



LUND UNIVERSITY

Modeling and Estimation Topics in Robotics

Bagge Carlson, Fredrik

2017

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Bagge Carlson, F. (2017). *Modeling and Estimation Topics in Robotics*. [Licentiate Thesis, Department of Automatic Control]. Department of Automatic Control, Lund Institute of Technology, Lund University.

Total number of authors:

1

Creative Commons License:

Unspecified

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Modeling and Estimation Topics in Robotics

Fredrik Bagge Carlson



LUND
UNIVERSITY

Department of Automatic Control

Lic. Tech. Thesis TFRT-3272
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2017 by Fredrik Bagge Carlson. All rights reserved.
Printed in Sweden by Media-Tryck.
Lund 2017

Abstract

The field of robotics offers a wide array of estimation problems, ranging from kinematic and dynamic calibration to pose estimation and computer vision. This thesis presents a set of methods to solve estimation problems encountered in robotics, with an emphasis on industrial robotics. The researched topics are all practically motivated and have found immediate use in applications.

Industrial robotics often require high accuracy in the control of the tool position, applied forces etc. This thesis presents a set of methods to solve commonly encountered estimation problems, including accurate friction estimation, spectral analysis of disturbances in electrical motors, kinematic calibration and pose estimation under the influence of high external forces.

Common themes among the articles, such as the linear least-squares procedure, are introduced in greater detail in the beginning of the thesis for the uninitiated reader.

Acknowledgements

I would most likely not have been in the position to write this thesis without the influence of my PhD thesis supervisor Prof. Rolf Johansson and my Master's thesis advisor Dr. Vuong Ngoc Dung at SIMTech, who both encouraged me to pursue the PhD degree, for which I am very thankful. Prof. Johansson has continuously supported my ideas and let me define my work with great freedom, which makes sure that I can spend time on what I believe in and that I look forward to every new day, thank you.

My thesis co-supervisor, Prof. Anders Robertsson, thank you for your never-ending enthusiasm, source of good mood and encouragement. When working 100% overtime during hot July nights in the robotlab, it helps to know that one is never alone.

MARTINKA, THE ALMIGHTY GOD OF PROGRAMMING AND ENTERTAINMENT AND ALSO ICE CREAM AND INTERNATIONAL BUSINESS RELATIONS, thank you for many good times, discussion and execution of many ideas, good and bad (bad in the best sense of the word), and of course, for sharing your great knowledge and excellent material about defects in semiconductors.

The Department of Automatic Control at Lund University is a great place, not because the building we are sitting in is so spectacular, or because the sun is always shining in Lund, but because of the excellent co-workers who share the corridors and offices. I appreciate the high level of motivation and interest people here have for their jobs, but equally much how everyone working here is contributing to the outstanding aura at the department. I like to think about the time when I was new at the department and I was living in a nearby student corridor. During Sundays, I was more or less the only one who did not dread the upcoming Monday, not only because I like my job, but because I like the environment in which I am working. I would also like to take this opportunity to thank my office mates in particular for making our office come to life.

Lisbit, you have been amazing in your efforts to help with everything from proof reading to keeping spirit on top, thank you!

Contents

1. Introduction	9
1.1 Friction and force estimation	10
1.2 Friction stir welding	10
2. Theoretical primer	12
2.1 Singular Value Decomposition	12
2.2 Least-Squares	13
2.3 Basis Function Expansions	17
3. Publications	19
Bibliography	22
Paper I. Modeling and Identification of Position and Temperature Dependent Friction Phenomena without Temperature Sensing	23
1 Introduction	24
2 Models and Identification Procedures	26
3 Simulations	32
4 Experiments	32
5 Discussion	38
6 Conclusions	39
References	40
Paper II. Linear Parameter-Varying Spectral Decomposition	41
1 Introduction	42
2 LPV Spectral Decomposition	43
3 Experimental Results	50
4 Conclusions	54
A Proofs	54
B Additional proofs not part of the original article	54
References	55
Paper III. Six DOF Eye-to-Hand Calibration from 2D Measurements Using Planar Constraints	57
1 Introduction	58
2 Preliminaries	59
3 Method	60

4	Results	63
5	Conclusions	67
	References	68
A	Calibration of point lasers	69
Paper IV. Particle Filter Framework for 6D Seam Tracking Under Large External Forces Using 2D Laser Sensors		71
1	Introduction	72
2	Method	73
3	Simulation framework	81
4	Discussion	85
5	Conclusions	86
	References	87
4.	Discussion and future work	89
4.1	Paper I	89
4.2	Paper II	90
4.3	Paper III	91
4.4	Paper IV	91

1

Introduction

The pending ubiquity of robots in everyday life, on the workshop floor and in traffic, provides researchers with a seemingly never ending stream of research problems. Many interesting problems in robotics can be categorized as *estimation* problems, in which some quantity, relation or property of the robot or its environment is to be estimated using *data* collected from the robots own or external sensors. A common way of framing estimation problems is in the framework of *optimization*. When formulating an estimation problem as an optimization problem, the goal is to minimize the model *residuals*, i.e., the misfit between the data and the model output.

This thesis will consider a series of estimation problems that can be further categorized as *system identification*, *calibration* and *state estimation* problems. We consider the simple setup depicted in Fig. 1.1, where an input signal $u(t)$ is fed into a system G that in response produces the output $y(t)$. System identification considers the problem of estimating the properties of the system G , given input-output data collected during experiments. Calibration is closely related to system identification, and deals with estimation of a property of the system that can somehow change, e.g., by changes in environmental conditions, aging of components or system reconfiguration. State estimation deals with estimation of the internal state of the system G as it operates, e.g., the positions and velocities of the links in a robot arm. To this end, the state estimation algorithm makes use of measurements of both u and y as well as a model of the system G .

The motivation for the research presented in this thesis mainly comes from the projects Flexifab and SARAFun. The Flexifab project investigates the use of industrial robots for friction stir welding, whereas the SARAFun project considers robotic assembly. The following two sections present some background of the challenges that have led to the presented work.



Figure 1.1 A system G that when subjected to the input u , produces the output y .

1.1 Friction and force estimation

Robotic applications have for a long time been concentrated around non-contact operations, such as picking and placing objects, arc welding and painting. Recently, robots have appeared also in applications that require establishing and maintaining contact with the environment. Such applications include assembly, machining, friction stir welding and surgery. The contact between the robot and its environment gives rise to *contact forces*, the magnitude and direction of which are of great importance for the quality of the job performed.

A straightforward way of measuring contact forces is to somehow equip the robot, tool or workpiece with a force sensor. While often straight forward, a force sensor typically adds cost, bulk, weight and complexity to the setup, all of which are undesirable.

Another strategy to monitor the contact forces is to estimate them. Using equations describing the relations between forces and motions in the robot, one can in theory estimate the external forces originating from contact with the environment. A huge obstacle to this approach is *friction*.

All mechanical systems with moving parts are subject to friction. The friction force is a product of interaction forces on an atomic level and is always resisting relative motion between two elements in contact. Due to the nature of friction, the force acting on a moving object is easy to estimate from observations of the motion of the object and the forces acting upon it. When the object is at rest, however, the motion is zero, but the forces acting on it can assume any value inside a non-zero interval. This phenomenon is the reason objects placed on a table won't slide off, even if the table is slightly tilted. This many-to-one relationship between forces and velocity when the sliding surfaces are at rest creates a problem when we invert the relationship, which now becomes a one-to-many relationship between velocity and force. When an object is at rest, we can thus not say anything about the force acting upon it, other than that the force is smaller than the so-called stiction force.

When estimating the external forces, accurate estimation of the friction force is key to an accurate force estimate.

1.2 Friction stir welding

During traditional welding, the metal in the pieces to be welded are melted together to form a bond. This bond is typically a weak part in the construction, and the pieces have to be made thicker, stronger and heavier to compensate for this weak link. Performing fusion welding does not require any significant forces other than to lift the tool, and can be performed by both humans and robots.

Friction stir welding (FSW) is a relatively recent welding technique in which a solid state merging of the work pieces is obtained by a rotating tool inserted between the pieces using a high force. The properties of the joint obtained with FSW are usually far superior to those obtained using traditional fusion welding, but

the high process forces involved have limited the applications of FSW. Historically, special purpose machines with high stiffness and low flexibility have been used to overcome the problems with high process forces. This has naturally made FSW a much more expensive joining technique compared to fusion welding. Recent research has, however, suggested that heavy-duty industrial robots might offer a low-cost, flexible alternative to special purpose FSW equipment, while overcoming problems with limited stiffness through modeling, perception and control [De Backer, 2014].

The promise of greatly reduced initial investment and increased flexibility of an industrial robot over a traditional FSW machine has sparked great research interest into robotic FSW. The implications of a greater availability of FSW in the industry are savings in both cost and environmental impact, as products produced with FSW can be made lighter and more energy efficient. This is particularly true for the transport and aviation sectors, where even a small increase in fuel efficiency can have a large impact on cost and emissions.

2

Theoretical primer

This chapter serves as an introduction to the reader unfamiliar with the concepts of singular value decomposition and the linear least-squares problem. These methods will be used extensively in the upcoming collection of articles, where they are only briefly introduced as needed. Readers familiar with these topics can skip this chapter.

2.1 Singular Value Decomposition

The singular value decomposition (SVD) was first developed in the late 1800s for bilinear forms, and later extended to rectangular matrices by [Eckart and Young, 1936]. The SVD is a factorization of a matrix $A \in \mathbb{R}^{N \times M}$ on the form

$$A = USV^T \tag{2.1}$$

where the matrices $U \in \mathbb{R}^{N \times N}$ and $V \in \mathbb{R}^{M \times M}$ are orthonormal, such that $U^T U = U U^T = I_N$ and $V^T V = V V^T = I_M$, and $S = \text{diag}(\sigma_1, \dots, \sigma_m) \in \mathbb{R}^{N \times M}$ is a rectangular, diagonal matrix with the singular values on the diagonal. The singular values are the square roots of the eigenvalues of the matrices AA^T and $A^T A$ and are always non-negative and real. The orthonormal matrices U and V can be shown to have columns consisting of a set of orthonormal eigenvectors of AA^T and $A^T A$ respectively.

One of many applications of the SVD that will be exploited in this thesis is to find the equation for a plane that minimizes the sum of squared distances between the plane and a set of points. The normal to this plane is simply the singular vector corresponding to the smallest singular value of a matrix composed of all point coordinates. The smallest singular value will in this case correspond to the mean squared distance between the points and the plane, i.e., the variance of the residuals.

2.2 Least-Squares

This thesis will frequently deal with the estimation of models which are linear in the parameters, and can thus be written on the form

$$y = \mathbf{A}k \quad (2.2)$$

where \mathbf{A} denotes the regressor matrix and k denotes a vector of coefficients to be identified. Models on the form (2.2) are commonly identified with the well-known least-squares procedure [Johansson, 1993]. As an example, we consider the model $y_n = k_1 u_n + k_2 v_n$, where a measured signal y is a linear combination of two input signals u and v . The identification task is to identify the parameters k_1 and k_2 . In this case, the procedure amounts to arranging the data according to

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} u_1 & v_1 \\ \vdots & \vdots \\ u_N & v_N \end{bmatrix} \in \mathbb{R}^{N \times 2}, \quad k = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

and solving the optimization problem of Eq. (2.3) with solution (2.4).

THEOREM 1

The vector k^* of parameters that solves the optimization problem

$$k^* = \underset{k}{\operatorname{argmin}} \|y - \mathbf{A}k\|_2^2 \quad (2.3)$$

is given by the closed-form expression

$$k^* = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top y \quad (2.4)$$

Proof Completion of squares in the least-squares cost function J yields

$$\begin{aligned} J &= \|y - \mathbf{A}k\|_2^2 = (y - \mathbf{A}k)^\top (y - \mathbf{A}k) \\ &= y^\top y - y^\top \mathbf{A}k - k^\top \mathbf{A}^\top y + k^\top \mathbf{A}^\top \mathbf{A}k \\ &= \left(k - (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top y \right)^\top \mathbf{A}^\top \mathbf{A} \left(k - (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top y \right) + y^\top (I - \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top) y \end{aligned}$$

where we identify the last expression as a sum of two terms, one that does not depend on k , and a term which is a positive definite quadratic form ($\mathbf{A}^\top \mathbf{A}$ is always positive (semi)definite). The estimate k^* that minimizes J is thus the value that makes the quadratic form equal to zero. \square

Equation (2.4) is known as the least-squares solution and the full-rank matrix $(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$ is commonly referred to as the pseudo inverse of \mathbf{A} . If \mathbf{A} is a square matrix, the pseudo inverse reduces to the standard matrix inverse. If \mathbf{A} however is a tall matrix, the equation $y = \mathbf{A}k$ is over determined and Eq. (2.4) produces the solution k^* that minimizes Eq. (2.3).

Consistency

The consistency of the least-squares estimate can be analyzed by calculating the bias and variance properties. Consider the standard model, with an added noise term ν , for which consistency is given by the following theorem:

THEOREM 2

$\hat{k} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top y$ is an unbiased and consistent estimate of k in the model

$$\begin{aligned} y &= \mathbf{A}k + \nu \\ \nu &\sim \mathcal{N}(0, \sigma^2) \\ \mathbb{E}\{\mathbf{A}^\top \nu\} &= 0 \end{aligned}$$

Proof The bias and variance of the resulting least-squares based estimate are:

Bias We begin by rewriting the expression for the estimate \hat{k} as

$$\begin{aligned} \hat{k} &= (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top y \\ &= (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top (\mathbf{A}k + \nu) \\ &= k + (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \nu \end{aligned}$$

If the regressors are uncorrelated with the noise, $\mathbb{E}\{(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \nu\} = 0$, we can conclude that $\mathbb{E}\{\hat{k}\} = k$ and the estimate is unbiased.

Variance The variance is given by

$$\begin{aligned} \mathbb{E}\{(\hat{k} - k)(\hat{k} - k)^\top\} &= \mathbb{E}\{(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \nu \nu^\top \mathbf{A} (\mathbf{A}^\top \mathbf{A})^{-1}\} \\ &= \mathbb{E}\{\mathbf{A}^{-1} \nu \nu^\top \mathbf{A}^{-\top}\} \\ &= \sigma^2 \mathbb{E}\{(\mathbf{A}^\top \mathbf{A})^{-1}\} \\ &= \sigma^2 (\mathbf{A}^\top \mathbf{A})^{-1} \end{aligned}$$

where the third equality holds if ν and \mathbf{A} are uncorrelated. As $N \rightarrow \infty$, we have $\sigma^2 (\mathbf{A}^\top \mathbf{A})^{-1} \rightarrow 0$, provided that the Euclidean length of all columns in \mathbf{A} increases as N increases. \square

Other loss functions

The least-squares loss function

$$k^* = \underset{k}{\operatorname{argmin}} \|y - \mathbf{A}k\|_2^2 \quad (2.5)$$

is convex and admits a particularly simple, closed-form expression for the minimum. If another norm is used instead of the L_2 norm, the estimate will have

different properties. The choice of other norms will, in general, not admit a solution on closed form, but for many norms of interest, the optimization problem remains *convex*. This fact will in practice guarantee that a global minimum can be found easily using iterative methods. Many of the methods described in this thesis could equally well be solved with another convex loss function, such as the L_1 norm for increased robustness, or the L_∞ norm for a minimum worst-case scenario. For an introduction to convex optimization and a description of the properties of different convex loss functions, see [Boyd and Vandenberghe, 2004].

Ridge regression

For certain problems, it might be desirable to add a term to the cost function Eq. (2.3) that penalizes the size of the estimated parameter vector. This might be the case if the problem is *ill-posed*, or if we have the *a priori knowledge* that the parameter vector is small. Depending on the norm in which we measure the size of the parameter vector, this procedure has many names. For the common L_2 norm, the resulting method is commonly referred to as Tikhonov regularized regression, ridge regression or weight decay if one adopts an optimization perspective, or maximum a posteriori (MAP) estimation with a Gaussian prior, if one adopts a Bayesian view on the estimation problem. The solution to the resulting optimization problem remains on a closed form, as indicated by the following theorem. Here, we demonstrate an alternative way of proving the least-squares solution, based on differentiation instead of completion of squares.

