Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Signal and Information Processing Laboratory
Institut für Signal- und Informationsverarbeitung

Spring Semester 2014 Prof. H.-A. Loeliger

Semester Thesis

Phase-Locked Loops

Daniel Gilgen

and

Fabio Marti

Advisor: Nour Zalmai

Co-Advisor: Lukas Bruderer

(Here the project description may be put in . . . page 1)

(Here the project description may be put in . . . page 2)

Abstract

Here comes the abstract ...

(The abstract is better included by the command `\include{file}`.)

Zusammenfassung

Das ist die Zusammenfassung …

(Die Zusammenfassung sollte besser mittels des Befehls `\include{file}` eingebunden werden.)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This is the introduction.

Chapter 2

Mathematical Fundamentals of Phase-Locked Loops

Here comes some introducing text.

2.1 Phase-Locked Loops

2.1.1 General Purpose

- Wozu braucht man Phase Locked Loops?

- Wie werden sie technisch realisiert? -> Quellen angeben!

- Zeige den Regelkreis auf -> Danach: Link zu Kalman Filtering.

2.1.2 The Kalman View on Phase Locked Loops

- Wie funktioniert ein Kalman Filter im Allgemeinen? -> Und wieder: Quellen!

- Wie sieht ein PLL mit Kalman Filter Implementierung aus? -> Block-Schaltbild

- Verweise auf nächste Section und kündige an, dass nun die Theorie entwickelt wird, um einen PLL zu implementieren

2.2 The Factor Graph Approach

2.2.1 Discrete State Space Model for Harmonic Signals

So far, we have investigated PLLs only conceptually without giving any insights into concrete mathematical details. We now do so by starting with the notion of *harmonic signals*. The general representation of a harmonic signal $s(t)$ is the following [?]

$$s(t) = \sum_{k=0}^{N} a_k \cos\left(2\pi f_k t + \varphi_k\right). \tag{2.1}$$

The lowest frequency f_0 is usually called the *fundamental frequency* and the higher frequencies $f_k,\ k = 1, \ldots, N$ are the *upper harmonics* and are integer multiples of f_0, i.e.,

$$f_k = (k+1) f_0, \qquad k = 1, \ldots, N.$$

- Beginne mit continous time model der harmonischen Signale.
- Verweise auf Nyquist Theorem und stelle fest, dass harmonische Signale ganz toll abgetastet werden können, wenn $f_s \geq 2 f_g$, wobei f_s die Abtastfrequenz ist und f_g die höchste Frequenz im harmonischen Signal -> Noch was? Genau: QUELLEN!
- Konstruiere anschliessend ein diskretes Model.
- Erstelle nun das state space model.
- Füge noise hinzu -> Danach: Link zum minimization problem

2.2.2 The Phase-Locked Loop as a Minimization Problem

- Formuliere nun den PLL als minimization problem, d.h. erstelle die cost function
- Erweitere die cost function um den decay factor
- Forme das Ganze um, bis die cost function faktorisiert ist -> Danach: Link zu Solution via Gaussian Message Passing

2.2.3 Solution via Gaussian Message Passing

- Zeichne den factor graph zu der betrachteten cost function -> Q-U-E-L-L-E-N!!!

- Zeige auf, dass belief propagation, d.h. Max-product message passing das problem löst.
- Führe Berechnungen durch und rechne $\overrightarrow{W}_{X_k}$ und $\overrightarrow{W}_{X_k}\overrightarrow{m}_{X_k}$
- Füge den Pseudocode hinzu
- Erwähne kurz, dass man aus factor graphs interessante Eigenschaften herauslesen kann -> Danach: Link zu Properties of Gaussian Message Passing

2.2.4 Properties of Gaussian Message Passing

- Füge explizite Formeln für die messages hinzu
- Zeige, dass $\overrightarrow{m}_{X_k}$ unabhängig ist vom noise
- Berechne $\overrightarrow{V}_{X_k}$ und $\overrightarrow{m}_{X_k}$ und zeige ultimativ auf, dass Kalman filtering und factor graph ein und dasselbe sind -> Wie geil ist denn das?
- Berechne steady state precision matrix und erwähne, dass dies nun computation power spart -> Herleitung oder die berühmt berüchtigen QUELLEN!

Chapter 3

Tracking of a Sinusoidal Signal

3.1 Algorithm

In 3.1 we can see the most simple approach to construct a PLL with forward message passing.

Algorithm 1 PLL implementation with forward message passing.

