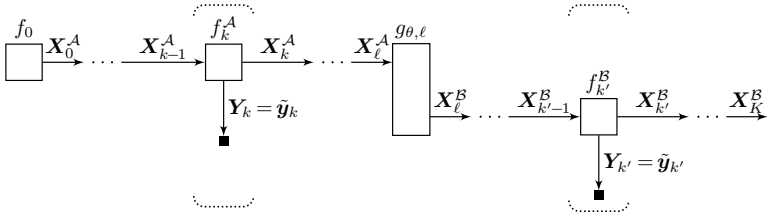
or

$$g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \stackrel{\triangle}{\propto} p\big(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta} \big| \ell\big)\,. \tag{7.21}$$

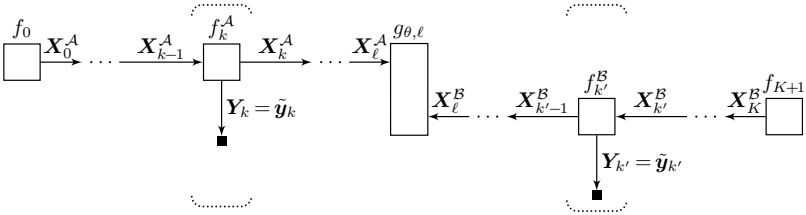
In Figure 7.3b, we define the glue factor to be proportional to a conditional



(a) The glue factor is a prior PDF.



(b) The glue factor is a conditional PDF.



(c) The glue factor is a copula function.

Figure 7.3: Three cases in which $\beta^{\circ}_{U_\ell}$ is constant. The edge directions indicate conditional PDFs as explained in Section 1.5.2.

PDF as

$$g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \stackrel{\triangle}{\propto} p\big(\boldsymbol{x}_\ell^{\mathcal{B}} \big| \boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{\theta}, \ell\big)\,, \tag{7.22}$$

or

$$g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \stackrel{\triangle}{\propto} p\big(\boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta} \big| \boldsymbol{x}_\ell^{\mathcal{A}}, \ell\big)\,. \tag{7.23}$$

Of course, we can alternatively consider a time-reversed variant of the factor graph in Figure 7.3b.

In all the above cases Figures 7.3a and 7.3b, $\beta^\circ_{U_\ell}(\boldsymbol{\theta})$ and $\beta^\circ_{U_\ell}$ are constant with respect to ℓ. In the case of Equations (7.20) and (7.22), $\beta^\circ_{U_\ell}(\boldsymbol{\theta})$ is also constant with respect to $\boldsymbol{\theta}$. This follows directly from the chain rule of probability.

Finally, in Figure 7.3c, the glue factor is assumed to be a copula function [78] with an internal parameter $\boldsymbol{\theta}$. For our purposes we define that $g_\ell(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell)$ is a copula function for the factor graph in Figure 7.3c if

$$\overrightarrow{p}^\circ_{X_\ell^{\mathcal{A}}}(\boldsymbol{x}_\ell^{\mathcal{A}})\, \overrightarrow{p}^\circ_{X_\ell^{\mathcal{B}}}(\boldsymbol{x}_\ell^{\mathcal{B}})\, g_\ell(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \propto p\big(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}} \big| \boldsymbol{\theta}, \ell\big)\,, \tag{7.24}$$

or

$$\overrightarrow{p}^\circ_{X_\ell^{\mathcal{A}}}(\boldsymbol{x}_\ell^{\mathcal{A}})\, \overrightarrow{p}^\circ_{X_\ell^{\mathcal{B}}}(\boldsymbol{x}_\ell^{\mathcal{B}})\, g_\ell(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \propto p\big(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta} \big| \ell\big)\,. \tag{7.25}$$

It follows directly from this definition that $\beta^\circ_{U_\ell}(\boldsymbol{\theta})$ and $\beta^\circ_{U_\ell}$ are constant with respect to ℓ and $\beta^\circ_{U_\ell}(\boldsymbol{\theta})$ is additionally constant with respect to $\boldsymbol{\theta}$ in the case of (7.24).

In general, none of these three cases applies. Recall that in (7.9) and (7.13) we have proposed to re-normalize the function represented by the factor graph, and at least for ℓ this may be a desirable strategy. The normalization function $\zeta(\boldsymbol{\theta}, \ell) = \beta^\circ_{U_\ell}(\boldsymbol{\theta}) = \beta^\circ_{\Theta_\ell}(\boldsymbol{\theta})$ or $\zeta(\ell) = \beta^\circ_{U_\ell}$ can, in principle, be computed by message passing.

Alternatively, in certain cases, re-normalization can be accomplished by a re-parameterization of the glue factor. As an example, the general case of Figure 7.1 can be turned into the constant normalization case of Figure 7.3b if f_0 is proper (i.e. integrable), $f_{K+1}(\boldsymbol{x}_K^{\mathcal{B}}) \propto 1$, $f_k^{\mathcal{A}}$ and $f_{k'}^{\mathcal{B}}$ are conditional PDFs, and if the glue factor $g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell)$ is proper with respect to $\boldsymbol{X}_\ell^{\mathcal{A}}$ and $\boldsymbol{X}_\ell^{\mathcal{B}}$. Then the original glue factor $g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell)$ can be substituted by a re-parameterized glue factor

$$g'(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \stackrel{\triangle}{\propto} \frac{g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell)}{\sum_{\ell'=0}^{K} \iint g(\boldsymbol{x}, \boldsymbol{x}_{\ell'}^{\mathcal{B}}, \boldsymbol{\theta}', \ell')\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{\theta}'}\,, \tag{7.26}$$

or

$$g'(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \overset{\triangle}{\propto} \frac{g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell)}{\sum_{\ell'=0}^{K} \int g(\boldsymbol{x}, \boldsymbol{x}_{\ell'}^{\mathcal{B}}, \boldsymbol{\theta}, \ell') \,\mathrm{d}\boldsymbol{x}}\,, \tag{7.27}$$

which represents a conditional PDF as in (7.22) or (7.23) respectively.

In most cases, the original glue factor g does not depend on ℓ such that (7.26) and (7.27) can be simplified to

$$g'(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \overset{\triangle}{\propto} \frac{g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell)}{\iint g(\boldsymbol{x}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}', \ell) \,\mathrm{d}\boldsymbol{x} \,\mathrm{d}\boldsymbol{\theta}'}\,, \tag{7.28}$$

and

$$g'(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \overset{\triangle}{\propto} \frac{g(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell)}{\int g(\boldsymbol{x}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{\theta}, \ell) \,\mathrm{d}\boldsymbol{x}}\,, \tag{7.29}$$

respectively.

As a second example, we consider re-normalization of the general model family in Figure 7.1 to the constant normalization case of Figure 7.3a. First, for this case we must have $f_0(\boldsymbol{x}_0^{\mathcal{A}}) \propto 1$ and $f_{K+1}(\boldsymbol{x}_K^{\mathcal{B}}) \propto 1$. Next, we must be able to re-parameterize every factor $f_k^{\mathcal{A}}$ into a (potentially scaled) conditional PDF

$$f_k'^{\mathcal{A}}(\boldsymbol{x}_k^{\mathcal{A}}, \boldsymbol{x}_{k-1}^{\mathcal{A}}, \boldsymbol{y}_k) \overset{\triangle}{\propto} \frac{f_k^{\mathcal{A}}(\boldsymbol{x}_k^{\mathcal{A}}, \boldsymbol{x}_{k-1}^{\mathcal{A}}, \boldsymbol{y}_k)}{\int f_k^{\mathcal{A}}(\boldsymbol{x}_k^{\mathcal{A}}, \boldsymbol{x}, \boldsymbol{y}_k) \,\mathrm{d}\boldsymbol{x}}\,, \tag{7.30}$$

i.e., we have to be able to reverse the direction of time in model $\mathcal{A}$. For linear SSMs that comply with Assumption 3.1, this is in general feasible. Finally, the glue factor g must be proper with respect to $\boldsymbol{X}_\ell^{\mathcal{A}}$, $\boldsymbol{X}_\ell^{\mathcal{B}}$ to act as a prior PDF as in (7.20) or it must be additionally proper with respect to $\boldsymbol{\theta}$ to act as a prior PDF as in (7.21).

7.3 Glue Factors with Fixed Position

In this section we let the position ℓ of any glue factor be known or otherwise fixed. With this setup, we elaborate on two topics. First, we make some general considerations about learning glue factor parameters. Second, we consider the usage of a likelihood filter to track several hypotheses that differ only in the glue factor.

We apply these general concepts to pulse modeling and array signal processing.

7.3.1 Learning Glue Factor Parameters

Consider situations, in which a glue factor position (or the position of several glue factors) is given, and from observed data we would like to estimate a glue factor parameter (vector) $\boldsymbol{\theta}$. In principle, this topic is not directly connected to likelihood computation for a model family, but rather to the concept of a glue factor in general. We distinguish the following three scenarios:

a) Given an observed signal $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ and a glue factor position ℓ, we make an ML or a MAP estimate of $\boldsymbol{\theta}$ by maximizing (7.9) or (7.11) respectively.

b) Given M independently observed signals $\boldsymbol{Y}^{(m)} = \tilde{\boldsymbol{y}}^{(m)}$ for $m = 1, \ldots, M$, each with a glue factor $g_\theta^{(m)}$ positioned at ℓ_m and we make a joint estimate of $\boldsymbol{\theta}$ as

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{m=1}^{M} p\big(\tilde{\boldsymbol{y}}^{(m)} \big| \boldsymbol{\theta}\big)\,, \tag{7.31}$$

or

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{m=1}^{M} p\big(\boldsymbol{\theta} \big| \tilde{\boldsymbol{y}}^{(m)}\big)\,. \tag{7.32}$$

c) We are given one signal $\boldsymbol{Y} = \tilde{\boldsymbol{y}}$ with M glue factor locations $\ell_1, \ldots, \ell_M$ (spaced sufficiently far apart). Strictly, such a situation cannot be modeled accurately by one glue factor and two models $\mathcal{A}$ and $\mathcal{B}$. Nevertheless, we envisage estimating a common parameter $\boldsymbol{\theta}$ of all glue factors.

Scenario (a) requires no further explanation at this level of abstraction. We have described this scenario in the simple model of Example 7.1 and we will revisit it in Section 7.4 in a more complex context.

For Scenario (b) we can envisage the factor graph of Figure 7.4. In this factor graph, the parameter for the m-th glue factor in the m-th signal

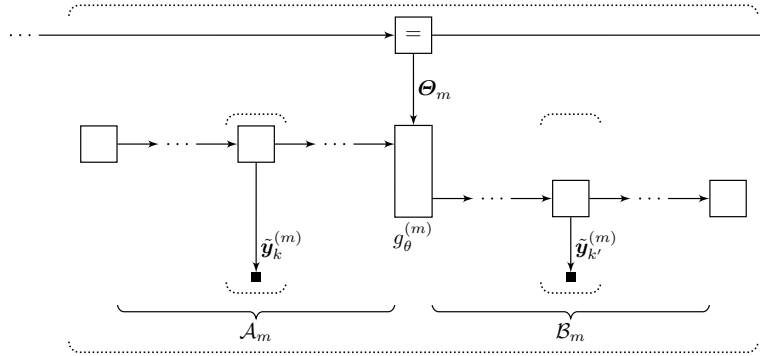


Figure 7.4: Learning a glue factor parameter $\boldsymbol{\theta}$ from several signals $\tilde{\boldsymbol{y}}^{(m)}$.

is modeled by an edge $\boldsymbol{\Theta}_m$ and all edges $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_M$ are connected by equality nodes.

Since the graph in Figure 7.4 has no cycles, the estimation procedures in Equation (7.31) or (7.32) can be done in the following way. First, sum-product message passing is done in each of the M SSMs in order to compute $\overleftarrow{\mu}_{\Theta_m}$. Then, max-product message passing is done in the chain of equalities (upper part of the factor graph) and the estimates (7.31) or (7.32) are obtained by maximizing the max-marginal.

If the glue factors are such that a sum-product message $\overleftarrow{\mu}_{\Theta_m}$ cannot easily be computed, then we may resort to the techniques described in Chapter 4, i.e., to EM, CM, or to local Taylor approximation.

Finally, we elaborate on Scenario (c). A factor graph that conforms to this scenario in the general case is shown in Figure 7.5, in which a sequence of $M+1$ SSMs $\mathcal{M}_0, \ldots, \mathcal{M}_M$ is concatenated by M glue factors $g_\theta^{(1)}, \ldots, g_\theta^{(M)}$ all of which share the same parameter $\boldsymbol{\theta}$. In contrast to Scenario (b) the factor graph in Figure 7.5 has cycles and hence, one option is to consider iterative message-passing algorithms.

For signal processing applications and with regard to online algorithms, simplified versions of this scenario may, however, be more relevant. Specifically, let us assume that the glue factor positions are far enough apart such that, locally, the signal can be modeled by one glue factor $g_\theta^{(m)}$. In other words, we suggest to substitute the factor graph of Figure 7.5 by the one of Figure 7.4 with $\mathcal{A}_m = \mathcal{M}_{m-1}$ and $\mathcal{B}_m = \mathcal{M}_m$.

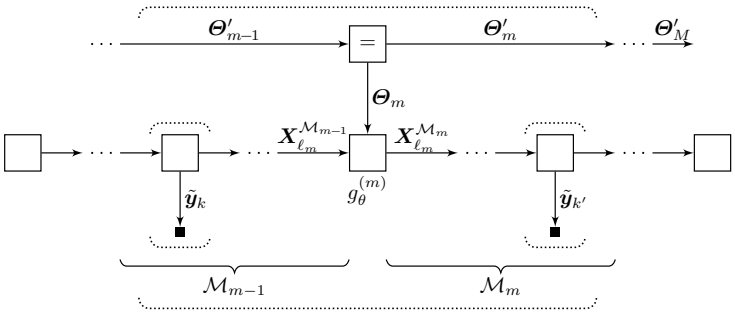


Figure 7.5: Learning a glue factor parameter $\boldsymbol{\theta}$ from one signal.

Furthermore, when adopting this local view, we can even break the continuity $\mathcal{A}_m = \mathcal{B}_{m-1}$ in the sequence of models and, e.g., consider a case in which $\mathcal{A}_m = \mathcal{A}$ and $\mathcal{B}_m = \mathcal{B}$ for all $m = 1, \ldots, M$. We then have, however, lost the generative form of the model in Figure 7.5. To clarify this restricted version of the localized Scenario (c), we define an online algorithm for estimating the glue factor parameter $\boldsymbol{\theta}$. In the following, let D be a delay parameter that is smaller than the time distance between glue factor positions. Also, let m be the currently last glue factor position. The following algorithm is formulated with respect to Figure 7.5 in which we substitute $\mathcal{M}_{m-1} = \mathcal{A}$ and $\mathcal{M}_m = \mathcal{B}$.

Online Glue Factor Parameter Learning for Known Glue Factor Positions:

a) Increment m and fetch the next data items $\tilde{\boldsymbol{y}}_{\ell_{m-1}+D+1}, \ldots, \tilde{\boldsymbol{y}}_{\ell_m+D}$.

b) Do forward message passing in the factor graph of model $\mathcal{A}$, i.e., compute $\overrightarrow{\mu}_{X_k^{\mathcal{A}}}$ for $k = \ell_{m-1} + D + 1, \ldots, \ell_m$.

c) Do backward message passing in the factor graph of model $\mathcal{B}$, i.e., compute $\overleftarrow{\mu}_{X_{k'}^{\mathcal{B}}}$ for $k' = \ell_m + D, \ldots, \ell_m$.

d) From $\overrightarrow{\mu}_{X_{\ell_m}^{\mathcal{A}}}$ and $\overleftarrow{\mu}_{X_{\ell_m}^{\mathcal{B}}}$ compute $\overleftarrow{\mu}_{\Theta_m}$.

e) From $\overrightarrow{\mu}_{\Theta'_{m-1}}$ and $\overleftarrow{\mu}_{\Theta_m}$ compute $\overrightarrow{\mu}_{\Theta'_m}$ using max-product message passing and make an estimate $\hat{\boldsymbol{\theta}}_m = \operatorname{argmax}_{\boldsymbol{\theta}} \overrightarrow{\mu}_{\Theta'_m}(\boldsymbol{\theta})$.

f) Go to Step (a).

In Step (e), a forgetting factor may be used to track slow changes in the parameter $\boldsymbol{\theta}$.

7.3.2 Tracking Several Hypotheses

The idea of the glue factor and likelihood filtering can be used to analyze a signal with respect to several (or even a continuum of) hypotheses simultaneously. We envisage a situation in which the models $\mathcal{A}$ and $\mathcal{B}$ in our general model family of Figure 7.1 are fixed but for each hypothesis of interest we may consider a different glue factor. For this section, we restrict the setting to such hypotheses that are independent of the glue factor position.

For any two hypotheses $\mathcal{H}_1 \colon \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}_1$ and $\mathcal{H}_2 \colon \boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}_2$ and a given glue factor position, an LLR $\log \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_1)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_2)}$ or $\log \frac{p(\tilde{\boldsymbol{y}},\mathcal{H}_1)}{p(\tilde{\boldsymbol{y}},\mathcal{H}_2)}$ and a corresponding LLR test can be devised. This scenario can be extended to multiple hypothesis testing. Note that, since the hypotheses do not depend on the models $\mathcal{A}$ and $\mathcal{B}$, the message passing in Steps (a) and (b) in the offline algorithm or in Steps (b) and (c) of the likelihood filter can be done without specifying the hypothesis.

We recall that Theorem 6.4 tells us in detail how to compute three commonly used (G)LLRs. Others that are more suited for communication scenarios as, e.g., in [28, 29] can be devised.

7.4 Application to Array Processing

In this section we give an example application of estimating glue factor parameters. Specifically, we formulate ML estimation and LLR computation for the detection of coupled sinusoids [88]. In contrast to the general glue factor model of Figure 7.1 we now glue more than two SSMs, each of second order.

7.4.1 Setup and Uncoupled Case

In a sensor array with J sensors, vector data $\tilde{\boldsymbol{y}}_k \in \mathbb{R}^J$ is recorded in parallel, with the aim of finding some structure in the joint signal $\tilde{\boldsymbol{y}} \triangleq (\tilde{\boldsymbol{y}}_1, \dots, \tilde{\boldsymbol{y}}_K)$ from all sensors. Consider J discrete-time sinusoidal

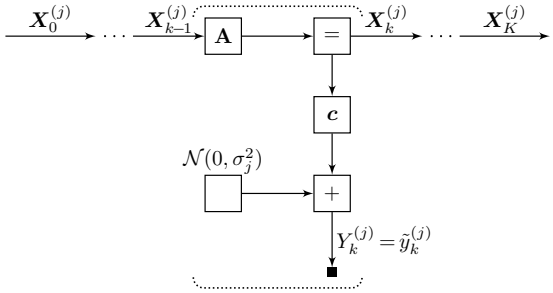


Figure 7.6: Autonomous second-order state-space model (SSM) for the j-th sinusoid.

signals

$$Y_k^{(j)} = \alpha_j \cos(\Omega k + \psi_j) + Z_k^{(j)}\,, \tag{7.33}$$

for $j = 1, \ldots, J$. All J signals have the same, known frequency Ω but differ in amplitude α_j and phase ψ_j. The Gaussian noises $Z_k^{(j)} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_j^2)$ are allowed to have differing noise variances σ_j^2 for $j = 1, \ldots, J$.

ML estimation of the parameters of a single sinusoid from an observed signal $\tilde{y}_k^{(j)}$ signal was treated in Example 2.3 and the case of several uncoupled sinusoids is a straightforward extension. In the following example, we assume that $j = 1$ in both Equation (7.33) and Figure 7.6 and we drop j from our notation.

Example 7.2: Log-Likelihood Ratios (LLR) and Generalized Log-Likelihood Ratios (GLLR) for a Single Sinusoid
Consider a given signal $\tilde{y}_1, \ldots, \tilde{y}_K$ for which we want to compute the LLR with respect to the hypotheses $\mathcal{H}_1$ and $\mathcal{H}_0$, where $\mathcal{H}_1$ is the presence of a sinusoid observed under additive white Gaussian noise and $\mathcal{H}_0$ is the presence of noise only. We can model a sinusoid (7.33) by means of the autonomous second-order SSM of Figure 7.6. The SSM is defined by $\mathbf{A} \triangleq \operatorname{rotm} \Omega$ and $\boldsymbol{c} \triangleq [1, 0]$.

For the noise-only hypothesis $\mathcal{H}_0$ we can still employ the same SSM but with fixed state $\boldsymbol{X}_K = \boldsymbol{0}$. Now we note that the LLRs of interest are covered in Theorem 6.4 in which we identify $\boldsymbol{\Theta} \triangleq \boldsymbol{X}_K$. For the Gaussian case at hand:

a) The LLR for unknown phase and amplitude $\boldsymbol{X}_K (= \boldsymbol{\Theta})$ is given in (6.98).

b) The LLR for known fixed phase and amplitude $\boldsymbol{X}_K = \tilde{\boldsymbol{x}}_K (= \tilde{\boldsymbol{\theta}})$ is given in (6.99).

c) The GLLR for plugged-in ML estimate $\boldsymbol{X}_K = \hat{\boldsymbol{x}}_{K,\mathrm{ML}} = \vec{\boldsymbol{m}}_{X_K} (= \boldsymbol{m}_\Theta)$ is given in (6.100).

With the analytic solution (3.48) for $\vec{\boldsymbol{m}}_{X_K}$ the connection with the discrete Fourier transform (DFT) (or the discrete-time Fourier transform) becomes evident. E.g., for DFT frequencies $\Omega_n = 2\pi n/K$ we get in the above cases (a) and (c)

$$\mathrm{LLR} \propto |\breve{y}_n| \,, \tag{7.34}$$

where $\breve{y}_0, \ldots, \breve{y}_{K-1}$ is the DFT of $\tilde{y}_1, \ldots, \tilde{y}_K$. This result is standard and can be found in many textbooks [54]. ◊

Detection of J uncoupled sinusoids is a straightforward extension to Example 7.2. In the following, we turn to the case in which the amplitudes and phases of the sinusoids are coupled.

7.4.2 Estimation and Detection of Coupled Sinusoids

For coupled sinusoids of the form (7.33), we assume that the amplitudes α_j and phase shifts ψ_j are constrained by a parameter vector $\boldsymbol{\theta}$ via some mapping

$$\Gamma : \boldsymbol{\theta} \mapsto \big((\alpha_1, \psi_1), \ldots, (\alpha_J, \psi_J)\big) \,. \tag{7.35}$$

In this thesis we will not give any explicit expression for Γ, since this is largely application-dependent.

Consider, for example, the estimation of a seismic wave field measured by a sensor array. In this example, $\boldsymbol{\theta}$ may contain wave field parameters such as wave type, velocity of propagation, angle of arrival, etc. In this example, several waves with differing parameter vectors are assumed to be present concurrently. The mapping Γ includes sensor characteristics and positions. This setting is treated in more detail in [71, 72] for the estimation and detection of Rayleigh wave and Love wave parameters using triaxial vector-sensors.

The joint signal $\boldsymbol{Y} \triangleq (\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(J)})$, where $\boldsymbol{Y}^{(j)} = (Y_1^{(j)}, \ldots, Y_K^{(j)})$ is modeled by J SSMs, which are connected by a glue factor g_θ at time K as shown in Figure 7.7. Each SSM is an autonomous second-order

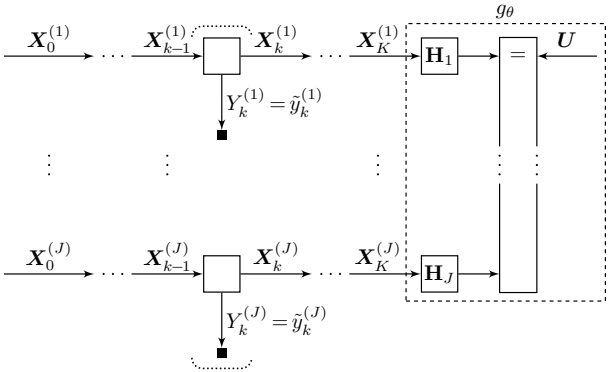


Figure 7.7: A glue factor view for array processing.

model as in Figure 7.6 with $\mathbf{A} \triangleq \operatorname{rotm} \Omega$, and $\boldsymbol{c} = [1, 0]$. We recall that the noise variance σ_j^2 in each such model is allowed to differ.

In the glue factor, the reference state

$$\boldsymbol{U} \triangleq \rho_0 \begin{bmatrix} \cos(\Omega K + \phi_0) \\ \sin(\Omega K + \phi_0) \end{bmatrix} \tag{7.36}$$

contains the reference amplitude ρ_0 and the reference phase ϕ_0. The matrices

$$\mathbf{H}_j \triangleq \rho_j \operatorname{rotm}(\phi_j)\,, \tag{7.37}$$

with $\rho_j \triangleq \rho_0/\alpha_j$ and $\phi_j \triangleq \phi_0 - \psi_j$ for $j = 1, \ldots, J$ thus model the coupling between the signals. The matrices $\mathbf{H}_j$ depend on the parameter vector $\boldsymbol{\theta}$ via the mapping Γ. We refrain from reflecting this dependence in our notation.

We note that the model at hand is strictly conditional with respect to $\boldsymbol{U}$ but not with respect to $\boldsymbol{\theta}$. The former can be proved by noting that $\overleftarrow{\mathbf{W}}_U^\circ = \mathbf{0}$ for any nonsingular matrices $\mathbf{H}_1, \ldots, \mathbf{H}_J$. To see the latter we note that Rules (II.3) and (II.5) allow us to write

$$\overleftarrow{\gamma}_U^\circ = \prod_{j=1}^{J} \frac{\overrightarrow{\gamma}_{X_K^{(j)}}^\circ}{|\det \mathbf{H}_j|} = \prod_{j=1}^{J} \frac{1}{|\det \mathbf{H}_j|}\,, \tag{7.38}$$

where the second equality follows from Rules (II.2), (II.5), (II.3), and by noting that $\det \mathbf{A} = 1$. The above quantity depends on $\boldsymbol{\theta}$ via the matrices $\mathbf{H}_j$. In the present example, re-normalization is equivalent with re-parameterization of the model by inverting $\mathbf{H}_j$ and reversing the arrows of the edges $\boldsymbol{X}_k^{(j)}$ for all j. This re-parameterized version is used in [71, 88].

The log-likelihood function under re-normalization follows from (7.9) and in our Gaussian case at hand from (6.93) as

$$\ln p(\tilde{\boldsymbol{y}}|\boldsymbol{u},\boldsymbol{\theta}) = \ln \frac{\overleftarrow{\mu}_U(\boldsymbol{u})}{\overleftarrow{\gamma}_U^\circ} \tag{7.39}$$

$$= \ln \frac{\overleftarrow{\gamma}_U}{\overleftarrow{\gamma}_U^\circ} - \boldsymbol{u}^\mathsf{T}\overleftarrow{\mathbf{W}}_U\boldsymbol{u}/2 + \boldsymbol{u}^\mathsf{T}\overleftarrow{\mathbf{W}}_U\overleftarrow{\boldsymbol{m}}_U\,, \tag{7.40}$$

with

$$\overleftarrow{\gamma}_U = \prod_{j=1}^{J} \frac{\overrightarrow{\gamma}_{X_K^{(j)}}}{|\det \mathbf{H}_j|} \tag{7.41}$$

$$\overleftarrow{\mathbf{W}}_U = \sum_{j=1}^{J} \mathbf{H}_j^{-\mathsf{T}}\overrightarrow{\mathbf{W}}_{X_K^{(j)}}\mathbf{H}_j^{-1}\,, \tag{7.42}$$

$$\overleftarrow{\mathbf{W}}_U\overleftarrow{\boldsymbol{m}}_U = \sum_{j=1}^{J} \mathbf{H}_j^{-\mathsf{T}}\overrightarrow{\mathbf{W}}_{X_K^{(j)}}\,, \tag{7.43}$$

where $\mathbf{H}_1, \ldots, \mathbf{H}_J$ depend implicitly on $\boldsymbol{\theta}$ via the mapping (7.35). The dependency of $\overleftarrow{\gamma}_U$ on $\boldsymbol{\theta}$ cancels with the dependency of $\overleftarrow{\gamma}_U^\circ$ on $\boldsymbol{\theta}$ such that (7.40) simplifies to

$$\ln p(\tilde{\boldsymbol{y}}|\boldsymbol{u},\boldsymbol{\theta}) = \sum_{j=1}^{J} \ln \overrightarrow{\gamma}_{X_K^{(j)}} - \boldsymbol{u}^\mathsf{T}\overleftarrow{\mathbf{W}}_U\boldsymbol{u}/2 + \boldsymbol{u}^\mathsf{T}\overleftarrow{\mathbf{W}}_U\overleftarrow{\boldsymbol{m}}_U\,, \tag{7.44}$$

where the scale factors $\overrightarrow{\gamma}_{X_K^{(j)}}$ do not depend on $\boldsymbol{\theta}$.

The objective of the present application is to estimate the wave field parameters in $\boldsymbol{\theta}$ and the reference amplitude and reference phase in $\boldsymbol{U}$. The ML estimate of the latter simply is

$$\hat{\boldsymbol{u}}_{\mathrm{ML}} = \overleftarrow{\boldsymbol{m}}_U\,. \tag{7.45}$$

Plugging in (7.45) into the log-likelihood function of (7.44) we get the partially maximized log-likelihood function

$$\ln p(\tilde{\boldsymbol{y}}|\hat{\boldsymbol{u}}_{\mathrm{ML}}, \boldsymbol{\theta}) = \overleftarrow{\boldsymbol{m}}_U^{\mathsf{T}} \overleftarrow{\mathbf{W}}_U \overleftarrow{\boldsymbol{m}}_U/2 + \sum_{j=1}^{J} \ln \overrightarrow{\gamma}_{X_K^{(j)}} . \tag{7.46}$$

In [71,72] the remaining maximization to find $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$ is done by exhaustive search. Let us emphasize that with this approach we can formulate the log-likelihood function of any wave type. The state-space model serves only to assemble a sufficient statistic in the form of the messages $\overrightarrow{\mu}_{X_K^{(j)}}$, while the wave type is specified in the glue factor only.

For the purpose of detection we observe that the cases of interest conform with Cases (a) and (c) of Theorem 6.4. For the LLR in Case (a), we consider the signal and noise hypotheses

$$\mathcal{H}_1 \colon \boldsymbol{u} \neq \boldsymbol{0} \tag{7.47}$$

$$\mathcal{H}_0 \colon \boldsymbol{u} = \boldsymbol{0} . \tag{7.48}$$

Note that since the SSMs are autonomous, the noise hypothesis can indeed be formulated as in (7.48). For the GLLR in Case (c), the signal hypothesis changes to

$$\mathcal{H}_1 \colon \boldsymbol{u} = \hat{\boldsymbol{u}}_{\mathrm{ML}} . \tag{7.49}$$

The corresponding LLR and GLLR are given in Equation (6.98) and Equation (6.100) respectively, which we repeat here to avoid notational confusion as

a) $$\mathrm{LLR} = \overleftarrow{\boldsymbol{m}}_U^{\mathsf{T}} \overleftarrow{\mathbf{W}}_U \overleftarrow{\boldsymbol{m}}_U/2 + \frac{1}{2} \ln \frac{(2\pi)^n}{\det \overleftarrow{\mathbf{W}}_U} , \tag{7.50}$$

c) $$\mathrm{GLLR} = \overleftarrow{\boldsymbol{m}}_U^{\mathsf{T}} \overleftarrow{\mathbf{W}}_U \overleftarrow{\boldsymbol{m}}_U/2 , \tag{7.51}$$

respectively.

7.4.3 Noise Variance Estimation

We consider a setting in which the noise variances σ_j^2 for $j = 1, \ldots, J$ are unknown and must be estimated from the observations. This applies for

both hypotheses $\mathcal{H}_1$ and $\mathcal{H}_0$ separately. We collect the noise variances in a parameter vector

$$\boldsymbol{\eta} \triangleq (\sigma_1^2, \ldots, \sigma_J^2)\,, \tag{7.52}$$

and we let $\hat{\boldsymbol{\eta}}_{\mathrm{ML}}^{(1)}$ and $\hat{\boldsymbol{\eta}}_{\mathrm{ML}}^{(0)}$ be the ML estimates under the hypotheses $\mathcal{H}_1$ and $\mathcal{H}_0$ respectively. The target GLLR can now be written as

$$\mathrm{GLLR} = \ln \frac{p\big(\tilde{\boldsymbol{y}} \,\big|\, \hat{\boldsymbol{u}}_{\mathrm{ML}}, \hat{\boldsymbol{\theta}}_{\mathrm{ML}}, \hat{\boldsymbol{\eta}}_{\mathrm{ML}}^{(1)}, \mathcal{H}_1\big)}{p\big(\tilde{\boldsymbol{y}} \,\big|\, \hat{\boldsymbol{\eta}}_{\mathrm{ML}}^{(0)}, \mathcal{H}_0\big)}\,. \tag{7.53}$$

In our notation we let $\hat{\sigma}_{1,j}^2$ and $\hat{\sigma}_{0,j}^2$ for $j = 1, \ldots, J$ be estimates of σ_j^2 under the hypotheses $\mathcal{H}_1$ and $\mathcal{H}_0$ respectively.

For the noise hypothesis $\mathcal{H}_0$, the ML estimate of the noise variance in the j-th signal is the signal power

$$\hat{\sigma}_{0,j}^2 = \xi_j^2 \triangleq \frac{1}{K} \sum_{k=1}^{K} \Big(\tilde{y}_k^{(j)}\Big)^2\,. \tag{7.54}$$

The log-likelihood can thus be written as

$$\ln p\big(\tilde{\boldsymbol{y}} \,\big|\, \hat{\boldsymbol{\eta}}_{\mathrm{ML}}^{(0)}, \mathcal{H}_0\big) = \ln \overleftarrow{\gamma}_U / \overleftarrow{\gamma}_U^{\circ} \tag{7.55}$$

$$= -\frac{K}{2} \sum_{j=1}^{J} \ln(2\pi e \xi_j^2)\,, \tag{7.56}$$

where we have used the relation (6.19) between the γ-type scale factor and the signal energy for autonomous models (cf. Example 6.1).