THEOREM 3

The vector k^* of parameters that solves the optimization problem

$$k^* = \frac{1}{2} \operatorname{argmin}_k \|y - \mathbf{A}k\|_2^2 + \frac{\lambda}{2} \|k\|_2^2 \quad (2.6)$$

is given by the closed-form expression

$$k^* = (\mathbf{A}^\top \mathbf{A} + \lambda I)^{-1} \mathbf{A}^\top y \quad (2.7)$$

Proof Differentiation of the cost function yields

$$J = \frac{1}{2} \|y - \mathbf{A}k\|_2^2 + \frac{\lambda}{2} \|k\|_2^2 = \frac{1}{2} (y - \mathbf{A}k)^\top (y - \mathbf{A}k) + \frac{\lambda}{2} k^\top k$$

$$\frac{dJ}{dk} = -\mathbf{A}^\top (y - \mathbf{A}k) + \lambda k$$

If we equate this last expression to zero we get

$$\frac{dJ}{dk} = -\mathbf{A}^\top (y - \mathbf{A}k) + \lambda k = 0$$

$$(\mathbf{A}^\top \mathbf{A} + \lambda I)k = \mathbf{A}^\top y$$

$$k = (\mathbf{A}^\top \mathbf{A} + \lambda I)^{-1} \mathbf{A}^\top y$$

Since $\mathbf{A}^\top \mathbf{A}$ is positive semi-definite, both first- and second-order conditions for a minimum are satisfied by $k^* = (\mathbf{A}^\top \mathbf{A} + \lambda I)^{-1} \mathbf{A}^\top y$. \square

We immediately notice that the solution to the regularized problem (2.6) reduces to the solution of the ordinary least-squares problem (2.3) in the case $\lambda = 0$. The regularization adds the positive term λ to all diagonal elements of $\mathbf{A}^\top \mathbf{A}$, which reduces the condition number of the matrix to be inverted and ensures that the problem is well posed [Golub and Van Loan, 2012]. The regularization reduces the variance in the estimate at the expense of the introduction of a bias.

Computation

Although the solutions to the least-squares problems are available in closed form, it is ill-advised to actually perform the calculation $k = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top y$ [Golub and Van Loan, 2012]. Numerically more robust strategies include

- performing a Cholesky factorization of the symmetric matrix $\mathbf{A}^\top \mathbf{A}$.
- performing a QR-decomposition of \mathbf{A} .
- performing a singular value decomposition (SVD) of \mathbf{A} .

where the latter two methods avoid the calculation of $\mathbf{A}^\top \mathbf{A}$ altogether, which can be subject to numerical difficulties if \mathbf{A} has a high condition number [Golub and Van Loan, 2012]. In fact, the method of performing a Cholesky decomposition of $\mathbf{A}^\top \mathbf{A}$ can be implemented using the QR-decomposition since triangular matrix R obtained by a QR-decomposition of \mathbf{A} is a Cholesky factor of $\mathbf{A}^\top \mathbf{A}$:

$$\mathbf{A}^\top \mathbf{A} = (QR)^\top (QR) = R^\top R$$

Many numerical computation tools, including Julia, Matlab and numpy, provide numerically robust methods to calculate the solution to the least-squares problem, indicated in Algorithm 1. These methods typically analyze the matrix \mathbf{A} and choose a suitable numerical algorithm to execute based on its properties [Julialang, 2017].

Algorithm 1 Syntax for solving the least-squares problem $k = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top y$ in different programming languages.

<code>k = A \ y</code>	<code># Julia</code>
<code>k = A \ y</code>	<code>% Matlab</code>
<code>k = numpy.linalg.solve(A, y)</code>	<code># Python with numpy</code>
<code>k <- solve(A, y)</code>	<code># R</code>

For numerically robust methods of solving the ridge regression problem, see, e.g., the excellent manual by [Hansen, 1994].

2.3 Basis Function Expansions

When estimating a functional relationship between two or more variables, i.e., $y = f(v)$, a standard initial approach is *linear regression* using the least-squares procedure. A strong motivation for this is the fact that the optimal linear combination of the chosen basis functions, or *regressors*, is available in closed form. A typical choice of basis functions are low order monomials, e.g., a decomposition of a signal y according to

$$y = \phi(v)k = k_0 + k_1 v^1 + k_2 v^2 + \dots + k_J v^J \quad (2.8)$$

where $\phi(v) = [v^0 \ v^1 \ \dots \ v^J]$ is the set of basis function activations. The function $f(v) = \phi(v)k$ can be highly nonlinear and even discontinuous in v , but is *linear in the parameters*, making it easy to fit to data.

While the low order monomials v^i are easy to work with and provide reasonable fit when the relationship between y and v is simple, they tend to perform worse when the relationship is complex.

Intuitively, a basis function expansion decomposes an intricate function or signal as a linear combination of simple basis functions. The Fourier transform can be given this interpretation, where an arbitrary signal is decomposed as a sum of complex-valued sinusoids, similarly, a stair function can be decomposed as a sum of step functions.

In many situations, there is no a priori information regarding the relationship between the free variable v and the dependent variable y , and it might be hard to choose a reasonable set of basis functions to use for a decomposition of the signal y . In such situations, an alternative is to choose a set of functions with *local support*, spread out to cover the domain of v . Some examples of basis functions with local support are: radial basis functions $\kappa(v) = \exp(-\gamma(v - \mu)^2)$, triangular functions $\kappa(v) = \max(0, 1 - \gamma|v - \mu|)$ and rectangular functions¹ $\kappa(v) = |v - \mu| < \Delta\mu$. In all cases, μ determines the center of the basis function and γ determines the width. Examples of decompositions using these basis functions are shown in Fig. 2.1.

The concept of basis function expansions will be used extensively in the initial part of the thesis.²

¹ Here we interpret the Boolean values true/false as 1/0.

² The open-source software accompanying many of the papers in this thesis makes use of basis function expansions. This functionality has been externalized into the package <https://github.com/baggepinnen/BasisFunctionExpansions.jl>, which provides many convenient methods for working with basis function expansions.

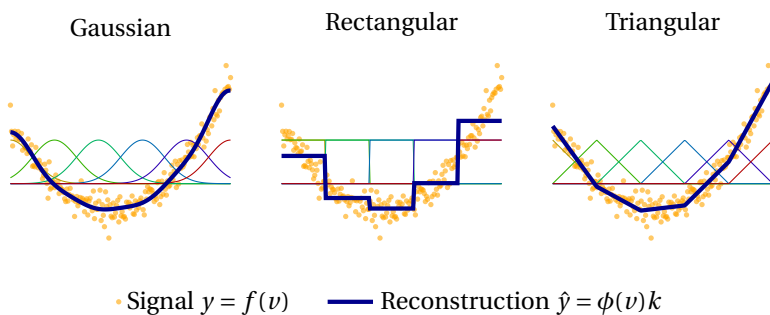


Figure 2.1 Reconstructions of a sampled signal $y = f(v)$ using different sets of basis functions. The basis functions used for the decomposition of y is shown in the background.

3

Publications

This thesis is based on the following publications:

Paper I

Bagge Carlson, F., A. Robertsson, and R. Johansson (2015). “Modeling and identification of position and temperature dependent friction phenomena without temperature sensing”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

The problem of friction estimation for systems with temperature and position dependence is considered. Due to well-known issues with adaptive estimation techniques in friction estimation, a classical system identification view is adopted and a first order dependence between input power (heat generated by friction) and the friction parameters is identified. The article further considers position dependent friction, which is shown to be substantial in certain common industrial robots. Combined, the two modeling and identification approaches greatly reduce the friction modeling error, allowing for, e.g., more accurate estimation of external forces. Open-source implementations of the algorithms presented are provided in [Bagge Carlson, 2015].

Paper II

Bagge Carlson, F., A. Robertsson, and R. Johansson (2017). “Linear parameter-varying spectral decomposition”. In: *2017 American Control Conference (ACC)*. Accepted.

During analysis of modeling residuals from the experiments conducted in Paper I, periodic patterns in the residuals were easily identified visually, but Fourier-based spectral estimation methods failed to identify the spectral contents. Several issues related to standard spectral estimation techniques were identified, where the underlying problem was found related to a dependence between the spectral

properties and an auxiliary signal, in this case, the angular velocity of the robot joint under investigation. This paper develops a novel spectral estimation technique that allows the spectral properties (phase and amplitude) of the analyzed signal to vary with an auxiliary signal. Apart from a standard spectrum, a functional relationship between the scheduling signal and the amplitude and phase of each frequency is identified, providing high levels of detail from which the origin of the signal is easier to identify. An open-source implementation of the algorithm presented is provided in [Bagge Carlson, 2016].

Paper III

Bagge Carlson, F., R. Johansson, and A. Robertsson (2015). “Six DOF eye-to-hand calibration from 2D measurements using planar constraints”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

We consider the calibration of a 2D laser scanner for localization of weld seams. The paper introduces a calibration algorithm tailored to the properties of a class of sensors commonly referred to as laser-stripe profilers. While standard eye-to-hand calibration methods could be used in theory, the properties of the data recorded by the sensor creates the need for complex pre-processing and/or additional estimation steps in order to obtain the desired calibration matrix. The developed algorithm is easy to apply without pre-processing and is shown to find the desired calibration given a wide range of initial guesses and under the influence of noise. The sensor described in this paper reappears as the main method of feedback in the following Paper IV. An open-source implementation of the algorithm presented is provided in [Bagge Carlson, 2015].

Paper IV

Bagge Carlson, F., M. Karlsson, A. Robertsson, and R. Johansson (2016). “Particle filter framework for 6D seam tracking under large external forces using 2D laser sensors”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

The problem of pose estimation during friction stir welding (FSW) is considered. We present an open-source library for simulation of seam-tracking together with a particle-filter based state estimation algorithm, utilizing stiffness models and feedback from a class of laser sensors presented in Paper III. The developed framework helps the user configure the sensor setup for the considered seam geometry as well as tune the particle-filter based state estimator. An open-source implementation of the framework presented is provided in [Bagge Carlson and Karlsson, 2016].

Other publications

The following papers, authored or co-authored by the author of this thesis, cover related topics in robotics but are not included in this thesis:

- Bagge Carlson, F., N. D. Vuong, and R. Johansson (2014). “Polynomial reconstruction of 3D sampled curves using auxiliary surface data”. In: *2014 IEEE International Conference on Robotics and Automation*.
- Karlsson, M., F. Bagge Carlson, J. De Backer, M. Holmstrand, A. Robertsson, and R. Johansson (2016). “Robotic seam tracking for friction stir welding under large contact forces”. In: *7th Swedish Production Symposium (SPS)*.
- Karlsson, M., F. Bagge Carlson, J. De Backer, M. Holmstrand, A. Robertsson, R. Johansson, L. Quintino, and E. Assuncao (n.d.). “Robotic friction stir welding, challenges and solutions”. *Welding in the World, The International Journal of Materials Joining*. ISSN: 0043-2288. Submitted.
- Karlsson, M., F. Bagge Carlson, A. Robertsson, and R. Johansson (2017). “Two-degree-of-freedom control for trajectory tracking and perturbation recovery during execution of dynamical movement primitives”. In: *20th IFAC World Congress*. Accepted.
- Stolt, A., F. Bagge Carlson, M. M. G. Ardakani, I. Lundberg, A. Robertsson, and R. Johansson (2015). “Sensorless friction-compensated passive lead-through programming for industrial robots”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Bibliography

- Bagge Carlson, F. (2015). *Robotlib.jl*. Dept. Automatic Control. URL: <https://gitlab.control.lth.se/cont-frb/robotlib>.
- Bagge Carlson, F. (2016). *Lpvspectral.jl*. Dept. Automatic Control. URL: <https://github.com/baggepinnen/LPVSpectral.jl>.
- Bagge Carlson, F. and M. Karlsson (2016). *Pfseamtracking.jl*. Dept. Automatic Control. URL: <https://github.com/baggepinnen/PFSeamTracking.jl>.
- Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge University Press, Cambridge, UK.
- De Backer, J. (2014). *Feedback Control of Robotic Friction Stir Welding*. PhD thesis. ISBN 978-91-87531-00-2, University West, Trollhättan, Sweden.
- Eckart, C. and G. Young (1936). “The approximation of one matrix by another of lower rank”. *Psychometrika* **1**:3, pp. 211–218.
- Golub, G. H. and C. F. Van Loan (2012). *Matrix computations*. Vol. 3. Johns Hopkins University Press, Baltimore.
- Hansen, P. C. (1994). “Regularization tools: a matlab package for analysis and solution of discrete ill-posed problems”. *Numerical algorithms* **6**:1, pp. 1–35. URL: <http://www2.compute.dtu.dk/~pcha/Regutools/RTv4manual.pdf> (visited on 2017-01).
- Johansson, R. (1993). *System modeling & identification*. Prentice-Hall, Englewood Cliffs, NJ.
- Julialang (2017). *Julia standard library*. URL: <http://docs.julialang.org/en/stable/stdlib/linalg/> (visited on 2017-01).

Paper I

Modeling and Identification of Position and Temperature Dependent Friction Phenomena without Temperature Sensing

Fredrik Bagge Carlson Anders Robertsson Rolf Johansson

Abstract

This paper investigates both positional dependence in systems with friction and the influence an increase in temperature has on the friction behavior. The positional dependence is modeled with a Radial Basis Function network and the temperature dependence is modeled as a first order system with the power loss due to friction as input, eliminating the need for temperature sensing. The proposed methods are evaluated in both simulations and experiments on two industrial robots with strong positional and temperature friction dependence.

Originally published in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Reprinted with permission. Open-source implementations of the algorithms presented are provided in [Bagge Carlson, 2015].

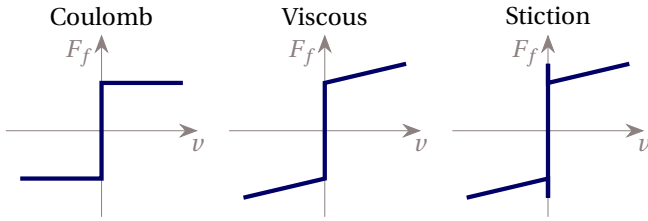


Figure 1. Illustrations of simple friction models.

1. Introduction

All mechanical systems with moving parts are subject to friction. The friction force is a product of interaction forces on an atomic level and is always resisting relative motion between two elements in contact. Due to the complex nature of the interaction forces, friction is usually modeled based on empirical observations. The simplest model of friction is the Coulomb model, Eq (1), which assumes a constant friction force acting in the reverse direction of motion

$$F_f = k_c \text{sign}(v) \quad (1)$$

where k_c is the Coulomb friction constant and v is the relative velocity between the interacting surfaces.

A slight extension to the Coulomb model includes also velocity dependent terms

$$F_f = k_v v + k_c \text{sign}(v) \quad (2)$$

where k_v is the viscous friction coefficient. The Coulomb model and the viscous model are illustrated in Fig. 1. If the friction is observed to vary with $\text{sign}(v)$, the model (2) can be extended to

$$F_f = k_v v + k_c^+ \text{sign}(v^+) + k_c^- \text{sign}(v^-) \quad (3)$$

where the sign operator is defined to be zero for $v = 0$, $v^+ = \max(0, v)$ and $v^- = \min(0, v)$.

It is commonly observed that the force needed to initiate movement from a resting position is higher than the force required to maintain a low velocity. This phenomenon, called stiction, is illustrated in Fig. 1. The friction for zero velocity and an external force F_e can be modeled as

$$F_f = \begin{cases} F_e & \text{if } v = 0 \text{ and } |F_e| < k_s \\ k_s \text{sign} F_e & \text{if } v = 0 \text{ and } |F_e| \geq k_s \end{cases} \quad (4)$$

where k_s is the stiction friction coefficient. An external force greater than the stiction force will, according to model (4), cause an instantaneous acceleration and a discontinuity in the friction force.