Parameters: A, C, n, σ^2

Input: $\{y_k\}_{k=1,\ldots,M}$

Output: $\{\tilde{\phi}_k\}_{k=1,\ldots,M}$

$k \leftarrow 0$

$\overrightarrow{W}_{X_0}\overrightarrow{m}_{X_0} \leftarrow 0^{2n\times 1}$

$\overrightarrow{W}_{X_0} \leftarrow 10^{10} \cdot \texttt{identityMatrix}(2n)$

for $k < M$ **do** ▷ Message passing algorithm

  $\overrightarrow{W}_{X_{k+1}}\overrightarrow{m}_{X_{k+1}} \leftarrow \texttt{updateMeanVector}(\overrightarrow{W}_{X_k}\overrightarrow{m}_{X_k}, y_{k+1}, A, C, \sigma^2)$

  $\overrightarrow{W}_{X_{k+1}} \leftarrow \texttt{updatePrecisionMatrix}(\overrightarrow{W}_{X_k}, A, C, \sigma^2)$

  $\tilde{\phi}_{k+1} \leftarrow \texttt{estimateCurrentPhase}(\overrightarrow{W}_{X_{k+1}}\overrightarrow{m}_{X_{k+1}}, \overrightarrow{W}_{X_{k+1}})$

  $k \leftarrow k + 1$

end for

return $\{\tilde{\phi}_k\}_{k=1,\ldots,M}$

Chapter 4

PLL for Harmonic Signals

4.1 Introduction

4.2 Blabla

Chapter 5

BlaBlaBla

Appendix A

Fancy factor graphs

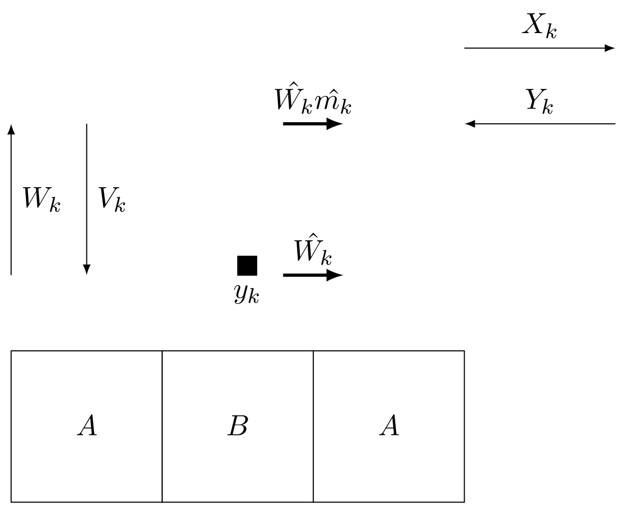

Figure A.1: All elements from the file factorGraph.sty.

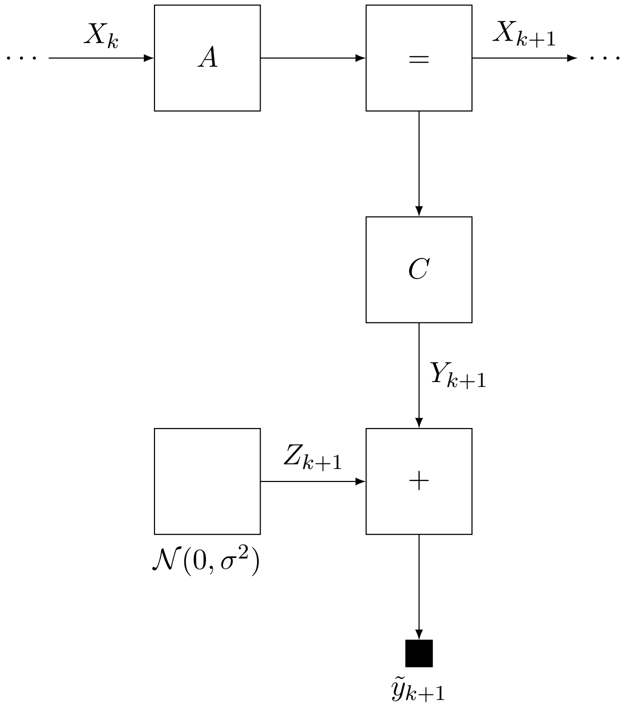

Figure A.2: This factor graph has been constructed with custom-built TikZ macros.