For the signal hypothesis $\mathcal{H}_1$ we should ideally maximize over both $(\boldsymbol{u}, \boldsymbol{\theta})$ and $\boldsymbol{\eta}$ jointly. Since this cannot be done easily, we propose to use CM by alternating between

$$\hat{\boldsymbol{\eta}}^{(1)} = \operatorname*{argmax}_{\boldsymbol{\eta}^{(1)}} p\big(\tilde{\boldsymbol{y}} \,\big|\, \hat{\boldsymbol{u}}, \hat{\boldsymbol{\theta}}, \boldsymbol{\eta}^{(1)}, \mathcal{H}_1\big)\,, \tag{7.57}$$

and

$$(\hat{\boldsymbol{u}}, \hat{\boldsymbol{\theta}}) = \operatorname*{argmax}_{\boldsymbol{u}, \boldsymbol{\theta}} p\big(\tilde{\boldsymbol{y}} \,\big|\, \boldsymbol{u}, \boldsymbol{\theta}, \hat{\boldsymbol{\eta}}^{(1)}, \mathcal{H}_1\big)\,. \tag{7.58}$$

The estimation in (7.58) was treated in Section 7.4.2 and hence we focus on (7.57).

As an initial estimate $\hat{\boldsymbol{\eta}}^{(1)}$ we propose to assume that the signals are decoupled, i.e., that the glue factor is not present. In this case, the ML estimate is the residual noise power

$$\hat{\sigma}_{1,j}^2 = \frac{1}{K}\sum_{k=1}^{K}\Big(\tilde{y}_k^{(\ell)} - \boldsymbol{c}\,\mathbf{A}^{k-K}\overrightarrow{\boldsymbol{m}}_{X_K^{(j)}}\Big)^2 . \tag{7.59}$$

Once we have $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\theta}}$ we propose to use a joint estimate

$$\hat{\sigma}_{1,j}^2 = \frac{1}{K}\sum_{k=1}^{K}\Big(\tilde{y}_k^{(\ell)} - \boldsymbol{c}\,\mathbf{A}^{k-K}\hat{\mathbf{H}}_j^{-1}\hat{\boldsymbol{u}}\Big)^2 . \tag{7.60}$$

In the above, note that $\hat{\mathbf{H}}_j^{-1}\hat{\boldsymbol{u}}$ is an estimate of $\boldsymbol{X}_K^{(j)}$ for which we have plugged $\hat{\boldsymbol{\theta}}$ into the matrices $\mathbf{H}_j$ and for which both $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{u}} = \overleftarrow{\boldsymbol{m}}_U$ have been computed with the previously assigned noise variance estimates for hypothesis $\mathcal{H}_1$.

Once the algorithm has converged, the resulting $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\theta}}$ induce a signal estimate

$$\hat{y}_k^{(j)} = \boldsymbol{c}\,\mathbf{A}^{k-K}\hat{\mathbf{H}}_j^{-1}\hat{\boldsymbol{u}}\,, \tag{7.61}$$

and a corresponding noise variance estimate

$$\hat{\sigma}_{1,j}^2 = \frac{1}{K}\sum_{k=1}^{K}\big(\tilde{y}_k^{(j)} - \hat{y}_k^{(j)}\big)^2 . \tag{7.62}$$

Hence, the log-likelihood can be written as

$$\begin{aligned}
\ln p\big(\tilde{\boldsymbol{y}}\big|\hat{\boldsymbol{u}},\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\eta}}^{(1)},\mathcal{H}_1\big) &= -\sum_{j=1}^{J}\sum_{k=1}^{K}\left(\frac{1}{2}\ln(2\pi\hat{\sigma}_{1,j}^2) + \frac{\big(\tilde{y}_k^{(j)} - \hat{y}_k^{(j)}\big)^2}{2\hat{\sigma}_{1,j}^2}\right)\\
&= -\sum_{j=1}^{J}\left(\frac{K}{2}\ln(2\pi\hat{\sigma}_{1,j}^2) + \sum_{k=1}^{K}\frac{\big(\tilde{y}_k^{(j)} - \hat{y}_k^{(j)}\big)^2}{2\hat{\sigma}_{1,j}^2}\right)\\
&= -\frac{K}{2}\sum_{j=1}^{J}\ln(2\pi e\hat{\sigma}_{1,j}^2)\,.
\end{aligned} \tag{7.63}$$

Alternatively, we can compute the likelihood via scale factors from (7.46)

as

$$\ln p\big(\tilde{\boldsymbol{y}}\,\big|\,\hat{\boldsymbol{u}},\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\eta}}^{(1)},\mathcal{H}_1\big) = \overleftarrow{\boldsymbol{m}}_U^{\mathsf{T}}\overleftarrow{\mathbf{W}}_U\overleftarrow{\boldsymbol{m}}_U/2 - \frac{K}{2}\sum_{j=1}^{J}\left(\ln(2\pi\hat{\sigma}_{1,j}^2) + \frac{\xi_j^2}{\hat{\sigma}_{1,j}^2}\right), \tag{7.64}$$

where we have made use of (6.19) from Example 6.1. From (7.56) and (7.63) or (7.64) we can write the GLLR in (7.53) as

$$\mathrm{GLLR} = \frac{K}{2}\sum_{j=1}^{J}\ln\frac{\xi_j^2}{\hat{\sigma}_{1,j}^2}\,, \tag{7.65}$$

or

$$\mathrm{GLLR} = \overleftarrow{\boldsymbol{m}}_U^{\mathsf{T}}\overleftarrow{\mathbf{W}}_U\overleftarrow{\boldsymbol{m}}_U/2 - \frac{K}{2}\sum_{j=1}^{J}\left(\frac{\xi_j^2-\hat{\sigma}_{1,j}^2}{\hat{\sigma}_{1,j}^2} - \ln\frac{\xi_j^2}{\hat{\sigma}_{1,j}^2}\right). \tag{7.66}$$

As a side remark, let us note that the last term in brackets is a differential entropy and hence nonnegative. Therefore the present GLLR cannot exceed the GLLR of (7.51).

If we have very long signals, then we might want to avoid the direct computation of $\hat{\sigma}_{1,j}^2$ as in (7.59) and (7.60). We propose to approximate the initial estimate by

$$\hat{\sigma}_{1,j}^2 \approx \xi_j^2 - \overrightarrow{\boldsymbol{m}}_{X_K^{(j)}}^{\mathsf{T}}\overrightarrow{\boldsymbol{m}}_{X_K^{(j)}}/2\,, \tag{7.67}$$

and the joint estimate by

$$\hat{\sigma}_{1,j}^2 \approx \xi_j^2 - \hat{\boldsymbol{u}}^{\mathsf{T}}\hat{\mathbf{H}}_j^{-T}\hat{\mathbf{H}}_j^{-1}\hat{\boldsymbol{u}}\,, \tag{7.68}$$

where we recall that the matrices $\hat{\mathbf{H}}_j$ are computed with plugged-in $\hat{\boldsymbol{\theta}}$.

With these approximations, message passing in the autonomous SSMs needs to be done only once and the quantities $\overrightarrow{\boldsymbol{m}}_{X_K^{(j)}}$, $\overrightarrow{\mathbf{W}}_{X_K^{(j)}}$, and ξ_j^2 for $j=1,\ldots,J$ suffice to perform CM as in (7.57) and (7.58). To see this, note that $\overrightarrow{\boldsymbol{m}}_{X_K^{(j)}}$ does not depend on any chosen noise variance σ_j^2, and $\overrightarrow{\mathbf{W}}_{X_K^{(j)}}$ depends linearly on σ_j^{-2} as detailed in Equation (3.45).

7.4.4 Extension to Wave Superposition

In the example of seismic wave fields [71, 72], several waves of different type and with differing parameters but same frequency Ω may be present simultaneously. In this case, the signal model (7.33) changes to

$$Y_k^{(j)} = \sum_{m=1}^{M} \left(\alpha_j^{(m)} \cos\bigl(\Omega k + \psi_j^{(m)}\bigr) \right) + Z_k^{(j)} , \tag{7.69}$$

where M is the number of waves. The m-th wave is described by a parameter vector $\boldsymbol{\theta}_m$ and gives rise to amplitudes $\alpha_j^{(m)}$ and phase shifts $\psi_j^{(m)}$ for $j = 1, \ldots, J$ via M mappings

$$\Gamma_m \colon \boldsymbol{\theta}_m \to \left(\bigl(\alpha_1^{(m)}, \psi_1^{(m)}\bigr), \ldots, \bigl(\alpha_J^{(m)}, \psi_J^{(m)}\bigr) \right) \tag{7.70}$$

for $m = 1, \ldots, M$.

We collect the parameters in a vector $\boldsymbol{\theta} \triangleq (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M)$. It is straightforward to model all waves simultaneously by extending the reference state (7.36) to

$$\boldsymbol{U}^{\mathsf{T}} \triangleq \left[\boldsymbol{U}_1^{\mathsf{T}}, \ldots, \boldsymbol{U}_M^{\mathsf{T}} \right] , \; \boldsymbol{U}_m \triangleq \rho_0^{(m)} \begin{bmatrix} \cos\Bigl(\Omega K + \phi_0^{(m)}\Bigr) \\ \sin\Bigl(\Omega K + \phi_0^{(m)}\Bigr) \end{bmatrix} \tag{7.71}$$

for $m = 1, \ldots, M$, where $\rho_0^{(m)}$ and $\phi_0^{(m)}$ are the reference amplitude and phase of the m-th wave. Analogously, we have to extend (7.37) to

$$\mathbf{H}_j \triangleq \left[\mathbf{H}_j^{(1)}, \ldots, \mathbf{H}_j^{(M)} \right] , \; \mathbf{H}_j^{(m)} \triangleq \rho_j^{(m)} \operatorname{rotm}(\phi_j^{(m)}) , \tag{7.72}$$

where $\rho_j^{(m)} \triangleq \rho_0^{(m)} / \alpha_j^{(m)}$ and $\phi_j^{(m)} \triangleq \phi_0^{(m)} - \psi_j^{(m)}$ for $m = 1, \ldots, M$.

The partially maximized log-likelihood function is still given by (7.46) with given definitions in (7.71) and (7.72). The space over which to maximize has, however, increased approximately M fold. In the example of seismic waves estimation, exhaustive search over this increased parameter space is not feasible anymore.

As an alternative we propose an iterative algorithm based on CM. To this end, we formulate the glue factor of Figure 7.8a, in which the factor g_{θ_m} contains the glue factor for the m-th wave as depicted in Figure 7.8b.

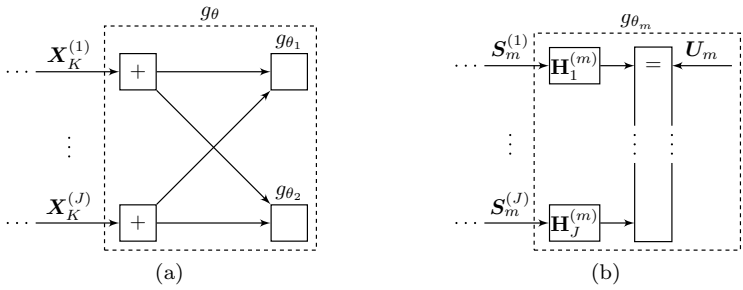


Figure 7.8: Glue factor for wave superposition based on cyclic maximization (CM) – an example for $M = 2$. The details of g_{θ_1} and g_{θ_2} in (a) are given in (b) for $m = 1, 2$.

In this latter figure note the definition of the edges $\boldsymbol{S}_m^{(j)}$ for $j = 1, \ldots, J$ and $m = 1, \ldots, M$.

CM can be applied to the addition nodes in this glue factor as follows. Assume that we have some initial estimate $\hat{\boldsymbol{\theta}}$. We pick some $m \in \{1, \ldots, M\}$ and update the estimate of $\boldsymbol{\theta}_m$ while keeping fixed $\{\boldsymbol{\theta}_q = \hat{\boldsymbol{\theta}}_q\}_{q \in \{1,\ldots,M\} \setminus \{m\}}$, i.e., we fix $\boldsymbol{S}_q^{(j)} = \hat{\boldsymbol{s}}_q^{(j)}$ for all $j = 1, \ldots, J$ and for all $q \in \{1, \ldots, M\} \setminus \{m\}$. The resulting log-likelihood function with respect to $\boldsymbol{\theta}_m$ is analogous to (7.46) given by

$$\ln p(\tilde{\boldsymbol{y}} | \hat{\boldsymbol{u}}_{\mathrm{ML}}, \boldsymbol{\theta}_m) = \overleftarrow{\boldsymbol{m}}_{U_m}^{\mathsf{T}} \overleftarrow{\mathbf{W}}_{U_m} \overleftarrow{\boldsymbol{m}}_{U_m} / 2 + \mathrm{const}\,. \tag{7.73}$$

To apply this algorithm we propose the following greedy-type procedure. Initially, set $M = 1$ and use the glue factor of Figure 7.7 to find $\hat{\boldsymbol{\theta}}_1$. Then repeatedly do the following. Increment M and use CM in the glue factor of Figure 7.8a starting with $m = M$ to find $\hat{\boldsymbol{\theta}}_M$, and iterate finding $\hat{\boldsymbol{\theta}}_m$ for $m \in \{1, \ldots, M\}$ until convergence.

The outlined procedure can be stopped, e.g., by assuming an upper limit for M, or by incorporating a model complexity term, e.g., the Bayesian information criterion [96]. The former is used in [71] and the latter is used in [72].

7.5 Pulse Modeling with Sinusoids

In this section we consider modeling a pulse in two parts, one decaying towards the past and one decaying towards the future. Both parts consist of sum of exponentially decaying sinusoids. We show how we can devise a glue factor to ensure smoothness of the pulse at the glueing position. Different re-parameterizations of this glue factor can serve the different purposes of generating pulses, learning pulse shapes, and locating a pulse.

7.5.1 Pulse Model

Recall that a one-sided superposition of sinusoids can be modeled as in Example 3.1 by an autonomous system with system matrices as given in (3.13) and (3.14). Here, we consider modeling pulses by gluing two such one-sided pulses by means of a glue factor.

Note that a separate treatment of the two sides results in general in discontinuities at the gluing position. We therefore propose to design a glue factor that ensures equality of the first N time-derivatives (including the 0-th) of the left and the right side of the pulse at the gluing position.

From Example 3.1, we recall the sum of M exponentially decaying sinusoids

$$s(t) = \sum_{m=1}^{M} \operatorname{Re} \xi_m(t)\,, \tag{7.74}$$

where

$$\xi_m(t) \triangleq e^{\alpha_m t + i(\omega_m t + \phi_m)}\,. \tag{7.75}$$

The n-th time-derivative of $s(t)$ is

$$\overset{(n)}{s}(t) = \sum_{m=1}^{M} \operatorname{Re} \overset{(n)}{\xi}_m(t) \tag{7.76}$$

$$= \sum_{m=1}^{M} \operatorname{Re}\Big((\alpha_m + i\omega_m)^n \, e^{\alpha_m t + i(\omega_m t + \phi_m)} \Big)\,. \tag{7.77}$$

For any $n \in \mathbb{N}$ the term $(\alpha_m + i\omega_m)^n$ can be expanded into a real term

$a_m^{(n)}$ and an imaginary term $b_m^{(n)}$ as follows:

$$(\alpha_m + i\omega_m)^n = \sum_{j=0}^{n} \binom{n}{j} \alpha_m^{n-j} (i\omega_m)^j \tag{7.78}$$

$$= a_m^{(n)} + i b_m^{(n)} , \tag{7.79}$$

with

$$a_m^{(n)} \triangleq \sum_{j \in \mathcal{E}} (-1)^{j/2} \binom{n}{j} \alpha_m^{n-j} \, \omega_m^j , \tag{7.80}$$

$$b_m^{(n)} \triangleq \sum_{j \in \mathcal{O}} (-1)^{(j-1)/2} \binom{n}{j} \alpha_m^{n-j} \, \omega_m^j , \tag{7.81}$$

where $\mathcal{E}$ and $\mathcal{O}$ form a partition of the set $\{0, \ldots, n\}$ into even and odd integers respectively. We hence can write

$$\overset{\scriptscriptstyle (n)}{s}(t) = \sum_{m=1}^{M} \mathrm{Re}\Big(\big(a_m^{(n)} + i b_m^{(n)}\big) \xi_m(t) \Big) \tag{7.82}$$

$$= \sum_{m=1}^{M} a_m^{(n)} \, \mathrm{Re}\big(\xi_m(t)\big) - b_m^{(n)} \, \mathrm{Im}\big(\xi_m(t)\big) , \tag{7.83}$$

$$\overset{\scriptscriptstyle (n)}{s}(t_\ell) = \boldsymbol{h}^{(n)} \boldsymbol{X}_\ell , \tag{7.84}$$

where

$$\boldsymbol{h}^{(n)} \triangleq \Big[a_1^{(n)}, -b_1^{(n)}, \cdots, a_M^{(n)}, -b_M^{(n)} \Big] . \tag{7.85}$$

For example, in the case $n = 1$ we have $a_m = \alpha_m$, $b_m = \omega_m$ and hence $\boldsymbol{h}^{(1)} = [\alpha_1, -\omega_1, \ldots, \alpha_M, -\omega_M]$.

Now we formulate the complete model in Figure 7.1 by defining each factor. The factors $f_0(\boldsymbol{x}_0^{\mathcal{A}}) = 1$ and $f_{K+1}(\boldsymbol{x}_K^{\mathcal{B}}) = 1$ are neutral. For the models $\mathcal{A}$ and $\mathcal{B}$ we recall the uniform sampling equivalent to Example 3.1 in which we have modeled exponentially decaying sinusoids. Specifically, each factor $f_k^{\mathcal{M}}$ contains an autonomous SSM as in Figure 7.9 with

$$\mathbf{A}_{\mathcal{M}} \triangleq \begin{bmatrix} \rho_1^{\mathcal{M}} \, \mathrm{rotm}\, \Omega_1^{\mathcal{M}} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \rho_M^{\mathcal{M}} \, \mathrm{rotm}\, \Omega_M^{\mathcal{M}} \end{bmatrix} , \tag{7.86}$$

$$\boldsymbol{c} \triangleq \big[1, 0, \cdots, 1, 0\big] , \tag{7.87}$$

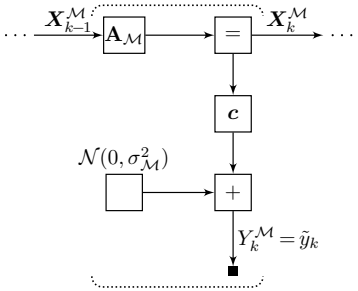


Figure 7.9: Autonomous state-space model (SSM) for models $\mathcal{M} = \mathcal{A}$ or $\mathcal{M} = B$.

where $\rho_m^{\mathcal{M}} \triangleq e^{\alpha_m^{\mathcal{M}}}$, $\Omega_m^{\mathcal{M}} \triangleq \omega_m^{\mathcal{M}} T$ for $m = 1, \ldots, M_{\mathcal{M}}$ and for each model $\mathcal{M} = \mathcal{A}$ and $\mathcal{M} = \mathcal{B}$, where T is the sampling interval. This model can be generalized to the non-uniform sampling setting of Example 3.1, but we refrain from doing so.

Note, that the two systems $\mathcal{A}$ and $\mathcal{B}$ have different parameters, i.e., frequencies $\omega_m^{\mathcal{M}}$, decay-factors $\alpha_m^{\mathcal{M}}$, and system order $M_{\mathcal{M}}$ for $\mathcal{M} = \mathcal{A}, \mathcal{B}$. Specifically, we must have $\alpha_m^{\mathcal{A}} > 0$, since in $\mathcal{A}$ we model increasing sinusoids. Likewise we must have $\alpha_m^{\mathcal{B}} < 0$ because in $\mathcal{B}$ we model decreasing sinusoids.

Finally, the glue factor is defined by Figure 7.10a, where

$$\mathbf{H}_{\mathcal{M}} \triangleq \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{h}_{\mathcal{M}}^{(1)} \\ \vdots \\ \boldsymbol{h}_{\mathcal{M}}^{(N)} \end{bmatrix}, \tag{7.88}$$

with $\boldsymbol{h}_{\mathcal{M}}^{(n)}$ constructed as in (7.85) for $\mathcal{M} = \mathcal{A}, \mathcal{B}$.

7.5.2 Different Glue Factor Parameterizations

Although both $f_k^{\mathcal{A}}$ and $f_{k'}^{\mathcal{B}}$ are conditional PDFs, the defined model does not comply with any of the strictly conditional cases in Figure 7.3 (cf. Section 7.2.3). In the present case, re-normalization can be accomplished easily by glue factor re-parameterization as explained in Section 7.2.3.

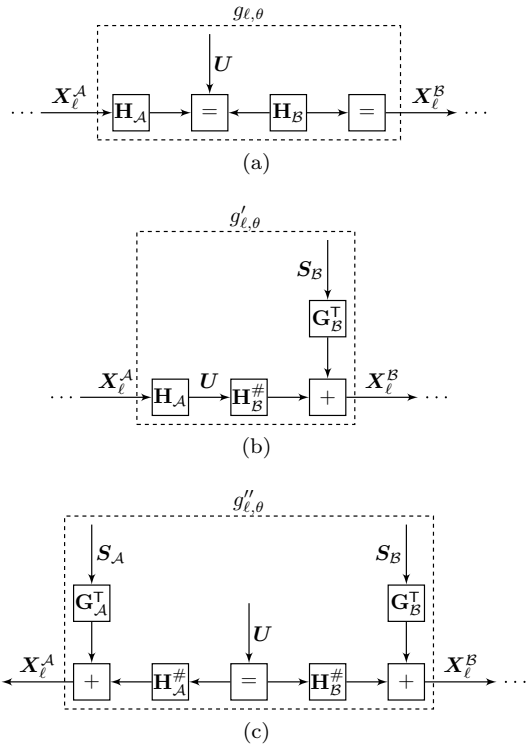


Figure 7.10: Glue factors for two-sided sinusoidal pulses.

In the following we use this technique to construct two cases of constant normalization.

We start with the target parameterization of 7.3b. First, we assume that $f_{K+1}(\boldsymbol{x}_K^{\mathcal{B}}) \propto 1$ and that f_0 is a (potentially scaled) PDF; in our example, the steady-state message may be a good candidate for the latter. Second, note that the glue factor does not depend on ℓ and hence (7.29) can be applied.

In the present case, the re-parameterization can, however, be accomplished by using [65, Table 5] to reverse the matrix multiplication by $\mathbf{H}_{\mathcal{B}}$. The resulting glue factor is shown in 7.10b, in which $\mathbf{H}_{\mathcal{B}}^{\#}$ is the pseudo-inverse of $\mathbf{H}_{\mathcal{B}}$ and $\mathbf{G}_{\mathcal{B}}^{\mathsf{T}}$ is the kernel (nullspace) of $\mathbf{H}_{\mathcal{B}}$. The glue

factor g' indeed is a conditional PDF

$$g'(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{s}_{\mathcal{B}}, \ell) = p\big(\boldsymbol{x}_\ell^{\mathcal{B}} \,\big|\, \boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{s}_{\mathcal{B}}\big)\,, \tag{7.89}$$

which can be verified by inspection since the edge directions indicate conditional PDFs as explained in Section 1.5.2. By identifying $\boldsymbol{\theta} = \boldsymbol{s}_{\mathcal{B}}$ we have established the model parameterization of Figure 7.3b. Note that the substitution of g by g' does change the normalization factor. This change has, however, exactly the same effect as re-normalization.

The parameterization of Figure 7.10b is especially well suited for pulse position estimation as we shall see in Section 7.6.2, Example 7.6, and we will show simulated examples in Section 7.9.

Our second target parameterization is the one shown in Figure 7.3a. The re-parameterization is only feasible if $f_0(\boldsymbol{x}_0^{\mathcal{A}}) \propto 1$ and $f_{K-1}(\boldsymbol{x}_K^{\mathcal{B}}) \propto 1$. First, the autonomous SSM of Figure 7.9 for $\mathcal{M} = \mathcal{A}$ is reversed in time which can easily be done since $\mathbf{A}_{\mathcal{A}}$ is invertible. Second, we substitute the glue factor of Figure 7.10a by the one in Figure 7.10c, in which $\mathbf{H}_{\mathcal{M}}^{\#}$ is the pseudo-inverse of $\mathbf{H}_{\mathcal{M}}$ and $\mathbf{G}_{\mathcal{M}}^{\mathsf{T}}$ is the kernel (nullspace) of $\mathbf{H}_{\mathcal{M}}$ for $\mathcal{M} = \mathcal{A}, \mathcal{B}$. Both alterations change the normalization factor thus achieving the targeted re-normalization. The glue factor now is in the form of a conditional PDF

$$g''(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}}, \boldsymbol{s}_{\mathcal{A}}, \boldsymbol{s}_{\mathcal{B}}, \boldsymbol{u}, \ell) = p\big(\boldsymbol{x}_\ell^{\mathcal{A}}, \boldsymbol{x}_\ell^{\mathcal{B}} \,\big|\, \boldsymbol{s}_{\mathcal{A}}, \boldsymbol{s}_{\mathcal{B}}, \boldsymbol{u}\big)\,, \tag{7.90}$$

where we associate

$$\boldsymbol{\theta} = (\boldsymbol{s}_A, \boldsymbol{s}_B, \boldsymbol{u})\,. \tag{7.91}$$

The glue factor of Figure 7.10c is especially attractive for the generation of a pulse. Specifically, any assignment of $\boldsymbol{\theta}$ completely defines a pulse. Note that the dimensionality of $\boldsymbol{\theta}$ as defined in (7.91) depends on the number N of derivatives to equate for the two sides of the pulse as

$$\dim \boldsymbol{\theta} = n_X^{\mathcal{A}} + n_X^{\mathcal{B}} - 2N\,, \tag{7.92}$$

where $n_X^{\mathcal{M}}$ is the dimensionality of the state in model $\mathcal{M} = \mathcal{A}, \mathcal{B}$.

Figure 7.11 shows 10 example pulses that were generated by setting $\boldsymbol{S}_{\mathcal{A}}$, $\boldsymbol{S}_{\mathcal{B}}$, and $\boldsymbol{U}$ randomly and normalizing the pulse to unit energy. The model parameters used are listed in Table 7.1.

This second re-parameterization of the glue factor (Figure 7.10c) may also be useful for estimating the pulse shape because the latter is defined by $\boldsymbol{\theta}$.

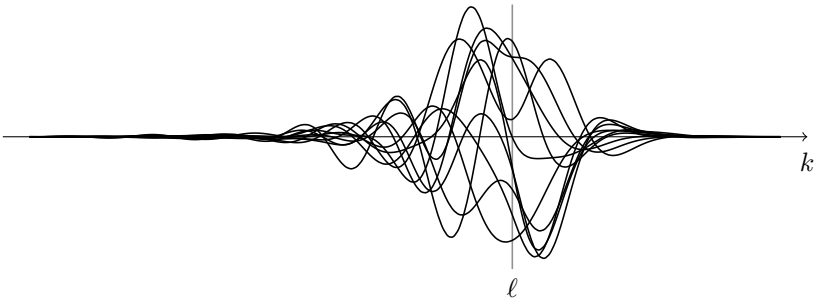


Figure 7.11: 10 examples of unit-energy pulses positioned at ℓ for the two-sided system with parameters given in Table 7.1.

Parameters for model $\mathcal{A}$		Value
$M_{\mathcal{A}}$	Number of sinusoids	4
$\boldsymbol{\omega}^{\mathcal{A}}$	Frequencies	$[0.2, 0.299, 0.447, 0.669]\pi$
$\boldsymbol{\alpha}^{\mathcal{A}}$	Decay factors	[0.8, 0.5769, 0.4160, 0.3]
Parameters for model $\mathcal{B}$		
$M_{\mathcal{B}}$	Number of sinusoids	3
$\boldsymbol{\omega}^{\mathcal{B}}$	Frequencies	$[0.25, 0.397, 0.630]\pi$
$\boldsymbol{\alpha}^{\mathcal{B}}$	Decay factors	[-0.7, -0.75, -0.8]
N	Number of derivatives	4
T	Sampling interval	0.2

Table 7.1: Model parameters for two-sided pulses in Figure 7.11.

In principle, one can envisage all three learning scenarios in Section 7.3.1. Most notably, Scenario (c) for online learning is feasible although we always must have different SSM parameters for $\mathcal{A}$ and $\mathcal{B}$. Furthermore, a combination of an online learning scenario and a pulse detection scenario (cf. Section 7.8) can be considered.

As an alternative to the two-sided model for pulses we can envisage a one sided model, a sum of exponentially decreasing sinusoids towards the past. This setting is well suited for likelihood filtering with forward message passing only, i.e., with $D_1 = D_2 = 0$. This setting is used in [28, 29] throughout. Also in this setting, we may want to set some derivatives at the ending location of the pulse to zero. The two versions

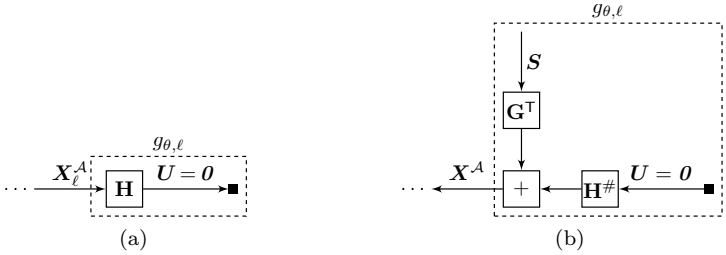


Figure 7.12: Glue factors for one-sided sinusoidal pulses.

of glue factors that accomplish this are shown in Figure 7.12. Equivalent remarks concerning learning and detection as for two-sided pulses apply to this case too.

7.6 Estimating Glue Factor Positions

In this section we consider estimation of the glue factor position ℓ from observed data. Strictly speaking this makes only sense for a finite block of data. An online algorithm that solves a related problem for an indefinitely extended stream of data is presented in Section 7.7. Throughout this section, we usually assume that the glue factor does not depend on any parameter $\boldsymbol{\theta}$. Extensions to include such parameters are, however, feasible.

In this section, we start by elaborating on principles of ML estimation of the glue factor position. Next, we identify cases in which the normalization factor $\zeta^{\text{ℵ}}(\boldsymbol{\theta})$ does not depend on ℓ. A straightforward generalization to MAP estimation of a model-change position is presented. Finally, we sketch an approximate method for estimating the time positions of several glue factors.

7.6.1 Principles

Given a block of data $\tilde{\boldsymbol{y}} \triangleq (\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_K)$ we consider ML estimation of ℓ in the model family of Figure 7.1. The likelihood function in Equation (7.13) allows us to compute $\hat{\ell}_{\text{ML}}$ in Step (c) in the "offline likelihood computation"

algorithm in Section 7.2.2 as

$$\hat{\ell}_{\mathrm{ML}} = \underset{\ell\in\{0,\ldots,K\}}{\operatorname{argmax}}\; p(\tilde{\boldsymbol{y}}|\ell) \tag{7.93}$$

$$= \underset{\ell\in\{0,\ldots,K\}}{\operatorname{argmax}}\; \beta_{U_\ell}/\beta^{\circ}_{U_\ell}, \tag{7.94}$$

where we recall that $\boldsymbol{U}_\ell$ is either $\boldsymbol{X}^{\mathcal{A}}_\ell$, $\boldsymbol{X}^{\mathcal{B}}_\ell$, or any edge within the glue factor. We realize that, in general, we have to abide with scale factor computation. As mentioned in Section 6.4, message scale factors can quickly tend to extreme values.

This motivates us to consider an alternative way to (7.93) for ML estimation of ℓ as

$$\hat{\ell}_{\mathrm{ML}} = \underset{\ell\in\{0,\ldots,K\}}{\operatorname{argmax}}\; \mathrm{LLR}_\ell\,, \tag{7.95}$$

based on LLRs

$$\mathrm{LLR}_\ell \triangleq \log\frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_\ell)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\not\ell})}\,. \tag{7.96}$$

In the above, $\mathcal{H}_\ell$ is the hypothesis for the glue factor sitting at position ℓ, i.e.,

$$p(\tilde{\boldsymbol{y}}|\mathcal{H}_\ell) \triangleq p(\tilde{\boldsymbol{y}}|\ell)\,, \tag{7.97}$$

and the signal model under the null hypothesis $\mathcal{H}_{\not\ell}$ can be chosen arbitrarily as long as the likelihood under this hypothesis does not depend on ℓ. Note that for special cases, the appropriate choice of $\mathcal{H}_{\not\ell}$ can influence the resulting expression for the LLR considerably.

For the general case at hand we choose the null hypothesis $\mathcal{H}_{\not\ell} \triangleq \mathcal{H}_{\mathcal{A}}$, which corresponds to the model represented by the factor graph in Figure 7.13. In this model, we attribute the data to the model $\mathcal{A}$ solely. Let us remark that if f_0 is a (potentially scaled) prior PDF and the factors $f^{\mathcal{A}}_1, \ldots, f^{\mathcal{A}}_\ell$ are (potentially scaled) conditional PDFs in the sense as explained in Section 1.5.2, then the normalization factor $\zeta^{\not\ell}_{\mathcal{A}}$ for this factor graph depends on neither ℓ nor K and hence is constant in both the offline and the online scenario.

In order to distinguish quantities computed in the factor graph of Figure 7.1 from quantities computed for $\mathcal{H}_{\mathcal{A}}$ in the factor of Figure 7.13 we denote the latter by $(\cdot)|\mathcal{A}$; e.g. $\boldsymbol{X}_k|\mathcal{A}$ denotes the edge $\boldsymbol{X}_k$ in Figure 7.13.

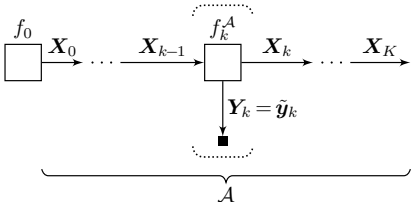


Figure 7.13: Model for hypothesis $\mathcal{H}_{\mathcal{A}}$.

We now can write the LLR of Equation (7.96) as

$$\mathrm{LLR}_\ell = \log \frac{\beta_{U_\ell}\, \beta^{\circ}_{X_\ell|\mathcal{A}}}{\beta^{\circ}_{U_\ell}\, \beta_{X_\ell|\mathcal{A}}}\,, \tag{7.98}$$

where $\boldsymbol{U}_\ell$ is either $\boldsymbol{X}_\ell^{\mathcal{A}}$, $\boldsymbol{X}_\ell^{\mathcal{B}}$ or any edge within the glue factor in Figure 7.1. In (7.98), neither $\beta_{X_\ell|\mathcal{A}}$ nor $\beta^{\circ}_{X_\ell|\mathcal{A}}$ does not depend on ℓ (cf. Theorem 6.1). For the general case, we nevertheless keep these terms because some of the message scale factors cancel. Specifically, the LLR of Equation (7.98) can be written as

$$\begin{aligned}\mathrm{LLR}_\ell = \log \frac{\overleftarrow{\gamma}_{X_\ell^{\mathcal{B}}}\, \overleftarrow{\gamma}^{\circ}_{X_\ell|\mathcal{A}}}{\overleftarrow{\gamma}^{\circ}_{X_\ell^{\mathcal{B}}}\, \overleftarrow{\gamma}_{X_\ell|\mathcal{A}}} + \log \frac{\int \overleftarrow{\nu}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x})\, g(\boldsymbol{0}, \boldsymbol{x}, \ell)\, \mathrm{d}\boldsymbol{x}}{\int \overleftarrow{\nu}^{\circ}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x}')\, g(\boldsymbol{0}, \boldsymbol{x}', \ell)\, \mathrm{d}\boldsymbol{x}'} \\ + \log \frac{\int \nu_{X_\ell^{\mathcal{A}}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}\ \int \nu^{\circ}_{X_\ell|\mathcal{A}}(\boldsymbol{x}')\, \mathrm{d}\boldsymbol{x}'}{\int \nu^{\circ}_{X_\ell^{\mathcal{A}}}(\boldsymbol{x}'')\, \mathrm{d}\boldsymbol{x}''\ \int \nu_{X_\ell|\mathcal{A}}(\boldsymbol{x}''')\, \mathrm{d}\boldsymbol{x}'''}\,.\end{aligned} \tag{7.99}$$

A similar expression with β-type scale factors can be formulated. The proof of (7.99) is given in Appendix D.2.