The models above suffice for many purposes but can not explain several commonly observed friction-related phenomena, such as the Stribeck effect and dynamical behavior etc. [Olsson et al., 1998]. To explain more complicated behavior, dynamical models such as the Dahl model [Dahl, 1968] and the LuGre model [De Wit et al., 1995] have been proposed.

Most proposed friction models include velocity-dependent effects, but no position dependence. A dependence upon position is however often observed, and may stem from, for instance, imperfect assembly, irregularities in the contact surfaces or application of lubricant etc. [Armstrong-Hélouvry et al., 1994]. Modeling of the position dependence is unfortunately nontrivial due to an often irregular relationship between the position and the friction force. Several authors have however made efforts in the area. In [Armstrong, 1988] the author uses accurate friction measurements to implement a look-up table for the position dependence and in [Huang et al., 1998] the authors adaptively identify a sinusoidal position dependence.

More recent endeavors include [Kruif and Vries, 2002] where an Iterative Learning Control approach is used to learn a feedforward model including position dependent friction terms.

In [Bittencourt and Gunnarsson, 2012], no significant positional dependence of the friction in a robot joint was found, however, a clear dependence upon the temperature of contact region was reported. To allow for temperature sensing, the grease in the gear box was replaced by an oil-based lubricant which allowed for temperature sensing in the oil flow circuit.

A standard approach in dealing with systems with varying parameters is recursive identification during normal operation [Johansson, 1993]. Recursive identification of the models (1) and (2) could account for both position- and temperature dependence. Whereas straight forward in theory, it is often hard to perform in a robust manner in practical situations. Presence of external forces, accelerating motions etc. require either a break in the adaptation, or an accurate model of the additional dynamics. Many control programs, such as time-optimal programs, never exhibit zero acceleration, and thus no chance for parameter adaptation.

This paper suggests a model that incorporates positional friction dependence as well as a temperature dependent term. Since many industrially relevant systems lack temperature sensing in areas of importance for friction modeling, a sensorless approach is proposed. Both models are used for identification of friction in the joint of an ABB YuMi robot, see Fig. 2, and special aspects of position dependence are verified on an ABB IRB140. The models and identification procedures are introduced in Sec. 2 and verification is performed in Sec. 3 and Sec. 4. The paper is summarized in Sec. 6.

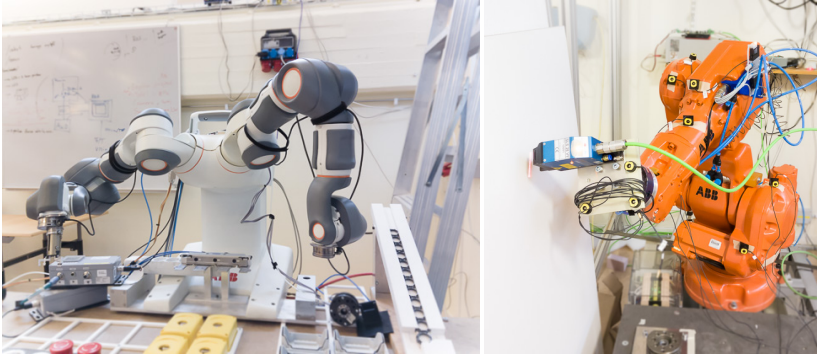


Figure 2. ABB YuMi and ABB IRB140 used for experimental verification of proposed models and identification procedures.

2. Models and Identification Procedures

This section first introduces a general identification procedure for linear models, based on the least-squares method, followed by the introduction of a model which allows for the friction to vary with position. Third, a model which accounts for temperature varying friction phenomena is introduced. Here, a sensor-less approach where the power loss due to friction is used as an input to a first order system, is adopted.

As the models are equally suited for friction due to linear and angular movements, the terms force and torque are here used interchangeably.

2.1 Least-Squares Identification

A standard model of the torques in rigid-body dynamical systems, such as industrial robots, is [Spong et al., 2006]

$$\tau = M(p)a + C(p, v)v + G(p) + F(v) \quad (5)$$

where $a = \dot{v} = \ddot{p}$ is the acceleration, τ the control torque, M, C, G are matrices representing inertia-, Coriolis-, centrifugal- and gravitational forces and F is a friction model. If a single joint at the time is operated, at constant velocity, Coriolis effects disappear [Spong et al., 2006] and

$$\left. \begin{array}{l} C(p, v) = 0 \\ a = 0 \end{array} \right\} \Rightarrow \tau = G(p) + F(v) \quad (6)$$

To further simplify the presentation, it is assumed that $G(p) = 0$. This can easily be achieved by either aligning the axis of rotation with the gravitational vector such that gravitational forces vanish, by identifying and compensating for a gravity

model¹ or, as in [Bittencourt and Gunnarsson, 2012], performing a symmetric experiment with both positive and negative velocities and calculating the torque difference.

The simple models described in Sec. 1 are commonly identified with the well-known least-squares procedure [Johansson, 1993; Golub and Van Loan, 2012; Rugh, 1996]. For the model (2), this amounts to arranging data that satisfies Eq. (6) according to

$$y = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_N \end{bmatrix}, \mathbf{A} = \begin{bmatrix} v_1 & \text{sign}(v_1) \\ \vdots & \vdots \\ v_N & \text{sign}(v_N) \end{bmatrix} \in \mathbb{R}^{N \times 2}, k = \begin{bmatrix} k_v \\ k_c \end{bmatrix} \quad (7)$$

and solving optimization problem (8) with solution (9).

$$k^* = \underset{k}{\operatorname{argmin}} \|\mathbf{A}k - y\| \quad (8)$$

$$k^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T y \quad (9)$$

2.2 Position Dependent Model

As mentioned in Sec. 1, a positional, repeatable friction dependence is often observed in mechanical systems. This section extends the simple nominal models presented in Sec. 1 with position dependent terms, where the position dependence is modeled with a radial basis function network (RBFN)² [Murphy, 2012].

Define the Gaussian RBF kernel κ and the kernel vector ϕ

$$\kappa(p, \mu, \sigma) = \exp\left(-\frac{(p - \mu)^2}{2\sigma^2}\right) \quad (10)$$

$$\phi(p) : (p \in \mathcal{P}) \rightarrow \mathbb{R}^{1 \times K}$$

$$\phi(p) = [\kappa(p, \mu_1, \sigma), \dots, \kappa(p, \mu_K, \sigma)] \quad (11)$$

where $\mu_i \in \mathcal{P}, i = 1, \dots, K$ is a set of K evenly spaced centers. For each input position $p \in \mathcal{P} \subseteq \mathbb{R}$, the kernel vector $\phi(p)$ will have activated (>0) entries for the kernels with centers close to p . The parameter σ in Eq. (10) determines the bandwidth of the RBFs. A large value of σ will result in a smooth estimate of the position dependence with low variance. Smaller values increase the variance but are able to capture finer detail. Refer to Fig. 3 for an illustration of RBFs. The kernel vector is appended the matrix \mathbf{A} from Sec. 2.1 such that

$$\mathbf{A} = \begin{bmatrix} v_1 & \text{sign}(v_1) & \phi(p_1) \\ \vdots & \vdots & \vdots \\ v_N & \text{sign}(v_N) & \phi(p_N) \end{bmatrix} \in \mathbb{R}^{N \times (2+K)}, k = \begin{bmatrix} k_v \\ k_c \\ k_\kappa \end{bmatrix} \quad (12)$$

¹ For a single joint, this simply amounts to appending the regressor matrix \mathbf{A} in Eq. (7) with $\begin{bmatrix} \sin(p) & \cos(p) \end{bmatrix}$

² Other common terms are Kernel Machines and RBF expansions.

where $k_\kappa \in \mathbb{R}^K$ denotes the parameters corresponding to the kernel vector entries. The number of RBFs to include and the bandwidth σ is usually chosen based on evidence maximization or cross validation [Murphy, 2012].

The position dependent model can now be summarized as

$$F_f = F_n + \phi(p)k_\kappa \quad (13)$$

where F_n is one of the nominal models from Sec. 1.

The above method is valid for position-varying Coulomb friction. It is conceivable that the position dependence is affected by the velocity, in which case the model (13) will produce a sub-optimal result. The RBF network can however be designed to cover the space $(\mathcal{P} \times \mathcal{V}) \subseteq \mathbb{R}^2$. The inclusion of velocity dependence comes at the cost of an increase in the number of parameters from K_p to $K_p K_v$, where K_p and K_v denote the number of basis function centers in the position and velocity input spaces respectively.

The expression for the RBF kernel will in this extended model assume the form

$$\kappa(x, \mu, \Sigma) = \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right) \quad (14)$$

where $x = [p \quad v]^\top \in \mathcal{P} \times \mathcal{V}$, $\mu \in \mathcal{P} \times \mathcal{V}$ and Σ is the covariance matrix determining the bandwidth. The kernel vector will be

$$\begin{aligned} \phi(x) : (x \in \mathcal{P} \times \mathcal{V}) &\rightarrow \mathbb{R}^{1 \times (K_p K_v)} \\ \phi(x) &= [\kappa(x, \mu_1, \Sigma), \dots, \kappa(x, \mu_{K_p K_v}, \Sigma)] \end{aligned} \quad (15)$$

This concept extends to higher dimensions, at the cost of an exponential growth in the number of model parameters.

Normalization For some applications, it may be beneficial to normalize the kernel vector for each input point [Bugmann, 1998] such that

$$\bar{\phi}(x) = \left(\sum_{i=1}^{K_p K_v} \kappa(x, \mu_i, \Sigma) \right)^{-1} \phi(x) \quad (16)$$

One major difference between a standard RBF network and a normalized RBF network (NRBFN) is the behavior far (in terms of Mahalanobis distance) from the training data. The prediction of an RBFN will tend towards zero, whereas the prediction from an NRBFN keeps its value. Figure 3 shows two networks fit to the function $f(t) = 0.3t^2 - 0.5$ together with the basis functions used. The RBF tends towards zero both outside the data points and in the interval of missing data in the center. The NRBF on the other hand generalizes better and keeps its current prediction trend outside the data. The performance of NRBF networks is studied in detail in [Bugmann, 1998].

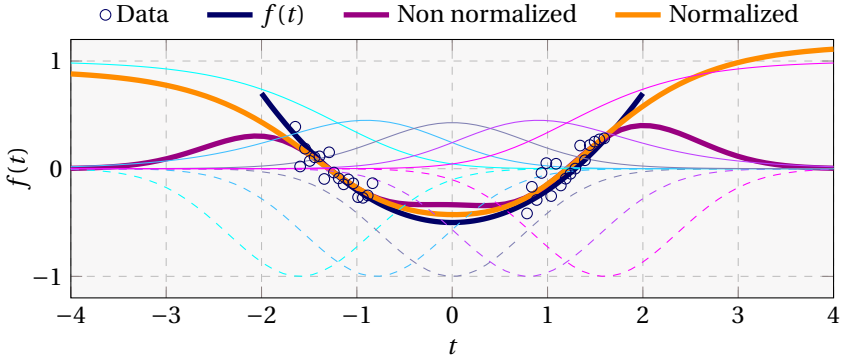


Figure 3. RBF networks fit to noisy data from the function $f(t) = 0.3t^2 - 0.5$ using normalized (-) and non-normalized (- -) basis functions. Non-normalized basis functions are shown mirrored in the x -axis.

2.3 Energy Dependent Model

Friction is often observed to vary with the temperature of the contact surfaces and lubricants involved [Bittencourt and Gunnarsson, 2012]. Many systems of industrial relevance lack the sensors needed to measure the temperature of the contact regions, thus rendering temperature dependent models unusable.

The rise in temperature that occurs during operation is mostly due to friction losses. This section introduces a model which includes the generated energy, and estimates its influence on the friction.

A simple model for the temperature change in a system with temperature T , surrounding temperature T_s , and power input W , is given by

$$\frac{dT(t)}{dt} = k_s(T_s - T(t)) + k_W W(t) \quad (17)$$

for some constants $k_s > 0, k_W > 0$. After the variable change $\Delta T(t) = T(t) - T_s$, and transformation to the Laplace domain, the model (17) can be written

$$\Delta T(s) = \frac{k_W}{s + k_s} W(s) \quad (18)$$

where the power input generated by friction losses is equal to the product of the friction force and the velocity

$$W(t) = |F_f(t) v(t)| \quad (19)$$

We are now ready to introduce the proposed model, which takes on the form

$$F_f = F_n + \text{sign}(v) E \quad (20)$$

$$E(s) = G(s) W(s) = \frac{\bar{k}_e}{1 + s\bar{\tau}_e} W(s) \quad (21)$$

where the friction force F_f has been divided into the nominal friction F_n and the signal E , corresponding to the influence of the thermal energy stored in the joint. The nominal model F_n can be chosen as any of the models previously introduced, including (13). The energy is assumed supplied by the instantaneous power due to friction, W , and is dissipating as a first order system with time constant $\bar{\tau}_e$. A discrete representation is obtained after Zero-Order-Hold (ZOH) sampling [Wittenmark et al., 2002] according to

$$E(z) = H(z)W(z) = \frac{k_e}{z - \tau_e} W(z) \quad (22)$$

In the suggested model form, Eqs. (20) to (22), the transfer function $H(z)$ incorporates both the notion of energy being stored and dissipated, as well as the influence of the stored energy on the friction.

The proposed model suggests that the change in friction due to the temperature change occurs in the Coulomb friction. This assumption is always valid for the nominal model (1), and a reasonable approximation for the model (2) if $k_c \gg k_v v$ or if the system is both operated and identified in a small interval of velocities. If, however, the temperature change has a large effect on the viscous friction or on the position dependence, a 3D basis function expansion can be performed in the space $\mathcal{P} \times \mathcal{V} \times \mathcal{E}$, $E \in \mathcal{E}$. This general model can handle arbitrary nonlinear dependencies between position, velocity and estimated temperature. The energy signal E can then be estimated using a simple nominal model, and included in the kernel expansion for an extended model. Further discussion on this is held in Sec. 5.

Denote by $\hat{\tau}_n$ the output of the nominal model F_n . Estimation of the signal E can now be done by rewriting Eq. (20) in two different ways

$$\hat{E} = (\tau - \hat{\tau}_n) \text{sign}(v) \quad (23)$$

$$F_n = \tau - \text{sign}(v)\hat{E} \quad (24)$$

The joint estimation of the parameters in the nominal model and in $H(z)$ in Eq. (22) can be carried out in an Expectation-Maximization like fashion [Murphy, 2012]. This amounts to iteratively finding an estimate \hat{F}_n of the nominal model, using \hat{F}_n to find an estimate \hat{E} of E according to Eq. (23), using \hat{E} to estimate $H(z)$ in Eq. (22) and, using $H(z)$, filter $\hat{E} = H(z)W$.