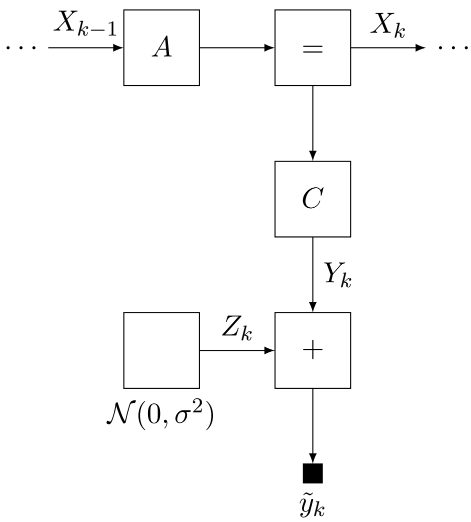

Figure A.3: This factor graph has been constructed with the factor graph macro from factorGraphs.sty.

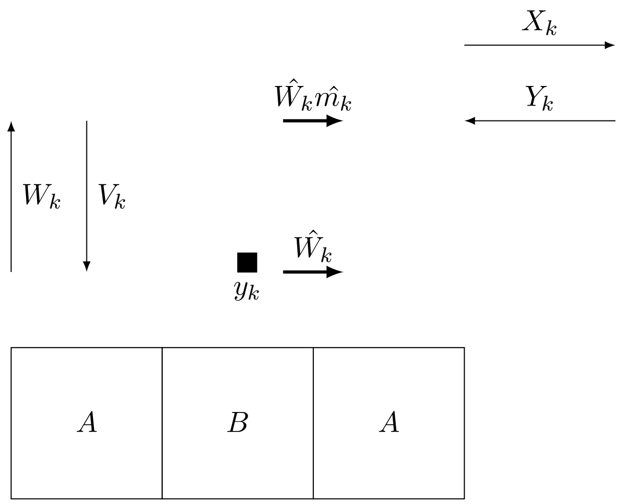

Figure A.4: All elements from the file factorGraph.sty.

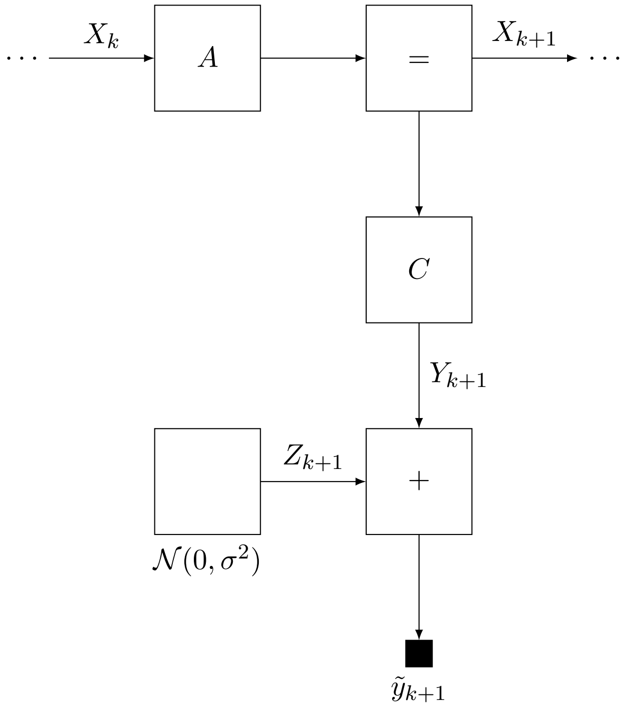

Figure A.5: This factor graph has been constructed with custom-built TikZ macros.

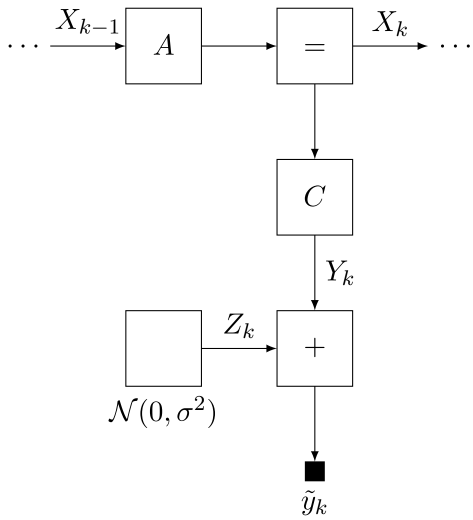

Figure A.6: This factor graph has been constructed with the factor graph macro from factorGraphs.sty.

Bibliography

[1] H.-A. Loeliger. *Signal and Information Processing: Modeling, Filtering, Learning.* Signal and Information Processing Laboratory, ETH Zurich, 2013.

[2] Erik Hampus Malmberg. Kalman filter-based phase-locked loops for harmonic signals. semester thesis. ISI, ETH Zurich, 2013.