In Equation (7.99) we note that message scale factors need not be computed in the forward pass. In contrast, it is in general necessary to compute the message scale factors in the backward pass. The reason for this is that for $k = \ell + 1, \ldots, K$ the models in the two hypotheses are different. If, however, the two models are the same, i.e. if $\mathcal{A} = \mathcal{B}$, then the scale factors in the first term of (7.99) cancel too and no scale factor needs to be computed at all.

The LLR approach in Equation (7.98) to ML estimation of ℓ when compared with the approach of (7.94) has thus potentially the advantage that less message scale factors have to be computed, even in the most general case.

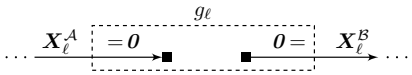


Figure 7.14: Glue factor for noise-only hypothesis $\mathcal{H}_{\mathcal{N}}$.

In certain cases, it is worth while to consider a second alternative for the null hypothesis $\mathcal{H}_{\not 0}$. Specifically, we define $\mathcal{H}_{\mathcal{N}}$ to be the hypothesis of a noise-only model under which the likelihood can be written as

$$p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\mathcal{N}}) = \prod_{k=1}^{K} \mathcal{N}(\tilde{\boldsymbol{y}}_k|\,\boldsymbol{0}, \mathbf{V}_Z)\,. \tag{7.100}$$

Furthermore, we restrict the models $\mathcal{A}$ and $\mathcal{B}$ to be such that (7.100) can be achieved by positioning the glue factor depicted in Figure 7.14 at any position ℓ. In particular, this is the case if both models $\mathcal{A}$ and $\mathcal{B}$ are autonomous linear SSMs and if $\mathbf{V}_Z^{\mathcal{A}} = \mathbf{V}_Z^{B}$.

In these special cases we can choose $\mathcal{H}_{\not 0} \triangleq \mathcal{H}_{\mathcal{N}}$ and the corresponding LLRs

$$\mathrm{LLR}_\ell = \log \frac{\beta_{U_\ell} \overrightarrow{\gamma}^{\circ}_{X_\ell^{\mathcal{A}}} \overleftarrow{\gamma}^{\circ}_{X_\ell^{\mathcal{B}}}}{\beta^{\circ}_{U_\ell} \overrightarrow{\gamma}_{X_\ell^{\mathcal{A}}} \overleftarrow{\gamma}_{X_\ell^{\mathcal{B}}}} \tag{7.101}$$

simplify from (7.99) to

$$\mathrm{LLR}_\ell = \log \frac{\int \overleftarrow{\nu}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x})\, g(\boldsymbol{0}, \boldsymbol{x}, \ell)\, \mathrm{d}\boldsymbol{x}}{\int \overleftarrow{\nu}^{\circ}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x}')\, g(\boldsymbol{0}, \boldsymbol{x}', \ell)\, \mathrm{d}\boldsymbol{x}'} + \log \frac{\int \nu_{X_\ell^{\mathcal{A}}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}}{\int \nu^{\circ}_{X_\ell^{\mathcal{A}}}(\boldsymbol{x}')\, \mathrm{d}\boldsymbol{x}'}\,. \tag{7.102}$$

Note that all message scale factors have cancelled. Equation (7.102) is proved in Appendix D.2.

We henceforth restrict the setting to linear time-invariant (LTI) SSMs for $\mathcal{A}$ and $\mathcal{B}$ as shown in Figure 7.15. The generalization to linear time-varying (LTV) SSMs is straightforward. We take a closer look at the first term in Equation (7.99). Specifically, we define

$$\overleftarrow{\gamma}_k \triangleq \frac{\overleftarrow{\gamma}_{X_k^{\mathcal{B}}} \overleftarrow{\gamma}^{\circ}_{X_k|\mathcal{A}}}{\overleftarrow{\gamma}^{\circ}_{X_k^{\mathcal{B}}} \overleftarrow{\gamma}_{X_k|\mathcal{A}}} \tag{7.103}$$

for $k = \ell, \ldots, K$. The quantity $\ln \overleftarrow{\gamma}_k$ can be computed recursively in the

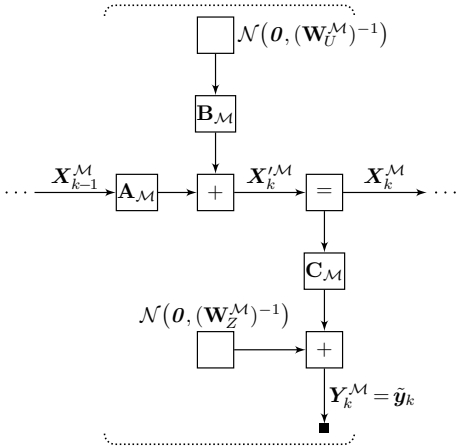


Figure 7.15: Linear time-invariant (LTI) model for $\mathcal{M} = \mathcal{A}, \mathcal{B}$.

backward pass as

$$\ln \overleftarrow{\gamma}_{K+1} = 0\,, \tag{7.104}$$

$$\begin{aligned} \ln \overleftarrow{\gamma}_{k-1} = \ln \overleftarrow{\gamma}_k + \frac{1}{2} \ln \frac{\det \mathbf{W}_Z^{\mathcal{B}} \, \det \mathbf{W}_{\mathcal{B}}^{\circ} \, \det \mathbf{W}_{\mathcal{A}}}{\det \mathbf{W}_Z^{\mathcal{A}} \, \det \mathbf{W}_{\mathcal{A}}^{\circ} \, \det \mathbf{W}_{\mathcal{B}}} \\ - \tilde{\boldsymbol{y}}_k^{\mathsf{T}} \big(\mathbf{W}_Z^{\mathcal{B}} - \mathbf{W}_Z^{\mathcal{A}} \big) \tilde{\boldsymbol{y}}_k / 2 \\ + \boldsymbol{m}_{\mathcal{B}}^{\mathsf{T}} \mathbf{W}_{\mathcal{B}} \boldsymbol{m}_{\mathcal{B}} / 2 - \boldsymbol{m}_{\mathcal{A}}^{\mathsf{T}} \mathbf{W}_{\mathcal{A}} \boldsymbol{m}_{\mathcal{A}} / 2\,, \end{aligned} \tag{7.105}$$

for $k = K, \ldots, \ell$, where

$$\mathbf{W}_{\mathcal{M}} \triangleq \mathbf{W}_U^{\mathcal{M}} + \mathbf{B}_{\mathcal{M}}^{\mathsf{T}} \overleftarrow{\mathbf{W}}_{X_k'^{\mathcal{M}}} \mathbf{B}_{\mathcal{M}}\,, \tag{7.106}$$

$$\mathbf{W}_{\mathcal{M}} \boldsymbol{m}_{\mathcal{M}} \triangleq \mathbf{B}_{\mathcal{M}}^{\mathsf{T}} \overleftarrow{\mathbf{W}}_{X_k'^{\mathcal{M}}} \overleftarrow{\boldsymbol{m}}_{X_k'^{\mathcal{M}}}\,, \tag{7.107}$$

$$\mathbf{W}_{\mathcal{M}}^{\circ} \triangleq \mathbf{W}_U^{\mathcal{M}} + \mathbf{B}_{\mathcal{M}}^{\mathsf{T}} \overleftarrow{\mathbf{W}}_{X_k'^{\mathcal{M}}}^{\circ} \mathbf{B}_{\mathcal{M}}\,, \tag{7.108}$$

for $\mathcal{M} = \mathcal{A}, \mathcal{B}$. We recall that $\mathbf{W}_U^{\mathcal{M}}$ and $\mathbf{W}_Z^{\mathcal{M}}$ are the state noise and observation noise precision matrices for model $\mathcal{M}$ respectively. Also, note the definition of the edge $\boldsymbol{X}_k'^{\mathcal{M}}$ in Figure 7.15. The recursion (7.105) is proved in Appendix D.3. We highlight that this recursion can only be achieved by means of Rule (IV.3) for the composite addition and multiplication block.

In certain cases it can occur that $\beta_{U_\ell}^{\circ}$ tends to infinity. In all such cases considered in this chapter, the reason for this is that some or all of the

hidden variables should be treated as parameters. One way of dealing with these parameters is to turn them into random variables by assuming a prior PDF, e.g. a uninformative Gaussian prior, whose variance tends to infinity. The viewpoint of Equation (7.98) is more attractive with this respect, because the respective (inverse) covariance matrices cancel before taking the limit.

7.6.2 Cases with Constant Normalization Factor

Recall the case in which the normalization factor $\beta^{\circ}_{U_\ell}$ does not depend on ℓ. Such a case arises either, in a strictly conditional setting (cf. Section 7.2) or by re-normalizing via re-parameterization of the glue factor and potentially the SSMs $\mathcal{A}$ and $\mathcal{B}$ (cf. Section 7.2.3).

In such a case, Equation (7.94) simplifies to $\hat{\ell}_{\mathrm{ML}} = \operatorname{argmax}_\ell \beta_{U_\ell}$ and there is no need to compute $\beta^{\circ}_{U_\ell}$. Furthermore, for LLR-based ML estimation, the LLR in Equation (7.98) can now be written as

$$\begin{aligned} \mathrm{LLR}_\ell &= \log \frac{\beta_{U_\ell}}{\beta_{X_\ell|\mathcal{A}}} + \text{const} && (7.109)\\ &= \log \frac{\overleftarrow{\gamma}_{X_\ell^{\mathcal{B}}}}{\overleftarrow{\gamma}_{X_\ell|\mathcal{A}}} + \log \int \overleftarrow{\nu}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x})\, g(\boldsymbol{0}, \boldsymbol{x}, \ell)\, \mathrm{d}\boldsymbol{x} \\ &\quad + \log \frac{\int \nu_{X_\ell^{\mathcal{A}}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}}{\int \nu_{X_\ell|\mathcal{A}}(\boldsymbol{x}')\, \mathrm{d}\boldsymbol{x}'} + \text{const}\,. && (7.110) \end{aligned}$$

Similarly, the LLR with respect to the noise-only hypothesis in Equation (7.101) can be written as

$$\begin{aligned} \mathrm{LLR}_\ell &= \log \frac{\beta_{U_\ell}}{\overrightarrow{\gamma}_{X_\ell^{\mathcal{A}}}\, \overleftarrow{\gamma}_{X_\ell^{\mathcal{B}}}} + \text{const} && (7.111)\\ &= \log \int \nu_{X_\ell^{\mathcal{A}}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} + \log \int \overleftarrow{\nu}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x})\, g(\boldsymbol{0}, \boldsymbol{x}, \ell)\, \mathrm{d}\boldsymbol{x} \\ &\quad + \text{const}\,. && (7.112) \end{aligned}$$

Furthermore, in the case of LTI SSMs, the quantity

$$\overleftarrow{\gamma}'_k \triangleq \overleftarrow{\gamma}_{X_k^{\mathcal{B}}} / \overleftarrow{\gamma}_{X_k|\mathcal{A}} \tag{7.113}$$

can be computed recursively for $k = K, \dots, \ell$ as

$$\ln \overleftarrow{\gamma}'_{K+1} = 0\,, \tag{7.114}$$

$$\begin{aligned} \ln \overleftarrow{\gamma}'_{k-1} = \ln \overleftarrow{\gamma}'_k + \frac{1}{2} \ln \frac{\det \mathbf{W}_Z^{\mathcal{B}} \det \mathbf{W}_U^{\mathcal{B}} \det \mathbf{W}_{\mathcal{A}}}{\det \mathbf{W}_Z^{\mathcal{A}} \det \mathbf{W}_U^{\mathcal{A}} \det \mathbf{W}_{\mathcal{B}}} \\ - \tilde{\boldsymbol{y}}_k^{\mathsf{T}} \big(\mathbf{W}_Z^{\mathcal{B}} - \mathbf{W}_Z^{\mathcal{A}}\big) \tilde{\boldsymbol{y}}_k / 2 \\ + \boldsymbol{m}_{\mathcal{B}}^{\mathsf{T}} \mathbf{W}_{\mathcal{B}} \boldsymbol{m}_{\mathcal{B}} / 2 - \boldsymbol{m}_{\mathcal{A}}^{\mathsf{T}} \mathbf{W}_{\mathcal{A}} \boldsymbol{m}_{\mathcal{A}} / 2\,, \end{aligned} \tag{7.115}$$

with the definitions (7.106) and (7.107). The recursion (7.115) is proved in Appendix D.3.

In the following we give some examples of glue factor position estimation for models with constant normalization with respect to the glue factor position ℓ. In some of the examples, a clever choice for the null hypothesis $\mathcal{H}_\ell$ leads to simplified expressions for the LLR of (7.96).

Example 7.3: Locating a Model Parameter Change
In this example, we assume that the observed data $\tilde{\boldsymbol{y}}_1, \dots, \tilde{\boldsymbol{y}}_K$ has been generated by a LTI SSM model (3.7), whose parameters have changed at some unknown time ℓ. Such a parameter change can, e.g., be a change in the covariance matrix of the state noise or the observation noise, or a change of the system poles. The resulting model family thus is of the form in Figure 7.1 with $\mathcal{A}$ and $\mathcal{B}$ both being LTI SSMs (Figure 7.15) of the same order but with differing parameters. The glue factor is a simple connection of the states $\boldsymbol{X}_\ell^{\mathcal{A}} = \boldsymbol{X}_\ell^{\mathcal{B}}$. Assuming that all the system parameters are known, ML estimation of ℓ from observed data thus can be done by the offline likelihood computation method of Section 7.2.2, followed by (7.95) where we choose $\boldsymbol{U}_\ell \triangleq \boldsymbol{X}_\ell$.

In this example, the second term in (7.110) evaluates to

$$\ln \int \overleftarrow{\nu}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x})\, g(\boldsymbol{0}, \boldsymbol{x}, \ell) = \ln \overleftarrow{\nu}_{X_\ell^{\mathcal{B}}}(\boldsymbol{0}) = 0 \tag{7.116}$$

such that the LLR (7.110) can be written as

$$\begin{aligned} \mathrm{LLR}_\ell = \ln \frac{\overleftarrow{\gamma}_{X_\ell^{\mathcal{A}}}}{\overleftarrow{\gamma}_{X_{\ell|\mathcal{A}}}} + \frac{1}{2} \ln \frac{\det \mathbf{W}_{X_{\ell|\mathcal{A}}}}{\det \mathbf{W}_{X_\ell^{\mathcal{A}}}} \\ + \frac{1}{2} \Big(\boldsymbol{m}_{X_\ell^{\mathcal{A}}}^{\mathsf{T}} \mathbf{W}_{X_\ell^{\mathcal{A}}} \boldsymbol{m}_{X_\ell^{\mathcal{A}}} - \boldsymbol{m}_{X_{\ell|\mathcal{A}}}^{\mathsf{T}} \mathbf{W}_{X_{\ell|\mathcal{A}}} \boldsymbol{m}_{X_{\ell|\mathcal{A}}} \Big)\,. \end{aligned} \tag{7.117}$$

where we have made use of (6.16).

Note that in the case at hand, $\boldsymbol{X}_K^{\mathcal{B}}$ is an open edge, carrying a neutral message $\overleftarrow{\mu}_{X_K^{\mathcal{B}}}$ with $\overleftarrow{\mathbf{W}}_{X_K^{\mathcal{B}}} = \mathbf{0}$, $\overleftarrow{\boldsymbol{m}}_{X_K^{\mathcal{B}}} = \boldsymbol{0}$, and $\overleftarrow{\gamma}_{X_K^{\mathcal{B}}} = 1$. The computation of $\overleftarrow{\mathbf{W}}_{X_\ell^{\mathcal{A}}}$ and $\overleftarrow{\mathbf{W}}_{X_\ell^{\mathcal{A}}}\overleftarrow{\boldsymbol{m}}_{X_\ell^{\mathcal{A}}}$ is done by the standard "information filter" backward message passing using update rules in [65, Table 4] for the composite block (a) in Figure 6.1. The computation of $\ln \overleftarrow{\gamma}'_k$ as defined in (7.113) can be done alongside with the recursion (7.115). ◊

Example 7.4: Locating a Parameter Change of an Autonomous Model

As a special case of Example 7.3, we assume that the SSMs $\mathcal{A}$ and $\mathcal{B}$ are autonomous. The LLR of (7.117) from the previous example still applies. In this case, however, the update rules (II.2) and (II.5) suffice to formulate the recursion (7.115) for computing the quantity $\overleftarrow{\gamma}'_k$ as

$$\ln \overleftarrow{\gamma}'_{k-1} = \ln \overleftarrow{\gamma}'_k + \frac{1}{2} \ln \frac{\det \mathbf{W}_Z^{\mathcal{B}}}{\det \mathbf{W}_Z^{\mathcal{A}}} - \tilde{\boldsymbol{y}}_k^{\mathsf{T}} (\mathbf{W}_Z^{\mathcal{B}} - \mathbf{W}_Z^{\mathcal{A}}) \tilde{\boldsymbol{y}}_k / 2 \,. \tag{7.118}$$

If the observation noise covariance matrices of both models $\mathcal{A}$ and $\mathcal{B}$ are the same, i.e., if $\mathbf{V}_Z^{\mathcal{A}} = \mathbf{V}_Z^{\mathcal{B}}$ then we can choose the noise-only hypothesis $\mathcal{H}_{\text{ł}} = \mathcal{H}_{\mathcal{N}}$. Clearly, in this case, a model for this hypothesis is achieved by the glue factor in Figure 7.14. Hence, the simplified LLR of Equation (7.112) applies. As in the previous example the second term in (7.112) simplifies as in (7.116) and we can use (6.16) to write the LLR of (7.112) as

$$\mathrm{LLR}_\ell = \boldsymbol{m}_{X_\ell^{\mathcal{A}}}^{\mathsf{T}} \mathbf{W}_{X_\ell^{\mathcal{A}}} \boldsymbol{m}_{X_\ell^{\mathcal{A}}} / 2 - \ln \det \mathbf{W}_{X_\ell^{\mathcal{A}}} \,. \qquad \text{◊}$$

Example 7.5: Locating an Additional Input

Consider a LTI SSM (3.7) with given system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and covariance matrices $\mathbf{V}_U$, $\mathbf{V}_Z$. Assume that at some unknown time $\ell \in \{0, \ldots, K\}$, an additional input has been applied, i.e. the state update equation at time $k = \ell$ changes to

$$\boldsymbol{X}_\ell = \mathbf{A}\, \boldsymbol{X}_{\ell-1} + \mathbf{B}\, (\boldsymbol{U}_\ell + \boldsymbol{U}'_\ell) \,, \tag{7.119}$$

where $\boldsymbol{U}'_\ell$ is the additional input. Given observations $\boldsymbol{Y}_k = \tilde{\boldsymbol{y}}_k$ for $k = 1, \ldots, K$ we would like to make a ML estimate of ℓ.

We can cast this problem into the form of Figure 7.3b as follows. First, we set the models equal, i.e. $\mathcal{A} = \mathcal{B}$. Second, we define the factors $f_k^{\mathcal{A}}$ and $f_K^{\mathcal{B}}$ as in Figure 7.16, i.e. as a "shuffled" version of the standard factor

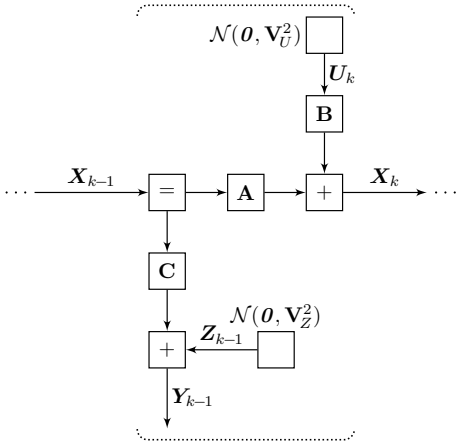


Figure 7.16: Shuffled linear time-invariant (LTI) state-space model (SSM).

graph representation in Figure 3.2 of a LTI SSM. Note that apart from the graph borders (at $k = 0$ and $k = K$) and a relabeling of the hidden variables, Figures 7.16 and 3.2 are the same. Finally, this shuffling allows us to model the additional input by the glue factor of Figure 7.17a.

Now we consider ML estimation based on LLRs as in (7.96). In principle, we could choose $\mathcal{H}_{\not 0} = \mathcal{H}_{\mathcal{A}}$, but for this example there is a more adequate choice. Recall that the model for $\mathcal{H}_{\mathcal{A}}$ is depicted in Figure 7.13. Now, we can insert the "glue factor" of Figure 7.17b into edge $\boldsymbol{X}_k$ at any position k in this model without changing neither the likelihood nor the normalization factor. (This type of neutral modification is treated in Appendix C.1.) We choose this modified factor graph with the glue factor at position ℓ as our model for the null hypothesis $\mathcal{H}_{\not 0}$. Hence, the LLR in (7.96) can be written as

$$\mathrm{LLR}_\ell = \ln \frac{\beta_{U'_\ell}}{\beta_{U''_\ell}} = \ln \frac{\overleftarrow{\beta}_{U'_\ell}}{\overleftarrow{\gamma}_{U''_\ell}} \tag{7.120}$$

$$= \frac{1}{2} \ln \frac{(2\pi)^n}{\det \overleftarrow{\mathbf{W}}_{U'_\ell}} + \overleftarrow{\boldsymbol{m}}_{U'_\ell}^{\mathsf{T}} \overleftarrow{\mathbf{W}}_{U'_\ell} \overleftarrow{\boldsymbol{m}}_{U'_\ell} / 2 \,, \tag{7.121}$$

where n is the dimensionality of $\boldsymbol{U}'_\ell$. For (7.121) we have used the fact that $\overleftarrow{\mu}_{U'_\ell}(\boldsymbol{u}) = \overleftarrow{\mu}_{U''_\ell}(\boldsymbol{u})$ and we have applied the conversion (6.16).

(a) Additional input. (b) No additional input.

Figure 7.17: Glue factors for locating an additional input into a linear time-invariant (LTI) state-space model (SSM).

In case we have forward and backward steady state messages $\vec{\mu}_{X_0}$ and $\overleftarrow{\mu}_{X_K}$, then $\overleftarrow{\mathbf{W}}_{U'_\ell}$ does not depend on ℓ. Furthermore, in this case, it is evident that for a scalar input ($n = 1$) we have

$$\hat{\ell}_{\mathrm{ML}} = \operatorname*{argmax}_{\ell \in \{0,\dots,K\}} |\hat{u}_\ell| \,, \tag{7.122}$$

which is based on input estimation alone.

Also, it is evident how to work out the LLR in (7.109) in the case of a general input $\boldsymbol{U}'_\ell$ to the state, i.e., if we substitute (7.119) by

$$\boldsymbol{X}_\ell = \mathbf{A}\,\boldsymbol{X}_{\ell-1} + \mathbf{B}\,\boldsymbol{U}_\ell + \boldsymbol{U}'_\ell \,. \qquad \lozenge$$

Example 7.6: Locate a Pulse
We recall the setting of Section 7.5, in which we have modelled a pulse as a sum of sinusoids, decaying both towards the future and towards the past. Furthermore, we have formulated a constant normalization case by defining the factors in Figure 7.3b as follows. The factors $f_k^{\mathcal{M}}$ are autonomous SSMs as in Figure 7.9, potentially with differing parameters for $\mathcal{M} = \mathcal{A}, \mathcal{B}$. The initial factor f_0 is a Gaussian prior PDF $\mathcal{N}(\boldsymbol{0}, \vec{\mathbf{V}}_X)$ where $\vec{\mathbf{V}}_X$ is the steady-state covariance matrix for forward message passing in model $\mathcal{A}$. Finally, for the glue factor, we consider two variants

a) The pulse shape is unknown. The corresponding glue factor is depicted in Figure 7.10b.

b) The pulse shape is known. The corresponding glue factor is depicted in Figure 7.18.

Figure 7.18: Glue factor for the hypothesis $\mathcal{H}_\ell$ for fixed pulse shape.

To understand the glue factor in Figure 7.18 for Variant (b) we recall that a pulse is completely described by specifying $\boldsymbol{S}_{\mathcal{A}} = \tilde{\boldsymbol{s}}_{\mathcal{A}}$, $\boldsymbol{S}_B = \tilde{\boldsymbol{s}}_{\mathcal{B}}$, and $\boldsymbol{U} = \tilde{\boldsymbol{u}}$ in the glue factor of Figure 7.10c. These three values define corresponding states

$$\tilde{\boldsymbol{x}}_{\mathcal{A}} = \mathbf{H}_{\mathcal{A}}^{\#}\tilde{\boldsymbol{u}} + \mathbf{G}_{\mathcal{A}}^{\mathsf{T}}\tilde{\boldsymbol{s}}_{\mathcal{A}} \tag{7.123}$$

$$\tilde{\boldsymbol{x}}_{\mathcal{B}} = \mathbf{H}_{B}^{\#}\tilde{\boldsymbol{u}} + \mathbf{G}_{\mathcal{B}}^{\mathsf{T}}\tilde{\boldsymbol{s}}_{\mathcal{B}}\,. \tag{7.124}$$

For the present case, if we assume $\sigma_{\mathcal{A}}^2 = \sigma_{\mathcal{B}}^2$, then in the LLR of (7.96) we can choose $\mathcal{H}_{\ell\!\!\!/} = \mathcal{H}_{\mathcal{N}}$ where we recall that $\mathcal{H}_{\mathcal{N}}$ is the hypothesis for the noise-only signal model. Hence, the LLR of Equation (7.112) can be applied for both variants resulting in

a)
$$\begin{aligned}\mathrm{LLR}_\ell = {} & \boldsymbol{m}_{X_\ell^{\mathcal{A}}}^{\mathsf{T}}\mathbf{W}_{X_\ell^{\mathcal{A}}}\boldsymbol{m}_{X_\ell^{\mathcal{A}}}/2 + \boldsymbol{m}^{\mathsf{T}}\mathbf{W}\boldsymbol{m}/2 \\ & - \big(\ln\det\mathbf{W} + \ln\det\mathbf{W}_{X_\ell^{\mathcal{A}}}\big)/2 + \mathrm{const}\,,\end{aligned} \tag{7.125}$$

b)
$$\begin{aligned}\mathrm{LLR}_\ell = {} & \tilde{\boldsymbol{x}}_{\mathcal{A}}^{\mathsf{T}}\overrightarrow{\mathbf{W}}_{X_\ell^{\mathcal{A}}}\overrightarrow{\boldsymbol{m}}_{X_\ell^{\mathcal{A}}} + \tilde{\boldsymbol{x}}_{\mathcal{B}}^{\mathsf{T}}\overleftarrow{\mathbf{W}}_{X_\ell^{\mathcal{B}}}\overleftarrow{\boldsymbol{m}}_{X_\ell^{\mathcal{B}}} \\ & - \tilde{\boldsymbol{x}}_{\mathcal{A}}^{\mathsf{T}}\overrightarrow{\mathbf{W}}_{X_\ell^{\mathcal{A}}}\tilde{\boldsymbol{x}}_{\mathcal{A}}/2 - \tilde{\boldsymbol{x}}_{\mathcal{B}}^{\mathsf{T}}\overleftarrow{\mathbf{W}}_{X_\ell^{\mathcal{B}}}\tilde{\boldsymbol{x}}_{\mathcal{B}}/2\,,\end{aligned} \tag{7.126}$$

respectively, where

$$\mathbf{W} \triangleq \mathbf{G}_{\mathcal{B}}\overleftarrow{\mathbf{W}}_{X_\ell^{\mathcal{B}}}\mathbf{G}_{\mathcal{B}}^{\mathsf{T}}\,, \tag{7.127}$$

$$\mathbf{W}\boldsymbol{m} \triangleq \mathbf{G}_{\mathcal{B}}\overleftarrow{\mathbf{W}}_{X_\ell^{\mathcal{B}}}\overleftarrow{\boldsymbol{m}}_{X_\ell^{\mathcal{B}}}\,. \tag{7.128}$$

Equations (7.125)–(7.128) are proved in Appendix D.4. ◊

7.6.3 Maximum a Posteriori estimation of a Glue Factor Position

We return to the general question of locating a glue factor, i.e., of estimating the position ℓ within the model family of Figure 7.1. In Section 7.6.1, we have treated ℓ as a parameter, which has lead us to ML estimation.

In many situations we may have prior knowledge about the glue factor position. In these situations, it is more appropriate to treat this position as a random variable L that has a prior PMF $p(\ell)$. In this section we look at the resulting MAP estimation problem.

Given the data $\tilde{\boldsymbol{y}} \triangleq (\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_K)$ and a prior PDF $p(\ell)$ we can compute a MAP estimate of ℓ in Step (c) of the "offline likelihood computation" algorithm in Section 7.2.2 as

$$\hat{\ell}_{\mathrm{MAP}} = \operatorname*{argmax}_{\ell \in \{0,\ldots,K\}} p(\tilde{\boldsymbol{y}}|\ell)\, p(\ell) \tag{7.129}$$

$$= \operatorname*{argmax}_{\ell \in \{0,\ldots,K\}} p(\ell)\, \beta_{U_\ell}/\beta^{\circ}_{U_\ell}\,, \tag{7.130}$$

where $\boldsymbol{U}_\ell$ is either $\boldsymbol{X}^{\mathcal{A}}_\ell$, $\boldsymbol{X}^{\mathcal{B}}_\ell$, or any edge within the glue factor. We can regard Equations (7.129) and (7.130) as a straightforward extension of (7.93) and (7.94), and indeed if $p(\ell)$ is uniform then $\hat{\ell}_{\mathrm{MAP}} = \hat{\ell}_{\mathrm{ML}}$.

For the sake of clarity we recall that the ratio $\beta_{U_\ell}/\beta^{\circ}_{U_\ell}$ in (7.130) originates from (7.13). This latter equation applies in the strictly conditional case, i.e. if the function represented by the factor graph is proportional to a conditional PDF $p(\boldsymbol{x}, \boldsymbol{y}|\ell)$, as well as in the re-normalized case. In this latter case the function represented by the factor graph is proportional to a joint PDF/PMF $p(\boldsymbol{x}, \boldsymbol{y}, \ell)$ but in the ratio $\beta_{U_\ell}/\beta^{\circ}_{U_\ell}$ we have removed any prior knowledge about ℓ. In (7.130) we add such prior knowledge again by the factor $p(\ell)$.

As in Section 7.6.1, we can formulate an alternative to (7.129) by means of a log-ratio

$$\log R_\ell \triangleq \log \frac{p(\tilde{\boldsymbol{y}}|\ell)\, p(\ell)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\text{ø}})} \tag{7.131}$$

for any null hypothesis $\mathcal{H}_{\text{ø}}$ that must not depend on ℓ. For the general case, we may choose $\mathcal{H}_{\text{ø}} = \mathcal{H}_{\mathcal{A}}$ where the model for the hypothesis $\mathcal{H}_{\mathcal{A}}$ is still given by the factor graph depicted in Figure 7.13. Alternatively, in certain cases, the noise-only model $\mathcal{H}_{\mathcal{N}}$ may be chosen for $\mathcal{H}_{\text{ø}}$.

Once we have decided upon the choice for $\mathcal{H}_{\text{ø}}$, the corresponding MAP estimate (7.129) can be written as

$$\hat{\ell}_{\mathrm{MAP}} = \operatorname*{argmax}_{\ell \in \{0,\ldots,K\}} \ln R_\ell \tag{7.132}$$

$$= \operatorname*{argmax}_{\ell \in \{0,\ldots,K\}} \big(\mathrm{LLR}_\ell + \log p(\ell)\big)\,. \tag{7.133}$$

We recall that the expressions for the LLR in (7.133) are given in (7.99) and (7.102) for $\mathcal{H}_{\ell} = \mathcal{H}_{\mathcal{A}}$ and $\mathcal{H}_{\ell} = \mathcal{H}_{\mathcal{N}}$ respectively. In cases of constant normalization, the respective LLRs are (7.110) and (7.112).

Finally, note that all the glue factor position estimation problems described in Examples 7.3–7.6 generalize immediately to situations in which we are given a prior PMF $p(\ell)$ by simply plugging in the respective expressions for the LLRs into (7.133).

7.6.4 Estimation of Multiple Glue Factor Positions

Assume that we are given a signal $\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_K$ in which we would like to locate r model changes. In terms of a factor graph model we can consider a SSM in which r glue factors connect several SSM segments, each of which corresponds to a (potentially unique) model $\mathcal{M}_q$ for $q = 0, \ldots, r$.

The parameter to be estimated is now a vector $\boldsymbol{\ell} \in \mathbb{N}^r$ of glue factor positions. Here, we assume that the number of glue factors and the model parameters are known. In order to compute the likelihood function of $\boldsymbol{\ell}$ the number of sum-product messages to be computed increases exponentially with r and with the signal length K. While this is in principle feasible, it is not very attractive in practice.

Consider the special case in which all the SSM segments contain the same SSM with the same parameters ($\mathcal{M}_q = \mathcal{A}$ for all q), and all the glue factors have the same form. For this case, we propose to use CM for approximate ML estimation of $\boldsymbol{\ell}$ as follows:

a) Set the number of currently modeled glue factors $q = 0$. We thus start with a single segment of one model for the whole block of data.

b) Assume that we have q glue factors in our model and all messages have been computed. Consider each of the $q + 1$ model segments as a single factor graph of the form in Figure 7.7 with f_0 and f_{K+1} being the messages that originate from adjacent segments. In each of this factor graphs make a ML estimate of the position of the glue factor and compute the corresponding likelihood. Let the likelihood in the j-th segment be p_j and the corresponding glue factor position estimate $\hat{i}_j$.

c) Choose to insert a new glue factor into the whole model at position

$\hat{i}_{\hat{j}}$ where $\hat{j} = \operatorname{argmax}_j p_j$. Recompute all the messages in the factor graph that have changed due to this insertion. (Potentially, another step can be inserted here in which all the glue factor positions are re-estimated for fine-adjustment.)

d) If $q < r$ then increment q and go to Step (b). Otherwise stop the algorithm.