Initial Guess For this scheme to work, an initial estimate of the parameters in $H(z)$ is needed. This can be easily obtained by observing the raw torque data from an experiment. Consider for example Fig. 4, where the system (20) and (21) has been simulated. The figure depicts the torque signal as well as the energy signal E . The envelope of the torque signal decays approximately as the signal E , which allows for easy estimation of the gain \bar{k}_e and the time constant $\bar{\tau}_e$. The time constant $\bar{\tau}_e$ is determined by the time it takes for the signal to reach $(1 - e^{-1}) \approx 63\%$ of its final value. Since $G(s)$ is essentially a low-pass filter, the output $E = G(s)W$ will approximately reach $E_\infty = G(0)\mathbb{E}(W) = \bar{k}_e\mathbb{E}(W)$ if sent a stationary, stochastic input W with fast enough time constant ($\ll \bar{\tau}_e$). Here, $\mathbb{E}(\cdot)$ denotes the statistical

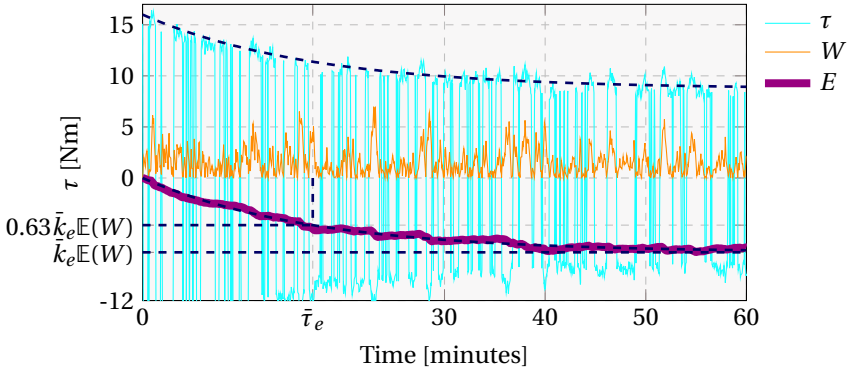


Figure 4. A realization of simulated signals. The figure shows how the envelope of the applied torque approximately decays as the signal E . Dashed, blue lines are drawn to illustrate the determination of initial guesses for the time constant $\bar{\tau}_e$ and the gain \bar{k}_e .

expectation operator and E_∞ is the final value of the signal E . An initial estimate of the gain \bar{k}_e can thus be obtained from the envelope of the torque signal as

$$\bar{k}_e \approx \frac{E_\infty}{\mathbb{E}(W)} \approx \frac{E_\infty}{\frac{1}{N} \sum_n W_n} \quad (25)$$

Refer to Fig. 4 for an illustration, where dashed guides have been drawn to illustrate the initial guesses.

The discrete counterpart to $G(s)$ can be obtained by discretization with relevant sampling time [Wittenmark et al., 2002].

Estimating the Model An algorithm for the estimation of all parameters in Eqs. (20) to (22) is given in Algorithm 2. The estimation of $\hat{H}(z)$ in Eq. (22) can be done with, e.g., the Output Error Method [Johansson, 1993] and the estimation of the nominal model is carried out using the LS procedure from Sec. 2.1.

Algorithm 2 Estimation of the parameters and the signal E in the energy dependent friction model.

Require: Initial estimate $\hat{H}(z, k_e, \tau_e)$;

repeat

 Calculate \hat{E} according to Eq. (23);

 Update $\hat{H}(z)$ using Eq. (22)

 ▷ E.g., command `oe()` in Matlab;

$\hat{E} \leftarrow \hat{H}(z)W$

 ▷ Filter W through $\hat{H}(z)$;

 Update F_n according to (24) using Eq. (9);

until Convergence

3. Simulations

To analyze the validity of the proposed technique for estimation of the energy dependent model, a simulation experiment was performed. The system described by (20) and (21) was simulated to create 50 realizations of the relevant signals, and the proposed method was run for 50 iterations to identify the model parameters. The parameters used in the simulation are provided in Table 1. Initial guesses were chosen at random from the uniform distributions $\hat{k}_e \sim \mathcal{U}(0, 3\bar{k}_e)$ $\hat{\tau}_e \sim \mathcal{U}(0, 3\bar{\tau})$.

Table 1. Parameter values used in simulation. Values given on the format x/y represent continuous/discrete values.

Parameter	Value
k_v	5
k_c	15
k_e	-3/-0.5
τ_e	10/0.9983
Measurement noise σ_τ	0.5 Nm
Sample time h	1 s
Duration	3600 s
Iterations	50

Figure 5 shows that the estimated parameters converge rapidly to their true values, and Fig. 6 indicates that the Root Mean Square output Error (RMSE) converges to the level of the added measurement noise. Figure 6 further shows that the errors in the parameter estimates, as defined by Eq. (26), were typically below 5 % of the parameter values.

$$\text{NPE} = \sqrt{\sum_{i=1}^{N_p} \left(\frac{\hat{x}_i - x_i}{|x_i|} \right)^2} \quad (26)$$

4. Experiments

The proposed models and identification procedures were applied to data from experiments with the ABB YuMi, and the ABB IRB140 industrial robots, see Fig. 2.

4.1 Procedure

For IRB140, the first joint was used. The rest of the arms were positioned so as to minimize the moment of inertia. For YuMi, joint four in one of the arms was positioned such that the influence of gravity vanished.

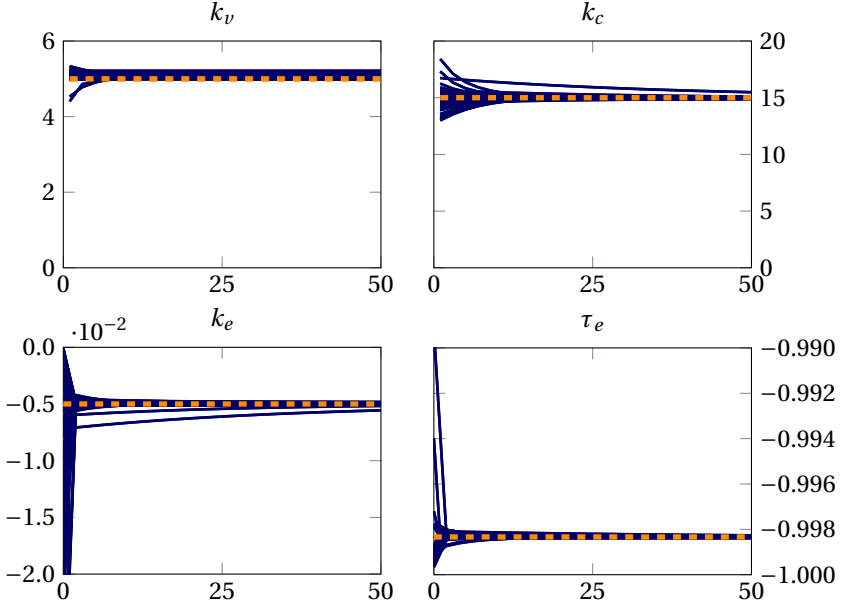


Figure 5. Estimated parameters during 50 simulations. The horizontal axis displays the iteration number and the vertical axis the current parameter value. True parameter values are indicated with dashed lines.

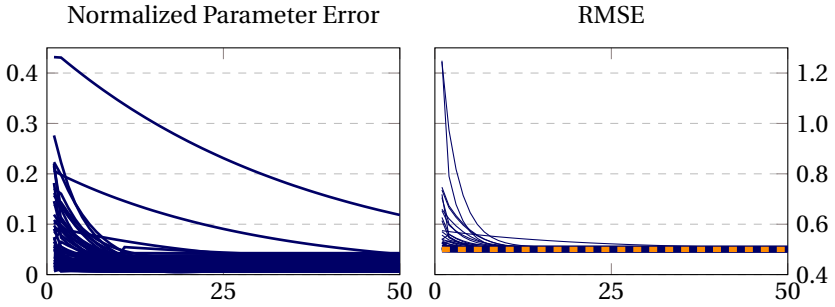


Figure 6. Evolution of errors during the simulation experiment, the horizontal axis displays the iteration number. The left plot shows normalized norms of parameter errors, defined in Eq. (26), and the right plot shows the RMS output error using the estimated parameters. The standard deviation of the added measurement noise is shown with a dashed line.

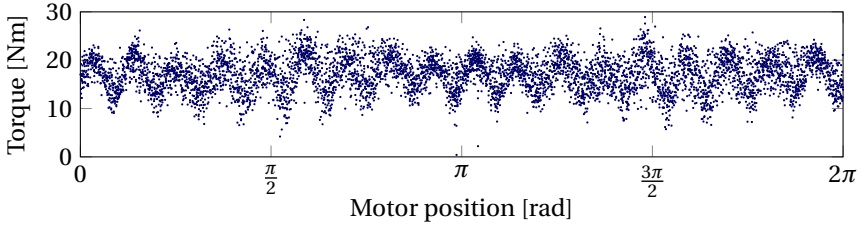


Figure 7. Illustration of the torque dependence upon the motor position for the IRB140 robot.

A program which moved the selected joint at piecewise constant velocities between the two joint limits was executed for approximately 20 min. Torque-, velocity-, and position data were sampled and filtered at 250 Hz and subsequently sub-sampled and stored at 20 Hz, resulting in 25 000 data points. Points approximately satisfying Eq. (6) were selected for identification, resulting in a set of 16 000 data points.

Nominal Model The viscous model (3) was fit using the ordinary LS procedure from Sec. 2.1. This model was also used as the nominal model in the subsequent fitting of position model (13) and energy model, Eqs. (20) to (22).

Position Model For the position dependent model, the number of basis functions and their bandwidth was determined using cross validation. A large value of σ has a strong regularizing effect and resulted in a model that generalized well outside the training data. The model was fit using normalized basis functions.

Due to the characteristics of the gear box in many industrial robots, there is a clear dependence not only on the arm position, but also on the motor position. Figure 7 shows the torque versus the motor position when the joint is operated at constant velocity. This is especially strong on the IRB140 and results are therefore illustrated for this robot. Both arm and motor positions are available through the simple relationship $p_{motor} = \text{mod}_{2\pi}(g \cdot p_{arm})$, where g denotes the gear ratio. This allows for basis function expansion also for the motor positions. To illustrate this, p_{motor} was expanded into $K_{p_m} K_v = 36 \times 6$ basis functions, corresponding to the periodicity observed in Fig. 7. The results for the model with motor position dependence are reported separately.

To reduce variance in the estimated kernel parameters, all position-dependent models were estimated using ridge regression [Murphy, 2012], where a Gaussian prior was put on the kernel parameters. The strength of the prior was determined using cross validation. All basis function expansions were performed with normalized basis functions.

Energy Model The energy dependent model was identified for YuMi using the procedure described in Algorithm 2. The initial guesses for $H(z)$ were $\bar{\tau}_e = 10$ min and $\bar{k}_e = -0.1$. The nominal model was chosen as the viscous friction model Eq. (3). Once the signal E was estimated, a kernel expansion in the space $\mathcal{P} \times \mathcal{V} \times \mathcal{E}$

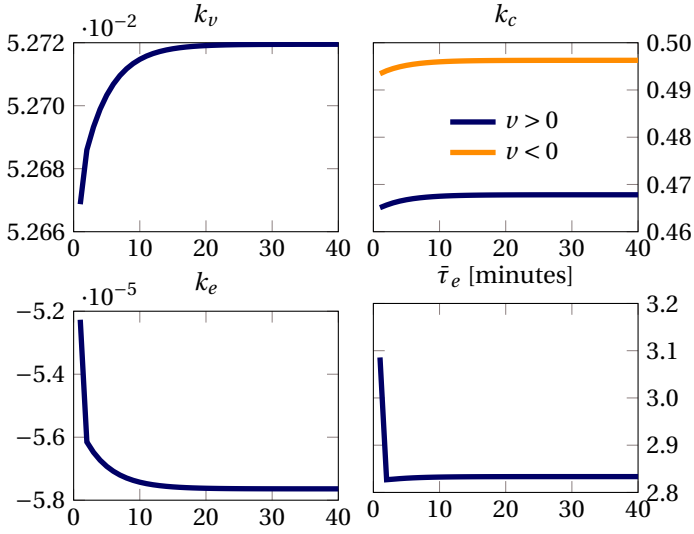


Figure 8. Estimated parameters from experimental data. The horizontal axis displays the iteration number and the vertical axis the current parameter value.

Table 2. Performance indicators for the identified models, YuMi.

	Nominal	Position	Position + Energy
Fit	86.968	93.193	96.674
FPE	3.63e-03	1.03e-03	2.65e-04
RMSE	6.03e-02	3.15e-02	1.54e-02
MAE	4.71e-02	2.36e-02	1.22e-02

with $40 \times 6 \times 3$ basis functions was performed to capture temperature dependent effects in both the Coulomb and viscous friction parameters.

4.2 Results

The convergence of the model parameters is shown in Fig. 8 and Fig. 9 illustrates how the models identified for YuMi fit the experimental data. The upper plot shows an early stage of the experiment when the joint is cold. At this stage, the model without the energy term underestimates the torque needed, whereas the energy model does a better job. The lower plot shows a later stage of the experiment where the mean torque level is significantly lower. Here, the model without energy term is instead slightly over estimating the friction torque. The observed behavior is expected, since the model without energy dependence will fit the average friction level during the entire experiment. The two models correspond well in the middle of the experiments (not shown). The nominal model (3), can not account for

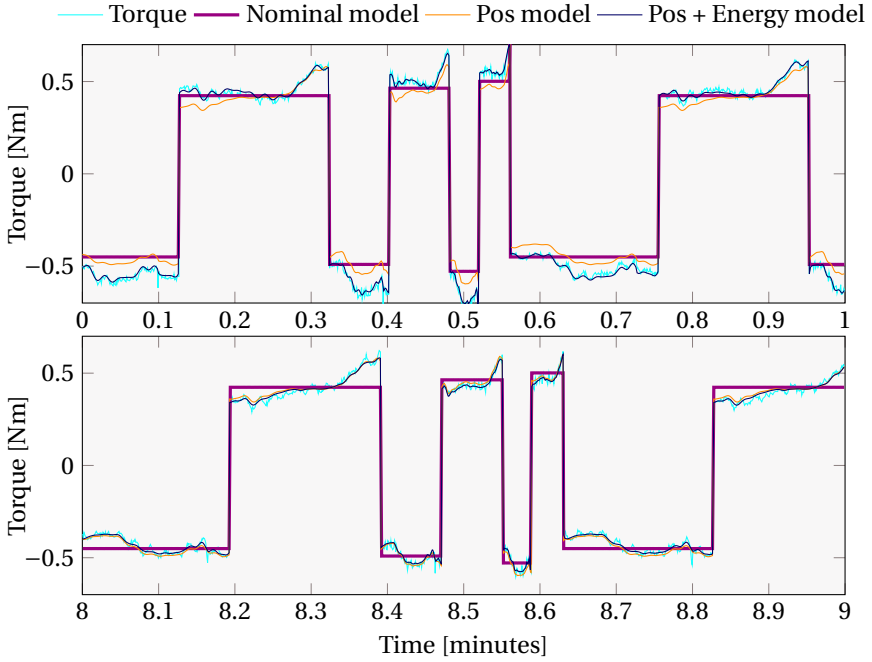


Figure 9. Model fit to experimental data (YuMi). Upper plot shows an early stage of the experiment when the joint is cold. Lower plot a later stage, when the joint has been warmed up.

any of the positional effects and produces an overall, much worse fit. Different measures of model fit for the three models are presented in Table 2 and Fig. 11 (Fit (%), Final Prediction Error, Root Mean Square Error, Mean Absolute Error). For definitions, see e.g. [Johansson, 1993].

For the IRB140, three models are compared. The nominal model Eq. (3), a model with a basis function expansion in the space \mathcal{P}_{arm} and a model with an additional basis function expansion in the space $\mathcal{P}_{motor} \times \mathcal{V}$. The resulting model fits are shown in Fig. 10. What may seem like random measurement noise in the torque signal is in fact predictable using a relatively small set of parameters. Figure 12 illustrates that the large dependence of the torque on the motor position results in large errors. The inclusion of a basis function expansion of the motor position in the model reduces the error significantly.

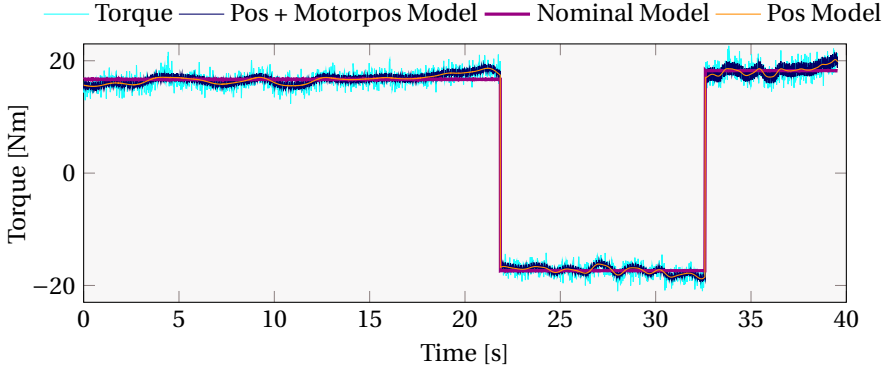


Figure 10. Model fit including kernel expansion for motor position on IRB140. During $t = [0\text{ s}, 22\text{ s}]$, the joint traverses a full revolution of 2π rad. The same distance was traversed backwards with a higher velocity during $t = [22\text{ s}, 33\text{ s}]$. Notice the repeatable pattern as identified by the position dependent models.

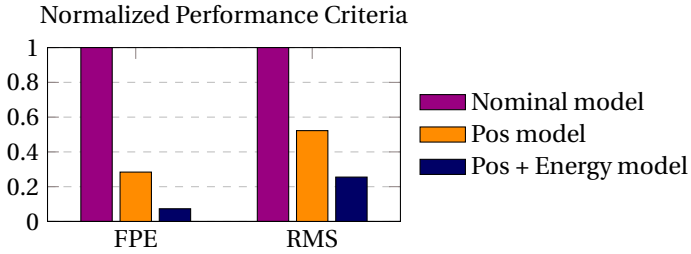


Figure 11. Performance indicators for the identified models, YuMi.

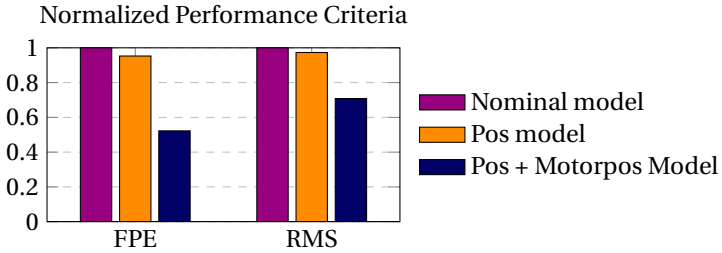


Figure 12. Performance indicators for the identified models, IRB140.

5. Discussion

The proposed models try to increase the predictive power of common friction models by incorporating position- and temperature dependence. Systems with varying parameters can in theory be estimated with recursive algorithms, so called online identification. As elaborated on in Sec. 1, online or observer-based identification of friction models is often difficult in practice due to the presence of additional dynamics or external forces. The proposed methods are identified offline, during a controlled experiment, and are thus not subject to the problems associated with online identification. However, apart from the temperature related parameters, all suggested models are linear in the parameters, and could be updated recursively using for instance the well-known recursive least squares or Kalman filter algorithms [Johansson, 1993].

Although outside the scope of this work, effects of joint load on the friction behavior can be significant [Bittencourt and Gunnarsson, 2012]. Such dependencies could be incorporated in the proposed models using the same RBF approach as for the incorporation of position dependence, i.e. through an RBF expansion in the joint load ($l \in \mathcal{L}$) dimension according to $\phi(x) : (x \in \mathcal{P} \times \mathcal{E} \times \mathcal{L}) \rightarrow \mathbb{R}^{1 \times (K_p K_e K_l)}$, with K_l basis function centers along dimension \mathcal{L} . This strategy would capture possible position and temperature dependencies in the load-friction interaction.

In its simplest form, the proposed energy dependent model assumes that the change in friction occurs in the Coulomb friction level. This is always valid for the Coulomb model, and a reasonable approximation for the viscous friction model if $k_c \gg k_v v$ or if the system is both operated and identified in a small interval of velocities. If the viscous friction $k_v v$ is large, the approximation will be worse. This suggests modeling the friction as

$$F_f = k_v(E) v + k_c(E) \text{sign}(v) \quad (27)$$

where the Coulomb- and viscous constants are seen as functions of the estimated energy signal E , i.e., a Linear Parameter-Varying model (LPV). To accomplish this, a kernel expansion including the estimated energy signal was suggested and evaluated experimentally.

Although models based on the internally generated power remove the need for temperature sensing in some scenarios, they do not cover significant variations in the surrounding temperature. The power generated in, for instance, an industrial robot is, however, often high enough to cause a much larger increase in temperature than the expected temperature variations of its surrounding [Bittencourt and Gunnarsson, 2012].

6. Conclusions

The modeling of both position and temperature dependence in systems with friction have been investigated. To model position varying friction, a Radial Basis Function network approach was adopted. It has been experimentally verified that taking position dependence into account can significantly reduce the model output error. It has also been reported that friction phenomena on both sides of a gearbox can be modeled using the proposed approach.

The influence of an increase in temperature due to power generated by friction has been modeled and estimated. The proposed approach was based on a first-order temperature input-output model where the power generated by friction was used as input. The model together with the proposed identification procedure was shown to capture the decrease in friction seen in an industrial robot during a long term experiment, this was accomplished without the need of temperature sensing.

References

- Armstrong, B. (1988). "Friction: experimental determination, modeling and compensation". In: *Robotics and Automation, Proc. 1988 IEEE Int. Conf. Pennsylvania*, pp. 1422–1427.
- Armstrong-Hélouvry, B., P. Dupont, and C. C. De Wit (1994). "A survey of models, analysis tools and compensation methods for the control of machines with friction". *Automatica* **30**:7, pp. 1083–1138.
- Bagge Carlson, F. (2015). *Robotlib.jl*. Dept. Automatic Control. URL: <https://gitlab.control.lth.se/control-frb/robotlib>.
- Bittencourt, A. C. and S. Gunnarsson (2012). "Static friction in a robot joint - modeling and identification of load and temperature effects". *Journal of Dynamic Systems, Measurement, and Control* **134**:5.
- Bugmann, G. (1998). "Normalized gaussian radial basis function networks". *Neurocomputing* **20**:13, pp. 97–110. ISSN: 0925-2312. DOI: [http://dx.doi.org/10.1016/S0925-2312\(98\)00027-7](http://dx.doi.org/10.1016/S0925-2312(98)00027-7). URL: <http://www.sciencedirect.com/science/article/pii/S0925231298000277>.
- Dahl, P. (1968). *A solid friction model*. Tech. rep. DTIC.
- De Wit, C. C., H. Olsson, K. J. Åström, and P. Lischinsky (1995). "A new model for control of systems with friction". *Automatic Control, IEEE Trans. on* **40**:3, pp. 419–425.
- Golub, G. H. and C. F. Van Loan (2012). *Matrix computations*. Vol. 3. Johns Hopkins University Press, Baltimore.
- Huang, P.-Y., Y.-Y. Chen, and M.-S. Chen (1998). "Position-dependent friction compensation for ballscrew tables". In: *Control Applications, 1998. Proc. 1998 IEEE Int. Conf., Trieste, Italy*. Vol. 2, pp. 863–867.
- Johansson, R. (1993). *System modeling & identification*. Prentice-Hall, Englewood Cliffs, NJ.
- Kruif, B. J. de and T. J. de Vries (2002). "Support-vector-based least squares for learning non-linear dynamics". In: *Decision and Control, 2002, Proc. IEEE Conf., Las Vegas*. Vol. 2, pp. 1343–1348.
- Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press, Cambridge, Massachusetts.
- Olsson, H., K. J. Åström, C. C. de Wit, M. Gäfvert, and P. Lischinsky (1998). "Friction models and friction compensation". *European Journal of Control* **4**:3, pp. 176–195.
- Rugh, W. J. (1996). *Linear system theory*. Prentice-Hall, Englewood Cliffs.
- Spong, M. W., S. Hutchinson, and M. Vidyasagar (2006). *Robot modeling and control*. Vol. 3. Wiley, New York.
- Wittenmark, B., K. J. Åström, and K.-E. Årzén (2002). "Computer control: an overview". *IFAC Professional Brief*.

Paper II

Linear Parameter-Varying Spectral Decomposition

Fredrik Bagge Carlson Anders Robertsson Rolf Johansson

Abstract

We develop a linear parameter-varying (LPV) spectral decomposition method, based on least-squares estimation and kernel expansions. Statistical properties of the estimator are analyzed and verified in simulations. The method is linear in the parameters, applicable to both the analysis and modeling problems and is demonstrated on both simulated signals as well as measurements of the torque in an electrical motor.

Originally published in The 2017 American Control Conference. Reprinted with permission. An open-source implementation of the algorithm presented is provided in [Bagge Carlson, 2016].

1. Introduction

Standard spectral density estimations techniques such as the discrete Fourier transform (DFT) exhibit several well-known limitations. These methods are typically designed for data sampled equidistantly in time or space. Whenever this property fails to hold, typical approaches employ some interpolation technique in order to perform spectral estimation on equidistantly sampled data. Other possibilities include employing a method suitable for non-equidistant data, such as least-squares spectral analysis [Wells et al., 1985]. Fourier transform-based methods further suffer from spectral leakage due to the assumption that all sinusoidal basis functions are orthogonal over the data window [Puryear et al., 2012]. Least-squares spectral estimation takes the correlation of the basis functions into account and further allows for estimation of arbitrary/known frequencies without modification [Wells et al., 1985].

In some applications, the spectral content is varying with an external variable, for instance, a controlled input. As motivating example, we consider the torque ripple induced by the rotation of an electrical motor. Spectral analysis of the torque signal is made difficult by equidistant, time-based sampling, which causes the spectrum to vary with the velocity of the motor, both due to the frequency of the ripple being directly proportional to the velocity, but also due to the properties of an electric DC-motor. A higher velocity both induces higher magnitude torque ripple, but also a higher filtering effect due to the inertia of the rotating parts. The effect of a sampling delay on the phase of the measured ripple is similarly proportional to the velocity.

Time-frequency analysis traditionally employ windowing techniques [Johansson, 1993] in order to reduce spectral leakage [Harris, 1978; Stoica and Moses, 2005], mitigate effects of non-stationarity, reduce the influence of ill-posed auto-correlation estimates [Stoica and Moses, 2005] and allow for time-varying spectral estimates [Puryear et al., 2012]. The motivating example considers estimation of the spectral content of a signal which is periodic over the space of angular positions \mathcal{X} , with a spectral content varying with time solely due to the fact that the velocity is varying with time. Time does thus not hold any intrinsic meaning to the modification of the spectrum, and the traditional windowing in time is no longer essential.

This paper develops a spectral estimation technique using basis function expansions identified with the least-squares method, that allows the spectral properties (phase and amplitude) of the analyzed signal to vary with an external signal. Apart from a standard spectrum, functional relationships between the scheduling signal and the amplitude and phase of each frequency will be identified.

2. LPV Spectral Decomposition

2.1 Basis function expansions

In order to decompose the spectrum along an external dimension, we consider basis function expansions. Intuitively, a basis function expansion decomposes an intricate function or signal as a linear combination of simple basis functions. The Fourier transform can be given this interpretation, where an arbitrary signal is decomposed as a sum of complex-valued sinusoids, similarly, a stair function can be decomposed as a sum of step functions. With this intuition, we aim for a method which allow decomposition of the spectrum of a signal along an external dimension, in LPV terminology called the scheduling dimension, \mathcal{V} . If we consider a single sinusoid in the spectrum, the functional dependence decomposed by the basis function expansion will thus be the complex-valued coefficient k in $ke^{i\omega}$ as a function of the scheduling variable, v , which in the motivating example is the angular velocity of the motor. Using complex valued calculations, we simultaneously model the dependence of both amplitude and phase of a real frequency by considering the complex frequency.

Radial Basis Functions (RBFs) have been widely used in nonlinear modeling through RBF expansions or RBF networks [Murphy, 2012]. The motivation for considering RBFs as opposed to other basis functions include their, in practice, local support, which often make the modeling more intuitive and the result easier to interpret. This in contrast to basis functions with global support, such as sigmoid-type functions. Another motivation for the use of RBFs is the implicit assumption that the underlying functional dependence is smooth. The method proposed in this paper is not limited to the use of RBFs as basis functions, and extend without difficulty to other basis functions when motivated. A typical set of RBFs is shown in Fig. 1.

2.2 Least-squares identification of periodic signals

This paper deals with estimations of models which are linear in the parameters, and can thus be written on the form

$$y = \mathbf{A}k \quad (1)$$

where \mathbf{A} denotes a regressor matrix and k denotes a vector of coefficients to be identified. Models on the form (1) are commonly identified with the well-known least-squares procedure [Johansson, 1993]. For the model $y(n) = k_1 \sin(\omega n) + k_2 \cos(\omega n)$, this amounts to arranging the data according to

$$y = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \sin(\omega 1) & \cos(\omega 1) \\ \vdots & \vdots \\ \sin(\omega N) & \cos(\omega N) \end{bmatrix} \in \mathbb{R}^{N \times 2}, \quad k = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

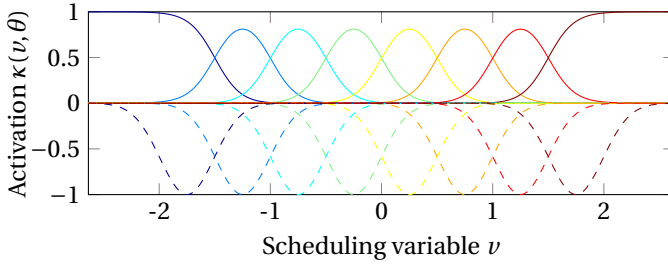


Figure 1. Gaussian (dashed) and normalized Gaussian (solid) windows. Regular windows are shown mirrored in the x -axis for clarity.

and solving the optimization problem¹ of Eq. (2) with solution (3).

$$k^* = \underset{k}{\operatorname{argmin}} \|\mathbf{A}k - y\| \quad (2)$$

$$= (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T y \quad (3)$$

This can be written in compact form by noting that $e^{i\omega} = \cos \omega + i \sin \omega$, which will be used extensively throughout the paper to simplify notation.²

We will now proceed to formalize a method for spectral decomposition using a perspective based on basis function expansions.

2.3 Model

We start by establishing some notation. Let k denote the Fourier series coefficients of interest. The kernel activation vector $\phi(v_n) : (v \in \mathcal{V}) \rightarrow \mathbb{R}^K$ maps the input to a set of basis function activations and is given by

$$\phi(v_n) = [\kappa(v_n, \theta_1) \quad \cdots \quad \kappa(v_n, \theta_J)]^T \in \mathbb{R}^J \quad (4)$$

$$\kappa(v, \theta_j) = \kappa_j(v) = \exp\left(-\frac{(v - \mu_j)^2}{2\sigma_j^2}\right) \quad (5)$$

where κ is a basis function parameterized by $\theta_j = (\mu_j, \sigma_j)$, $\mu \in \mathcal{V}$ is the center of the kernel and σ^2 is determining the width.

Let y denote the signal to be decomposed and denote the location of the sampling of y_n by $x_n \in \mathcal{X}$. The space \mathcal{X} is commonly time or space, in the motivating example of the electrical motor, \mathcal{X} is the space of motor positions.³ Let the

¹ This problem can easily be solved also for other norms or convex loss functions.

² Note that solving the complex LS problem using complex regressors $e^{i\omega}$ is not equivalent to solving the real LS problem using sin/cos regressors.

³ We note at this stage that $x \in \mathcal{X}$ can be arbitrarily sampled and are not restricted to lie on an equidistant grid, as is the case for, e.g., Fourier transform-based methods.

intensities of a set of complex frequencies $i\omega \forall \omega \in \Omega$ be given by basis function expansions along \mathcal{V} , according to

$$\hat{y}_n = \sum_{\omega \in \Omega} \sum_{j=1}^J k_{\omega,j} \kappa_j(v_n) e^{-i\omega x_n} \quad (6)$$

$$= \sum_{\omega \in \Omega} k_{\omega}^T \phi(v_n) e^{-i\omega x_n}, \quad k_{\omega} \in \mathbb{C}^J \quad (7)$$

The complex coefficients to be estimated, $k \in \mathbb{C}^{O \times J}$, $O = \text{card}(\Omega)$, constitute the Fourier series coefficients, with intensities split over \mathcal{V} . This formulation reduces to the standard Fourier style spectral relation (8) in the case $\phi(v) \equiv 1$

$$\hat{y} = \sum_{\omega \in \Omega} k_{\omega} e^{-i\omega x} = \Phi k \quad (8)$$

where $\Phi = [e^{-i\omega_1 x} \dots e^{-i\omega_O x}]$. If the number J of basis functions equals the number of data points N , the model will exactly interpolate the signal, i.e., $\hat{y} = y$. If in addition to $J = N$, the basis function centers are placed at $\mu_j = v_j$, we obtain a Gaussian process regression interpretation where κ is the covariance function. Due to the numerical properties of the analytical solution of the least-squares problem, it is often beneficial to reduce the number of parameters significantly, so that $J \ll N$. If the chosen basis functions are suitable for the signal of interest, the error induced by this dimensionality reduction is small. In a particular case, the number of RBFs to include, J , and the bandwidth Σ is usually chosen based on evidence maximization or cross validation [Murphy, 2012].