This approach can be used to extend Examples 7.5 and 7.6 on input and pulse position estimation.

7.7 Detection-Inspired Estimation of Glue Factor Positions

In the previous section we have treated the estimation of the position ℓ at which the glue factor g_ℓ in the factor graph of Figure 7.1 is most likely. In this section we assume a scenario in which we do not know whether a glue factor is present at all. We hence are facing two problems:

a) A detection problem: Is the glue factor present at all?

b) An estimation problem: If we detect the presence of a glue factor, what is its position?

In an online filtering scenario (cf. Section 7.2.2) we can envisage a localized version of the above, in which we try to solve problems (a) and (b) in parallel to the filtering action.

7.7.1 Principles

We formalize the scenario envisaged by defining the following hypotheses for a block of given data $\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_K$:

$\mathcal{H}_g$: A glue factor is present. (7.134)

$\mathcal{H}_\ell$: The glue factor is at position ℓ. (7.135)

$\mathcal{H}_\mathcal{A}$: No glue factor is present, only model $\mathcal{A}$ is active. (7.136)

The model family for $\mathcal{H}_\ell$ for $\ell = 0, \ldots, K$ is defined by our family of factor graphs including the glue factor g_ℓ as depicted in Figure 7.1:

$$p(\boldsymbol{y}|\mathcal{H}_\ell) = p(\boldsymbol{y}|\ell) \tag{7.137}$$

We assume that the glue factor does not depend on any parameter $\boldsymbol{\theta}$. Extensions to include such parameters are, however, feasible.

We recall that the model for $\mathcal{H}_\mathcal{A}$ is defined by the factor graph in Figure 7.13. The PDF under hypothesis $\mathcal{H}_g$ is defined as

$$p(\boldsymbol{y}|\mathcal{H}_g) \triangleq \sum_{\ell=0}^{K} p(\boldsymbol{y}|\ell)\, p(\ell)\,, \tag{7.138}$$

where $L \in \{0, \ldots, K\}$ is now a random glue factor position and $p(\ell)$ is a prior PMF on L.

Based on these definitions we formulate the LLR for $\mathcal{H}_g$ versus $\mathcal{H}_\mathcal{A}$ as

$$\begin{aligned} \mathrm{LLR} &\triangleq \log \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_g)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_\mathcal{A})} && (7.139)\\ &= \log \frac{\sum_{\ell=1}^{K} p(\tilde{\boldsymbol{y}}|\mathcal{H}_\ell)\, p(\ell)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_\mathcal{A})} && (7.140)\\ &= \log \sum_{\ell=1}^{K} R_\ell\,, && (7.141) \end{aligned}$$

where we repeat the definition (7.131) of R_ℓ for the choice $\mathcal{H}_\ell = \mathcal{H}_\mathcal{A}$ of the null hypothesis as

$$R_\ell \triangleq \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_\ell)\, p(\ell)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_\mathcal{A})}\,. \tag{7.142}$$

Based on this LLR we can devise a test

$$\mathrm{LLR} \overset{?}{\gtrless} \vartheta\,, \tag{7.143}$$

where ϑ is a threshold. If we decide in favor of $\mathcal{H}_g$, i.e. if $\mathrm{LLR} > \vartheta$, then we can formulate the MAP estimate

$$\hat{\ell}_\mathrm{MAP} = \operatorname*{argmax}_{\ell \in \{0,\ldots,K\}} p(\tilde{\boldsymbol{y}}|\mathcal{H}_\ell)\, p(\ell)\,. \tag{7.144}$$

As an alternative to the LLR, we formulate the GLLR

$$\text{GLLR} \triangleq \log \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\hat{\ell}_{\text{ML}}})}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\mathcal{A}})} \tag{7.145}$$

$$= \log \frac{\max_{\ell\in\{0,\dots,K\}} p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\ell})}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\mathcal{A}})} \tag{7.146}$$

$$= \max_{\ell\in\{0,\dots,K\}} \log R_{\ell}\,, \tag{7.147}$$

where ℓ is treated as a parameter with ML estimate

$$\hat{\ell}_{\text{ML}} = \operatorname*{argmax}_{\ell\in\{0,\dots,K\}} p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\ell})\,, \tag{7.148}$$

and R_ℓ is defined in (7.142) for a uniform prior $p(\ell) = 1/K$. Note that we have encountered $\ln R_\ell = \text{LLR}_\ell$ in Equation (7.96) where the LLR is computed with the null hypothesis $\mathcal{H}_{\ell} = \mathcal{H}_{\mathcal{A}}$.

Both the LLR and the GLLR above can be computed via message passing as

$$\text{LLR} = \log \sum_{\ell=0}^{K} p(\ell)\, e^{\text{LLR}_\ell}\,, \tag{7.149}$$

$$\text{GLLR} = \max_{\ell\in\{0,\dots,K\}} \text{LLR}_\ell\,, \tag{7.150}$$

where the quantity LLR_ℓ is given in Equation (7.99), or in cases of constant normalization in Equation (7.110). Note that all of the previously derived LLRs that have been computed with a null hypothesis $\mathcal{H}_{\mathcal{A}}$ can be applied in (7.149) or (7.150). In particular, this applies to the following two examples.

Example 7.7: Detection of a Model Parameter Change
We recall the setting of Example 7.3, in which two LTI SSMs are given, differing only in the parameters for model $\mathcal{A}$ and $\mathcal{B}$. In contrast to Example 7.3, we first have to decide, whether the parameters have changed at all. If we detect a parameter change, we want to estimate of the change point.

If we have a prior PMF $p(\ell)$ on the point of change then we can compute the LLR of Equation (7.149) where the quantity LLR_ℓ is given in (7.117). This LLR is then compared with a threshold ϑ and if the threshold is exceeded then the MAP estimate $\hat{\ell}_{\text{MAP}}$ is computed as in Equation (7.133).

If we do not have any prior information about the point of change then we can compute the GLLR of Equation (7.150) where the quantity LLR_ℓ is still given in (7.117). This GLLR is then compared with a threshold ϑ and if the threshold is exceeded, then the ML estimate $\hat{\ell}_{\mathrm{ML}}$ is simply the argument that maximizes LLR_ℓ. ◊

Example 7.8: Detection of an Input
We recall the setting of Example 7.5, in which a LTI SSM (3.7) is given with the system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and the covariance matrices $\mathbf{V}_U$, $\mathbf{V}_Z$. In contrast to Example 7.5, we first have to decide whether an additional input $\boldsymbol{U}'_\ell$ (modeled as in Equation (7.119)) is present at all. If such an input is detected, we want to estimate its position on the time axis.

As in Example 7.5, we define our model family for $\mathcal{H}_\ell$ by the factor graph in Figure 7.3b with Figures 7.16 and 7.17a inserted, and the model for $\mathcal{H}_\mathcal{A}$ by Figure 7.16 with Figure 7.17b inserted at time position ℓ.

If we have a prior PMF $p(\ell)$ on the time of the additional input then we can compute the LLR of Equation (7.149) where the quantity LLR_ℓ is given in (7.121). This LLR is then compared with a threshold ϑ and if the threshold is exceeded then the MAP estimate $\hat{\ell}_{\mathrm{MAP}}$ is computed as in Equation (7.133).

If we do not have any prior information about the point of change then we can compute the GLLR of Equation (7.150) where the quantity LLR_ℓ is still given in (7.121). This GLLR is then compared with a threshold ϑ and if the threshold is exceeded, then the ML estimate $\hat{\ell}_{\mathrm{ML}}$ is simply the argument that maximizes LLR_ℓ. ◊

In the case of steady-state forward and backward messages $\vec{\mu}_{X_0}$ and $\overleftarrow{\mu}_{X_K}$ or when considering an online algorithm, the term $\frac{1}{2}\ln\big(\det \overleftarrow{\mathbf{W}}_{U'_\ell}/(2\pi)^n\big)$ in (7.121) is constant and hence can be absorbed into the detection threshold ϑ.

Finally, let us note that in contrast to Example 7.7, there is no need to keep track of the scale factors neither for backward messages nor for forward messages in the scenario at hand, because the models $\mathcal{A} = \mathcal{B}$ are the same. ◊

We close this section by mentioning that the original scenario (7.134)–

(7.136) can be changed to the following:

$$\mathcal{H}_{\theta_1}: \text{A glue factor with parameter } \tilde{\boldsymbol{\theta}}_1 \text{ is present.} \tag{7.151}$$

$$\mathcal{H}_{\theta_0}: \text{A glue factor with parameter } \tilde{\boldsymbol{\theta}}_0 \text{ is present.} \tag{7.152}$$

Additionally we may be given some prior PMFs $p(\ell|\mathcal{H}_{\theta_1})$ and $p(\ell|\mathcal{H}_{\theta_0})$. Starting from (7.98) we can formulate

$$\begin{aligned} \text{LLR} &= \log \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\theta_1})}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\theta_0})} && (7.153)\\ &= \log \sum_{\ell=0}^{K} \frac{\beta_{\Theta_\ell}(\tilde{\boldsymbol{\theta}}_1)\, p(\ell|\mathcal{H}_{\theta_1})}{\beta^{\circ}_{\Theta_\ell}(\tilde{\boldsymbol{\theta}}_1)} \\ &\quad - \log \sum_{\ell'=0}^{K} \frac{\beta_{\Theta_{\ell'}}(\tilde{\boldsymbol{\theta}}_0)\, p(\ell'|\mathcal{H}_{\theta_0})}{\beta^{\circ}_{\Theta_{\ell'}}(\tilde{\boldsymbol{\theta}}_0)}\,, && (7.154) \end{aligned}$$

and

$$\begin{aligned} \text{GLLR} &= \log \frac{\max_{\ell\in\{0,\ldots,K\}} p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\theta_1}, \ell)}{\max_{\ell'\in\{0,\ldots,K\}} p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\theta_0}, \ell')} && (7.155)\\ &= \max_{\ell\in\{0,\ldots,K\}} \log \frac{\beta_{\Theta_\ell}(\tilde{\boldsymbol{\theta}}_1)}{\beta^{\circ}_{\Theta_\ell}(\tilde{\boldsymbol{\theta}}_1)} - \max_{\ell'\in\{0,\ldots,K\}} \log \frac{\beta_{\Theta'_\ell}(\tilde{\boldsymbol{\theta}}_0)}{\beta^{\circ}_{\Theta'_\ell}(\tilde{\boldsymbol{\theta}}_0)}\,. && (7.156) \end{aligned}$$

These expressions can be expanded in a similar fashion as in Section 7.6.

7.7.2 Extension to Detection of Multiple Glue Factors

Assume that we are given a signal $\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_K$ in which we would like to detect several glue factors. In terms of a factor graph model we can think of a SSM in which several glue factors connect several segments, each of which corresponds to a (potentially unique) model. The main difference between this setup and the setup of Section 7.6.4 is that here the number of glue factors is not known.

In a statistical decision framework, the detection of several changes leads to multiple hypotheses testing [61]. In order to compute all the needed test statistics for say r model changes, we would have to envisage a factor graph with r glue factors at all possible time positions. The number of sum-product messages to be computed in such a situation grows exponentially with r and with the signal length K.

In a Bayesian setting we may consider r to be a parameter to be estimated in a ML sense. In principle, the corresponding likelihood function can be computed by message passing with the number of glue factors r ranging from 0 to K and where now we ought to sum the likelihoods over all possible glue factor positions. Again, the complexity of such a procedure grows exponentially with K.

Instead of these rather complicated approaches, we may envisage a similar approach as in Section 7.6.4 using CM. We recall that in this approach, all glue factors are assumed to have the same structure, and all segments between the glue factors are modeled by the same SSM. The algorithm envisaged here is essentially the same as in Section 7.6.4, with the main difference being that in every iteration, before ML estimation of a glue factor position, a detection based on a LLR (7.140) or a GLLR (7.145) is performed in order to decide whether to insert a new glue factor at all or whether to stop the algorithm.

When following the above procedure we face, however, the problem of choosing appropriate detection thresholds in each iteration. We never have mentioned how to compute a detection threshold for a given alarm rate in the glue factor model, and we will not do so in this thesis. In the envisaged procedure though, we face a series of detection problems, which, when treated consistently should have related (if not the same) false alarm probability. We conjecture that there are many cases in which a detection threshold can indeed be found for a given false alarm probability.

7.8 Online Estimation of Glue Factor Positions

We recall the online setting in which the data $\tilde{\boldsymbol{y}}_k$ arrives in a stream for an indefinite time. In this scenario the hypothesis test $\mathcal{H}_g$ in (7.134) versus $\mathcal{H}_{\mathcal{A}}$ in (7.136) may not make much sense anymore. In this section we propose algorithms that approach the problem of detecting and locating a glue factor as soon as enough data has arrived. Put differently, we draw from our formulation of detecting a glue factor for the purpose of online estimation of the glue factor position. The structure of these algorithms still follow the "online likelihood filtering" procedure in Section 7.2.2.

We remark that the problem at hand – locating a glue factor in an online scenario – is not directly related to the setup of sequential detection as discussed in [14, 112], in which the goal is to decide (usually from

observing data that is independent and identically distributed), which of two underlying stationary distributions the data can be attributed to. In this section, in contrast, we want to locate the position of the glue factor, an event that breaks any stationarity of the underlying process. This setup is much closer to what is known as "quickest detection" [85]. We will, however, give no guarantees on the optimality of our algorithms.

In the following we describe two detection-inspired online algorithms, one for the GLLR (7.150) and one for the LLR (7.149), both based on the "online likelihood filtering" procedure of Section 7.2.2. In this procedure we let $D_1 = D_2 \triangleq D$ for both variants. We conjecture that, as long as D is large enough, we do not loose much by doing this simplification.

Towards the end of this section we look at extensions of the algorithms to estimate several glue factors in an online algorithm.

7.8.1 Online Estimation from GLLR

First, we define the algorithm inspired by the GLLR of Equation (7.150). In Step (b) of the "online likelihood filtering" procedure, a single value

$$\ln \eta_{K-D} = \mathrm{LLR}_\ell \,. \tag{7.157}$$

is computed where LLR_ℓ is given in (7.98). Here we introduce the new notation to distinguish between the online and the offline setting. Indeed, for a given block of data, the sequence of values $\ln \eta_{K-D}$ for increasing K is not the same as the values LLR_ℓ, because for every K, the data has changed.

Each time a new value $\ln \eta_{K-D}$ as in (7.157) is computed this value is compared with the detection threshold ϑ. If the threshold is exceeded, then we change the procedure for Step (b) to the following:

b) Compute the value (7.157) and if $\ln \eta_{K-D} < \ln \eta_{K-D-1}$ then declare a glue factor at time $\hat{\ell} = K - D$ and stop the algorithm.

This change of strategy is a simple way of computing the next occurring maximum. In [28, 29], the continuous-time equivalent to this method is described, in which the time-derivative of the GLLR is used to locate the next occurring maximum.

7.8.2 Online Estimation from LLR

For the formulation of the LLR based algorithm, a problem arises because there exists no uniform prior $p(\ell)$ on the whole range of $\mathbb{N}_0$ that we could employ in (7.149). We propose a time-varying prior

$$p^{(K)}(\ell) \overset{\triangle}{\propto} \begin{cases} (1-\lambda)^{K-D-\ell} & \text{for } \ell = 0, \ldots, K-D \\ 0 & \text{else} \end{cases} \tag{7.158}$$

for each K. The function $p^{(K)}$ is a truncated geometric PMF that weighs the position $K-D$ the most and decays towards the past. As the algorithm proceeds, i.e. as $K \to \infty$, this PMF converges to the true geometric distribution with mean $1/\lambda$.

By virtue of this prior PMF we can, in principle, compute a LLR as in Equation (7.149) for every K. Let us denote this LLR at time K by LLR_{K-D} because the glue factor position is $K-D$. The computation of LLR_{K-D} can be done by the "offline likelihood computation" procedure of Section 7.2.2. This is, however, not attractive in practice because when incrementing K, all the backward messages have to be recomputed.

Therefore, we propose to compute an approximate quantity $\ln \eta'_{K-D}$ of LLR_{K-D} in a recursive manner. Specifically, we define an online algorithm based on the "online likelihood filtering" procedure of Section 7.2.2 by specifying that in Step (b) we update

$$\eta'_{K-D} = e^{\mathrm{LLR}_\ell} + \lambda\, \eta'_{K-D-1}\,, \tag{7.159}$$

with a starting value of $\eta'_0 = 1$, where LLR_ℓ is again given in (7.98).

If $\ln \eta'_{K-D}$ exceeds a threshold ϑ then we declare that a glue factor has been detected and stop the algorithm. The actual estimate $\hat{\ell}$ of the glue factor position has to be computed either by stepping back to an offline procedure and computing (7.98) for all values ℓ of interest followed by (7.95), or by a method similar to the one in the previous section 7.8.1.

7.8.3 Online Estimation of Several Glue Factors

We consider a similar scenario as in Section 7.7.2, where the task is to locate several model changes. The main difference is that we now are given a infinitely long data stream $\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \ldots$ in which potentially infinitely many model changes have to be located.

Clearly, a rigorous treatment of this problem in the general case is precluded because of the complexity. We therefore simplify the scenario to one of the following, for both of which we assume that all glue factors are sufficiently far apart from each other:

a) All the model segments between the changes have the same underlying model and all the glue factors are the same.

b) There are only two underlying models $\mathcal{A}$ and $\mathcal{B}$ that are used alternatingly and they are connected by two types of glue factors, one for the transition from model $\mathcal{A}$ to model $\mathcal{B}$ and one for the transition from model $\mathcal{B}$ to model $\mathcal{A}$.

Scenario (b) can straightforwardly be extended to a cyclic sequence of n models and n glue factors, or indeed to any given sequence of models and glue factors. In both scenarios, true Bayesian inference of the number and location of the glue factors is still prohibited, as detailed in Section 7.7.2.

We therefore propose the following "forward-only" procedure to estimate from observations $\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \ldots$ the glue factor positions L_j for $j \in \mathbb{N}$:

a) Set $j = 1$.

b) Given fixed glue factor positions $L_i = \hat{\ell}_i$ for $i = 1, \ldots, j-1$ we make an estimate $\hat{\ell}_j$ of L_j.

c) Increment j and proceed with Step (b).

For Step (b) we propose to use the online estimation procedures using the GLLR or the LLR as described in Sections 7.8.1 and 7.8.2.

Once a glue factor position has been decided upon, this glue factor is placed permanently in the factor graph. The resulting forward message $\overrightarrow{\mu}_{X^{\mathcal{B}}_{\hat{\ell}_j}}$ becomes then the initial node f_0 in Figure 7.1 for the next iteration.

7.8.4 Online Estimation of a Hidden Bernoulli Process

Recall that for the online estimation procedure in Section 7.8.2 based on the LLR (7.149), one problem was that there exists no uniform prior $p(\ell)$ on the glue factor position for $L \in \mathbb{N}_0$. In the present section we look at a case in which indeed a non-uniform prior PMF makes sense.

A Bernoulli process [40] (the discrete-time equivalent to a Poisson process) is defined as follows. Let I_k be independent and identically distributed according to a Bernoulli distribution with mean λ, i.e. $I_k \in \{0,1\}$ for $k = 1, 2, \ldots$ and $p(i_k = 1) = \lambda$. A Bernoulli process B_k with rate λ is defined as $B_k = \sum_{k'=1}^{k} I_{k'}$, i.e., the number of ones occurred until time index k. The PMF of B_k is the binomial PMF

$$p_{B_k}(b) = \binom{k}{b} \lambda^b (1-\lambda)^{k-b} \,. \tag{7.160}$$

The connection to a Poisson process is as follows. Let t_S be the sampling interval and define $\lambda' \triangleq \lambda / t_S$. Using $t = k\, t_S$ we can write $\lambda = \lambda'\, t/k$. Now we can form the limit for $k \to \infty$ (or equivalently $t_S \to 0$).

$$\lim_{k\to\infty} p_{B_k}(b) = \lim_{k\to\infty} \frac{k!}{(k-b)!\, b!} \left(\frac{\lambda' t}{k}\right)^b \left(1 - \frac{\lambda' t}{k}\right)^{k-b} \tag{7.161}$$

$$= \lim_{k\to\infty} \frac{k!}{(k-b)!\, k^b} \frac{(\lambda' t)^b}{b!} \left(1 - \frac{\lambda' t}{k}\right)^{k} \left(1 - \frac{\lambda' t}{k}\right)^{-b} \tag{7.162}$$

$$= \frac{(\lambda' t)^b}{b!}\, e^{-\lambda' t} \,, \tag{7.163}$$

which is a Poisson distribution with mean $\lambda' t = \lambda k$. The last equality follows because as $k \to \infty$ we have

$$\frac{k!}{(k-b)!\, k^b} = \frac{k(k-1)\cdots(k-b+1)}{k^b} \to 1 \,, \tag{7.164}$$

$$\left(1 - \frac{\lambda' t}{k}\right)^{-b} \to 1 \,, \tag{7.165}$$

and

$$\left(1 - \frac{\lambda' t}{k}\right)^{k} \to e^{-\lambda' t} \,. \tag{7.166}$$

We define the set $\{L \in \mathbb{N} \colon I_L = 1\}$ of time indices of events and we let $\{L_j\}_{j\in\mathbb{N}}$ be an ordered list of the elements in this set. The inter-event times $M_j \triangleq L_j - L_{j-1}$ for $j = 2, 3, \ldots$ are independent and identically distributed according to a geometric distribution $p_{M_j}(m) = \lambda(1-\lambda)^{m-1}$ with mean $1/\lambda$, i.e., the probability of $m-1$ unsuccessful trials followed by

one success. Again, it can be shown that in the limit $t_S \to 0$ this geometric distribution converges to the exponential distribution $p_{T_j}(t) = \lambda' e^{-\lambda' t}$ with mean $1/\lambda'$.

We consider a situation in which the events $L_1, L_2, \ldots$ of the Bernoulli process are the glue factor positions. We make the same assumptions as we did for online estimation of several glue factors and thus we can immediately apply the "forward-only" procedure described in the previous section. In the situation at hand, however, we have more information about the inter-event times.

Specifically, we propose to implement Step (b) as a MAP estimation:

$$\hat{\ell}_j = \hat{\ell}_{j-1} + \operatorname*{argmax}_{\ell \in \mathbb{N}_0} p(\ell)\, p\big(\tilde{\boldsymbol{y}}^{(j)} \,\big|\, \mathcal{H}_\ell\big) \tag{7.167}$$

where we choose a geometric prior $p(\ell) = \lambda(1-\lambda)^{m-1}$ and $p\big(\tilde{\boldsymbol{y}}^{(j)} \,\big|\, \mathcal{H}_\ell\big)$ is defined by our factor graph model family in Figure 7.1 with observations $\tilde{\boldsymbol{y}}^{(j)} \triangleq (\tilde{\boldsymbol{y}}_{\hat{\ell}_{j-1}+1}, \tilde{\boldsymbol{y}}_{\hat{\ell}_{j-1}+2}, \ldots)$ and a glue factor positioned at $\hat{\ell}_{j-1} + \ell$. The initial message $\vec{\mu}_{X_0^A}$ is the message out of the glue factor $\vec{\mu}_{X_{\hat{\ell}_j}^B}$ that was estimated in the previous iteration.

Clearly, the MAP estimation rule (7.167) may not be optimal at all, because in reality there are several glue factors present. Moreover, a direct implementation of (7.167) is precluded because of the infinite support of ℓ.

Analogous to the approach taken in Section 7.8.2, an algorithm can be formulated based on a time-varying prior. It must, however, be noted that the detection threshold ϑ does now change considerably for a given constant false-alarm rate. This obstacle precludes, at present, the complete formulation and implementation of the algorithm.

7.9 Simulation Examples for Glue Factor Position Estimation

We report three example implementations of algorithms to estimate the glue factor position given an artificially generated signal. In all examples, the output of all SSMs is scalar. Both offline and online algorithms are implemented. We enumerate the examples as follows:

a) **Locating an additional input:**
The first example corresponds to Examples 7.5 and 7.8 and, in which we want to locate an additional input to a SSM. In this case, both models are the same, i.e. $\mathcal{B} = \mathcal{A}$, and the factor graph representation is given in Figure 7.3b with Figures 7.16 and 7.17 inserted, and with $f_0(\boldsymbol{x}_0^{\mathcal{A}}) = \mathcal{N}\big(\boldsymbol{x}_0^{\mathcal{A}} \big| \boldsymbol{0}, \overrightarrow{\mathbf{V}}_X\big)$, where $\overrightarrow{\mathbf{V}}_X$ is the steady-state solution to Equation (3.9). The SSM is a 6-th order IIR system with a scalar input.

All parameters of the model and of the algorithm are listed in Table 7.2 and the simulation results are shown in Figure 7.19.

b) **Locating a model change:**
The second example corresponds to Example 7.3, in which two SSMs of the same order but with different parameters are concatenated. In this case the glue factor is a direct connection of the states. The factor graph representation is the general model family in Figure 7.1 with two different SSMs each as in Figure 7.15, both 4-th order IIR systems but with different poles, zeros and noise variances. The factor f_0 is a zero-mean Gaussian with steady-state forward covariance matrix $\overrightarrow{\mathbf{V}}_X$. The factor f_K is neutral.

All parameters of the models and of the algorithm are listed in Table 7.3 and the simulation results are shown in Figure 7.20.

c) **Locating a pulse:**
The third example corresponds to Example 7.6 in which we want to locate a two-sided sinusoidal pulse, cf. Section 7.5. In this setup, both models $\mathcal{A}$ and $\mathcal{B}$ are autonomous IIR systems with order 8 and 6 respectively. The glue factor is depicted in Figures 7.10b and 7.18 for unknown and known pulse shape respectively.

All parameters of the models and of the algorithms are listed in Tables 7.1 and 7.4 and the simulation results are shown in Figure 7.21.

For all three examples, observations $\tilde{y}_k$ for $k = 1, \ldots, K$ are generated according to the given model and the three quantities LLR_ℓ as in (7.98), $\ln \eta_\ell$ as in (7.157), and $\ln \eta'_\ell$ as in (7.159) are computed. For Example (c) LLR_ℓ and η_ℓ are also computed for the variant in which the pulse shape is known.

In all examples, all SSMs are parameterized in real Jordan canonical form of (3.2) with unique complex conjugate eigenvalue pairs. Note that

the glue factor position ℓ (and if appropriate the glue factor parameter u, $\tilde{\boldsymbol{s}}_{\mathcal{A}}$, $\tilde{\boldsymbol{s}}_{\mathcal{B}}$, and $\tilde{\boldsymbol{u}}$) is used to generate $\tilde{y}_k$, but both are unknown to the algorithm.

In these examples it can be seen, that the glue factor approach can be used for very different purposes. Depending on the SSMs and the glue factor, and depending on the chosen LLR, the number D of backward steps has to be chosen differently in order to still be able to locate the glue factor. Also, we observe that the quantity $\ln \eta'_\ell$ has linearly shaped tails because the summation is taking place in the log domain.

We recall that LLR_ℓ is computed by an offline procedure, while $\ln \eta'_\ell$ and $\ln \eta'_\ell$ are computed by an online procedure. Also we highlight that the main difference between the examples is the appearance of the scale factors in the messages. In Example (a) (Figure 7.19) all scale factors cancel and all the quantities can be computed based on (7.121).

In Example (b) (Figure 7.20) only the scale factors of the forward messages cancel. In this case, all the quantities are computed based on (7.117) and the recursive computation of $\overleftarrow{\gamma}'_k$ in (7.115) is applied.

Finally, in Example (c) (Figure 7.21), scale factor computation is not necessary either. All the required quantities are computed from the LLRs in Equations (7.125) and (7.126) for unknown and known pulse shape respectively.

Although developed in the same framework, Examples (a) and (b) are quite different. For the former, a short delay D for the online algorithms suffices and the additional input can clearly be recognized. The parameters for the latter are chosen such that the offline algorithm can indeed estimate the rough location of the model switch. The offline algorithms are in this case, however, not as decisive.

In practice, the recursive computation of $\ln \overleftarrow{\gamma}'_k$ as in (7.115) (or $\ln \overleftarrow{\gamma}_k$ as in (7.105)) as well as the computation of the matrix determinants in (7.117) can be numerically unstable if done over many slices of the factor graph. The online quantities $\ln \eta_\ell$ and $\ln \eta'_\ell$ do not suffer from such instabilities because the scale factors are computed only over D slices of the factor graph.

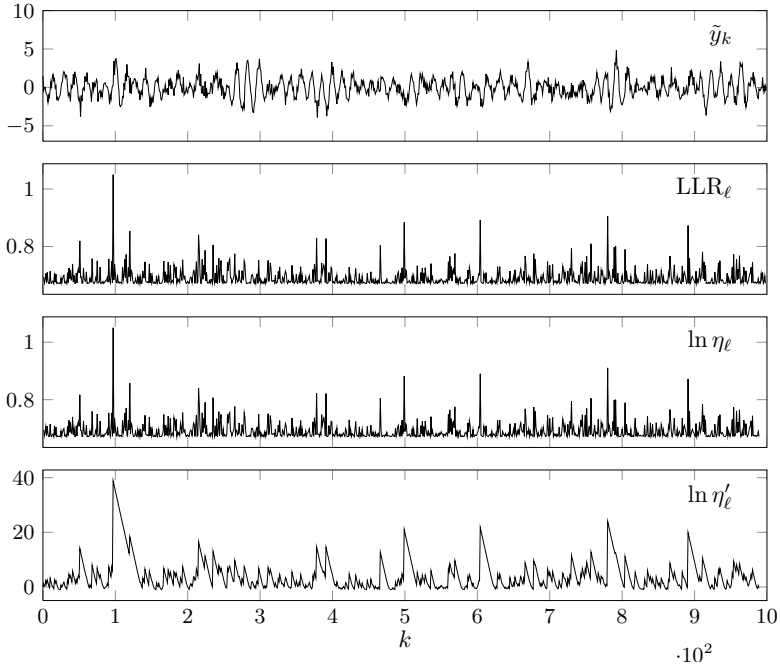


Figure 7.19: Example (a): Locating an additional input.

Model parameters		Value
ℓ	Glue factor position	97
u	Glue factor parameter	2
ϕ	Pole angles in $\mathbf{A}$	$\big[0.478, 0.137, 0.0420\big]\pi$
α	Pole magnitudes $\mathbf{A}$	$\big[0.648, 0.964, 0.164\big]$
$\boldsymbol{b}^\mathsf{T}$		$\big[1, 0, 1, 0, 1, 0\big]$
$\boldsymbol{c}$		$\big[0, 1, 0, 1, 0, 1\big]$
σ_U^2	Input noise variance	0.1
σ_Z^2	Observation noise variance	0.07
Parameters for $\ln \eta_\ell$ and $\ln \eta'_\ell$		
D	Backward steps	5
λ	Parameter for η'_ℓ (7.159)	0.3

Table 7.2: Parameter settings for Figure 7.19.

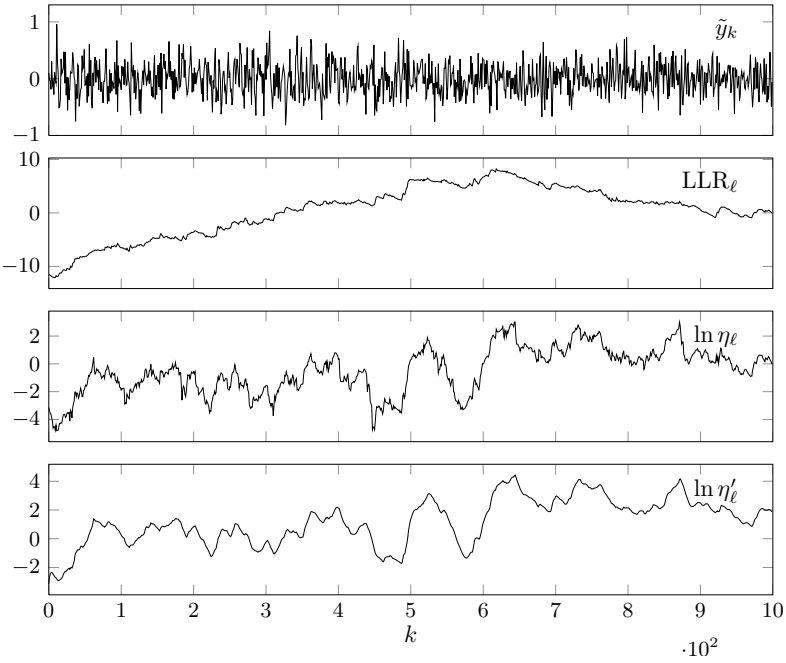
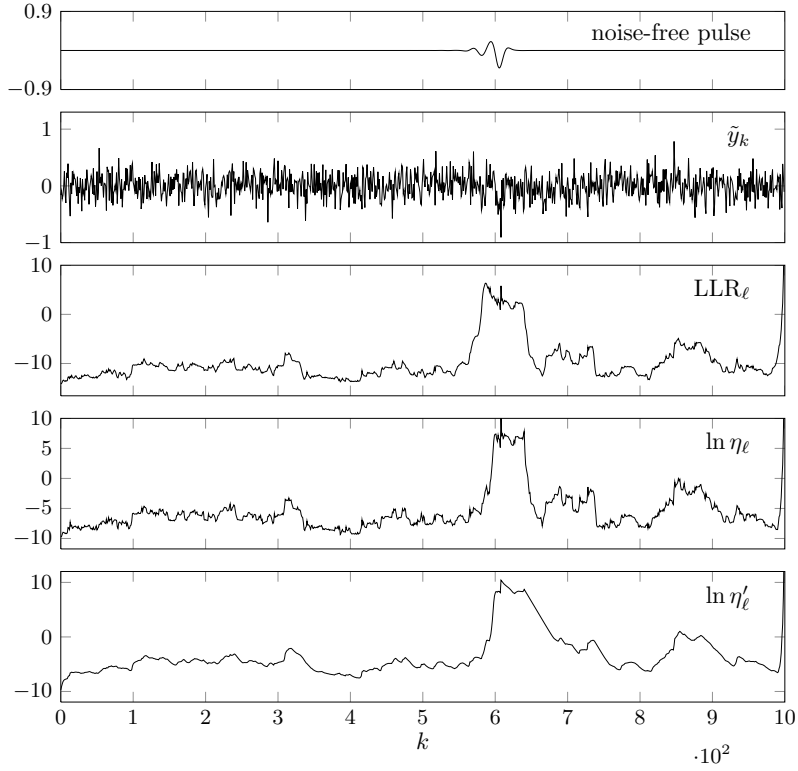


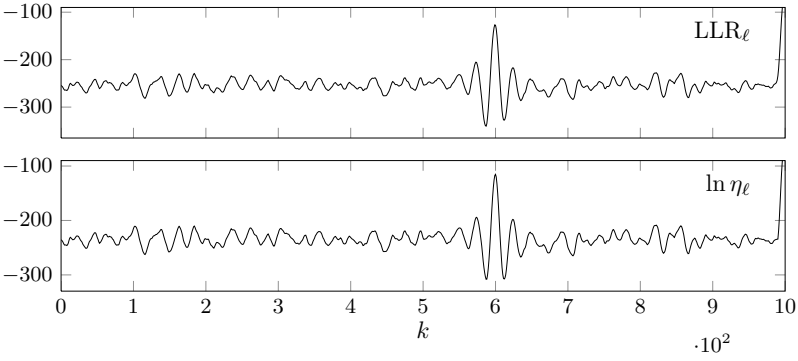
Figure 7.20: Example (b): Locating a model change.

Model Parameters			Value
ℓ	Glue factor position		600
		Model $\mathcal{A}$	Model $\mathcal{B}$
$\phi_\mathcal{M}$	Pole angles in $\mathbf{A}_\mathcal{M}$	$[0.3, 0.8]\pi$	$[0.4, 0.7]\pi$
$\alpha_\mathcal{M}$	Pole magnitudes in $\mathbf{A}_\mathcal{M}$	$[0.6, 0.5]$	$[0.5, 0.4]$
$\boldsymbol{b}_\mathcal{M}^\mathsf{T}$		$[0, 1, 0, 1]$	$[0, 1, 0, 1]$
$\boldsymbol{c}_\mathcal{M}$		$[1, 0, 1, 0]$	$[1, 0, 1, 0]$
$\sigma^2_{U,\mathcal{M}}$	Input noise variance	0.14	0.1
$\sigma^2_{Z,\mathcal{M}}$	Observation noise variance	0.1	0.1
Parameters for $\ln \eta_\ell$ and $\ln \eta'_\ell$			
D	Backward steps		50
λ	Parameter for η'_ℓ (7.159)		0.8

Table 7.3: Parameter settings for Figure 7.20.



(a) Unknown pulse shape.



(b) Known pulse shape.

Figure 7.21: Example (c): Locating a pulse. All model and algorithm parameters are set as in Tables 7.1 and 7.4.

Model Parameters		Value
ℓ	Glue factor position	600
$\tilde{\boldsymbol{s}}_{\mathcal{A}}^{\mathsf{T}}$	Pulse parameter	$[-0.4038, 0.5688, 1.252, -0.9366]$
$\tilde{\boldsymbol{s}}_{\mathcal{B}}^{\mathsf{T}}$	Pulse parameter	$[-0.2160, 1.330]$
$\tilde{\boldsymbol{u}}$	Pulse parameter	$[0.05434, -1.233, -1.037, 1.302]$
Parameters for $\ln \eta_\ell$ and $\ln \eta'_\ell$		
D	Backward steps	10
λ	Parameter for η'_ℓ (7.159)	0.8

Table 7.4: Parameter settings for Figure 7.21.

Chapter 8

Hierarchical Likelihood Filtering

8.1 Introduction

It is evident that for estimation and detection tasks involving complicated, structured signals, the complexity of a glue factor model as devised in the previous chapter may be very large. For example, in order to model a lengthy pulse by means of superposed sinusoids (cf. Section 7.5), many sinusoids may be required, or we may have to resort to models with several glue factors, distributed across time.

In this chapter we propose a hierarchical system [87] in which multiple instances of the likelihood filter of Section 7.2.2 feature. Roughly, the system consists of

- A population of linear second-order SSMs.
- A population of glue factors each of which is potentially connected to the state of every SSM.

The twist is that we allow quantities that computed by a glue factor to be reused as observations.

This twist induces a hierarchy of the signals with lower levels that are closer connected to the observed signal, and higher levels for which the observed signal has been processed by many glue factors. We conjecture that with such a system, signal structure on a longer timescale can be captured by levels higher up in the hierarchy.

This chapter must be understood as merely a sketch, a first step on a longer way towards a fully automated hierarchical likelihood filtering system.

Hierarchical modeling has a long tradition in neural networks [91,92], but these are usually block-processing based or in the case of recurrent neural networks [73] a distinction is made between the training phase and the actual usage phase. Other possibly related but more exotic hierarchical approaches include [42,43,113].

This chapter has only two sections. In the first section we note that the most meaningful quantity for the glue factor to compute in this situation is not a LLR but a posterior probability under a given hypothesis. In the second section we define the hierarchical likelihood filter, without mentioning how to put it to use.

8.2 From Log-Likelihood Ratios to Posterior Probabilities

We recall that likelihoods can assume extreme values ranging in the whole of $\mathbb{R}$. LLRs are somewhat better behaved but still they can occupy the whole range of $\mathbb{R}$. We conjecture that the usage of LLRs as observations for second-order SSMs is not suited, because any sudden change in scale can feed large amounts of energy into the message passing filter.

We therefore suggest that posterior probabilities, being confined between 0 and 1, are much better suited. In the following we show how we can convert a LLR into a posterior probability.

Let $\tilde{\boldsymbol{y}} \triangleq (\tilde{y}_1, \ldots, \tilde{y}_K)$ be an observed signal. Assume that we have two hypotheses $\mathcal{H}_1$ and $\mathcal{H}_0$ with an associated prior $p(\mathcal{H}_i)$ for $i = 0, 1$. Instead of the log-likelihood ratio

$$\mathrm{LLR} \triangleq \ln \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_1)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_0)} \tag{8.1}$$

we consider the posterior probability $p(\mathcal{H}_1|\tilde{\boldsymbol{y}})$. The quantity $p(\mathcal{H}_1|\tilde{\boldsymbol{y}})$ can be expressed in terms of the LLR and a quantity

$$\mathrm{LPR} \triangleq \ln \frac{p(\mathcal{H}_1)}{p(\mathcal{H}_0)}. \tag{8.2}$$

which we name *log-prior ratio*, as

$$p(\mathcal{H}_1|\tilde{\boldsymbol{y}}) = \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_1)\,p(\mathcal{H}_1)}{p(\tilde{\boldsymbol{y}})} \tag{8.3}$$

$$= \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_1)\,p(\mathcal{H}_1)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_1)\,p(\mathcal{H}_1) + p(\tilde{\boldsymbol{y}}|\mathcal{H}_0)\,p(\mathcal{H}_0)} \tag{8.4}$$

$$= \frac{e^{\mathrm{LLR+LPR}}}{1 + e^{\mathrm{LLR+LPR}}} \tag{8.5}$$

$$= \frac{1}{1 + e^{-\mathrm{LLR-LPR}}}\,. \tag{8.6}$$

This is a sigmoid-type function as used for neural networks [91].

The above can be generalized easily to the multiple hypotheses case. For the hypotheses $\mathcal{H}_1, \ldots, \mathcal{H}_M$, the posterior probability of any chosen hypothesis $\mathcal{H}_k$ can be expressed as

$$p(\mathcal{H}_k|\tilde{\boldsymbol{y}}) = \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_k)\,p(\mathcal{H}_k)}{p(\tilde{\boldsymbol{y}})} \tag{8.7}$$

$$= \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_k)\,p(\mathcal{H}_k)}{\sum_{i\in\{1,\ldots,M\}} p(\tilde{\boldsymbol{y}}, \mathcal{H}_i)} \tag{8.8}$$

$$= \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_k)\,p(\mathcal{H}_k)}{\sum_{i\in\{1,\ldots,M\}} p(\tilde{\boldsymbol{y}}|\mathcal{H}_i)\,p(\mathcal{H}_i)} \tag{8.9}$$

$$= \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_k)\,p(\mathcal{H}_k)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_k)\,p(\mathcal{H}_k) + \sum_{i\in\{1,\ldots,M\}\setminus\{k\}} p(\tilde{\boldsymbol{y}}|\mathcal{H}_i)\,p(\mathcal{H}_i)} \tag{8.10}$$

$$= \frac{R}{1+R} \tag{8.11}$$

$$= \frac{1}{1+1/R}\,, \tag{8.12}$$

where

$$R \triangleq \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_k)\,p(\mathcal{H}_k)}{\sum_{i\in\{1,\ldots,M\}\setminus\{k\}} p(\tilde{\boldsymbol{y}}|\mathcal{H}_i)\,p(\mathcal{H}_i)}\,. \tag{8.13}$$

We remark that in contrast to most scenarios of Chapter 7, we have introduced a prior PMF $p(\mathcal{H}_k)$ on the hypotheses. This PMF can be regarded as yet another parameter in the model family induced by a glue factor. We do not treat the estimation of this parameter in this thesis.

8.3 Concepts and Definitions

The goal is to analyze a given signal $\nu_k^{(0)} \in \mathbb{R}$, where k denotes the discrete time index. The hierarchical system considered here is defined by:

- A choice for $N \in \mathbb{N}$.
- N linear second-order linear SSMs with parameters $\boldsymbol{\theta}_n$ (cf. (8.17)) for $n = 1, \ldots, N$.
- M glue factor constraints with parameters $\boldsymbol{\upsilon}_m$ (cf. (8.24)) for $m = 1, \ldots, M$.
- A connection matrix $\mathbf{J} \in \{0,1\}^{N \times (M+1)}$.

The n-th SSM is of the form

$$\begin{aligned} \boldsymbol{X}_k^{(n)} &= \mathbf{A}^{(n)}\, \boldsymbol{X}_{k-1}^{(n)} + \boldsymbol{U}_k \\ Y_k^{(n)} &= \boldsymbol{c}\, \boldsymbol{X}_k^{(n)}\,, \end{aligned} \tag{8.14}$$

where

$$\mathbf{A}^{(n)} \triangleq \rho_n \operatorname{rotm}(\Omega_n)\,,\ \boldsymbol{c} \triangleq \big[1, 0\big]\,, \tag{8.15}$$

$$Z_k^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}\big(0, \sigma_{Z,n}^2\big)\,,\ \boldsymbol{U}_k^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}\big(\boldsymbol{0}, \sigma_{U,n}^2 \mathbf{I}_2\big)\,. \tag{8.16}$$

The parameter vector of the n-th model thus is

$$\boldsymbol{\theta}_n \triangleq (\rho_n, \Omega_n, \sigma_{Z,n}^2, \sigma_{U,n}^2)\,. \tag{8.17}$$

We define the combined state vector and the combined observation vector as

$$\boldsymbol{X}_k \triangleq \begin{bmatrix} \boldsymbol{X}_k^{(1)} \\ \vdots \\ \boldsymbol{X}_k^{(N)} \end{bmatrix} \in \mathbb{R}^{2N} \text{ and } \boldsymbol{Y}_k \triangleq \begin{bmatrix} \boldsymbol{Y}_k^{(1)} \\ \vdots \\ \boldsymbol{Y}_k^{(N)} \end{bmatrix} \in \mathbb{R}^{N} \tag{8.18}$$

respectively.

All N glue factors $g_1, \ldots, g_N$ take the form of the factor graph depicted in Figure 8.1 with $\mathbf{H}_n^{(m)} = \alpha_n^{(m)} \operatorname{rotm}(\phi_n^{(m)})$ for $n = 1, \ldots, N$ and $m =$

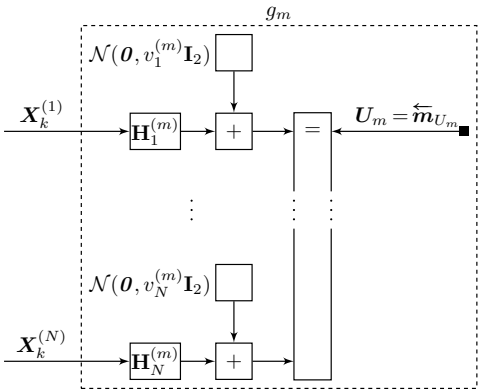


Figure 8.1: The glue factor for hierarchical likelihood filtering.

$1, \ldots, M$. For every such glue factor we consider the LLR

$$\mathrm{GLLR}_m = \ln \frac{p(\tilde{\boldsymbol{y}}|\mathcal{H}_1)}{p(\tilde{\boldsymbol{y}}|\mathcal{H}_0)}\,, \tag{8.19}$$

where the two hypotheses are defined as

$$\mathcal{H}_1 \colon \boldsymbol{u}_m = \operatorname*{argmax}_{\boldsymbol{u}_m} p(\tilde{\boldsymbol{y}}|\boldsymbol{u}_m) = \overleftarrow{\boldsymbol{m}}_{U_m} \tag{8.20}$$

$$\mathcal{H}_0 \colon \boldsymbol{u}_m = \boldsymbol{0}\,. \tag{8.21}$$

As detailed in Section 6.3.4, Equation (6.100), this GLLR can be computed as

$$\mathrm{GLLR}_m = \overleftarrow{\boldsymbol{m}}_{U_m}^{\mathsf{T}} \overleftarrow{\mathbf{W}}_{U_m} \overleftarrow{\boldsymbol{m}}_{U_m}/2\,. \tag{8.22}$$

We now define the signals $\nu_k^{(m)}$ for $m = 1, \ldots, M$ to be the posterior probability

$$\nu_k^{(m)} \triangleq p(\mathcal{H}_1|\tilde{\boldsymbol{y}}) = \frac{1}{1 + e^{-\mathrm{GLLR}_m - \mathrm{LPR}_m}}\,, \tag{8.23}$$

where the log-prior ratio LPR_m is also a parameter of the glue factor. The parameter vector of the m-th glue factor hence is

$$\boldsymbol{v}_m \triangleq \left(\mathrm{LPR}_m, \mathbf{H}_1^{(m)}, \ldots, \mathbf{H}_N^{(m)}, v_1^{(m)}, \ldots, v_N^{(m)}\right). \tag{8.24}$$

At first, it seems that every glue factor is connected to each state $\boldsymbol{X}_k^{(n)}$. Any such connection can however be "switched off" by letting $v_n^{(m)} \to \infty$.

Finally, we use the posterior probabilities as observations by making the connections

$$\boldsymbol{Y}_k = \mathbf{J}\,\boldsymbol{\nu}_k\,, \tag{8.25}$$

where $\boldsymbol{\nu}_k^{\mathsf{T}} \triangleq [\nu_k^{(0)}, \ldots, \nu_k^{(M)}]$. We might want to impose constraints on $\mathbf{J}$ such that in the resulting computation no cycle occurs. Specifically, we must have $J_{n,n+1} \stackrel{!}{=} 0$ and if $J_{n,m+1} = 1$ for some $m \neq n$ then the element $J_{m,n+1}$ must be 0. This condition can be written compactly as

$$\mathbf{J}'\,\mathbf{J}'^{\mathsf{T}} \stackrel{!}{=} \mathbf{0}\,, \tag{8.26}$$

where

$$\left[\boldsymbol{j}_0, \mathbf{J}'\right] \triangleq \mathbf{J}\,, \tag{8.27}$$

i.e., $\boldsymbol{j}_0$ is the first column of $\mathbf{J}$ and $\mathbf{J}' \in \mathbb{R}^{N \times M}$ is the remaining matrix.

Now we can modify the "online likelihood filtering" procedure of Section 7.2.2 to the following forward-only algorithm:

Hierarchical Likelihood Filtering:

a) Increment K and fetch the next data item $\nu_K^{(0)}$.

b) Compute the posterior probabilities $\nu_K^{(m)}$ for all $m = 1, \ldots, M$ by computing the messages $\vec{\mu}_{X_K^{(n)}}$ and applying the glue factor as in (8.23).

c) Go to Step (a).

Some remarks are due:

- The computation in Step (b) traverses the whole hierarchy and can only be done if no cyclic dependencies exist, i.e. if (8.26) is satisfied.

- If $\boldsymbol{j}_0 \triangleq \boldsymbol{1}$ and $\mathbf{J}' \triangleq \mathbf{0}$, then the algorithm degenerates to the "online likelihood filtering" procedure of Section 7.2.2 based on a decomposed model as in Section 3.5 Figure 3.11 whose states at time k are connected by a glue factor as in Figure 8.1 that has M different parameter settings.

- In the case of autonomous SSMs and when assuming a separate glue factor for each model with $\mathbf{H}_m^{(m)} = \mathbf{I}_2$, the signals $\nu_K^{(1)}, \ldots, \nu_K^{(M)}$ are related to the discrete-time Fourier transform at Ω_n of the modified signals $\rho_n^{k-K}\nu_k^{(0)}$ via (3.43). This setup is achieved if $M = N$, $\boldsymbol{j}_0 \triangleq \boldsymbol{1}$, $\mathbf{J}' \triangleq \mathbf{0}$, $\sigma_{U,n}^2 = 0$ for $n = 1, \ldots, N$, and $\mathbf{H}_m^{(m)} = \mathbf{I}$ for all $m = 1, \ldots, N$.

  Note that any further processing of such Fourier transform related quantities can be incorporated by adding further SSMs (increasing N to N'), adding further glue factors (increasing M to M') and making connections by adding columns in $\mathbf{J}$. Most notably, we can again compute Fourier transformed quantities of $\left(\nu_K^{(1)}, \ldots, \nu_K^{(M)}\right)$ by choosing appropriate matrices $\mathbf{H}_n^{(m)}$ for $n = N+1, \ldots, N'$ and $m = M+1, \ldots, M'$. This type of processing is in essence similar to the notion of a cepstrum [16].

- The described setup can easily be extended to vector observations $\boldsymbol{\nu}_k^{(0)} \in \mathbb{R}^{n_Y}$. In this case, $\mathbf{J} \in \{0,1\}^{N \times (M+n_Y)}$, and $\boldsymbol{j}_0$ changes into a matrix $\mathbf{J}_0 \in \{0,1\}^{N \times n_Y}$.

- Cyclic dependencies, i.e. matrices $\mathbf{J}$ that violate (8.26), can be considered if we are willing to introduce a delay by substituting (8.25) by

$$\boldsymbol{Y}_k = \mathbf{J}\,\boldsymbol{\nu}_{k-1}\,. \tag{8.28}$$

- Due to the lack of backward message passing, one can envisage continuous-time equivalents of this procedure. In this case, cyclic dependencies may be allowed without delay. There may, however, result uncontrolled oscillation or chaotic behavior.

In the algorithm as described above, all parameters (N, $\boldsymbol{\theta}_n$ for $n = 1, \ldots, N$, υ_m for $m = 1, \ldots, M$, and $\mathbf{J}$) are fixed. This may be well suited for situations in which we have complete knowledge about the observed signal $\nu_k^{(0)}$ and about the desired output in all circumstances.

The algorithm may be extended in many ways to incorporate estimation of some, or all of the parameters. In principle, the methods presented in Chapter 4 may be applied to estimate all the parameters in $\boldsymbol{\theta}_n$ and all the parameters except LPR_m in $\boldsymbol{\upsilon}_m$ for each $m = 1, \ldots, M$ separately. A general scenario of such glue factor learning for a single glue factor was elaborated on in Section 7.3.1. The situation at hand suggests the special case of forward-only message passing in the graph of Figure 7.5 for every parameter. This approach might be applicable to the analysis of signals about which we have partial knowledge, such as, e.g., quasi-periodic signals with an unknown number of harmonics.

The estimation of LPR_m is not treated in this thesis. This parameter may have a high impact on the performance of the algorithm.

The entries in $\mathbf{J}$ seem to be interlinked with the parameters $v_n^{(m)}$, and hence we believe that estimation of $\mathbf{J}$ is related to a hard decision version of estimation of $v_n^{(m)}$.

The scenario changes dramatically, if we furthermore let N and M vary over time. We now envisage a population of SSMs and glue factors, in which at each time step we can ask ourselves the question whether to add or drop members of this population. Care must be taken to guard against overfitting: If the N and M are estimated in an ML sense then these values will keep increasing until the number of parameters is comparable to the numbers of degrees of freedom in the signal. One way to fight this explosion of parameters may be the Bayesian view of Section 6.5.1, i.e. the assumption of a prior on N and M. More generally Dirichlet processes may be used for this task, cf. [102] and references therein.

Estimation of all the parameters (and possibly hyper-parameters) of this algorithm may lead to a genuine fully automated signal analysis framework, of which special cases are related to the algorithms of Chapter 7. It seems that this is a promising open direction for further research.

Chapter 9

Conclusion and Outlook

First and foremost, a formalism has been developed to deal consistently with scale factors in sum-product message passing. At the heart of this formalism stands the definition of the two types of scale factors and the update rules in Table 6.2–6.4 together with the proofs in Appendix C. Despite the fact that in many cases such scale factors can be neglected, they have prooved to be very useful in derivations throughout this chapter.

We have considered various procedures for computing quantities related to the likelihood of the given observation under some statistical model. In all these models, we have taken the common view of a glue factor. Clearly this view has lead to a wealth of possible quantities of interest that can be computed by the general procedure of likelihood filtering:

a) Do forward message passing.

b) Do backward message passing.

c) From the messages (and scale factors) compute the quantities of interest.

Although the general structure and complexity of this algorithm is the same as the computation of marginals by sum-product message passing, we are able to compute more elaborate quantities such as LLRs and solve more intricate problems such as model change detection.

As a special case of interest we mention the forward-only scenario, in which Step (b) is omitted from the above procedure and the glue factor always is located at the end of the signal. This special case has inspired us to propose a hierarchical likelihood filtering procedure, whose usability still has to be shown in the future.

We treat mostly the Gaussian case here. It would be worth while to take a closer look at the discrete case. It is suspected that in the course of the development of decoding algorithms based on factor graphs, many of the findings here have been already derived in the different scenario of the discrete case.

One of the severest short-comings of this thesis is the absence of a method to compute detection thresholds. Most notably, this precludes the completion of algorithms that are based on solving multiple detection problems.

On the same line, one can think of ways to introduce a measure of model complexity [5, 96, 100] in the framework of factor graphs. This would solve the same problem as in the previous paragraph but from a model selection point of view. It may well turn out that these problems can again be addressed in a factor graph framework.

Finally, we hope that many of the algorithms that are merely sketched in this thesis find applications in the real world. It is the authors believe that the potential of the approaches in this thesis are not yet fully exploited by far.

Appendices

"You learn by your mistakes. If you don't make mistakes, you are not trying hard enough."

Sir Robin Keith Saxby, (1947–)

Appendix A

Analytic Messages for Second-Order Autonomous Systems

A.1 Proofs for $\overrightarrow{\mu}_{X_K}$

In the following we give the proofs of the results in Section 3.3.3 for the forward message $\overrightarrow{\mu}_{X_K}$ in the factor graph of Figure 3.7.

Using the same approach as in Section 2.6 we can write $\overrightarrow{\mathbf{W}}_{X_K}$ as

$$\overrightarrow{\mathbf{W}}_{X_K} = \frac{1}{\sigma_Z^2}\sum_{k=1}^{K}\gamma^{K-k}\left(\mathbf{A}^{k-K}\right)^{\mathsf{T}}\boldsymbol{c}^{\mathsf{T}}\boldsymbol{c}\,\mathbf{A}^{k-K} \tag{A.1}$$

$$= \frac{1}{\sigma_Z^2}\sum_{k=0}^{K-1}\gamma^{k}\left(\mathbf{A}^{-k}\right)^{\mathsf{T}}\boldsymbol{c}^{\mathsf{T}}\boldsymbol{c}\,\mathbf{A}^{-k} \tag{A.2}$$

$$= \frac{1}{\sigma_Z^2}\sum_{k=0}^{K-1}\overrightarrow{v}^{k}\begin{bmatrix}\cos^2(k\Omega) & \cos(k\Omega)\sin(k\Omega)\\ \cos(k\Omega)\sin(k\Omega) & \sin^2(k\Omega)\end{bmatrix} \tag{A.3}$$

$$= \frac{1}{2\sigma_Z^2}\sum_{k=0}^{K-1}\overrightarrow{v}^{k}\left(\mathbf{I}_2 + \begin{bmatrix}\cos(2k\Omega) & \sin(2k\Omega)\\ \sin(2k\Omega) & -\cos(2k\Omega)\end{bmatrix}\right), \tag{A.4}$$

where $\overrightarrow{v} \triangleq \gamma/\rho^2$. The sum over the geometric series $\overrightarrow{v}^{k}$ is standard. For the sum involving the trigonometric terms $\overrightarrow{v}^{k}\cos(2k\Omega)$ and $\overrightarrow{v}^{k}\sin(2k\Omega)$, complex expansions of these terms can be used and the terms in (3.39)–

(3.41) result. In the case where $\overrightarrow{v} = 1$, Equations (3.39)–(3.41) simplify to

$$\frac{a}{d} = \frac{\sin(K\Omega)\cos(K\Omega - \Omega)}{\sin\Omega}, \tag{A.5}$$

$$\frac{b}{d} = \frac{\sin(K\Omega)\sin(K\Omega - \Omega)}{\sin\Omega}, \tag{A.6}$$

and (3.44) results.

In the case where $\overrightarrow{v} < 1$, the steady-state precision matrix $\overrightarrow{\mathbf{W}}_X$ given in Equation (3.50) is obtained by letting K go to infinity in Equation (A.4). Note that, alternatively, the same result can be obtained by solving the Lyapunov equation (cf. Section 3.3.1)

$$\overrightarrow{\mathbf{W}}_X = \overrightarrow{v}\,\mathbf{A}^{-\mathsf{T}}\overrightarrow{\mathbf{W}}_X\mathbf{A}^{-1} + \begin{bmatrix} \sigma_Z^{-2} & 0 \\ 0 & 0 \end{bmatrix}. \tag{A.7}$$

For the analytic expression (3.43) of the weighted mean we again use the approach of Section 2.6 as

$$\overrightarrow{\mathbf{W}}_{X_K}\overrightarrow{\boldsymbol{m}}_{X_K} = \frac{1}{\sigma_Z^2}\sum_{k=1}^{K}\gamma^{K-k}\tilde{y}_k\left(\mathbf{A}^{k-K}\right)^{\mathsf{T}}\boldsymbol{c}^{\mathsf{T}} \tag{A.8}$$

$$= \frac{\mathrm{rotm}(K\Omega)}{\sigma_Z^2}\sum_{k=1}^{K}(\gamma/\rho)^{K-k}\tilde{y}_k\begin{bmatrix}\cos(-k\Omega)\\ \sin(-k\Omega)\end{bmatrix} \tag{A.9}$$

$$= \frac{\mathrm{rotm}(K\Omega)}{\sigma_Z^2}\sum_{k=1}^{K}(\gamma/\rho)^{K-k}\tilde{y}_k\begin{bmatrix}\cos(k\Omega)\\ -\sin(k\Omega)\end{bmatrix}. \tag{A.10}$$

A.2 Proofs for $\overleftarrow{\mu}_{X_0}$

In the following we give the proofs of the results in Section 3.3.3 for the backward message $\overleftarrow{\mu}_{X_0}$ in the factor graph of Figure 3.8.

Using an analogous approach to the one in Section 2.6 we can write $\overleftarrow{\mathbf{W}}_{X_0}$

$$\overleftarrow{\mathbf{W}}_{X_0} = \frac{1}{\sigma_Z^2} \sum_{k=1}^{K} \gamma^k \left(\mathbf{A}^k\right)^{\mathsf{T}} \boldsymbol{c}^{\mathsf{T}} \boldsymbol{c}\, \mathbf{A}^k \tag{A.11}$$

$$= \frac{1}{\sigma_Z^2} \sum_{k=1}^{K} \overleftarrow{\upsilon}^k \begin{bmatrix} \cos^2(k\Omega) & -\cos(k\Omega)\sin(k\Omega) \\ -\cos(k\Omega)\sin(k\Omega) & \sin^2(k\Omega) \end{bmatrix} \tag{A.12}$$

$$= \frac{1}{2\sigma_Z^2} \sum_{k=1}^{K} \overleftarrow{\upsilon}^k \left(\mathbf{I}_2 - \begin{bmatrix} -\cos(2k\Omega) & \sin(2k\Omega) \\ \sin(2k\Omega) & \cos(2k\Omega) \end{bmatrix} \right), \tag{A.13}$$

where $\overleftarrow{\upsilon} \triangleq \gamma\rho^2$. The sum over the geometric series $\overleftarrow{\upsilon}^k$ is standard. For the sum involving the trigonometric terms $\overleftarrow{\upsilon}^k \cos(2k\Omega)$ and $\overleftarrow{\upsilon}^k \sin(2k\Omega)$, complex expansions of these terms can be used and the terms in (3.39), (3.52), and (3.53) result. In the case where $\overleftarrow{\upsilon} = 1$, these equations simplify to (A.5) and (A.6) and (3.56) results.

In the case where $\overleftarrow{\upsilon} < 1$, the steady-state precision matrix $\overrightarrow{\mathbf{W}}_X$ given in Equation (3.61) is obtained by letting K go to infinity in Equation (A.13).

For the analytic expression (3.55) of the weighted mean we again use an approach analogous to the one of Section 2.6 as

$$\overleftarrow{\mathbf{W}}_{X_0} \overleftarrow{\boldsymbol{m}}_{X_0} = \frac{1}{\sigma_Z^2} \sum_{k=1}^{K} \gamma^k \tilde{y}_k \left(\mathbf{A}^k\right)^{\mathsf{T}} \boldsymbol{c}^{\mathsf{T}} \tag{A.14}$$

$$= \frac{1}{\sigma_Z^2} \sum_{k=1}^{K} (\gamma\rho)^k \tilde{y}_k \begin{bmatrix} \cos(k\Omega) \\ -\sin(k\Omega) \end{bmatrix}. \tag{A.15}$$

Appendix B

Proofs for Chapter 4

B.1 About Rotation Matrices

We define the un-scaled rotation matrix for an angle α as

$$\mathrm{rotm}(\alpha) \triangleq \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}, \tag{B.1}$$

and the scaled rotation matrix for a vector $\boldsymbol{x} \triangleq \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right] \triangleq \rho \left[\begin{smallmatrix} \cos\alpha \\ \sin\alpha \end{smallmatrix}\right]$ as

$$\mathrm{rotm}(\boldsymbol{x}) \triangleq \begin{bmatrix} x_1 & -x_2 \\ x_2 & x_1 \end{bmatrix} = \rho\,\mathrm{rotm}(\alpha)\,, \tag{B.2}$$

where $\rho = \sqrt{\boldsymbol{x}^\mathsf{T}\boldsymbol{x}}$, and $\alpha = \arctan2(x_1, x_2)$. In (B.1) and (B.2), we have defined two different operators, for both of which we use the symbol $\mathrm{rotm}(\cdot)$. They are distinguished solely by the dimensionality of the argument.

A rotation matrix $\mathbf{R} \triangleq \rho\,\mathrm{rotm}(\alpha)$ maps a point in $\mathbb{R}^2$ as follows:

$$\mathbb{R}^2 \to \mathbb{R}^2 \tag{B.3}$$

$$\boldsymbol{x} \to \mathbf{R}\boldsymbol{x} \tag{B.4}$$

$$\nu \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} \to \nu\rho \begin{bmatrix} \cos(\phi+\alpha) \\ \sin(\phi+\alpha) \end{bmatrix}. \tag{B.5}$$

We do not consider rotations in higher dimensions than $\mathbb{R}^2$.

The following properties of rotation matrices can easily be verified:

$$\operatorname{rotm}(\boldsymbol{x})\ \operatorname{rotm}(\boldsymbol{y}) = \operatorname{rotm}(\boldsymbol{y})\ \operatorname{rotm}(\boldsymbol{x}) \tag{B.6}$$

$$\operatorname{rotm}(\boldsymbol{x})\,\boldsymbol{y} = \operatorname{rotm}(\boldsymbol{y})\,\boldsymbol{x} \tag{B.7}$$

$$\operatorname{rotm}\big(\operatorname{rotm}(\boldsymbol{x})\,\boldsymbol{y}\big) = \operatorname{rotm}(\boldsymbol{x})\ \operatorname{rotm}(\boldsymbol{y})\,. \tag{B.8}$$

The $\operatorname{rotm}_{n,m}$ operator as defined in (4.30) consists merely of a stack of rotm operators as in (B.2) and ordinary scalar multiplications. The properties (B.6)–(B.8) also hold in this case

$$\operatorname{rotm}_{n,m}(\boldsymbol{x})\ \operatorname{rotm}_{n,m}(\boldsymbol{y}) = \operatorname{rotm}_{n,m}(\boldsymbol{y})\ \operatorname{rotm}_{n,m}(\boldsymbol{x})\,, \tag{B.9}$$

$$\operatorname{rotm}_{n,m}(\boldsymbol{x})\,\boldsymbol{y} = \operatorname{rotm}_{n,m}(\boldsymbol{y})\,\boldsymbol{x}\,, \tag{B.10}$$

$$\operatorname{rotm}_{n,m}\big(\operatorname{rotm}_{n,m}(\boldsymbol{x})\,\boldsymbol{y}\big) = \operatorname{rotm}_{n,m}(\boldsymbol{x})\ \operatorname{rotm}_{n,m}(\boldsymbol{y})\,. \tag{B.11}$$

The $\operatorname{rotm}_{n,m}$ matrix operator can be decomposed as follows:

$$\operatorname{rotm}_{n,m}(\boldsymbol{x}) = \bar{\mathbf{I}}\ \operatorname{diag}(\boldsymbol{x})\,\bar{\mathbf{I}} + \underline{\mathbf{I}}_1\ \operatorname{diag}(\boldsymbol{x})\,\underline{\mathbf{I}}_2 + \underline{\mathbf{I}}_3\ \operatorname{diag}(\boldsymbol{x})\,\underline{\mathbf{I}}_4\,, \tag{B.12}$$

where

$$\bar{\mathbf{I}} \triangleq \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{2m} \end{bmatrix} \tag{B.13}$$

$$\underline{\mathbf{I}}_1 \triangleq \begin{bmatrix} \mathbf{0}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \otimes \left[\begin{smallmatrix} 1 & 1 \\ 0 & 0 \end{smallmatrix}\right] \end{bmatrix}, \qquad \underline{\mathbf{I}}_2 \triangleq \begin{bmatrix} \mathbf{0}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \otimes \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right] \end{bmatrix}, \tag{B.14}$$

$$\underline{\mathbf{I}}_3 \triangleq \begin{bmatrix} \mathbf{0}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \otimes \left[\begin{smallmatrix} 0 & 0 \\ 1 & 1 \end{smallmatrix}\right] \end{bmatrix}, \qquad \underline{\mathbf{I}}_4 \triangleq \begin{bmatrix} \mathbf{0}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \otimes \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right] \end{bmatrix}. \tag{B.15}$$

Property (B.12) is not the only way of decomposing a rotation matrix. All possible decompositions into matrices with unit Frobenius norm can be derived from

$$\operatorname{rotm}(\boldsymbol{x}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \operatorname{diag}(\boldsymbol{x}) \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & i \\ 1 & 0 \end{bmatrix} \operatorname{diag}(\boldsymbol{x}) \begin{bmatrix} 0 & 1 \\ 0 & i \end{bmatrix} \tag{B.16}$$

$$= \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \operatorname{diag}(\boldsymbol{x}) \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & i \\ 0 & 0 \end{bmatrix} \operatorname{diag}(\boldsymbol{x}) \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}. \tag{B.17}$$

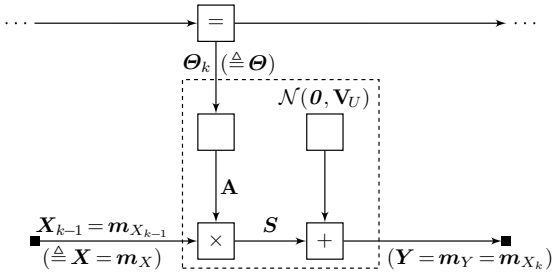


Figure B.1: Local view for cyclic maximization (CM).