To facilitate estimation of the parameters in Eq. (6), we rewrite the model by stacking the regressor vectors in a regressor matrix A , see Sec. 2.2, such that

$$\mathbf{A}_{n,:} = \text{vec}(\phi(v_n)\Phi^T)^T \in \mathbb{C}^{O \times J}, n = 1 \dots N$$

We further define $\tilde{\mathbf{A}}$ by expanding the regressor matrix into its real and imaginary parts

$$\tilde{\mathbf{A}} = [\Re \mathbf{A} \quad \Im \mathbf{A}] \in \mathbb{R}^{N \times 2OJ}$$

such that routines for real-valued least-squares problems can be used. The complex coefficients are, after solving the real-valued problem (3), retrieved as $k = \Re k + i \Im k$ where

$$[\Re k^T \quad \Im k^T]^T = \arg \min_{\tilde{k}} \|\tilde{\mathbf{A}} \tilde{k} - y\|$$

Since the purpose of the decomposition is spectral analysis, it is important to normalize the basis function activations such that the total activation over \mathcal{V} for

each data point is unity, to this end, the expressions (6) and (7) are modified to

$$\begin{aligned}\hat{y} &= \sum_{\omega \in \Omega} \sum_{j=1}^J k_{\omega,j} \bar{\kappa}_j(v) e^{-i\omega x} \\ &= \sum_{\omega \in \Omega} k_{\omega}^T \bar{\phi}(v) e^{-i\omega x} \\ \bar{\kappa}_j(v) &= \frac{\kappa_j(v)}{\sum_j \kappa_j(v)}, \quad \bar{\phi}(v) = \frac{\phi(v)}{\sum \phi(v)}\end{aligned}\tag{9}$$

This ensures that the spectral content for a single frequency ω is a convex combination of contributions from each basis function in the scheduling dimension. Without this normalization, the power of the spectrum would be ill-defined and depend on an arbitrary scaling of the basis functions. The difference between a set of Gaussian functions and a set of normalized Gaussian functions is demonstrated in Fig. 1. The normalization performed in Eq. (9) can be viewed as the kernel function being made data adaptive by normalizing $\phi(v)$ to sum to one.

2.4 Amplitude and phase functions

In spectral analysis, two functions of the Fourier series coefficients are typically of interest, the amplitude and phase functions, which are easily obtained and are stated here without proof⁴:

PROPOSITION 1

Let a signal y be composed by the linear combination $y = k_1 \cos(x) + k_2 \sin(x)$, then y can be written on the form

$$y = A \cos(x - \varphi)$$

with

$$A = \sqrt{k_1^2 + k_2^2} \quad \varphi = \arctan\left(\frac{k_2}{k_1}\right) \quad \square$$

From this we obtain the following two functions for a single frequency ω

$$\begin{aligned}A(\omega) &= |k_{\omega}| = \sqrt{\Re k_{\omega}^2 + \Im k_{\omega}^2} \\ \varphi(\omega) &= \arg(k_{\omega}) = \arctan(\Im k_{\omega} / \Re k_{\omega})\end{aligned}$$

In the proposed spectral decomposition method, these functions are further dependent on v , and are approximated by

$$A(\omega, v) = \left| \sum_{j=1}^J k_{\omega,j} \bar{\kappa}_j(v) \right| = |k_{\omega}^T \bar{\phi}(v)| \tag{10}$$

$$\varphi(\omega, v) = \arg\left(\sum_{j=1}^J k_{\omega,j} \bar{\kappa}_j(v)\right) = \arg(k_{\omega}^T \bar{\phi}(v)) \tag{11}$$

⁴ The proof is added to this thesis in the article appendix.

2.5 Covariance properties

We will now investigate and prove that Eqs. Eq. (10) and Eq. (11) lead to asymptotically unbiased and consistent estimates of A and φ and will provide a strategy to obtain confidence intervals. We will initially consider a special case for which analysis is simple, whereafter we invoke the RBF universal approximation results of Park [Park and Sandberg, 1991] to show that the estimators are well motivated for a general class of functions. We start by considering signals on the form Eq. (12), for which unbiased and consistent estimates of the parameters are readily available:

PROPOSITION 2

Let a signal y be given by

$$\begin{aligned} y &= a(v) \cos(x) + b(v) \sin(x) + e \\ a(v) &= \alpha^\top \phi_v \\ b(v) &= \beta^\top \phi_v \\ e &\in \mathcal{N}(0, \sigma^2) \end{aligned} \quad (12)$$

with $\phi_v = \phi(v)$ and let $\hat{\alpha}$ and $\hat{\beta}$ denote unbiased estimates of α and β , then

$$\hat{A}(\hat{\alpha}, \hat{\beta}) = \sqrt{(\hat{\alpha}^\top \phi_v)^2 + (\hat{\beta}^\top \phi_v)^2} \quad (13)$$

is a biased estimate of A with

$$A < \mathbb{E} \{ \hat{A} \} < \sqrt{A^2 + \phi_v^\top \Sigma_\alpha \phi_v + \phi_v^\top \Sigma_\beta \phi_v} \quad (14)$$

Proof Since α , β and e appear linearly in Eq. (12), unbiased and consistent estimates $\hat{\alpha}$ and $\hat{\beta}$ are available from the least-squares procedure [Johansson, 1993]. The expected value of \hat{A}^2 is given by

$$\begin{aligned} \mathbb{E} \{ \hat{A}^2 \} &= \mathbb{E} \{ (\hat{\alpha}^\top \phi_v)^2 + (\hat{\beta}^\top \phi_v)^2 \} \\ &= \mathbb{E} \{ (\hat{\alpha}^\top \phi_v)^2 \} + \mathbb{E} \{ (\hat{\beta}^\top \phi_v)^2 \} \end{aligned} \quad (15)$$

We further have

$$\mathbb{E} \{ (\hat{\alpha}^\top \phi_v)^2 \} = \mathbb{E} \{ \hat{\alpha}^\top \phi_v \}^2 + \mathbb{V} \{ \hat{\alpha}^\top \phi_v \} = (\alpha^\top \phi_v)^2 + \phi_v^\top \Sigma_\alpha \phi_v \quad (16)$$

where Σ_α and Σ_β are the covariance matrices of $\hat{\alpha}$ and $\hat{\beta}$ respectively. Calculations for β are analogous. From (15) and (16) we deduce

$$\begin{aligned} \mathbb{E} \{ \hat{A}^2 \} &= (\alpha^\top \phi_v)^2 + (\beta^\top \phi_v)^2 + \phi_v^\top \Sigma_\alpha \phi_v + \phi_v^\top \Sigma_\beta \phi_v \\ &= A^2 + \phi_v^\top \Sigma_\alpha \phi_v + \phi_v^\top \Sigma_\beta \phi_v \end{aligned} \quad (17)$$

Now, due to Jensen's inequality, we have

$$\mathbb{E}\{\hat{A}\} = \mathbb{E}\left\{\sqrt{\hat{A}^2}\right\} < \sqrt{\mathbb{E}\{\hat{A}^2\}} \quad (18)$$

which provides the upper bound on the expectation of \hat{A} . The lower bound is obtained by writing \hat{A} on the form

$$\hat{A}(k) = \sqrt{(\hat{\alpha}^\top \phi_v)^2 + (\hat{\beta}^\top \phi_v)^2} = \|\hat{k}\| \quad (19)$$

with $\hat{k} = [\hat{\alpha}^\top \phi_v \quad \hat{\beta}^\top \phi_v]$. From Jensen's inequality we have

$$\mathbb{E}\{\hat{A}\} = \mathbb{E}\{\|\hat{k}\|\} > \|\mathbb{E}\{\hat{k}\}\| = \|k\| = A \quad (20)$$

which concludes the proof. \square

COROLLARY 1

$$\hat{A} = \sqrt{(\hat{\alpha}^\top \phi_v)^2 + (\hat{\beta}^\top \phi_v)^2} \quad (21)$$

is an asymptotically unbiased and consistent estimate of A .

Proof Since the least-squares estimate upon which the estimated quantity is based, is unbiased and consistent, the variances in the upper bound in Eq. (14) will shrink as the number of datapoints increases and both the upper and lower bounds will become tight, hence

$$\mathbb{E}\{\hat{A}\} \rightarrow A \quad \text{as } N \rightarrow \infty \quad \square$$

Analogous bounds for the phase function are harder to obtain, but the simple estimator $\hat{\varphi} = \arg \hat{k}$ based on \hat{k} obtained from the least-squares procedure is still asymptotically consistent [Kay, 1993].

Estimates using the least-squares method (3) are, under the assumption of uncorrelated Gaussian residuals of variance σ^2 , associated with a posterior parameter covariance $\sigma^2(\mathbf{A}^\top \mathbf{A})^{-1}$. This will in a straightforward manner produce confidence intervals for a future prediction of y as a linear combination of the estimated parameters. Obtaining unbiased estimates of the confidence intervals for the functions $A(v, \omega)$ and $\varphi(v, \omega)$ is made difficult by their nonlinear nature. We therefore proceed to establish an approximation strategy.

The estimated parameters k are distributed according to a complex-normal distribution $\mathcal{CN}(x + iy, \Gamma, C)$, where Γ and C are obtained through

$$\begin{aligned} \Gamma &= \Sigma_{xx} + \Sigma_{yy} + i(\Sigma_{yx} - \Sigma_{xy}) \\ C &= \Sigma_{xx} - \Sigma_{yy} + i(\Sigma_{yx} + \Sigma_{xy}) \\ \Sigma &= \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} = \sigma^2(\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}})^{-1} \end{aligned} \quad (22)$$

For details on the \mathcal{CN} -distribution, see, e.g., [Picinbono, 1996]. A linear combination of squared variables distributed according to a complex normal (\mathcal{CN}) distribution, is distributed according to a generalized χ^2 distribution, a special case of the gamma distribution. Expressions for sums of dependent gamma-distributed variables exist, see, e.g., [Paris, 2011], but no expressions for the distribution of linear combinations of norms of Gaussian vectors, e.g., Eq. (10), are known to the authors. In order to establish estimates of confidence bounds on the spectral functions, one is therefore left with high-dimensional integration or Monte-Carlo techniques. Monte-Carlo estimates will be used in the results presented in this paper. The sampling from a \mathcal{CN} -distribution is outlined in Proposition 3:

PROPOSITION 3

The vector

$$z = \tilde{x} + i\tilde{y} \in \mathbb{C}^D$$

where

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = L \begin{bmatrix} x \\ y \end{bmatrix}, \quad x, y \sim \mathcal{N}(0, I) \in \mathbb{R}^D$$

and

$$\Sigma = LL^\top$$

is a Cholesky decomposition of the matrix

$$\Sigma = \frac{1}{2} \begin{bmatrix} \Re(\Gamma + C) & \Im(-\Gamma + C) \\ \Im(\Gamma + C) & \Re(\Gamma - C) \end{bmatrix} \in \mathbb{R}^{2D \times 2D}$$

is a sample from the complex normal distribution $\mathcal{CN}(0, \Gamma, C)$.

Proof Proof is given in the appendix. \square

By sampling from the posterior distribution $p(k_\omega|y)$ and propagating the samples through the non-linear functions $A(\omega, v)$ and $\varphi(\omega, v)$, estimates of relevant confidence intervals are easily obtained.

The quality of the estimate thus hinges on the ability of the basis function expansion to approximate the given functions a and b in Eq. (12). Park [Park and Sandberg, 1991] provides us with the required result that establishes RBF expansions as a universal function approximator.⁵

2.6 Comparison to related methods

The act of performing a basis function expansion in \mathcal{V} could be compared to performing windowing along \mathcal{V} with a Gaussian window, with the added constraint $\sum_j \kappa_j(v) = 1$ imposed by the formulation

$$\frac{\kappa_j(v)}{\sum_j \kappa_j(v)} \tag{23}$$

which implies an adaptation of the window to the windows surrounding it.

⁵ For well behaved functions.

3. Experimental Results

3.1 Simulated signals

To assess the qualities of the proposed spectral decomposition method, a test signal y_t is generated as follows

$$\begin{aligned} y_t &= \sum_{\omega \in \Omega} A(\omega, v) \cos(\omega x - \varphi(\omega, v)) + e \\ v_t &= \text{linspace}(0, 1, N) \\ x &= \text{sort}(\mathcal{U}(0, 10)) \\ e &\in \mathcal{N}(0, 0.1^2) \end{aligned} \quad (24)$$

where $\Omega = \{4\pi, 20\pi, 40\pi\}$, the scheduling variable v_t is generated as $N = 500$ equidistantly sampled points between 0 and 1 and x is a sorted vector of uniform random numbers. The sorting is carried out for visualization purposes and for the Fourier based methods to work, this property is not a requirement for the proposed method to work. The functions A and φ are defined as follows

$$\begin{aligned} A(4\pi, v) &= 2v^2 \\ A(20\pi, v) &= 2/(5v + 1) \\ A(40\pi, v) &= 3e^{-10(v-0.5)^2} \\ \varphi(\omega, v) &= 0.5A(\omega, v) \end{aligned} \quad (25)$$

where the constants are chosen to allow for convenient visualization. The signals y_t and v_t are visualized as functions of the sampling points x in Fig. 2 and the functions A and φ together with the resulting estimates and confidence intervals using $J = 50$ basis functions are shown in Fig. 2. The traditional power spectral density can be calculated from the estimated coefficients as

$$P(\omega) = \left| \sum_{j=1}^J k_{\omega,j} \right|^2 \quad (26)$$

and is compared to the periodogram and Welch spectral estimates in Fig. 3. This figure illustrates how the periodogram and Welch methods fail to clearly identify the frequencies present in the signal due to the dependence on the scheduling variable v . The LPV spectral method, however, correctly identifies all three frequencies present.

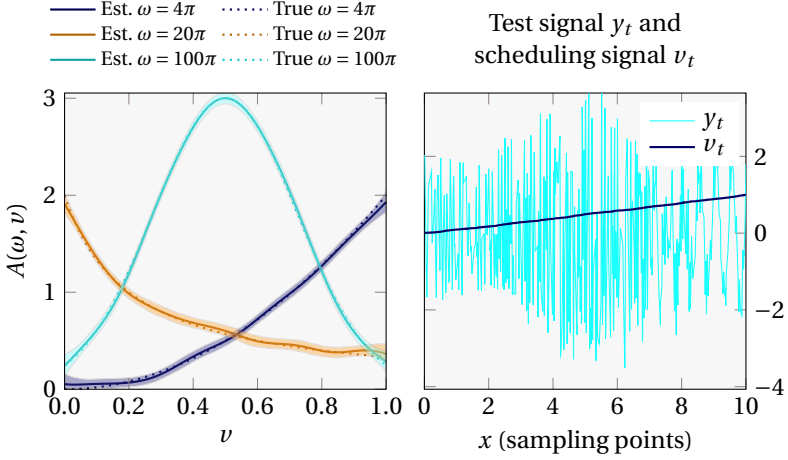


Figure 2. Left: True and estimated functional dependencies with 95% confidence intervals. Right: Test signal with $N = 500$ datapoints. The signal contains three frequencies, where the amplitude and phase are modulated by the functions (25) depicted in the left panel.