B.2 Proofs for Theorem 4.1

B.2.1 Proof of Equations (4.32)–(4.34)

In Figure B.1, we have drawn the factor graph for the second CM step (4.8). Note that all state variables are fixed as $\boldsymbol{X}_k = \boldsymbol{m}_{X_k}$. We use the local labels $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{\Theta}$ in the factor graph of Figure B.1. We compute $\overleftarrow{\mu}_{\Theta}$ from $\overleftarrow{\mu}_S(\boldsymbol{s}) = \mathcal{N}(\boldsymbol{s}|\boldsymbol{m}_Y, \mathbf{V}_U)$ as

$$\overleftarrow{\mu}_{\Theta}(\boldsymbol{\theta}) = \int \mathcal{N}(\boldsymbol{s}|\boldsymbol{m}_Y, \mathbf{V}_U)\, \delta\big(\boldsymbol{s} - \mathbf{A}(\boldsymbol{\theta})\, \boldsymbol{m}_X\big)\, \mathrm{d}\boldsymbol{s} \tag{B.18}$$

$$= \mathcal{N}\big(\mathbf{A}(\boldsymbol{\theta})\, \boldsymbol{m}_X \big| \boldsymbol{m}_Y, \mathbf{V}_U\big) \tag{B.19}$$

$$\propto \exp\Big(-(\mathbf{A}(\boldsymbol{\theta})\, \boldsymbol{m}_X - \boldsymbol{m}_Y)^{\mathsf{T}} \mathbf{W}_U (\mathbf{A}(\boldsymbol{\theta})\, \boldsymbol{m}_X - \boldsymbol{m}_Y)/2\Big) \tag{B.20}$$

$$= \exp\Big(-(\mathbf{R}\, \boldsymbol{\theta} - \boldsymbol{m}_Y)^{\mathsf{T}} \mathbf{W}_U (\mathbf{R}\, \boldsymbol{\theta} - \boldsymbol{m}_Y)/2\Big) \tag{B.21}$$

$$= \exp\big(-\big(\boldsymbol{\theta} - \mathbf{R}^{-1}\boldsymbol{m}_Y\big)\mathbf{R}^{\mathsf{T}} \mathbf{W}_U \mathbf{R}\, \big(\boldsymbol{\theta} - \mathbf{R}^{-1}\boldsymbol{m}_Y\big)/2\big)\,, \tag{B.22}$$

where $\mathbf{R} \triangleq \mathrm{rotm}_{n,m}(\boldsymbol{m}_X)$. In (B.21) we have used Property (B.10) of the $\mathrm{rotm}_{n,m}$ operator.

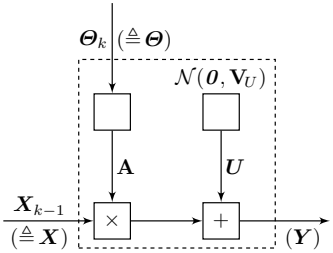


Figure B.2: Local view for expectation maximization (EM).

B.2.2 Proof of Equations (4.35) and (4.36)

We start directly from Equation (89) in [26]. The EM message with respect to the local factor graph in Figure B.2 is $\overleftarrow{\mu}_{\Theta}(\boldsymbol{\theta}) \propto e^{\eta(\boldsymbol{\theta})}$ with

$$\eta(\boldsymbol{\theta}) = -\frac{1}{2} \underbrace{\mathsf{E}\big[\boldsymbol{X}^{\mathsf{T}}\, \mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_Z\, \mathbf{A}(\boldsymbol{\theta})\, \boldsymbol{X}\big]}_{\triangleq E_1} + \underbrace{\mathsf{E}\big[\boldsymbol{X}^{\mathsf{T}}\, \mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U\, \boldsymbol{Y}\big]}_{\triangleq E_2}, \quad \text{(B.23)}$$

where the expectation in E_1 is with respect to

$$p\big(\boldsymbol{x}\,\big|\,\hat{\boldsymbol{\theta}}\big) \propto \overrightarrow{\mu}_X(\boldsymbol{x})\, \overleftarrow{\mu}_X(\boldsymbol{x})\,, \quad \text{(B.24)}$$

and the expectation in E_2 is with respect to

$$p\big(\boldsymbol{x}, \boldsymbol{y}\,\big|\,\hat{\boldsymbol{\theta}}\big) \propto \overrightarrow{\mu}_X(\boldsymbol{x})\, \overleftarrow{\mu}_Y(\boldsymbol{y})\, f_k(\boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{\theta}})\,. \quad \text{(B.25)}$$

We compute the first expectation as

$$\begin{aligned}
E_1 &= \mathsf{E}\big[(\boldsymbol{X} - \boldsymbol{m}_X)^{\mathsf{T}} \mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U \mathbf{A}(\boldsymbol{\theta})(\boldsymbol{X} - \boldsymbol{m}_X)\big] \\
&\quad + \boldsymbol{m}_X^{\mathsf{T}}\, \mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U\, \mathbf{A}(\boldsymbol{\theta})\, \boldsymbol{m}_X && \text{(B.26)} \\
&= \mathsf{E}\big[\mathrm{tr}\big((\boldsymbol{X} - \boldsymbol{m}_X)^{\mathsf{T}} \mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U\, \mathbf{A}(\boldsymbol{\theta})(\boldsymbol{X} - \boldsymbol{m}_X)\big)\big] \\
&\quad + \boldsymbol{m}_X^{\mathsf{T}} \mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U\, \mathbf{A}(\boldsymbol{\theta}) \boldsymbol{m}_X && \text{(B.27)} \\
&= \mathsf{E}\big[\mathrm{tr}\big(\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U\, \mathbf{A}(\boldsymbol{\theta})(\boldsymbol{X} - \boldsymbol{m}_X)(\boldsymbol{X} - \boldsymbol{m}_X)^{\mathsf{T}}\big)\big] \\
&\quad + \boldsymbol{m}_X^{\mathsf{T}}\, \mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U\, \mathbf{A}(\boldsymbol{\theta})\, \boldsymbol{m}_X && \text{(B.28)} \\
&= \mathrm{tr}\big(\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U\, \mathbf{A}(\boldsymbol{\theta})\, \mathsf{E}\big[(\boldsymbol{X} - \boldsymbol{m}_X)(\boldsymbol{X} - \boldsymbol{m}_X)^{\mathsf{T}}\big]\big) \\
&\quad + \boldsymbol{m}_X^{\mathsf{T}}\, \mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U\, \mathbf{A}(\boldsymbol{\theta})\, \boldsymbol{m}_X && \text{(B.29)} \\
&= \underbrace{\mathrm{tr}\big(\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U\, \mathbf{A}(\boldsymbol{\theta})\, \mathbf{V}_X\big)}_{\triangleq T_1} + \underbrace{\boldsymbol{m}_X^{\mathsf{T}}\, \mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\, \mathbf{W}_U\, \mathbf{A}(\boldsymbol{\theta})\, \boldsymbol{m}_X}_{\triangleq S_1}\,. && \text{(B.30)}
\end{aligned}$$

Using Property (B.10) we can write the term S_1 as

$$S_1 = \boldsymbol{\theta}^{\mathsf{T}} \operatorname{rotm}_{n,m}(\boldsymbol{m}_X)^{\mathsf{T}} \mathbf{W}_U \operatorname{rotm}_{n,m}(\boldsymbol{m}_X)\, \boldsymbol{\theta}\,. \tag{B.31}$$

By applying Property (B.12) to $\mathbf{A}(\boldsymbol{\theta})$, the trace T_1 in E_1 can be written as

$$\begin{aligned} T_1 &= \operatorname{tr}\Big(\big(\bar{\mathbf{I}}\,\mathbf{T}\,\bar{\mathbf{I}} + \underline{\mathbf{I}}_1\,\mathbf{T}\,\underline{\mathbf{I}}_2 + \underline{\mathbf{I}}_3\,\mathbf{T}\,\underline{\mathbf{I}}_4\big)^{\mathsf{T}} \mathbf{W}_U \\ &\qquad \cdot \big(\bar{\mathbf{I}}\,\mathbf{T}\,\bar{\mathbf{I}} + \underline{\mathbf{I}}_1\,\mathbf{T}\,\underline{\mathbf{I}}_2 + \underline{\mathbf{I}}_3\,\mathbf{T}\,\underline{\mathbf{I}}_4\big)\,\mathbf{V}_X \Big) \qquad \text{(B.32)} \\ &= \operatorname{tr}\big(\bar{\mathbf{I}}\,\mathbf{T}\,\mathbf{W}_U\,\mathbf{T}\,\bar{\mathbf{I}}\,\mathbf{V}_X\big) + 2\operatorname{tr}\big(\underline{\mathbf{I}}_2^{\mathsf{T}}\,\mathbf{T}\,\underline{\mathbf{I}}_1^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_3\,\mathbf{T}\,\underline{\mathbf{I}}_4\,\mathbf{V}_X\big) \\ &\quad + \operatorname{tr}\big(\underline{\mathbf{I}}_2^{\mathsf{T}}\,\mathbf{T}\,\underline{\mathbf{I}}_1^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_1\,\mathbf{T}\,\underline{\mathbf{I}}_2\,\mathbf{V}_X\big) + \operatorname{tr}\big(\underline{\mathbf{I}}_4^{\mathsf{T}}\,\mathbf{T}\,\underline{\mathbf{I}}_3^{\mathsf{T}}\,\mathbf{W}_U\underline{\mathbf{I}}_3\,\mathbf{T}\,\underline{\mathbf{I}}_4\,\mathbf{V}_X\big) \qquad \text{(B.33)} \\ &= \operatorname{tr}\big(\mathbf{T}\,\mathbf{W}_U\,\mathbf{T}\,\bar{\mathbf{I}}\,\mathbf{V}_X\,\bar{\mathbf{I}}\big) + 2\operatorname{tr}\big(\mathbf{T}\,\underline{\mathbf{I}}_1^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_3\,\mathbf{T}\,\underline{\mathbf{I}}_4\,\mathbf{V}_X\,\underline{\mathbf{I}}_2^{\mathsf{T}}\big) \\ &\quad + \operatorname{tr}\big(\mathbf{T}\,\underline{\mathbf{I}}_1^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_1\,\mathbf{T}\,\underline{\mathbf{I}}_2\mathbf{V}_X\,\underline{\mathbf{I}}_2^{\mathsf{T}}\big) + \operatorname{tr}\big(\mathbf{T}\,\underline{\mathbf{I}}_3^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_3\,\mathbf{T}\,\underline{\mathbf{I}}_4\,\mathbf{V}_X\,\underline{\mathbf{I}}_4^{\mathsf{T}}\big)\,, \qquad \text{(B.34)} \end{aligned}$$

where $\mathbf{T} \triangleq \operatorname{diag}(\boldsymbol{\theta})$. Next, we apply the identity (E.9), which we repeat here for convenience:

$$\operatorname{tr}\big(\operatorname{diag}(\boldsymbol{x})\,\mathbf{A}\,\operatorname{diag}(\boldsymbol{y})\,\mathbf{B}^{\mathsf{T}}\big) = \boldsymbol{x}^{\mathsf{T}}(\mathbf{A} \odot \mathbf{B})\,\boldsymbol{y}\,. \tag{B.35}$$

Equation (B.35) is proved in E.2 using the linear algebra interpretation of factor graphs.

$$\begin{aligned} T_1 &= \boldsymbol{\theta}^{\mathsf{T}}\big(\mathbf{W}_U \odot \big(\bar{\mathbf{I}}\,\mathbf{V}_X\,\bar{\mathbf{I}}\big)\big)\boldsymbol{\theta} + \boldsymbol{\theta}^{\mathsf{T}}\Big(2\big(\underline{\mathbf{I}}_1^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_3\big) \odot \big(\underline{\mathbf{I}}_4\,\mathbf{V}_X\,\underline{\mathbf{I}}_2^{\mathsf{T}}\big)\Big)\boldsymbol{\theta} \\ &\quad + \boldsymbol{\theta}^{\mathsf{T}}\Big(\big(\underline{\mathbf{I}}_1^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_1\big) \odot \big(\underline{\mathbf{I}}_2\,\mathbf{V}_X\,\underline{\mathbf{I}}_2^{\mathsf{T}}\big)\Big)\boldsymbol{\theta} \qquad \text{(B.36)} \\ &\quad + \boldsymbol{\theta}^{\mathsf{T}}\Big(\big(\underline{\mathbf{I}}_3^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_3\big) \odot \big(\underline{\mathbf{I}}_4\,\mathbf{V}_X\,\underline{\mathbf{I}}_4^{\mathsf{T}}\big)\Big)\boldsymbol{\theta} \\ &= \boldsymbol{\theta}^{\mathsf{T}}\Big(\mathbf{W}_U \odot \big(\bar{\mathbf{I}}\,\mathbf{V}_X\,\bar{\mathbf{I}}\big) + 2\big(\underline{\mathbf{I}}_1^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_3\big) \odot \big(\underline{\mathbf{I}}_4\,\mathbf{V}_X\,\underline{\mathbf{I}}_2^{\mathsf{T}}\big) \\ &\quad + \big(\underline{\mathbf{I}}_1^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_1\big) \odot \big(\underline{\mathbf{I}}_2\,\mathbf{V}_X\,\underline{\mathbf{I}}_2^{\mathsf{T}}\big) \qquad \text{(B.37)} \\ &\quad + \big(\underline{\mathbf{I}}_3^{\mathsf{T}}\,\mathbf{W}_U\,\underline{\mathbf{I}}_3\big) \odot \big(\underline{\mathbf{I}}_4\,\mathbf{V}_X\,\underline{\mathbf{I}}_4^{\mathsf{T}}\big)\Big)\boldsymbol{\theta}\,. \end{aligned}$$

We now turn to the second expectation:

$$
\begin{aligned}
E_2 &= \mathsf{E}\big[(\boldsymbol{X}-\boldsymbol{m}_X)^{\mathsf{T}}\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\,\mathbf{W}_U(\boldsymbol{Y}-\boldsymbol{m}_Y)\big] \\
&\quad + \boldsymbol{m}_X^{\mathsf{T}}\,\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\,\mathbf{W}_U\,\boldsymbol{m}_Y && \text{(B.38)}\\
&= \mathsf{E}\big[\mathrm{tr}\big((\boldsymbol{X}-\boldsymbol{m}_X)^{\mathsf{T}}\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\,\mathbf{W}_U(\boldsymbol{Y}-\boldsymbol{m}_Y)\big)\big] \\
&\quad + \boldsymbol{m}_X^{\mathsf{T}}\,\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\,\mathbf{W}_U\,\boldsymbol{m}_Y && \text{(B.39)}\\
&= \mathsf{E}\big[\mathrm{tr}\big(\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\,\mathbf{W}_U(\boldsymbol{Y}-\boldsymbol{m}_Y)(\boldsymbol{X}-\boldsymbol{m}_X)^{\mathsf{T}}\big)\big] \\
&\quad + \boldsymbol{m}_X^{\mathsf{T}}\,\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\,\mathbf{W}_U\,\boldsymbol{m}_Y && \text{(B.40)}\\
&= \mathrm{tr}\big(\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\,\mathbf{W}_U\,\mathsf{E}\big[(\boldsymbol{Y}-\boldsymbol{m}_Y)(\boldsymbol{X}-\boldsymbol{m}_X)^{\mathsf{T}}\big]\big) \\
&\quad + \boldsymbol{m}_X^{\mathsf{T}}\,\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\,\mathbf{W}_U\,\boldsymbol{m}_Y && \text{(B.41)}\\
&= \underbrace{\mathrm{tr}\big(\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\big)}_{\triangleq T_2} + \underbrace{\boldsymbol{m}_X^{\mathsf{T}}\,\mathbf{A}(\boldsymbol{\theta})^{\mathsf{T}}\,\mathbf{W}_U\,\boldsymbol{m}_Y}_{\triangleq S_2}\,. && \text{(B.42)}
\end{aligned}
$$

Using Property (B.10) we can write the term S_2 as

$$
S_2 = \boldsymbol{\theta}^{\mathsf{T}}\;\mathrm{rotm}_{n,m}(\boldsymbol{m}_X)^{\mathsf{T}}\,\mathbf{W}_U\,\boldsymbol{m}_Y\,. \tag{B.43}
$$

By applying Property (B.12) to $\mathbf{A}(\boldsymbol{\theta})$, the trace T_2 in E_2 can be written as

$$
\begin{aligned}
T_2 &= \mathrm{tr}\big(\big(\bar{\mathbf{I}}\,\mathbf{T}\,\bar{\mathbf{I}} + \mathbf{J}_1\,\mathbf{T}\,\underline{\mathbf{I}}_2 + \underline{\mathbf{I}}_3\,\mathbf{T}\,\underline{\mathbf{I}}_4\big)\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\big) && \text{(B.44)}\\
&= \mathrm{tr}\big(\bar{\mathbf{I}}\,\mathbf{T}\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\big) + \mathrm{tr}\big(\underline{\mathbf{I}}_1\,\mathbf{T}\,\underline{\mathbf{I}}_2\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\big) \\
&\quad + \mathrm{tr}\big(\underline{\mathbf{I}}_3\,\mathbf{T}\,\underline{\mathbf{I}}_4\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\big) && \text{(B.45)}\\
&= \mathrm{tr}\big(\mathbf{T}\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\,\bar{\mathbf{I}}\big) + \mathrm{tr}(\mathbf{T}\,\underline{\mathbf{I}}_2\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\,\underline{\mathbf{I}}_1) \\
&\quad + \mathrm{tr}(\mathbf{T}\,\underline{\mathbf{I}}_4\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\,\underline{\mathbf{I}}_3)\,, && \text{(B.46)}
\end{aligned}
$$

where $\mathbf{T} \triangleq \mathrm{diag}(\boldsymbol{\theta})$. Now we use again the identity (B.35):

$$
\begin{aligned}
T_2 &= \boldsymbol{\theta}^{\mathsf{T}}\,\mathrm{diag}\big(\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\,\bar{\mathbf{I}}\big) + \boldsymbol{\theta}^{\mathsf{T}}\,\mathrm{diag}\big(\underline{\mathbf{I}}_2\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\,\underline{\mathbf{I}}_1\big) \\
&\quad + \boldsymbol{\theta}^{\mathsf{T}}\,\mathrm{diag}\big(\underline{\mathbf{I}}_4\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\,\underline{\mathbf{I}}_3\big) && \text{(B.47)}\\
&= \boldsymbol{\theta}^{\mathsf{T}}\,\mathrm{diag}\big(\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\,\bar{\mathbf{I}} + \underline{\mathbf{I}}_2\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\,\underline{\mathbf{I}}_1 \\
&\qquad + \underline{\mathbf{I}}_4\,\mathbf{W}_U\,\mathbf{V}_{XY^\tau}\,\underline{\mathbf{I}}_3\big)\,. && \text{(B.48)}
\end{aligned}
$$

We finally insert the terms $E_i = T_i + S_i$ for $i = 1, 2$ in the exponent $\eta(\boldsymbol{\theta})$ (Equation (B.23) and compare the resulting EM message with the scaled

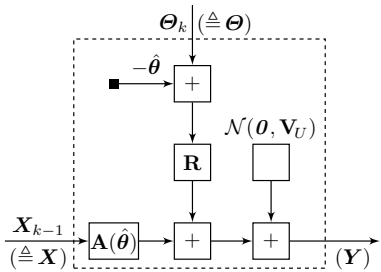


Figure B.3: Local view for linearization.

Gaussian PDF

$$\overleftarrow{\mu}_{\Theta}(\boldsymbol{\theta}) \propto \exp\Big(-\boldsymbol{\theta}^{\mathsf{T}}\, \overleftarrow{\mathbf{W}}_{\Theta}\, \boldsymbol{\theta}/2 + \boldsymbol{\theta}^{\mathsf{T}}\, \overleftarrow{\mathbf{W}}_{\Theta}\, \overleftarrow{\boldsymbol{m}}_{\Theta}\Big) \tag{B.49}$$

and Part (b) of Theorem 4.1 is proved.

B.2.3 Proof of Equations (4.39)–(4.41)

First we note that Example 4.2 generalizes without modification from the rotm operator to the $\mathrm{rotm}_{n,m}$ operator. We repeat the procedure in the following. Consider the constraint $f(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y}) = \delta\big(h(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y})\big)$, where $h(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y}) \triangleq \boldsymbol{y} - \mathrm{rotm}_{n,m}(\boldsymbol{\theta})\, \boldsymbol{x}$ with $\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y} \in \mathbb{R}^{n+2m}$. Since the operating point $\hat{\boldsymbol{x}} = \overrightarrow{\boldsymbol{m}}_X$, $\hat{\boldsymbol{\theta}}$ arbitrary, $\hat{\boldsymbol{y}} = \mathbf{A}(\hat{\boldsymbol{\theta}})\, \overrightarrow{\boldsymbol{m}}_X$ satisfies the constraint we can write (4.20) as

$$\tilde{h}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y}) = \underbrace{\begin{bmatrix} -\mathbf{A}(\hat{\boldsymbol{\theta}}) & -\mathbf{R} & \mathbf{I}_{n+2m} \end{bmatrix}}_{\triangleq \mathbf{H}} \begin{bmatrix} \boldsymbol{x} - \overrightarrow{\boldsymbol{m}}_X \\ \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \\ \boldsymbol{y} - \mathbf{A}(\hat{\boldsymbol{\theta}})\, \overrightarrow{\boldsymbol{m}}_X \end{bmatrix} \tag{B.50}$$

$$= \boldsymbol{y} - \mathbf{A}(\hat{\boldsymbol{\theta}})\, \overrightarrow{\boldsymbol{m}}_X - \mathbf{A}(\hat{\boldsymbol{\theta}})(\boldsymbol{x} - \overrightarrow{\boldsymbol{m}}_X) - \mathbf{R}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \tag{B.51}$$

$$= \boldsymbol{y} - \mathbf{A}(\hat{\boldsymbol{\theta}})\, \boldsymbol{x} - \mathbf{R}\, \boldsymbol{\theta} + \mathbf{R}\, \hat{\boldsymbol{\theta}} \tag{B.52}$$

$$= \boldsymbol{y} - \mathbf{A}(\hat{\boldsymbol{\theta}})\, \boldsymbol{x} - \mathbf{R}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\,, \tag{B.53}$$

where $\mathbf{R} \triangleq \mathrm{rotm}_{n,m}(\overrightarrow{\boldsymbol{m}}_X)$.

Figure B.3 shows the resulting local factor graph (including the state noise $\boldsymbol{U}$ for easier comparison with Figures 4.7, B.1, and B.2). The

expression for $\overleftarrow{\mathbf{V}}_\Theta$ in (4.39) follows from straightforward applications of update rules for Gaussian messages. Equation (4.41) is derived as

$$\begin{aligned}\overleftarrow{\boldsymbol{m}}_\Theta &= \mathbf{R}^{-1}\,\overleftarrow{\boldsymbol{m}}_Y - \mathbf{R}^{-1}\,\mathbf{A}(\hat{\boldsymbol{\theta}})\,\overrightarrow{\boldsymbol{m}}_X + \hat{\boldsymbol{\theta}} && \text{(B.54)}\\ &= \mathbf{R}^{-1}\,\overleftarrow{\boldsymbol{m}}_Y - \mathbf{R}^{-1}\,\mathbf{R}\,\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\theta}} && \text{(B.55)}\\ &= \mathbf{R}^{-1}\,\overleftarrow{\boldsymbol{m}}_Y\,, && \text{(B.56)}\end{aligned}$$

where we have used Property (B.10) in the second equality. Equation (4.40) results straightforwardly from (4.39) and (4.41).

B.3 Variance Estimation by Expectation Maximization

B.3.1 Inverse-Wishart and Inverse-Gamma Distribution

The inverse-Wishart PDF on a matrix $\mathbf{s} \in \mathbb{S}^n_{\succ 0}$ is

$$\mathcal{W}^{-1}(\mathbf{s}|\nu,\boldsymbol{\Psi}) = \zeta\,(\det \mathbf{s})^{-(\nu+n+1)/2}\,e^{-\operatorname{tr}(\boldsymbol{\Psi}\,\mathbf{s}^{-1})/2}\,, \tag{B.57}$$

where

$$\zeta = \frac{(\det \boldsymbol{\Psi})^{\nu/2}}{2^{\nu n/2}\,\Gamma_n\left(\frac{\nu}{2}\right)} \tag{B.58}$$

is the normalization constant in which Γ_n is the multivariate gamma function, $\boldsymbol{\Psi} \in \mathbb{S}^n_{\succ 0}$ is the inverse scale matrix, and $\nu > n-1$ are the degrees of freedom. The inverse-Wishart PDF has one mode at

$$\operatorname*{argmax}_{\mathbf{s}\in\mathbb{S}^n_{\succ 0}} \mathcal{W}^{-1}(\mathbf{s}|\nu,\boldsymbol{\Psi}) = \frac{\boldsymbol{\Psi}}{\nu+n+1}\,. \tag{B.59}$$

The inverse-Gamma PDF on a scalar $s \in \mathbb{R}_{>0}$ is

$$\mathcal{G}^{-1}(s|\alpha,\beta) = \zeta\,s^{-(\alpha+1)}\,e^{-\beta/s}\,, \tag{B.60}$$

where

$$\zeta = \frac{\beta^\alpha}{\Gamma(\alpha)} \tag{B.61}$$

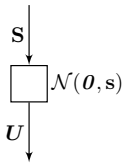


Figure B.4: Local view for expectation maximization (EM).

is the normalization constant in which Γ is the gamma function. The inverse-Gamma PDF has one mode at

$$\operatorname*{argmax}_{s \in \mathbb{R}_{>0}} \mathcal{G}^{-1}(s|\alpha, \beta) = \frac{\beta}{\alpha + 1}\,. \tag{B.62}$$

B.3.2 Expectation Maximization Message: the Matrix Case

According to Equation (4.10), the EM message with respect to the local factor graph in Figure B.4 is $\overleftarrow{\mu}_S(\mathbf{s}) \propto e^{\eta(\mathbf{s})}$ with

$$\eta(\mathbf{s}) = \mathsf{E}_{p_{\text{local}}}[\ln \mathcal{N}(\boldsymbol{U}\,|\,\boldsymbol{0}, \mathbf{s})]\,, \tag{B.63}$$

where

$$p_{\text{local}}\big(\boldsymbol{x}\,\big|\,\hat{\mathbf{s}}\big) \propto \overrightarrow{\mu}_U(\boldsymbol{u})\, \overleftarrow{\mu}_U(\boldsymbol{u})\,. \tag{B.64}$$

Now we formulate $\eta(\mathbf{s})$ up to a constant as

$$\begin{aligned}
\eta(\mathbf{s}) &= \ln(\det \mathbf{s})^{-1/2} - \mathsf{E}\big[\boldsymbol{X}^\mathsf{T}\mathbf{s}^{-1}\boldsymbol{X}\big]/2 + \text{const} && \text{(B.65)}\\
&= \ln(\det \mathbf{s})^{-1/2} - \mathsf{E}\big[\operatorname{tr}\big(\boldsymbol{X}^\mathsf{T}\mathbf{s}^{-1}\boldsymbol{X}\big)\big]/2 + \text{const} && \text{(B.66)}\\
&= \ln(\det \mathbf{s})^{-1/2} - \mathsf{E}\big[\operatorname{tr}\big(\boldsymbol{X}\boldsymbol{X}^\mathsf{T}\mathbf{s}^{-1}\big)\big]/2 + \text{const} && \text{(B.67)}\\
&= \ln(\det \mathbf{s})^{-1/2} - \operatorname{tr}\big(\mathsf{E}\big[\boldsymbol{X}\boldsymbol{X}^\mathsf{T}\big]\mathbf{s}^{-1}\big)/2 + \text{const} && \text{(B.68)}\\
&= \ln(\det \mathbf{s})^{-1/2} - \operatorname{tr}\Big(\big(\mathbf{V}_X - \boldsymbol{m}_X\boldsymbol{m}_X^\mathsf{T}\big)\mathbf{s}^{-1}\Big)/2 + \text{const}\,, && \text{(B.69)}
\end{aligned}$$

such that $\overleftarrow{\mu}_S$ is proportional to an inverse-Wishart PDF (B.57) with parameters $\overleftarrow{\nu}_S$ and $\overleftarrow{\boldsymbol{\Psi}}_S$ as given in (4.59) and (4.60) respectively.

B.4 Proof of Equations (4.51) and (4.52)

We start from Equation (4.47) by plugging in the message parameters $\overleftarrow{\mathbf{W}}_{\Theta_k}$ and $\overleftarrow{\mathbf{W}}_{\Theta_k}\overleftarrow{\boldsymbol{m}}_{\Theta_k}$ from (4.32) and (4.33) respectively and by using the definition $\mathbf{W}_U \triangleq \sigma_U^{-2}\mathbf{I}_{2M}$ as

$$\begin{aligned}\ln \overleftarrow{\mu}_{\Omega_k}(\omega) &= \frac{-\boldsymbol{\theta}(\omega)^{\mathsf{T}}\,\mathrm{rotm}_{0,M}(\boldsymbol{m}_{X_{k-1}})^{\mathsf{T}}\,\mathrm{rotm}_{0,M}(\boldsymbol{m}_{X_{k-1}})\,\boldsymbol{\theta}(\omega)}{\sigma_U^2} \\ &\quad + \frac{\boldsymbol{\theta}(\omega)^{\mathsf{T}}\,\mathrm{rotm}_{0,M}(\boldsymbol{m}_{X_{k-1}})^{\mathsf{T}}\boldsymbol{m}_{X_k}}{\sigma_U^2} + \mathrm{const} \qquad \text{(B.70)} \\ &\propto -\boldsymbol{m}_{X_{k-1}}^{\mathsf{T}}\,\mathrm{rotm}_{0,M}(\boldsymbol{\theta}(\omega))^{\mathsf{T}}\,\mathrm{rotm}_{0,M}(\boldsymbol{\theta}(\omega))\,\boldsymbol{m}_{X_{k-1}} \\ &\quad + \boldsymbol{m}_{X_k}^{\mathsf{T}}\,\mathrm{rotm}_{0,M}(\boldsymbol{m}_{X_{k-1}})\,\boldsymbol{\theta}(\omega) + \mathrm{const}' \qquad \text{(B.71)} \\ &= \boldsymbol{m}_{X_k}^{\mathsf{T}}\,\mathrm{rotm}_{0,M}(\boldsymbol{m}_{X_{k-1}})\,\boldsymbol{\theta}(\omega) + \mathrm{const}''\,. \qquad \text{(B.72)}\end{aligned}$$

To obtain (B.71) we have used the property (B.10) of the $\mathrm{rotm}_{n,m}(\cdot)$ operator and for (B.72) we note that due to the definition (4.46) of $\boldsymbol{\theta}(\omega)$

$$\mathrm{rotm}_{0,M}(\boldsymbol{\theta}(\omega))^{\mathsf{T}}\,\mathrm{rotm}_{0,M}(\boldsymbol{\theta}(\omega)) = \mathbf{I}_{2M}\,. \qquad \text{(B.73)}$$

The coefficients $\overleftarrow{\boldsymbol{\xi}}_{\Omega_k}$ in (4.51) and (4.52) follow directly from (B.72).

Appendix C

On Scale Factors

C.1 Neutral Modifications of Graphs

Here, we elaborate on transforming one factor graph into another by doing a local modification without changing the global function. Such transformations will be used later in some proofs. More general such transformations in a setting of discrete-valued variables are treated, e.g., in [1].

Reversing a multiplication

Assume that a factor graph contains the local factor $\delta(\boldsymbol{y} - \mathbf{A}\boldsymbol{x})$ as in Figure C.1 on the left. If $\mathbf{A}$ is nonsingular then this graph can be transformed into the one on the right, but an additional factor $|\det \mathbf{A}|^{-1}$ appears due to the fact that inside an integral we can substitute

$$\delta(\boldsymbol{y} - \mathbf{A}\boldsymbol{x}) = \frac{\delta\big(\boldsymbol{x} - \mathbf{A}^{-1}\boldsymbol{y}\big)}{|\det \mathbf{A}|} \,. \tag{C.1}$$

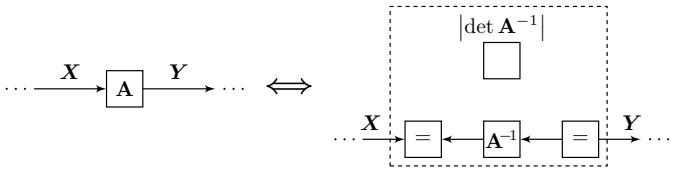


Figure C.1: Reversing a multiplication.

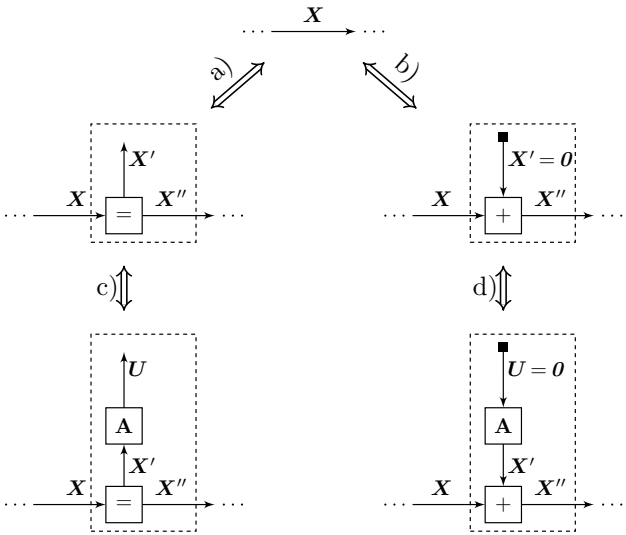


Figure C.2: Neutral modifications of a factor graph.

We emphasize that this transformation is only valid in the continuous case, i.e. if $\boldsymbol{X} \in \mathbb{R}^{n_X}$ and $\boldsymbol{Y} \in \mathbb{R}^{n_Y}$. In the discrete case, the factor $|\det \mathbf{A}|^{-1}$ does not appear. E.g., in a scalar discrete setup let $X \in \{x_1, \ldots, x_n\}$, $a \in \mathbb{R} \setminus \{0\}$, and $Y \in \{ax_1, \ldots, ax_n\}$. The constraint now is a Kronecker delta $\delta[y - ax]$ which can safely be substituted by $\delta[x - y/a]$ in the summation.