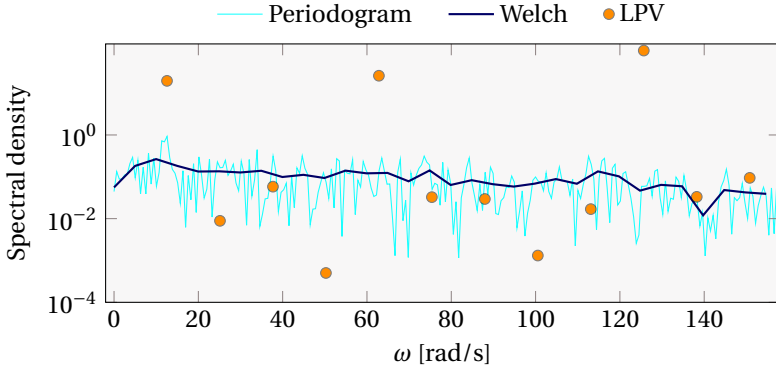


Figure 3. Estimated spectra, test signal. The periodogram and Welch methods fail to identify the frequencies present in the signal due to the dependence on the scheduling variable ν . The LPV spectral method correctly identifies all three frequencies present.

3.2 Measured signals

The proposed method was used to analyze measurements obtained from an ABB YuMi robot. Due to torque ripple and other disturbances, there is a velocity dependent periodic signal present in the velocity control error, which will serve as the subject of analysis. The analyzed signal is shown in Fig. 4.

The influence of Coulomb friction on the measured signal is mitigated by limiting the support of half of the basis functions to positive velocities and vice versa. A total number of 10 basis functions was used and the model was identified with ridge regression. The regularization parameter was chosen using the L-curve method [Hansen, 1994]. The identified spectrum is depicted in Fig. 5, where the dominant frequencies are identified. These frequencies correspond well with a visual inspection of the data. Figure 5 further illustrates the result of applying the periodogram and Welch spectral estimators to data that has been sorted and interpolated to an equidistant grid. These methods correctly identify the main frequency, 4 rev^{-1} , but fail to identify the lower amplitude frequencies at 7 rev^{-1} and 9 rev^{-1} visible in the signal. The amplitude functions for three strongest frequencies are illustrated in Fig. 6, where it is clear that the strongest frequency, 4 rev^{-1} , has most of its power distributed over the lower velocity datapoints, whereas the results indicate a slight contribution of frequencies at 7 rev^{-1} and 9 rev^{-1} at higher velocities, corresponding well with a visual inspection of the signal. Figure 6 also displays a histogram of the velocity values of the analyzed data. The confidence intervals are narrow for velocities present in the data, while they become wider outside the represented velocities.

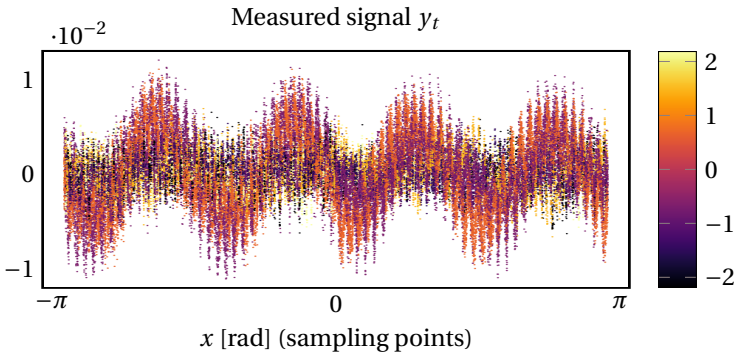


Figure 4. Measured signal as a function of sampling location, i.e., motor position. The color information indicates the value of the velocity/scheduling variable in each datapoint. Please note this is not a plot of the measured data sequentially in time. This figure indicates that there is a high amplitude periodicity of 4 rev^{-1} for low velocities, and slightly higher frequencies but lower amplitude signals at 7 rev^{-1} and 9 rev^{-1} for higher velocities.

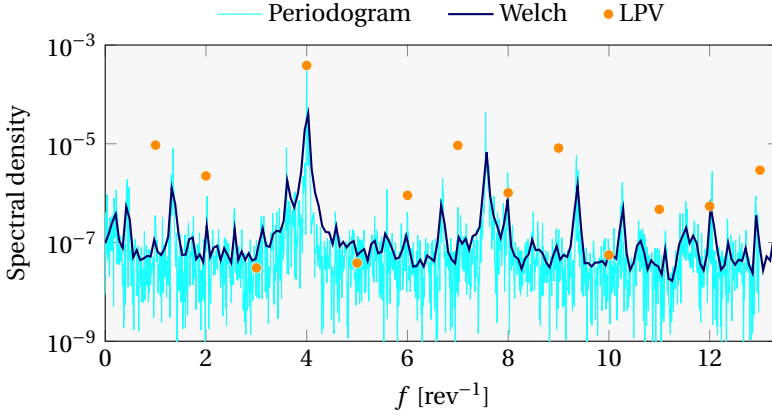


Figure 5. Estimated spectra, measured signal. The dominant frequencies are identified by the proposed method, while the Fourier based methods correctly identify the main frequency, 4 rev^{-1} , but fail to identify the lower amplitude frequencies at 7 rev^{-1} and 9 rev^{-1} visible in the signal in Fig. 4.

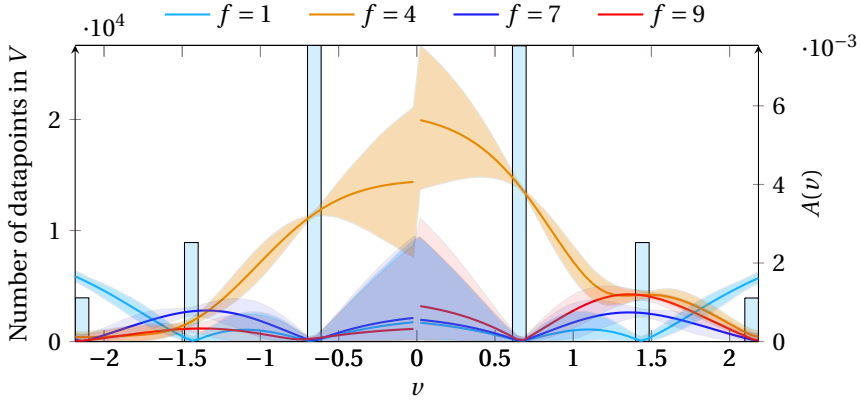


Figure 6. Estimated functional dependences with 99% confidence intervals. The left axis and histogram illustrates the number of datapoints available at each velocity v . The right axis illustrate the estimated amplitude functions together with their confidence intervals.

4. Conclusions

We have developed a spectral estimation method that can decompose the spectrum of a signal along an external dimension, which allows estimation of the amplitude and phase of the sinusoids as functions of the external variable. The method is linear in the parameters which allows for straight-forward calculation of the spectrum through solving a set of linear equations. The method does not impose limitations such as equidistant sampling, does not suffer from leakage and allows for estimation of arbitrary chosen frequencies. The closed-form calculation of the spectrum requires $\mathcal{O}(J^3 O^3)$ operations due to the matrix inversion associated with solving the LS-problem, which serve as the main drawback of the method if the number of frequencies to estimate is large (the product JO greater than a few thousands).

A. Proofs

Proof Proposition 3

Let $v^\top = [x^\top \ y^\top]$. The mean and variance of $\tilde{v} = Lv$ is given by

$$\begin{aligned}\mathbb{E}\{\tilde{v}\} &= L\mathbb{E}\{v\} = 0 \\ \mathbb{E}\{\tilde{v}\tilde{v}^\top\} &= \mathbb{E}\{Lv v^\top L^\top\} = LIL^\top = \Sigma\end{aligned}$$

The complex vector $z = x + iy \in \mathbb{C}^D$ composed of the elements of v is then $\mathcal{CN}(0, \Gamma, C)$ -distributed according to [Picinbono, 1996, Proposition 1]. \square

B. Additional proofs not part of the original article

Proof Proposition 1

The amplitude A is given by two trigonometric identities

$$A \cos(x - \varphi) = A \cos(\varphi) \cos(x) + A \sin(\varphi) \sin(x) \quad (27)$$

$$= k_1 \cos(x) + k_2 \sin(x) \quad (28)$$

$$k_1^2 + k_2^2 = A^2(\cos(\varphi)^2 + \sin(\varphi)^2) = A^2 \quad (29)$$

and the phase φ by

$$\arctan\left(\frac{k_2}{k_1}\right) = \arctan\left(\frac{A \sin \varphi}{A \cos \varphi}\right) = \varphi$$

where $k_1 = A \cos(\varphi)$, $k_2 = A \sin(\varphi)$ is identified from (27). \square

References

- Bagge Carlson, F. (2016). *Lpvspectral.jl*. Dept. Automatic Control. URL: <https://github.com/baggepinnen/LPVspectral.jl>.
- Hansen, P. C. (1994). “Regularization tools: a matlab package for analysis and solution of discrete ill-posed problems”. *Numerical algorithms* **6**:1, pp. 1–35.
- Harris, F. J. (1978). “On the use of windows for harmonic analysis with the discrete fourier transform”. *Proceedings of the IEEE* **66**:1, pp. 51–83.
- Johansson, R. (1993). *System modeling & identification*. Prentice-Hall, Englewood Cliffs, NJ.
- Kay, S. M. (1993). *Fundamentals of statistical signal processing, volume I: estimation theory*. Prentice Hall, Englewood Cliffs, NJ.
- Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press, Cambridge, Massachusetts.
- Paris, J. F. (2011). “A note on the sum of correlated gamma random variables”. *CoRR abs/1103.0505*. URL: <http://arxiv.org/abs/1103.0505>.
- Park, J. and I. W. Sandberg (1991). “Universal approximation using radial-basis-function networks”. *Neural computation* **3**:2, pp. 246–257.
- Picinbono, B. (1996). “Second-order complex random vectors and normal distributions”. *IEEE Transactions on Signal Processing* **44**:10, pp. 2637–2640.
- Puryear, C. I., O. N. Portniaguine, C. M. Cobos, and J. P. Castagna (2012). “Constrained least-squares spectral analysis: application to seismic data”. *Geophysics* **77**:5, pp. V143–V167.
- Stoica, P. and R. L. Moses (2005). *Spectral analysis of signals*. Pearson/Prentice Hall Upper Saddle River, NJ.
- Wells, D. E., P. Vaníček, and S. D. Pagiatakis (1985). *Least squares spectral analysis revisited*. Department of Surveying Engineering, University of New Brunswick Fredericton, Canada.

Paper III

Six DOF Eye-to-Hand Calibration from 2D Measurements Using Planar Constraints

Fredrik Bagge Carlson Rolf Johansson Anders Robertsson

Abstract

This article presents a linear, iterative method to solve the eye-to-hand calibration problem between a wrist-mounted laser scanner and the tool flange of a robot. Measurement data are acquired from a set of non parallel planes whereafter the plane equations and desired rigid transformation matrix are found in a two-step, iterative fashion. The method is shown to handle large error in the initial estimate of the transform and results are verified in both simulations and experiments using a seam tracking laser sensor for welding applications.

Originally published in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Reprinted with permission. An open-source implementation of the algorithm presented is provided in [Bagge Carlson, 2015].

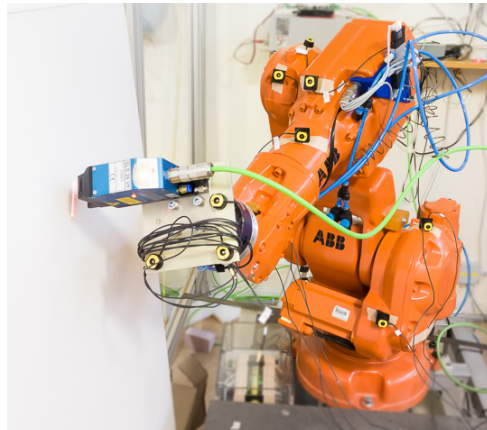


Figure 1. ABB IRB140 used for experimental verification.

1. Introduction

Laser scanners have been widely used for many years in the field of robotics. A large group of laser scanners, such as 2D laser range finders and laser stripe profilers, provide accurate distance measurements confined to a plane. By moving either the scanner or the scanned object, a 2D laser scanner can be used to build a 3D representation of an object or the environment. To this purpose, laser scanners are commonly mounted on mobile platforms or robots.

This work considers the calibration of a wrist mounted laser scanner for robotic 3D scanning and weld seam tracking applications. To relate the measurements of the scanner to the robot coordinate system, the rigid transformation between the scanner coordinate system and the tool flange of the robot is needed.

A naive approach to the stated calibration problem is to make use of the 4/5/6-point tool calibration routines commonly found in industrial robot systems. These methods suffer from the fact that the origin and the axes of the sensor coordinate system are invisible to the operator, which must rely on visual feedback from both the workspace and a computer monitor simultaneously. Further, the accuracy of these methods is very much dependent on the skill of the operator and data collection for even a small amount of points is very tedious.

Other well known algorithms for eye-to-hand calibration include [Daniilidis, 1999; Tsai and Lenz, 1989; Horaud and Dornaika, 1995], which are all adopted for calibration of a wrist-mounted camera using a calibration pattern. A laser scanner is fundamentally different in the information it captures, which must be considered by the calibration algorithm employed.

Kinematic calibration of robotic manipulator using planar constraints in various formats has been considered before. In [Zhuang et al., 1999], the proposed method begins with an initial estimate of the desired parameters, which is improved with a non-linear optimization algorithm. The authors also discuss ob-

servability issues related to identification using planar constraints. The method focuses on improving parameter estimates in the kinematic model of the robot, and convergence results are therefore only presented for initial guesses very close to their true values (0.01mm/0.01°).

In [Zhang and Pless, 2004], the transformation between a camera and a laser range finder is found using a checker board pattern and computer vision to estimate the location of the calibration planes. With the equations of the calibration planes known, the desired transformation matrix is obtained from a set of linear equations.

Planar constraints have also been considered in [Ikits and Hollerbach, 1997] where the authors employ a non-linear optimization technique to estimate the kinematic parameters. The method requires careful definition of the planes and can not handle arbitrary frame assignments.

A wrist mounted sensor can be seen as an extension of the kinematic chain of the robot. Initial guesses can be poor, especially if based on visual estimates. This paper presents a method based solely on solving linear sets of equations. The method accepts a very crude initial estimate of the desired kinematic parameters, which is refined in an iterative procedure. The placement of the calibration planes is assumed unknown, and their locations are found together with the desired transformation matrix.

The article is structured as follows, preliminary equations and notation are covered in Sec. 2 followed by the introduction of the proposed approach in Sec. 3. Section 4 presents a simulation study of convergence properties as well as experimental verification of the approach. Conclusions are finally given in Sec. 5.

2. Preliminaries

Throughout the paper the following notation will be used. A subscript denotes the frame of reference, such that the coordinates of a point p_A are given in the frame A . $T_A^B \in SE(4)$ denotes a transformation matrix from frame A to frame B such that [Hartenberg and Denavit, 1964]

$$T_A^B \mathbf{p}_B = \mathbf{p}_A \quad (1)$$

The matrix T_A^B can be decomposed into R_A^B and p_A^B such that

$$T_A^B = \begin{bmatrix} R_A^B & p_A^B \\ \mathbf{0} & 1 \end{bmatrix} \quad (2)$$

The normal of a plane from which measurement point i is taken, given in frame A , will be denoted n_A^i .

A plane is completely specified by

$$n^T p = d, \quad \|n\|_2 = 1 \quad (3)$$

where d is the orthogonal distance from the origin to the plane, n the plane normal and p is any point on the plane.

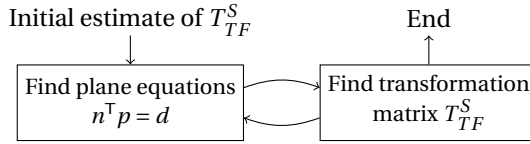


Figure 2. Illustration of the two-step, iterative method.