Neutral modifications

Figure C.2 shows two local modification that can be done to any edge in any factor graph without changing the global function. This is proved as follows. In all five factor graphs in Figure C.2, let $f_1(\boldsymbol{x}, \ldots)$ and $f_2(\boldsymbol{x}, \ldots)$ be the two parts of the factor graph, to which the edge $\boldsymbol{X}$ is connected (to the left and the right respectively).

Equivalences a) and c) are now proved as

$$\iiint f_1(\boldsymbol{x},\dots)\, f_2(\boldsymbol{x}'',\dots)\, \delta(\boldsymbol{x}''-\boldsymbol{x})\, \delta(\boldsymbol{x}'-\boldsymbol{x})\, \delta(\boldsymbol{u}-\mathbf{A}\boldsymbol{x}')\, \mathrm{d}\boldsymbol{u}\, \mathrm{d}\boldsymbol{x}''\, \mathrm{d}\boldsymbol{x}' \tag{C.2}$$

$$= \iint f_1(\boldsymbol{x},\dots)\, f_2(\boldsymbol{x}'',\dots)\, \delta(\boldsymbol{x}''-\boldsymbol{x})\, \delta(\boldsymbol{x}'-\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}''\, \mathrm{d}\boldsymbol{x}' \tag{C.3}$$

$$= f_1(\boldsymbol{x},\dots)\, f_2(\boldsymbol{x},\dots)\,, \tag{C.4}$$

and equivalences b) and d) are proved as

$$\iint f_1(\boldsymbol{x},\dots)\, f_2(\boldsymbol{x}'',\dots)\, \delta(\boldsymbol{x}''-\boldsymbol{x}-\boldsymbol{u})\, \delta(\boldsymbol{u}-\mathbf{A}\cdot\boldsymbol{0})\, \mathrm{d}\boldsymbol{u}\, \mathrm{d}\boldsymbol{x}'' \tag{C.5}$$

$$= \int f_1(\boldsymbol{x},\dots)\, f_2(\boldsymbol{x}'',\dots)\, \delta(\boldsymbol{x}''-\boldsymbol{x}-\boldsymbol{0})\, \mathrm{d}\boldsymbol{x}'' \tag{C.6}$$

$$= f_1(\boldsymbol{x},\dots)\, f_2(\boldsymbol{x},\dots)\,. \tag{C.7}$$

Note that the modifications in Figure C.2 are neutral also in the discrete setting, i.e., if $\boldsymbol{X}$ takes value in a finite set. In Equations (C.2)–(C.7) we simply substitute Dirac deltas by Kronecker deltas and integrations by summations.

C.2 Proofs for Table 6.1

General node $p(\boldsymbol{y}|\boldsymbol{x})$

Equation (I.1) is proved as follows:

$$\vec{\beta}_Y = \iint \vec{\mu}_X(\boldsymbol{x})\, p(\boldsymbol{y}|\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} \tag{C.8}$$

$$= \int \vec{\mu}_X(\boldsymbol{x}) \int p(\boldsymbol{y}|\boldsymbol{x})\, \mathrm{d}\boldsymbol{y}\, \mathrm{d}\boldsymbol{x} \tag{C.9}$$

$$= \int \vec{\mu}_X(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}\,. \tag{C.10}$$

Equation (I.2) is proved as follows:

$$\overleftarrow{\gamma}_X = \int \overleftarrow{\mu}_Y(\boldsymbol{y})\, p(\boldsymbol{y}|\boldsymbol{0})\, \mathrm{d}\boldsymbol{y} \tag{C.11}$$

$$= \int \overleftarrow{\mu}_Y(\boldsymbol{y})\, \delta(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y} \tag{C.12}$$

$$= \overleftarrow{\mu}_Y(\boldsymbol{0})\,. \tag{C.13}$$

Prediction node $p(\tilde{y}, \boldsymbol{z}|\boldsymbol{x})$

Equation (I.3) is proved as follows:

$$\overrightarrow{\beta}_Z = \iint \overrightarrow{\mu}_X(\boldsymbol{x})\, p(\tilde{\boldsymbol{y}}, \boldsymbol{z}|\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{z} \tag{C.14}$$

$$= \overrightarrow{\beta}_X \iint \overrightarrow{p}_X(\boldsymbol{x})\, p(\tilde{\boldsymbol{y}}, \boldsymbol{z}|\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{z} \tag{C.15}$$

$$= \overrightarrow{\beta}_X\, \overrightarrow{p}_{pY}(\tilde{\boldsymbol{y}})\,. \tag{C.16}$$

The first equality is due to the definition (6.2) of the scale factor $\overrightarrow{\beta}_X$ and the sum-product rule. In the second equality we merely have written the message $\overrightarrow{\mu}_X(\boldsymbol{x}) = \overrightarrow{\beta}_X \overrightarrow{p}_X(\boldsymbol{x})$ in terms of its scale factor and a properly scaled PDF. For the last equality, first note that the integral in Equation (C.15) is a properly scaled PDF. Second, a factor $\overleftarrow{\mu}_Z(\boldsymbol{z}) = 1$ can be inserted in the integrand without any consequences. Thus the prediction PDF is formed.

C.3 Proofs for Table 6.2

Multiplication node

Equations (II.1) and (II.2) are special cases of (I.1) and (I.2) for $p(\boldsymbol{y}|\boldsymbol{x}) = \delta(\boldsymbol{y} - \mathbf{A}\boldsymbol{x})$. Note that the condition $p(\boldsymbol{y}|\boldsymbol{0}) = \delta(\boldsymbol{y})$ for Equation (I.2) is satisfied.

For nonsingular $\mathbf{A}$, Equations (II.3) and (II.4) are proved by reversing the multiplication as in Figure C.1 and then applying (II.1) and (II.2) respectively.

Equality node

Equation (II.5) is a special case of (I.2). To prove this we rename in the graph of (II.5) the edges $\boldsymbol{X} \to \boldsymbol{U}$ and $\boldsymbol{Y} \to \boldsymbol{V}$. the special case is defined by $\boldsymbol{U} = \boldsymbol{X}$ and $\boldsymbol{V}^\mathsf{T} = \left[\boldsymbol{Y}^\mathsf{T}, \boldsymbol{Z}^\mathsf{T}\right]$, and

$$p(\boldsymbol{v}|\boldsymbol{u}) \triangleq \delta\left(\boldsymbol{v} - \begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix}\boldsymbol{u}\right). \tag{C.17}$$

With these definitions we get

$$p(\boldsymbol{v}|\boldsymbol{u}) = \delta(\boldsymbol{y} - \boldsymbol{x})\,\delta(\boldsymbol{z} - \boldsymbol{x})\,, \tag{C.18}$$

which is the constraint for the equality node.

Addition node

Equation (II.6) is a special case of (I.1). To prove this, we rename in the graph of (I.1) the edges $\boldsymbol{X} \to \boldsymbol{U}$ and $\boldsymbol{Y} \to \boldsymbol{V}$. The special case is defined by $\boldsymbol{U}^\mathsf{T} \triangleq \left[\boldsymbol{X}^\mathsf{T}, \boldsymbol{Y}^\mathsf{T}\right]$, $\boldsymbol{V} \triangleq \boldsymbol{Z}$, and

$$p(\boldsymbol{v}|\boldsymbol{u}) \triangleq \delta\big(\boldsymbol{v} - \begin{bmatrix}\mathbf{I} & -\mathbf{I}\end{bmatrix}\boldsymbol{u}\big)\,. \tag{C.19}$$

With these definitions we get

$$p(\boldsymbol{v}|\boldsymbol{u}) = \delta(\boldsymbol{z} - \boldsymbol{x} - \boldsymbol{y})\,, \tag{C.20}$$

which is the constraint of the addition node.

C.4 Proofs for Tables 6.3 and 6.4

Multiplication node

Equation (III.1) is proved by starting with (II.1), converting both sides from β to γ using (6.16) as

$$\vec{\gamma}_Y \sqrt{\frac{(2\pi)^n}{\det \vec{\mathbf{W}}_Y}}\, e^{\vec{\boldsymbol{m}}_Y^\mathsf{T} \vec{\mathbf{W}}_Y \vec{\boldsymbol{m}}_Y/2} = \vec{\gamma}_X \sqrt{\frac{(2\pi)^m}{\det \vec{\mathbf{W}}_X}}\, e^{\vec{\boldsymbol{m}}_X^\mathsf{T} \vec{\mathbf{W}}_X \vec{\boldsymbol{m}}_X/2}\,, \tag{C.21}$$

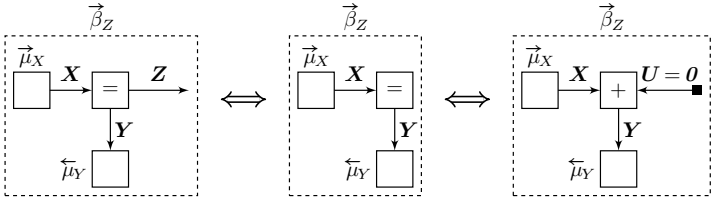


Figure C.3: Proof of (III.3).

and finally noting that

$$\vec{\boldsymbol{m}}_Y^{\mathsf{T}} \vec{\mathbf{W}}_Y \vec{\boldsymbol{m}}_Y = \vec{\boldsymbol{m}}_X^{\mathsf{T}} \mathbf{A}^{\mathsf{T}} \vec{\mathbf{W}}_Y \mathbf{A} \vec{\boldsymbol{m}}_X \,, \tag{C.22}$$

where $\vec{\mathbf{W}}_Y = \left(\mathbf{A} \vec{\mathbf{W}}_X^{-1} \mathbf{A}^{\mathsf{T}}\right)^{-1}$.

Similarly, Equation (III.2) is proved by starting with (II.2), converting both sides from γ to β using (6.16) as

$$\overleftarrow{\beta}_X \sqrt{\frac{\det \overleftarrow{\mathbf{W}}_X}{(2\pi)^m}}\, e^{-\overleftarrow{\boldsymbol{m}}_X^{\mathsf{T}} \overleftarrow{\mathbf{W}}_X \overleftarrow{\boldsymbol{m}}_X / 2} = \overleftarrow{\beta}_Y \sqrt{\frac{\det \overleftarrow{\mathbf{W}}_Y}{(2\pi)^n}}\, e^{-\overleftarrow{\boldsymbol{m}}_Y^{\mathsf{T}} \overleftarrow{\mathbf{W}}_Y \overleftarrow{\boldsymbol{m}}_Y / 2} \,, \tag{C.23}$$

and finally noting that

$$\overleftarrow{\boldsymbol{m}}_X^{\mathsf{T}} \overleftarrow{\mathbf{W}}_X \overleftarrow{\boldsymbol{m}}_X = \overleftarrow{\boldsymbol{m}}_Y^{\mathsf{T}} \overleftarrow{\mathbf{W}}_Y \mathbf{A} \overleftarrow{\mathbf{W}}_X \mathbf{A}^{\mathsf{T}} \overleftarrow{\mathbf{W}}_Y \overleftarrow{\boldsymbol{m}}_Y \,, \tag{C.24}$$

where $\overleftarrow{\mathbf{W}}_X = \left(\mathbf{A}^{\mathsf{T}} \overleftarrow{\mathbf{W}}_Y^{-1} \mathbf{A}\right)^{-1}$.

Equality node

For the proof of Equation (III.3) consider the sequence of equivalent graphs shown in Figure C.3. Clearly, the closed box value of the leftmost graph is the definition of $\vec{\beta}_Z$. The subsequent graphs are obtained using neutral modifications (cf. Section C.1). Now we write the closed box

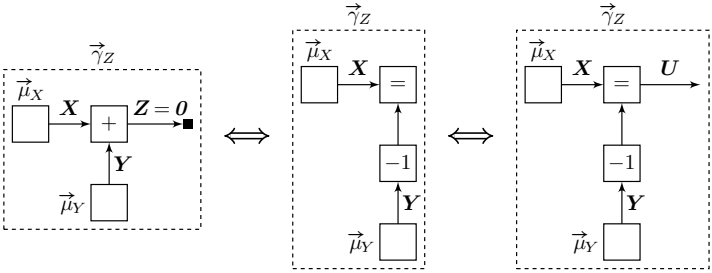


Figure C.4: Proof of (III.4) and (III.5).

value of the rightmost graph as

$$\vec{\beta}_Z = \overleftarrow{\mu}_U(\boldsymbol{0}) \tag{C.25}$$

$$= \overleftarrow{\gamma}_U \tag{C.26}$$

$$= \frac{\overleftarrow{\beta}_U}{\sqrt{(2\pi)^n \det \overleftarrow{\mathbf{V}}_U}} e^{-\overleftarrow{\boldsymbol{m}}_U^{\mathsf{T}} \overleftarrow{\mathbf{V}}_U^{-1} \overleftarrow{\boldsymbol{m}}_U / 2} \tag{C.27}$$

$$= \frac{\vec{\beta}_X \overleftarrow{\beta}_Y}{\sqrt{(2\pi)^n \det \overleftarrow{\mathbf{V}}_U}} e^{-\overleftarrow{\boldsymbol{m}}_U^{\mathsf{T}} \overleftarrow{\mathbf{V}}_U^{-1} \overleftarrow{\boldsymbol{m}}_U / 2} . \tag{C.28}$$

In the third equality we have used the translation (6.16) between β and γ. In the last equality we have used the update rule (II.7). Finally,

$$\overleftarrow{\mathbf{V}}_U = \vec{\mathbf{V}}_X + \overleftarrow{\mathbf{V}}_Y , \tag{C.29}$$

$$\overleftarrow{\boldsymbol{m}}_U = \vec{\boldsymbol{m}}_X + \overleftarrow{\boldsymbol{m}}_Y , \tag{C.30}$$

due to the standard update rule for the addition node.

Addition node

For the proof of Equation (III.4) consider the sequence of graphs shown in Figure C.4. Clearly, the closed box value of the leftmost graph is the definition of $\vec{\gamma}_Z$. The subsequent graphs are obtained using neutral modifications as in Section C.1. Now we write the closed box value of

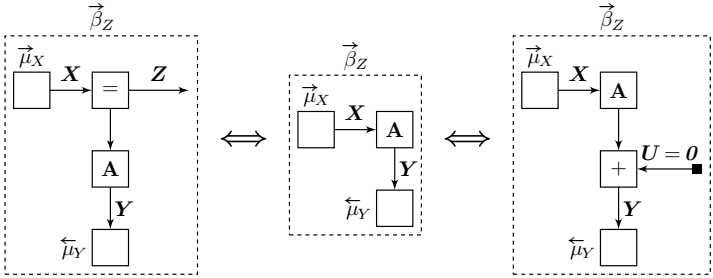


Figure C.5: Proof of (IV.1).

the rightmost graph as

$$\vec{\gamma}_Z = \int \vec{\mu}_U(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} \tag{C.31}$$

$$= \vec{\beta}_U \tag{C.32}$$

$$= \vec{\gamma}_U \sqrt{\frac{(2\pi)^n}{\det \vec{\mathbf{W}}_U}}\, e^{\vec{\boldsymbol{m}}_U^{\mathsf{T}} \vec{\mathbf{W}}_U \vec{\boldsymbol{m}}_U / 2} \tag{C.33}$$

$$= \vec{\gamma}_X \vec{\gamma}_Y \sqrt{\frac{(2\pi)^n}{\det \vec{\mathbf{W}}_U}}\, e^{\vec{\boldsymbol{m}}_U^{\mathsf{T}} \vec{\mathbf{W}}_U \vec{\boldsymbol{m}}_U / 2}\,. \tag{C.34}$$

In the third equality we have used the translation (6.16) between β and γ. In the last equality we have used the update rule (II.5) for γ through the equality node. Finally,

$$\vec{\mathbf{W}}_U = \vec{\mathbf{W}}_X + \vec{\mathbf{W}}_Y\,, \tag{C.35}$$

$$\vec{\mathbf{W}}_U \vec{\boldsymbol{m}}_U = \vec{\mathbf{W}}_X \vec{\boldsymbol{m}}_X - \vec{\mathbf{W}}_Y \vec{\boldsymbol{m}}_Y\,, \tag{C.36}$$

$$\tag{C.37}$$

due to the standard update rule for the equality node.

For the proof of Equation (III.5), an analogous sequence of equivalent graphs can be drawn. The only difference is the lack of the factor -1 because of the direction reversal.

Composite equality/multiplication block

For the proof of Equation (IV.1) consider the sequence of equivalent graphs shown in Figure C.5. Clearly, the closed box value of the leftmost graph is the definition of $\vec{\beta}_Z$. The subsequent graphs are obtained using neutral modifications (cf. Section C.1). Now we write the closed box value of the rightmost graph as

$$\begin{aligned}
\vec{\beta}_Z &= \overleftarrow{\mu}_U(\boldsymbol{0}) && \text{(C.38)}\\
&= \overleftarrow{\gamma}_U && \text{(C.39)}\\
&= \frac{\overleftarrow{\beta}_U}{\sqrt{(2\pi)^n \det \overleftarrow{\mathbf{V}}_U}}\, e^{-\overleftarrow{\boldsymbol{m}}_U^{\mathsf{T}} \overleftarrow{\mathbf{V}}_U^{\mathsf{T}} \overleftarrow{\boldsymbol{m}}_U/2} && \text{(C.40)}\\
&= \frac{\vec{\beta}_X\, \overleftarrow{\beta}_Y}{\sqrt{(2\pi)^n \det \overleftarrow{\mathbf{V}}_U}}\, e^{-\overleftarrow{\boldsymbol{m}}_U^{\mathsf{T}} \overleftarrow{\mathbf{V}}_U^{\mathsf{T}} \overleftarrow{\boldsymbol{m}}_U/2}\,. && \text{(C.41)}
\end{aligned}$$

In the third equality we have used the translation (6.16) between β and γ. In the last equality we have used the update rules (II.1) and (II.7). Finally,

$$\begin{aligned}
\overleftarrow{\mathbf{V}}_U &= \vec{\mathbf{V}}_U + \mathbf{A}\overleftarrow{\mathbf{V}}_X\mathbf{A}^{\mathsf{T}}\,, && \text{(C.42)}\\
\overleftarrow{\boldsymbol{m}}_U &= \mathbf{A}\vec{\boldsymbol{m}}_X + \overleftarrow{\boldsymbol{m}}_Y\,, && \text{(C.43)}
\end{aligned}$$

due to the standard update rules for the multiplication node and the addition node.

Composite addition/multiplication block

For the proof of Equation (IV.2) consider the sequence of equivalent graphs shown in Figure C.6. Clearly, the closed box value of the leftmost graph is the definition of $\vec{\gamma}_Z$. The subsequent graphs are obtained using neutral modifications (cf. Section C.1). We now write the closed box

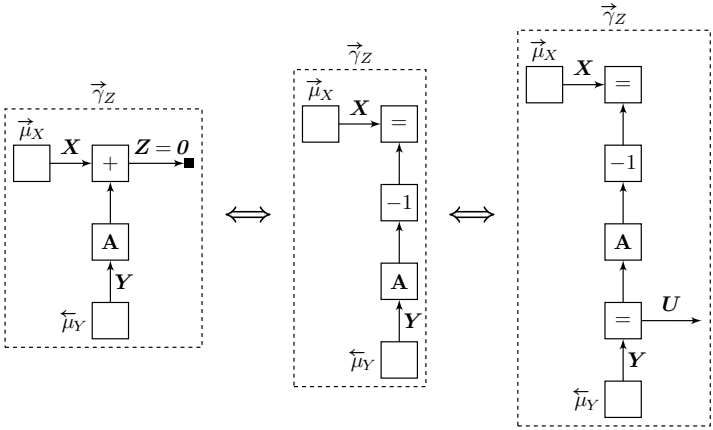


Figure C.6: Proof of (IV.3).

value of the rightmost graph as

$$\vec{\gamma}_Z = \int \vec{\mu}_U(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} \tag{C.44}$$

$$= \vec{\beta}_U \tag{C.45}$$

$$= \vec{\gamma}_U \sqrt{\frac{(2\pi)^n}{\det \vec{\mathbf{W}}_U}}\, e^{\vec{\boldsymbol{m}}_U^{\mathsf{T}} \vec{\mathbf{W}}_U \vec{\boldsymbol{m}}_U / 2} \tag{C.46}$$

$$= \vec{\gamma}_X\, \vec{\gamma}_Y \sqrt{\frac{(2\pi)^n}{\det \vec{\mathbf{W}}_U}}\, e^{\vec{\boldsymbol{m}}_U^{\mathsf{T}} \vec{\mathbf{W}}_U \vec{\boldsymbol{m}}_U / 2}\,. \tag{C.47}$$

In the third equality we have used the translation (6.16) between β and γ. In the last equality we have used the update rules (II.2) and (II.5). Finally,

$$\vec{\mathbf{W}}_U = \vec{\mathbf{W}}_Y + \mathbf{A}^{\mathsf{T}} \vec{\mathbf{W}}_X \mathbf{A}\,, \tag{C.48}$$

$$\vec{\mathbf{W}}_U \vec{\boldsymbol{m}}_U = \vec{\mathbf{W}}_Y \vec{\boldsymbol{m}}_Y - \mathbf{A}^{\mathsf{T}} \vec{\mathbf{W}}_X \vec{\boldsymbol{m}}_X\,, \tag{C.49}$$

due to the standard update rules for the multiplication node and the addition node.

For the proof of Equation (IV.3), an analogous sequence of equivalent graphs can be drawn. The only difference is the lack of the factor -1 because of the direction reversal.

Appendix D

Proofs for Chapter 7

D.1 Proof of Equations (7.8)–(7.11)

For Equation (7.8) we recall that the factor graph represents a joint PDF/PMF $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}, \ell)$ with some normalization constant ζ and we can write the conditional marginal as

$$p(\boldsymbol{\theta}, \ell | \tilde{\boldsymbol{y}}) = \frac{p(\tilde{\boldsymbol{y}}, \boldsymbol{\theta}, \ell)}{p(\tilde{\boldsymbol{y}})} = \frac{\mu_{\Theta_\ell}(\boldsymbol{\theta})}{\zeta} \cdot \frac{\zeta}{\sum_{j=0}^{K} \beta_{\Theta_\ell}} . \tag{D.1}$$

Equation (7.9) is a direct application of Theorem 6.2b.

For Equation (7.10) we note that, in the strictly conditional case, the factor graph represents a conditional PDF/PMF $p(\boldsymbol{x}, \boldsymbol{y}, \ell | \boldsymbol{\theta})$ with some normalization constant $\zeta^{\boldsymbol{\theta}}$ and we can write

$$p(\ell | \tilde{\boldsymbol{y}}, \boldsymbol{\theta}) = \frac{p(\tilde{\boldsymbol{y}}, \boldsymbol{\theta} | \ell)}{p(\tilde{\boldsymbol{y}} | \ell)} = \frac{\mu_{\Theta_\ell}(\boldsymbol{\theta})}{\zeta^{\boldsymbol{\theta}}} \cdot \frac{\zeta^{\boldsymbol{\theta}}}{\sum_{j=0}^{K} \mu_{\Theta_\ell}(\boldsymbol{\theta})} . \tag{D.2}$$

In the re-normalized case the factor graph is re-normalized by some normalization function $\zeta(\boldsymbol{\theta})$ and a similar expression as in (D.2) results with the sole difference that $\zeta^{\boldsymbol{\theta}}$ is exchanged by $\zeta(\boldsymbol{\theta})$.

Finally, for Equation (7.11) we note that, in the strictly conditional case, the factor graph represents a conditional PDF $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta} | \ell)$ with some normalization constant ζ^{ℓ} and we can write

$$p(\boldsymbol{\theta} | \tilde{\boldsymbol{y}}, \ell) = \frac{p(\tilde{\boldsymbol{y}}, \boldsymbol{\theta} | \ell)}{p(\tilde{\boldsymbol{y}} | \ell)} = \frac{\mu_{\Theta_\ell}(\boldsymbol{\theta})}{\zeta^{\ell}} \cdot \frac{\zeta^{\ell}}{\beta_{\Theta_\ell}} . \tag{D.3}$$

In the re-normalized case the factor graph is re-normalized by some normalization function $\zeta(\ell)$ and a similar expression as in (D.3) results with the sole difference that we ζ^{K} is exchanged by $\zeta(\ell)$.

D.2 Proof of Equations (7.99) and (7.102)

To prove (7.99) we choose $\boldsymbol{U}_\ell = \boldsymbol{X}_\ell^{\mathcal{A}}$. By using (6.11), the LLR of Equation (7.98) can be written as

$$\text{LLR}_\ell = \log \frac{\gamma_{X_\ell^{\mathcal{A}}}\, \gamma^{\circ}_{X_\ell|\mathcal{A}}}{\gamma^{\circ}_{X_\ell^{\mathcal{A}}}\, \gamma_{X_\ell|\mathcal{A}}} + \log \frac{\int \nu_{X_\ell^{\mathcal{A}}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \int \nu^{\circ}_{X_\ell|\mathcal{A}}(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'}{\int \nu^{\circ}_{X_\ell^{\mathcal{A}}}(\boldsymbol{x}'')\,\mathrm{d}\boldsymbol{x}'' \int \nu_{X_\ell|\mathcal{A}}(\boldsymbol{x}''')\,\mathrm{d}\boldsymbol{x}'''}\,. \tag{D.4}$$

We now concentrate on the first term and use (II.5) as

$$\frac{\gamma_{X_\ell^{\mathcal{A}}}\, \gamma^{\circ}_{X_\ell|\mathcal{A}}}{\gamma^{\circ}_{X_\ell^{\mathcal{A}}}\, \gamma_{X_\ell|\mathcal{A}}} = \frac{\overrightarrow{\gamma}_{X_\ell^{\mathcal{A}}}\, \overleftarrow{\gamma}_{X_\ell^{\mathcal{A}}}\, \overrightarrow{\gamma}^{\circ}_{X_\ell^{\mathcal{A}}}\, \overleftarrow{\gamma}^{\circ}_{X_\ell|\mathcal{A}}}{\overrightarrow{\gamma}^{\circ}_{X_\ell^{\mathcal{A}}}\, \overleftarrow{\gamma}^{\circ}_{X_\ell^{\mathcal{A}}}\, \overrightarrow{\gamma}_{X_\ell^{\mathcal{A}}}\, \overleftarrow{\gamma}_{X_\ell|\mathcal{A}}} = \frac{\overleftarrow{\gamma}_{X_\ell^{\mathcal{A}}}\, \overleftarrow{\gamma}^{\circ}_{X_\ell|\mathcal{A}}}{\overleftarrow{\gamma}^{\circ}_{X_\ell^{\mathcal{A}}}\, \overleftarrow{\gamma}_{X_\ell|\mathcal{A}}}\,, \tag{D.5}$$

where $\overrightarrow{\gamma}_{X_\ell^{\mathcal{A}}}$ and $\overrightarrow{\gamma}^{\circ}_{X_\ell^{\mathcal{A}}}$ have canceled. A similar argument can be made to show that $\overrightarrow{\beta}_{X_\ell^{\mathcal{A}}}$ and $\overrightarrow{\beta}^{\circ}_{X_\ell^{\mathcal{A}}}$ cancel as well. Equation (7.99) is established by noting that

$$\begin{aligned}
\overleftarrow{\gamma}_{X_\ell^{\mathcal{A}}} &= \overleftarrow{\mu}_{X_\ell^{\mathcal{A}}}(\boldsymbol{0}) && \text{(D.6)}\\
&= \int \overleftarrow{\mu}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x})\, g(\boldsymbol{0}, \boldsymbol{x}, \ell)\,\mathrm{d}\boldsymbol{x} && \text{(D.7)}\\
&= \overleftarrow{\gamma}_{X_\ell^{\mathcal{B}}} \int \overleftarrow{\nu}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x})\, g(\boldsymbol{0}, \boldsymbol{x}, \ell)\,\mathrm{d}\boldsymbol{x}\,, && \text{(D.8)}
\end{aligned}$$

and similarly

$$\begin{aligned}
\overleftarrow{\gamma}^{\circ}_{X_\ell^{\mathcal{A}}} &= \overleftarrow{\mu}^{\circ}_{X_\ell^{\mathcal{A}}}(\boldsymbol{0}) && \text{(D.9)}\\
&= \int \overleftarrow{\mu}^{\circ}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x}')\, g(\boldsymbol{0}, \boldsymbol{x}', \ell)\,\mathrm{d}\boldsymbol{x} && \text{(D.10)}\\
&= \overleftarrow{\gamma}^{\circ}_{X_\ell^{\mathcal{B}}} \int \overleftarrow{\nu}^{\circ}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x}')\, g(\boldsymbol{0}, \boldsymbol{x}', \ell)\,\mathrm{d}\boldsymbol{x}'\,. && \text{(D.11)}
\end{aligned}$$

To prove (7.102) we also choose $\boldsymbol{U}_\ell = \boldsymbol{X}_\ell^{\mathcal{A}}$. By using (6.11), the log-likelihood under hypothesis $\mathcal{H}_\ell$ can be written as

$$\log p(\tilde{\boldsymbol{y}}|\mathcal{H}_\ell) = \log \frac{\gamma_{\boldsymbol{X}_\ell^{\mathcal{A}}}}{\gamma^{\circ}_{\boldsymbol{X}_\ell^{\mathcal{A}}}} + \log \frac{\int \nu_{X_\ell^{\mathcal{A}}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\int \nu^{\circ}_{X_\ell^{\mathcal{A}}}(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'}\,. \tag{D.12}$$

Equation (7.102) is established by using (D.11), plugging into (7.101) and noting that all scale factors cancel.

D.3 Proof for Recursive Computation of $\ln \overleftarrow{\gamma}_k$ (Equations (7.104)–(7.108)) and $\ln \overleftarrow{\gamma}'_k$ (Equation (7.115))

First note for the initial value $\ln \overleftarrow{\gamma}_{K+1}$ that we have

$$\overleftarrow{\gamma}_{K+1} = \frac{\overleftarrow{\gamma}_{X^{\mathcal{B}}_{K+1}} \overleftarrow{\gamma}^{\circ}_{X_{K+1}|\mathcal{A}}}{\overleftarrow{\gamma}^{\circ}_{X^{\mathcal{B}}_{K+1}} \overleftarrow{\gamma}_{X_{K+1}|\mathcal{A}}} = \frac{f_{K+1}(\boldsymbol{0})}{f_{K+1}(\boldsymbol{0})} = 1\,. \tag{D.13}$$

To prove the recursion (7.105), we propagate the γ-type scale factor from edge $\boldsymbol{X}^{\mathcal{B}}_k$ to edge $\boldsymbol{X}^{\mathcal{B}}_{k-1}$ in the factor graph in Figure 7.15 for $\mathcal{M} = \mathcal{B}$. Using the update rules (II.2) and (IV.3), we obtain

$$\overleftarrow{\gamma}_{X^{\mathcal{B}}_{k-1}} = \overleftarrow{\gamma}_{X'^{\mathcal{B}}_k} \overrightarrow{\gamma}_{U^{\mathcal{B}}_k} \sqrt{\frac{(2\pi)^{n_U}}{\det \mathbf{W}_{\mathcal{B}}}}\, e^{\boldsymbol{m}^{\mathsf{T}}_{\mathcal{B}} \mathbf{W}_{\mathcal{B}} \boldsymbol{m}_{\mathcal{B}}/2}\,, \tag{D.14}$$

with the definitions given in (7.106) and (7.107), and where the edge $\boldsymbol{U}^{\mathcal{B}}_k$ is the input noise. Further, we have

$$\overleftarrow{\gamma}_{X'^{\mathcal{B}}_k} = \overleftarrow{\gamma}_{X^{\mathcal{B}}_k} \sqrt{\frac{\det \mathbf{W}^{\mathcal{B}}_Z}{(2\pi)^{n_Y}}}\, e^{-\tilde{\boldsymbol{y}}^{\mathsf{T}}_k \mathbf{W}^{\mathcal{B}}_Z \tilde{\boldsymbol{y}}_k/2}\,. \tag{D.15}$$

For the quantities without plugged-in observations we get

$$\overleftarrow{\gamma}^{\circ}_{X^{\mathcal{B}}_{k-1}} = \overleftarrow{\gamma}^{\circ}_{X^{\mathcal{B}}_k} \overrightarrow{\gamma}^{\circ}_{U^{\mathcal{B}}_k} \sqrt{\frac{(2\pi)^{n_U}}{\det \mathbf{W}^{\circ}_{\mathcal{B}}}}\,, \tag{D.16}$$

with the definition given in (7.108). Equivalent equations apply for $\overleftarrow{\gamma}_{X_k|\mathcal{A}}$. Finally, we observe that in the ratio (7.103), $\overrightarrow{\gamma}^{\circ}_{U^{\mathcal{B}}_k} = \overrightarrow{\gamma}_{U^{\mathcal{B}}_k}$ and $\overrightarrow{\gamma}^{\circ}_{U_k|\mathcal{A}} = \overrightarrow{\gamma}_{U_k|\mathcal{A}}$ cancel as well as all powers of 2π.

For the recursion of $\overleftarrow{\gamma}'_k$ in (7.115) we note that the terms $\overrightarrow{\gamma}^{\circ}_{U^{\mathcal{B}}_k}$ and $\overrightarrow{\gamma}^{\circ}_{U_k|\mathcal{A}}$ are missing and can therefore not be cancelled with $\overrightarrow{\gamma}_{U^{\mathcal{B}}_k}$ and $\overrightarrow{\gamma}_{U_k|\mathcal{A}}$. Therefore, we have to expand the term $\overrightarrow{\gamma}_{U^{\mathcal{M}}_k}$ in (D.14) as this case as

$$\overrightarrow{\gamma}_{U^{\mathcal{B}}_k} = \sqrt{\frac{\det \mathbf{W}^{\mathcal{B}}_U}{(2\pi)^{n_U}}}\,. \tag{D.17}$$

In the final ratio of Equation (7.115) all powers of 2π cancel again.

D.4 Proof of LLRs for Pulse Position Estimation

In the following we prove Equations (7.125)–(7.128). For Variant (a), for which the pulse shape is not known, we start from the LLR in Equation (7.112). The first term of (7.112) can be written as

$$\int \nu_{X_\ell^{\mathcal{A}}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \sqrt{\frac{(2\pi)^{n_X}}{\det \mathbf{W}_{X_\ell^{\mathcal{A}}}}}\, e^{\boldsymbol{m}_{X_\ell^{\mathcal{A}}}^{\mathsf{T}} \mathbf{W}_{X_\ell^{\mathcal{A}}} \boldsymbol{m}_{X_\ell^{\mathcal{A}}}/2}\,, \tag{D.18}$$

where we have used (6.16), and where n_X is the dimensionality of $\boldsymbol{X}_\ell^{\mathcal{A}}$. For the second term we note that

$$\int \overleftarrow{\mu}_{X_\ell^{\mathcal{B}}}(\boldsymbol{x})\, g(\boldsymbol{0}, \boldsymbol{x}, \ell)\,\mathrm{d}\boldsymbol{x} = \overleftarrow{\gamma}_{X_\ell^{\mathcal{A}}} \tag{D.19}$$

$$= \overleftarrow{\gamma}_{X_\ell^{\mathcal{B}}} \sqrt{\frac{(2\pi)^{n_S}}{\det \mathbf{W}}}\, e^{\boldsymbol{m}^{\mathsf{T}} \mathbf{W} \boldsymbol{m}}\,, \tag{D.20}$$

where $\mathbf{W}$ and $\mathbf{W}\boldsymbol{m}$ are defined as in (7.127) and (7.128) respectively and where n_S is the dimensionality of $\boldsymbol{S}_{\mathcal{B}}$ in Figure 7.10b. The LLRs of (7.125) follows directly.

For Variant (b), for which the pulse shape is known we write

$$p(\tilde{\boldsymbol{y}}|\mathcal{H}_\ell) = \overrightarrow{\mu}_{X_\ell^{\mathcal{A}}}(\tilde{\boldsymbol{x}}_{\mathcal{A}})\, \overleftarrow{\mu}_{X_\ell^{\mathcal{B}}}(\tilde{\boldsymbol{x}}_{\mathcal{B}}) \tag{D.21}$$

$$\begin{aligned} &= \overrightarrow{\gamma}_{X_\ell^{\mathcal{A}}}\, e^{-\tilde{\boldsymbol{x}}_{\mathcal{A}}^{\mathsf{T}} \overrightarrow{\mathbf{W}}_{X_\ell^{\mathcal{A}}} \tilde{\boldsymbol{x}}_{\mathcal{A}}/2 + \tilde{\boldsymbol{x}}_{\mathcal{A}}^{\mathsf{T}} \overrightarrow{\mathbf{W}}_{X_\ell^{\mathcal{A}}} \tilde{\boldsymbol{x}}_{\mathcal{A}}} \\ &\quad \cdot \overleftarrow{\gamma}_{X_\ell^{\mathcal{B}}}\, e^{-\tilde{\boldsymbol{x}}_{\mathcal{B}}^{\mathsf{T}} \overrightarrow{\mathbf{W}}_{X_\ell^{\mathcal{B}}} \tilde{\boldsymbol{x}}_{\mathcal{B}}/2 + \tilde{\boldsymbol{x}}_{\mathcal{B}}^{\mathsf{T}} \overrightarrow{\mathbf{W}}_{X_\ell^{\mathcal{B}}} \tilde{\boldsymbol{x}}_{\mathcal{B}}}\,, \end{aligned} \tag{D.22}$$

where we have used (6.13). The likelihood under the noise-only hypothesis can be written as

$$p(\tilde{\boldsymbol{y}}|\mathcal{H}_{\mathcal{N}}) = \overrightarrow{\mu}_{X_\ell^{\mathcal{A}}}(\boldsymbol{0})\, \overleftarrow{\mu}_{X_\ell^{\mathcal{B}}}(\boldsymbol{0}) \tag{D.23}$$

$$= \overrightarrow{\gamma}_{X_\ell^{\mathcal{A}}}\, \overleftarrow{\gamma}_{X_\ell^{\mathcal{A}}}\,. \tag{D.24}$$

The LLRs of (7.126) follows directly.

Appendix E

On Factor Graphs and Linear Algebra

The connection between factor graphs and matrix manipulations in linear algebra dates back to the 1950s (cf. references in [76]). This appendix does not present something new but instead translates known results into our factor graph notation. [1, 33, 68, 76, 83]

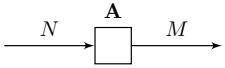
E.1 Definitions

In contrast to most texts on linear algebra (and to the rest of this thesis), we use exclusively zero-based indexing in this appendix, i.e., $[\mathbf{A}]_{0,0}$ denotes the first element in the first column in a matrix $\mathbf{A}$.

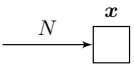
In order to define factor graph nodes for matrices and vectors, a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ is viewed as a mapping

$$\mathbf{A} \colon \{0, \dots, n-1\} \times \{0, \dots, m-1\} \to \mathbb{R} \colon (N, M) \to [\mathbf{A}]_{N,M} \,. \quad \text{(E.1)}$$

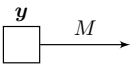
Similarly a column vector $\boldsymbol{x} \in \mathbb{R}^n$ and a row vector $\boldsymbol{y} \in \mathbb{R}^m$ are viewed



(a) Matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$



(b) Column vector $\boldsymbol{x} \in \mathbb{R}^n$



(c) Row vector $\boldsymbol{y} \in \mathbb{R}^m$

Figure E.1: Definitions of matrices and vectors as factors.

Figure E.2: Matrix transposition in factor graph notation.

as mappings

$$\boldsymbol{x}\colon \{0,\dots,n-1\} \to \mathbb{R}\colon N \to [\boldsymbol{x}]_N\,, \tag{E.2}$$

$$\boldsymbol{y}\colon \{0,\dots,m-1\} \to \mathbb{R}\colon M \to [\boldsymbol{y}]_M\,. \tag{E.3}$$

Figure E.1 shows the corresponding factor graph nodes.

The edges now represent column or row indices. The range of an index is implicitly defined by the size of the corresponding matrix (or vector). We choose the edge-direction such that for each node an incoming edge denotes a row index and an outgoing edge denotes a column index. With this definition we are able to represent transposition $(\cdot^{\mathsf{T}})$ by edge direction reversal as shown in Figure E.2.

Note that the usage of directed edges may not lead to the most compact notation. Specifically, it may be more convenient to label the ports of the nodes instead. Here, we stick with directed edges to be consistent with the rest of the thesis.

In Figure E.3 we list a range of matrix-vector operations which we will use in proofs in later subsections of this appendix. The factor graphs (a)–(j) in Figure E.3 are proved straightforwardly by writing the closed-box function. For the Kronecker product in (k) we can write

$$[\mathbf{A}]_{N,M}\,[\mathbf{D}]_{K,L} = [\mathbf{A} \otimes \mathbf{D}]_{Nk+K,M\ell+L}\,. \tag{E.4}$$

To prove the matrix to vector operations in (l) and (m) we can write

$$[\mathbf{A}]_{N,M} = [\mathrm{cvect}\,\mathbf{A}]_{N+Mn}\,, \tag{E.5}$$

$$[\mathbf{A}]_{N,M} = [\mathrm{rvect}\,\mathbf{A}]_{M+Nm}\,. \tag{E.6}$$

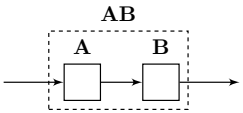
Similarly, the vector to matrix operation in (n) and (o) can be shown as

$$[\boldsymbol{c}]_{N+Mn} = [\mathbf{A}]_{N,M}\,, \tag{E.7}$$

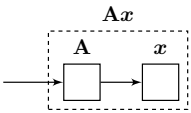
$$[\boldsymbol{r}]_{M+Nm} = [\mathbf{A}]_{N,M}\,. \tag{E.8}$$

Note that in the notation of the graphs (k)–(o) the range of all variables is implicitly defined by the admissible values of indices in (E.4)–(E.8).

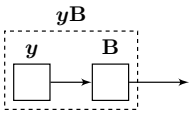
Specifically, one of the variables connected to the summation node is a nonnegative multiple of an integer.



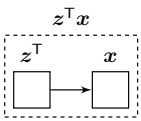
(a) Matrix-matrix product



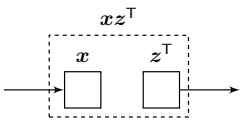
(b) Matrix-vector product



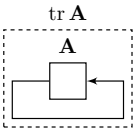
(c) Vector-matrix product



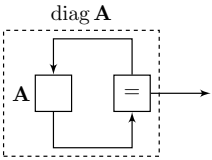
(d) Inner product



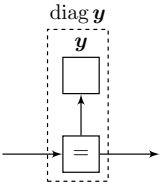
(e) Outer product



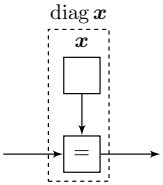
(f) Matrix trace



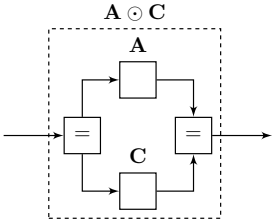
(g) Diagonal operator for a matrix



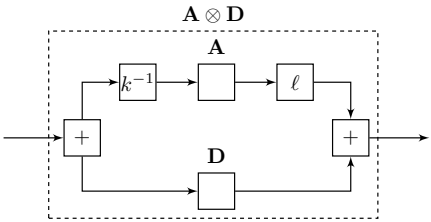
(h) Diagonal operator for a column vector



(i) Diagonal operator for a row vector

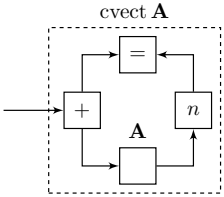


(j) Hadamard product

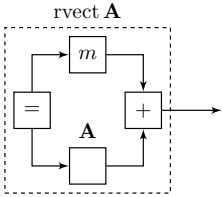


(k) Kronecker product

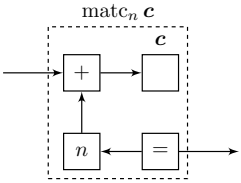
Figure E.3: Vector matrix operations in factor graph notation. Definitions for (a)–(k): $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^n$ (column vectors); $\boldsymbol{y} \in \mathbb{R}^m$ (row vector); $\mathbf{A}, \mathbf{C} \in \mathbb{R}^{n \times m}$; $\mathbf{B} \in \mathbb{R}^{m \times k}$; $\mathbf{D} \in \mathbb{R}^{k \times \ell}$.



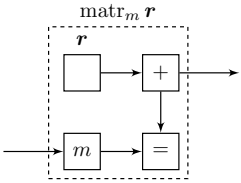
(l) Matrix to column vector operation



(m) Matrix to row vector operation



(n) Column vector to matrix operation



(o) Row vector to matrix operation

Figure E.3: (Continued) Vector matrix operations in factor graph notation. Definitions for (l)–(o): $\mathbf{A} \in \mathbb{R}^{n \times m}$; $\boldsymbol{c} \in \mathbb{R}^{nm}$ (column vector); $\boldsymbol{r} \in \mathbb{R}^{nm}$ (row vector).

E.2 Example: A Standard Expression in Linear Algebra

Here we prove the following identity

$$\boldsymbol{x}^{\mathsf{T}}(\mathbf{A} \odot \mathbf{B})\, \boldsymbol{y} = \operatorname{tr}\big(\operatorname{diag}(\boldsymbol{x})\, \mathbf{A}\, \operatorname{diag}(\boldsymbol{y})\, \mathbf{B}^{\mathsf{T}}\big)\,, \tag{E.9}$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ are column vectors and $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ are matrices. Figure E.4a shows the left-hand side of (E.9). In Figure E.4b the only change is the reversal of two edges and the corresponding transposition of $\mathbf{B}$. Reinterpreting this figure in terms of the building blocks (Figure E.3) yields the right-hand side of the identity (E.9).

E.3 Vectorization of a Lyapunov equation

We consider the equation

$$\mathbf{X} = \mathbf{A}\mathbf{X}\mathbf{B} + \mathbf{Q}\,, \tag{E.10}$$

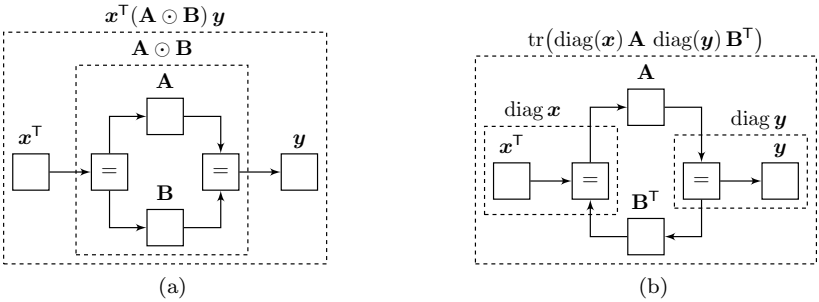


Figure E.4: Proof of (E.9).

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{Q}$ are given and $\mathbf{X}$ must be found. Note that this is a general form of the Lyapunov equation (3.16). This equation can be solved by vectorization as follows:

$$\operatorname{cvect} \mathbf{X} = \operatorname{cvect}(\mathbf{AXB}) + \operatorname{cvect} \mathbf{Q} \tag{E.11}$$

$$\operatorname{cvect} \mathbf{X} = \left(\mathbf{B}^\mathsf{T} \otimes \mathbf{A}\right) \operatorname{cvect} \mathbf{X} + \operatorname{cvect} \mathbf{Q} \tag{E.12}$$

$$\left(\mathbf{I} - \mathbf{B}^\mathsf{T} \otimes \mathbf{A}\right) \operatorname{cvect} \mathbf{X} = \operatorname{cvect} \mathbf{Q}\,. \tag{E.13}$$

In the second equality we have used the well-known identity

$$\operatorname{cvect}(\mathbf{AXB}) = \left(\mathbf{B}^\mathsf{T} \otimes \mathbf{A}\right) \operatorname{cvect} \mathbf{X}\,. \tag{E.14}$$

This identity is proved in Figure E.5 using factor graph notation. In this figure we start with the left-hand side of (E.14) and make successive modifications that do not change the closed box function. Finally we make use of various expressions from Figure E.3 to reinterpret the resulting graph.

The three graphs (a)–(c) are equivalent because we use neutral modifications as in Section C.1. For the equivalence of the graphs (c) and (d) note that the edges I, J, I', and J' (as labelled in the graph (d)) are subjected to the local constraint $I + J = I' + J'$ induced by the two addition nodes. Because I, I' are nonnegative multiples of m and $J, J' < m$, the only way to satisfy this constraint is by enforcing $I = I'$ and $J = J'$, which corresponds to the local constraint in the graph (c).

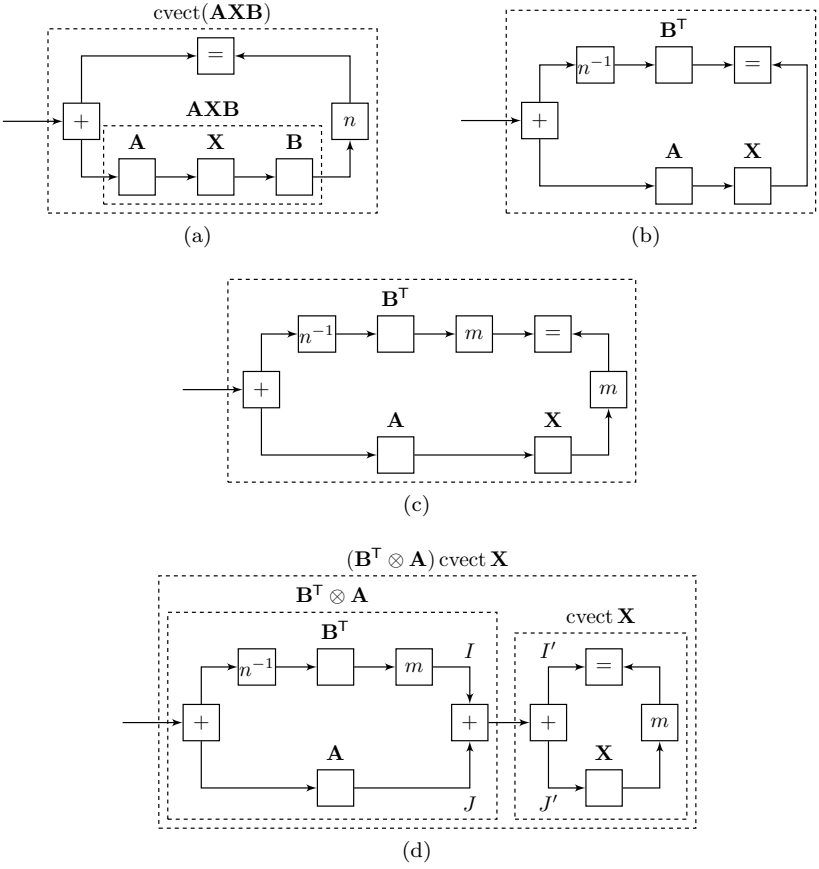


Figure E.5: Proof of (E.14) for $\mathbf{A} \in \mathbb{R}^{n \times m}$, $\mathbf{X} \in \mathbb{R}^{m \times k}$, and $\mathbf{B} \in \mathbb{R}^{k \times \ell}$.

Bibliography

[1] A. Al-Bashabsheh and Y. Mao, "Normal factor graphs and holographic transformations," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 752–763, 2011.

[2] D. Arnold, H.-A. Loeliger, P. Vontobel, A. Kavcic, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3498 –3508, aug. 2006.

[3] A. Azevedo-Filho and R. Shachter, "Laplace's method approximations for probabilistic inference in belief networks with continuous variables," in *Proceedings 10th Conference on Uncertainty in Artificial Intelligence (UAI)*, 1994, pp. 28–36.

[4] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[5] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2743–2760, Oct. 1998.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proceedings IEEE International Conference on Communications (ICC)*, vol. 2, 1993, pp. 1064–1070.

[7] C. M. Bishop, "The relevance vector machine," in *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2000, pp. 652–658.

[8] ——, *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] J. Biveroni, "On A/D converters with low-precision analog circuits and digital post-correction," Ph.D. dissertation, No. 20629, ETH Zurich, 2012.

[10] L. Bolliger, H.-A. Loeliger, and C. Vogel, "Simulation, MMSE estimation, and interpolation of sampled continuous-time signals using factor graphs," in *Proceedings Information Theory and Applications Workshop (ITA)*, UCSD, La Jolla, CA, USA, Jan.31 – Feb.2 2010.

[11] L. Bolliger, "Digital estimation of continuous-time signals using factor graphs," Ph.D. dissertation, No. 20123, ETH Zurich, 2012.

[12] L. Bolliger, H.-A. Loeliger, and C. Vogel, "LMMSE estimation and interpolation of continuous-time signals from discrete-time samples using factor graphs," 2013, arXiv:1301.4793.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, ser. Foundations and Trends in Machine Learning. Now Publishers, 2010, vol. 3, no. 1.

[14] J. Bussnang and D. Middleton, "Optimum sequential detection of signals in noise," *IRE Transactions on Information Theory*, vol. 1, no. 3, pp. 5–18, 1955.

[15] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe, "Bayesian inference for linear dynamic models with Dirichlet process mixtures," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 71–84, 2008.

[16] D. Childers, D. Skinner, and R. Kemerait, "The cepstrum: A guide to processing," *Proceedings of the IEEE*, vol. 65, no. 10, pp. 1428–1443, Oct. 1977.

[17] M. G. Christensen and A. Jakobsson, *Multi-Pitch Estimation*. Morgan & Claypool, 2009.

[18] C. K. Chui and G. Chen, *Kalman Filtering with Real-Time Applications*, 4th ed. Springer, 2009.

[19] J. Dauwels, S. Korl, and H.-A. Loeliger, "Expectation maximization as message passing," in *Proceedings IEEE International Symposion on Information Theory (ISIT)*, 2005, pp. 583–586.

[20] ——, "Steepest descent on factor graphs," in *Proceedings IEEE Information Theory Workshop (ITW)*, Rotorua, New Zealand, Aug. 28–Sep. 1 2005.

[21] J. Dauwels, S. Korl, and H. Loeliger, "Particle methods as message passing," in *Proceedings IEEE International Symposion on Information Theory (ISIT)*. IEEE, 2006, pp. 2052–2056.

[22] J. Dauwels and H.-A. Loeliger, "Computation of information rates by particle methods," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 406–409, 2008.

[23] J. Dauwels, "On graphical models for communications and machine learning: algorithms, bounds, and analog implementation," Ph.D. dissertation, No. 16365, ETH Zurich, 2006.

[24] ——, "On convergence properties of message-passing estimation algorithms," in *Proceedings IEEE International Symposion on Information Theory (ISIT)*, 2007, pp. 2561–2565.

[25] ——, "On variational message passing on factor graphs," in *Proceedings IEEE International Symposion on Information Theory (ISIT)*, 2007, pp. 2546–2550.

[26] J. Dauwels, A. Eckford, S. Korl, and H.-A. Loeliger, "Expectation maximization as message passing—part i: Principles and Gaussian messages," 2009, arXiv:0910.2832.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[28] M. V. R. S. Devarakonda and H.-A. Loeliger, "Joint synchronization and demodulation by forward filtering," in *Proceedings IEEE International Symposion on Information Theory (ISIT)*, Cambridge, MA, USA, Jul. 1–6 2012.

[29] M. V. R. S. Devarakonda, "Joint matched filitering, decoding, and timing synchronization," Ph.D. dissertation, No. 19882, ETH Zurich, 2012.

[30] P. Dragotti and M. Vetterli, "Wavelet footprints: theory, algorithms, and applications," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1306–1323, May 2003.

[31] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*. Oxford University Press, 2001.

[32] M.-A. El Mechat, "Statistical range estimation for optical time-of-flight 3d imaging," Ph.D. dissertation, No. 18989, ETH Zurich, 2010.

[33] G. D. Forney and P. O. Vontobel, "Partition functions of normal factor graphs," in *Proceedings Information Theory and Applications Workshop (ITA)*, UCSD, La Jolla, CA, USA, Feb. 2011.

[34] J. Forney, G. D., "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[35] ——, "Codes on graphs: normal realizations," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 520–548, 2001.

[36] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Non-parametric Bayesian learning of switching linear dynamical systems," in *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2009.

[37] R. Gallager, "Low-density parity-check codes," Ph.D. dissertation, Massachusetts Institute of Technology (MIT), 1963.

[38] W. A. Gardner, A. Napolitano, and L. Paura, "Cyclostationarity: Half a century of research," *Signal Processing*, vol. 86, no. 4, pp. 639–697, 2006.

[39] O. Goldshtein, H. Messer, and A. Zinevich, "Rain rate estimation using measurements from commercial telecommunications links," *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1616–1625, 2009.

[40] R. M. Gray and L. D. Davisson, *An Introduction to Statistical Signal Processing*. Cambridge University Press, 2004.

[41] S. L. Hahn, *Hilbert Transforms in Signal Processing*. Artech House, 1996.

[42] J. Hawkins, "Why can't a computer be more like a brain?" *IEEE Spectrum*, vol. 44, no. 4, pp. 21 –26, Apr. 2007.

[43] J. Hawkins and G. Dileep, "Hierarchical temporal memory: concepts, theory, and terminology," *Whitepaper, Numenta Inc*, 2006.

[44] E. Horita, K. Sumiya, H. Urakami, and S. Mitsuishi, "A leaky RLS algorithm: its optimality and implementation," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 2924–2936, 2004.

[45] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.

[46] J. Hu, "On Gaussian approximations in message passing algorithms with application to equalization," Ph.D. dissertation, No. 17804, ETH Zurich, 2008.

[47] E. Jacobsen and R. Lyons, "The sliding DFT," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 74–80, 2003.

[48] ——, "An update to the sliding DFT," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 110–111, 2004.

[49] 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. 3–233, 1999.

[50] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.

[51] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, pp. 34–45, Mar. 1960.

[52] E. Karseras, K. Leung, and W. Dai, "Tracking dynamic sparse signals using hierarchical Bayesian Kalman filters," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[53] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993, vol. 1.

[54] ——, *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice Hall, 1998, vol. 2.

[55] K. Kim and G. Shevlyakov, "Why Gaussianity?" *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 102–113, 2008.

[56] D. Koller, N. Friedman, L. Getoor, and B. Taskar, "Graphical models in a nutshell," in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2007.

[57] D. Koller and N. Friedman, *Probabilistic Graphical Models*. MIT Press, 2009.

[58] S. Korl, H. A. Loeliger, and A. G. Lindgren, "AR model parameter estimation: from factor graphs to algorithms," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, 2004.

[59] S. Korl, "A factor graph approach to signal modelling, system identification and filtering," Ph.D. dissertation, No. 16170, ETH Zurich, 2005.

[60] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[61] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*, 3rd ed. Springer New York, Mar. 2006.

[62] H. Leijnse, R. Uijlenhoet, and J. N. M. Stricker, "Hydrometeorological application of a microwave link: 2. precipitation," *Water Resources Research*, vol. 43, no. 4, p. W04417, Apr. 2007.

[63] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, 2004.

[64] H.-A. Loeliger, L. Bolliger, C. Reller, and S. Korl, "Localizing, forgetting, and likelihood filtering in state-space models," in *Proceedings Information Theory and Applications Workshop (ITA)*, 2009, pp. 184–186.

[65] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[66] H.-A. Loeliger, "Least squares and Kalman filtering on Forney graphs," in *Codes, Graphs, and Systems*, R. E. Blahut and R. Koetter, Eds. Kluwer, 2002, pp. 113–135, festschrift in honour of David Forney on the occasion of his 60th birthday.

[67] H.-A. Loeliger and C. Reller, "Signal processing with factor graphs: beamforming and Hilbert transform," in *Proceedings Information Theory and Applications Workshop (ITA)*, San Diego, CA, Feb.10–15 2013.

[68] H.-A. Loeliger and P. O. Vontobel, "A factor-graph representation of probabilities in quantum mechanics," 2012, arXiv:1201.5422.

[69] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed. Academic Press, 2009.

[70] S. Maranò, D. Fäh, C. Reller, and H.-A. Loeliger, "Maximum likelihood parameter estimation for surface waves: Application to ambient vibrations," in *4th IASPEI / IAEE International Symposium: Effects of Surface Geology on Strong Ground Motion (ESG)*, UCSB, Santa Barbara, CA, Aug. 23–26 2011.

[71] S. Maranò, C. Reller, D. Fäh, and H.-A. Loeliger, "Seismic waves estimation and wave field decomposition with factor graphs," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011.

[72] S. Maranò, C. Reller, H.-A. Loeliger, and D. Fäh, "Seismic waves estimation and wave field decomposition: Application to ambient vibrations," *Geophysical Journal International*, vol. 191, no. 1, pp. 175–188, 2012.

[73] L. R. Medsker and L. C. Jain, Eds., *Recurrent Neural Networks: Design and Applications*. CRC Press, 2001.

[74] A. Mertins, *Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications*. Wiley Publishing, 1999.

[75] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proceedings 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, San Francisco, CA, USA, 2001, pp. 362–369.

[76] S. Morse and E. Peterson, "Trace diagrams, matrix minors, and determinant identities," 2009, arXiv:0903.1373.

[77] R. E. Neapolitan, *Learning Bayesian Networks*. Prentice Hall, 2004.

[78] R. B. Nelsen, *An Introduction to Copulas*, 2nd ed. Springer, 2006.

[79] R. Olsen, D. Rogers, and D. Hodge, "The aR^b relation in the calculation of rain attenuation," *IEEE Transactions on Antennas and Propagation*, vol. 26, no. 2, pp. 318–329, 1978.

[80] P. Orbanz and Y. W. Teh, "Bayesian nonparametric models," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Springer US, 2010, pp. 81–89.

[81] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. McGraw-Hill, 1991.

[82] L. Perko, *Differential Equations and Dynamical Systems*. Springer, 1991.

[83] E. Peterson, "Unshackling linear algebra from linear notation," Oct. 2009, arXiv:0910.1362.

[84] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Springer, 1988.

[85] H. V. Poor and O. Hadjiliadis, *Quickest Detection*. Cambridge University Press, 2009.

[86] C. Reinsch, "Smoothing by spline functions," *Numerische Mathematik*, vol. 10, pp. 177–183, 1967, 10.1007/BF02162161.

[87] C. Reller, H.-A. Loeliger, and V. R. S. Devarakonda, Murthy, "Glue factors, likelihood computation, and filtering in state space models," in *Proceedings 50th Allerton Conference on Communication, Control, and Computing*, Monticello, Illinois, USA, Oct. 1–5 2012.

[88] C. Reller, H.-A. Loeliger, and S. Maranò, "Multi-sensor estimation and detection of phase-locked sinusoids," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011.

[89] C. Reller, H.-A. Loeliger, and J. P. Marín Díaz, "A model for quasi-periodic signals with application to rain estimation from microwave link gain," in *Proceedings 19th European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, Aug. 29–Sep. 2 2011.

[90] H. Robbins, "The empirical Bayes approach to statistical decision problems," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 1–20, 1964.

[91] R. Rojas, *Neural Networks: A Systematic Introduction*. Springer, 1996.

[92] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[93] S. Roweis and Z. Ghahramani, "A unifying review of linear Gaussian models," *Neural Computation*, vol. 11, no. 2, pp. 305–345, 1999.

[94] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 18–60, 1994.

[95] M. Schleiss and A. Berne, "Identification of dry and rainy periods using telecommunication microwave links," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 3, pp. 611–615, 2010.

[96] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[97] E. D. Sontag, *Mathematical Control Theory*, 2nd ed. Springer, 1998.

[98] F. Spitzer, "Markov random fields and Gibbs ensembles," *The American Mathematical Monthly*, vol. 78, no. 2, pp. 142–154, 1971.

[99] P. Stoica and Y. Selen, "Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: a refresher," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 112–114, 2004.

[100] ——, "Model-order selection: a review of information criterion rules," *IEEE Signal Processing Magazine*, vol. 21, no. 4, pp. 36–47, 2004.

[101] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533 – 547, Sep. 1981.

[102] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[103] E. Terhardt, "Fourier transformation of time signals: Conceptual revision," *Acustica*, vol. 57, pp. 242–265, 1985.

[104] M. E. Tipping, "The relevance vector machine," Microsoft Research, Tech. Rep., 2000.

[105] I. Trajkovic, "Modelling and filtering almost periodic signals by time-varying Fourier series with application to near-infrared spectroscopy," Ph.D. dissertation, No. 19420, ETH Zurich, 2010.

[106] G. Upton, A. Holt, R. Cummings, A. Rahimi, and J. Goddard, "Microwave links: The future for urban rainfall measurement?" *Atmospheric Research*, vol. 77, no. 1–4, pp. 300–312, 2005.

[107] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. Wiley Publishing, 2001, vol. 1.

[108] P. O. . Vontobel, "Counting graph covers: A combinatorial characterization of the Bethe entropy function," 2012, arXiv:1012.0065.

[109] P. O. Vontobel, "The Bethe permanent of a non-negative matrix," 2011, arXiv:1107.4196.

[110] P. O. Vontobel, D. Lippuner, and H.-A. Loeliger, "Kalman filtering, factor graphs and electrical networks," ETH Zurich, Tech. Rep., 2002.

[111] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference*, ser. Foundations and Trends in Machine Learning. Now Publishers, 2008, vol. 1, no. 1–2.

[112] A. Wald, "Sequential tests of statistical hypotheses," *The Annals of Mathematical Statistics*, vol. 16, no. 2, pp. 117–186, 1945.

[113] D. Weiss, B. Sapp, and B. Taskar, "Sidestepping intractable inference with structured ensemble cascades," *Advances in Neural Information Processing Systems*, vol. 23, pp. 2415–2423, 2010.

[114] N. Wiberg, H.-A. Loeliger, and R. Kotter, "Codes and iterative decoding on general graphs," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 513–525, 1995.

[115] J. C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Transactions on Automatic Control*, vol. 36, no. 3, pp. 259–294, 1991.

[116] J. Yedidia, W. Freeman, and Y. Weiss, "Generalized belief propagation," *Advances in Neural Information Processing Systems*, vol. 13, pp. 689–695, 2001.

Series in Signal and Information Processing

edited by Hans-Andrea Loeliger

Vol. 1: Hanspeter Schmid, **Single-Amplifier Biquadratic MOSFET-C Filters.** ISBN 3-89649-616-6

Vol. 2: Felix Lustenberger, **On the Design of Analog VLSI Iterative Decoders.** ISBN 3-89649-622-0

Vol. 3: Peter Theodor Wellig, **Zerlegung von Langzeit-Elektromyogrammen zur Prävention von arbeitsbedingten Muskelschäden.** ISBN 3-89649-623-9

Vol. 4: Thomas P. von Hoff, **On the Convergence of Blind Source Separation and Deconvolution.** ISBN 3-89649-624-7

Vol. 5: Markus Erne, **Signal Adaptive Audio Coding using Wavelets and Rate Optimization.** ISBN 3-89649-625-5

Vol. 6: Marcel Joho, **A Systematic Approach to Adaptive Algorithms for Multichannel System Identification, Inverse Modeling, and Blind Identification.** ISBN 3-89649-632-8

Vol. 7: Heinz Mathis, **Nonlinear Functions for Blind Separation and Equalization.** ISBN 3-89649-728-6

Vol. 8: Daniel Lippuner, **Model-Based Step-Size Control for Adaptive Filters.** ISBN 3-89649-755-3

Vol. 9: Ralf Kretzschmar, **A Survey of Neural Network Classifiers for Local Wind Prediction.** ISBN 3-89649-798-7

Vol. 10: Dieter M. Arnold, **Computing Information Rates of Finite State Models with Application to Magnetic Recording.** ISBN 3-89649-852-5

Vol. 11: Pascal O. Vontobel, **Algebraic Coding for Iterative Decoding.** ISBN 3-89649-865-7

Vol. 12: Qun Gao, **Fingerprint Verification using Cellular Neural Networks.** ISBN 3-89649-894-0

Vol. 13: Patrick P. Merkli, **Message-Passing Algorithms and Analog Electronic Circuits.** ISBN 3-89649-987-4

Vol. 14: Markus Hofbauer, **Optimal Linear Separation and Deconvolution of Acoustical Convolutive Mixtures.** ISBN 3-89649-996-3

Vol. 15: Sascha Korl, **A Factor Graph Approach to Signal Modelling, System Identification and Filtering.** ISBN 3-86628-032-7

Vol. 16: Matthias Frey, **On Analog Decoders and Digitally Corrected Converters.** ISBN 3-86628-074-2

Vol: 17: Justin Dauwels, **On Graphical Models for Communications and Machine Learning: Algorithms, Bounds, and Analog Implementation.** ISBN 3-86628-080-7

Vol. 18: Volker Maximillian Koch, **A Factor Graph Approach to Model-Based Signal Separation.** ISBN 3-86628-140-4

Vol. 19: Junli Hu, **On Gaussian Approximations in Message Passing Algorithms with Application to Equalization.** ISBN 3-86628-212-5

Vol. 20: Maja Ostojic, **Multitree Search Decoding of Linear Codes.** ISBN 3-86628-363-6

Vol. 21: Murti V.R.S. Devarakonda, **Joint Matched Filtering, Decoding, and Timing Synchronization.** ISBN 3-86628-417-9

Vol. 22: Lukas Bolliger, **Digital Estimation of Continuous-Time Signals Using Factor Graphs.** ISBN 3-86628-432-2

Hartung-Gorre Verlag Konstanz – http://www.hartung-gorre.de