2.1 Laser scanner characteristics

The laser scanner consists of a camera and a laser source emitting light in a plane which intersects a physical plane in a line. The three dimensional location of a point along the projected laser line may be calculated by triangulation, based on a known geometry between the camera and the laser emitter. A single measurement from the laser scanner typically yields the coordinates of a large number of points in the laser plane, alternatively, a measurement consists of a single point and the angle of the surface, which is easily converted to two points.

3. Method

The objective of the calibration is to find the transformation matrix $T_{TF}^S \in SE(4)$ that relates the measurements of the laser scanner to the coordinate frame of the tool flange of the robot.

The kinematic chain of a robot manipulator will here consist of the transformation from the robot base frame to the tool flange $T_{RB}^{TF_i}$, given by the manipulator forward kinematics in pose i , and the transformation from the tool flange to the sensor T_{TF}^S . The sensor, in turn, projects laser light onto a plane with unknown equation. A point observed by the sensor can be translated to the robot base frame by

$$\mathbf{p}_{RB_i} = T_{RB}^{TF_i} T_{TF}^S \mathbf{p}_{S_i} \quad (4)$$

where i denotes the index of the pose.

To find T_{TF}^S , an iterative two-step method is proposed, which starts with an initial guess of the matrix. In each iteration, the equations for the planes are found using eigendecomposition, whereafter a set of linear equations is solved for an improved estimate of the desired transformation matrix. The scheme, illustrated in Fig. 2, is iterated until convergence.

3.1 Finding the calibration planes

Consider initially a set of measurements, $\mathcal{P}_S = [p_1, \dots, p_{N_p}]_S$, gathered from a single plane. The normal can be found by Principal Component Analysis (PCA), which amounts to performing an eigendecomposition of the covariance matrix C of the points [Pearson, 1901]. The eigenvector corresponding to the smallest eigenvalue of C will be the desired estimate of the plane normal.¹ To this purpose, all points

¹ This eigenvalue will correspond to the mean squared distance from the points to the plane.

are transformed to a common frame, the robot base frame, using (4) and the current estimate of T_{TF}^S .

To fully specify the plane equation, the center of mass μ of \mathcal{P}_{RB} is calculated. The distance d to the plane is then calculated as the length of the projection of the vector μ onto the plane normal

$$d = \|\bar{n}(\bar{n}^\top \mu)\| \quad (5)$$

where \bar{n} is a normal with unit length given by PCA. This distance can be encoded into the normal by letting $\|n\| = d$. The normal is then simply found by

$$n = \bar{n}(\bar{n}^\top \mu) \quad (6)$$

This procedure is repeated for all measured calibration planes and results in a set of normals \mathcal{N} that will be used to find the optimal T_{TF}^S .

3.2 Solving for T_{TF}^S

All measured points should fulfill the equation for the plane they were obtained from. This means that for a single point p

$$\bar{n}^\top p = d \iff n^\top p = \|n\|^2 \quad (7)$$

A measurement point obtained from the sensor in the considered setup should thus fulfill the following set of linear equations

$$\mathbf{p}_{RB_i} = T_{RB}^{TF_i} T_{TF}^S \mathbf{p}_{S_i} \quad (8)$$

$$n^\top \mathbf{p}_{RB_i} = \|n\|^2 \quad (9)$$

$$\mathbf{p}_{S_i} = [p_{S_i}^\top \quad 1]^\top = [x_{S_i} \quad y_{S_i} \quad z_{S_i} \quad 1]^\top \quad (10)$$

where bold-face notation denotes a point expressed in homogeneous coordinates according to (10). Without loss of generality, the points p_S can be assumed to lie on the plane $z_S = 0$. As a result, the third column in T_{TF}^S can not be solved for directly. The constraints on R_{TF}^S to belong to $SO(3)$, will however allow for reconstruction of the third column in R_{TF}^S from the first two columns.

Let \tilde{T} denote the remainder of T_{TF}^S after removing the third column and the last row. Solving the linear equations (8)-(9) for the parameters in \tilde{T} can be expressed as

$$A_i k + q_i = \|n_i\| \iff A_i k = \|n_i\| - q_i \quad (11)$$

where $k = \text{vec}(\tilde{T}) \in \mathbb{R}^{9 \times 1}$ consists of the stacked columns of \tilde{T} and

$$A_i = [n_i^\top R_{RB}^{TF_i} x_{S_i} \quad n_i^\top R_{RB}^{TF_i} y_{S_i} \quad n_i^\top R_{RB}^{TF_i}] \in \mathbb{R}^{1 \times 9} \quad (12)$$

$$q_i = n_i^\top p_{RB}^{TF_i} \in \mathbb{R} \quad (13)$$

Since (8) and (9) are linear in the parameters, all elements of T_{TF}^S can be extracted into k and A_i can be obtained by performing the matrix multiplications in (8) and (9) and identifying terms containing any of the elements of k . Terms with which do not include any parameter to be identified are associated with q_i . The final expressions for A_i and q_i given above can then be obtained by identifying matrix multiplication structures among the elements of A_i and q_i .

Equation (11) does not have a unique solution. A set of at least nine points gathered from at least three planes is required in order to obtain a unique solution to the vector k . This can be obtained by stacking the entries in Eq. (11) according to

$$\mathbf{A}k = \mathbf{Y}, \quad \mathbf{A} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_{N_p} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \|n_1\| - q_1 \\ \|n_2\| - q_2 \\ \vdots \\ \|n_{N_p}\| - q_{N_p} \end{bmatrix} \quad (14)$$

The vector k^* of parameters that minimizes

$$k^* = \underset{k}{\operatorname{argmin}} \|\mathbf{A}k - \mathbf{Y}\|_2 \quad (15)$$

can then be obtained from the equation² [Rugh, 1996; Golub and Van Loan, 2012]

$$k^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Y} \quad (16)$$

Eq. (16) is known as the least-squares solution and the full-rank matrix $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ is commonly referred to as the pseudo inverse of \mathbf{A} . If \mathbf{A} is a square matrix, the pseudo inverse reduces to the standard matrix inverse. If \mathbf{A} however is a tall matrix, the equation $\mathbf{A}k = \mathbf{Y}$ is over determined and Eq. (16) produces the solution k^* that minimizes Eq. (15).

Since k only contains the first two columns of R_{TF}^S , the third column is formed as

$$R_3 = R_1 \times R_2 \quad (17)$$

where \times denotes the cross product between R_1 and R_2 , which produces a vector orthogonal to both R_1 and R_2 . The resulting R_{TF}^S will in general not belong to $SO(3)$. The closest valid rotation matrix can be found by Singular Value Decomposition according to [Eggert et al., 1997]

$$R = USV^T \quad (18)$$

$$R^\perp = U \begin{bmatrix} 1 & & \\ & 1 & \\ & & \det(UV^T) \end{bmatrix} V^T \quad (19)$$

² Commonly solved by the command $k = \mathbf{A} \backslash \mathbf{Y}$ in Julia and Matlab

or using the matrix square root³ as [Eggert et al., 1997]

$$R^\perp = R(R^\top R)^{-\frac{1}{2}} \quad (20)$$

The procedure of orthogonalizing R will change the corresponding entries in k^* and the resulting coefficients will no longer solve the problem (15). A second optimization problem can thus be formed to re-estimate the translational part of k , given the orthogonalized rotational part. Let k be decomposed according to

$$k = [\tilde{R}^* \quad p] \quad \tilde{R}^* \in \mathbb{R}^{1 \times 6}, \quad p \in \mathbb{R}^{1 \times 3} \quad (21)$$

and denote by $\mathbf{A}_{n:k}$ columns n to k of \mathbf{A} . The optimal translational vector, given the orthonormal rotation matrix, is found by solving the following optimization problem

$$\tilde{\mathbf{Y}} = \mathbf{Y} - \mathbf{A}_{1:6} \tilde{R}^* \quad (22)$$

$$p^* = \underset{p}{\operatorname{argmin}} \|\mathbf{A}_{7:9} p - \tilde{\mathbf{Y}}\|_2 \quad (23)$$

with the solution

$$p^* = (\mathbf{A}_{7:9}^\top \mathbf{A}_{7:9})^{-1} \mathbf{A}_{7:9}^\top \tilde{\mathbf{Y}} \quad (24)$$

3.3 Final Refinement

As noted in [Zhang and Pless, 2004], solving an optimization problem like (15) is equivalent to minimizing the algebraic distance between the matrix, parameterized by k , and the data. There is no direct minimization of the distances from measurements to planes involved. Given the result from the above procedure as initial guess, any suitable, iterative minimization strategy can be employed to further minimize a cost function on the form

$$J(T_{TF}^S) = \sum_{i=1}^{N_p} (n_i^\top p_{RB_i}(T_{TF}^S) - \|n_i\|)^2 \quad (25)$$

which is the squared distance from the measurement point to the plane. Here, p_{RB_i} is seen as a function of T_{TF}^S according to (8).

4. Results

The performance of the method was initially studied in simulations, Sec. 4.1, which allows for a comparison between the obtained estimate and the ground truth. The simulation study is followed by experimental verification, Sec. 4.2, using a real laser scanner mounted on the wrist of an industrial manipulator.

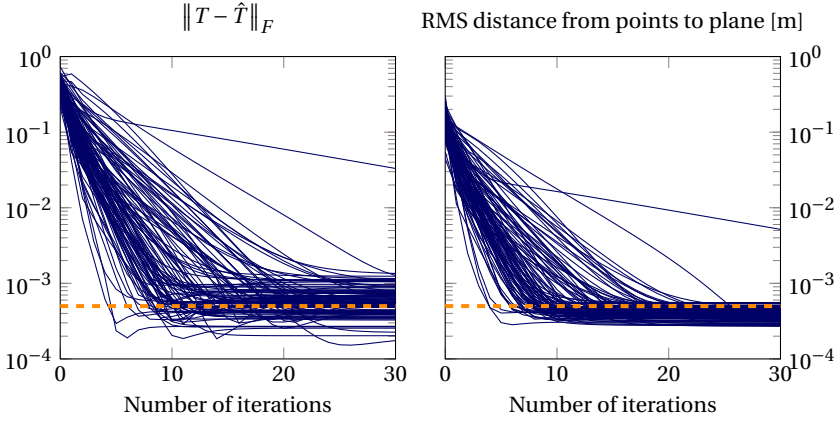


Figure 3. Convergence results for simulated data during 100 realizations. 30 poses in total on 3 planes. Measurement noise level $\sigma = 0.5\text{mm}$ is marked with a dashed line. On the left, the Frobenius norm between the true matrix and the estimated, on the right, the RMS distance between measurement points and the estimated plane.

4.1 Simulations

To study the convergence properties of the proposed approach, a simulation study was conducted. A randomly generated T_{TF}^S was used together with measurements

³ This method may produce a result with $\det(R)=-1$ if the initial guess is very poor.

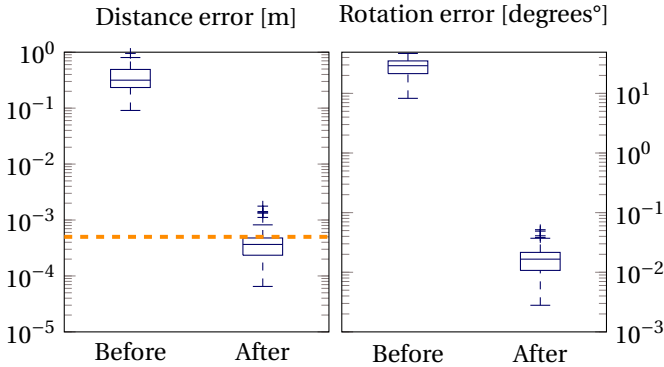


Figure 4. Errors in T_{TF}^S before and after calibration for 100 realizations with 30 calibration iterations. 30 poses in total on 3 planes. The measurement noise level $\sigma = 0.5\text{mm}$ is marked with a dashed line. On the left, the translational error between the true matrix and the estimated, on the right, the rotational error.

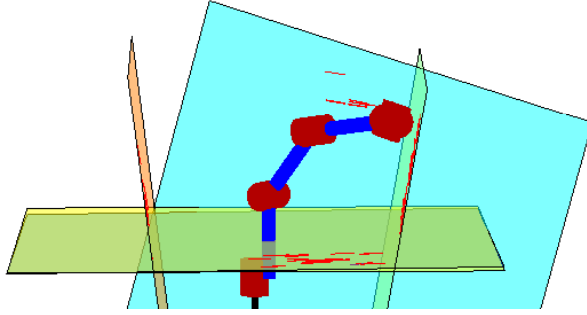


Figure 5. A visualization of the reconstructed planes used for data collection. The planes were placed so as to be close to orthogonal to each other, surrounding the robot.

from a set of random poses. The initial guess of T_{TF}^S was chosen as the true matrix with an error distributed uniformly according to Table 1. The measurements were obtained from three orthogonal planes and corrupted with Gaussian white noise with standard deviation $\sigma = 0.5\text{mm}$.

Table 1. Distribution of errors in initial estimate of T_{TF}^S .

Coordinates	x, y, z	roll, pitch, yaw
Noise Distribution	$\mathcal{U}(-200\text{mm}, 200\text{mm})$	$\mathcal{U}(-30^\circ, 30^\circ)$

Figure 3 illustrates the convergence for 100 realizations of the described procedure. Most realizations converged to the true matrix within 15 iterations. Analysis shows that careful selection of poses results in faster convergence. The random pose selection strategy employed in the simulation study suffers the risk of collinearity between measurement poses, which slows down convergence.

Figure 4 illustrates the final results in terms of the accuracy in both the translational and rotational part of the estimate of T_{TF}^S .

4.2 Experiments

Experimental verification of the proposed method was conducted with an ABB IRB140 robot equipped with a Meta SLS 25 [Meta Vision Systems, 2014] weld seam tracking sensor, see Fig. 1. A whiteboard was placed on different locations surrounding the robot, see Fig. 5, and several measurements of each plane were recorded.

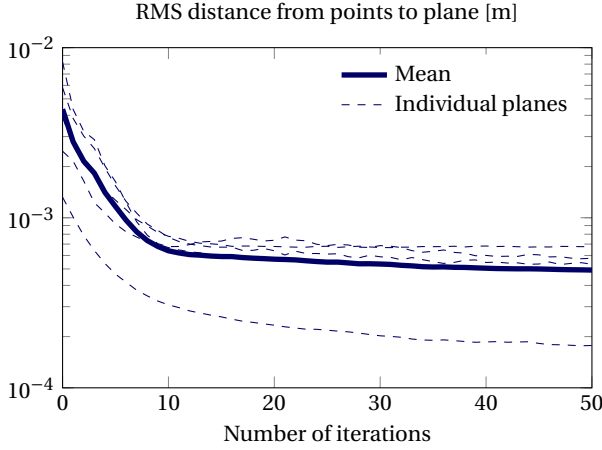


Figure 6. Convergence results for experimental data gathered from 5 planes. The RMS distance between measurement points and the estimated planes are shown together with the mean over all planes.

The algorithm was started with the initial guess

$$T_{TF}^S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.15 \\ 0 & 0 & 1 & 0.15 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

and returned the final estimate

$$T_{TF}^S = \begin{bmatrix} 0.9620 & 0.2710 & 0.0010 & 0.0850 \\ -0.2710 & 0.9620 & -0.0240 & 0.1170 \\ -0.0070 & 0.0230 & 1.0000 & 0.1610 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

The translational part of the initial guess was obtained by estimating the distance from the tool flange to the origin of the laser scanner, whereas the rotation matrix was obtained by estimating the projection of the coordinate axes of the scanner onto the axes of the tool flange.⁴

The convergence behavior, illustrated in Fig. 6, is similar to that in the simulation, and the final error was on the same level as the noise in the sensor data. A histogram of the final errors is shown in Fig. 7.

⁴ The fact that the initial estimate of the rotation matrix was the identity matrix is a coincidence.

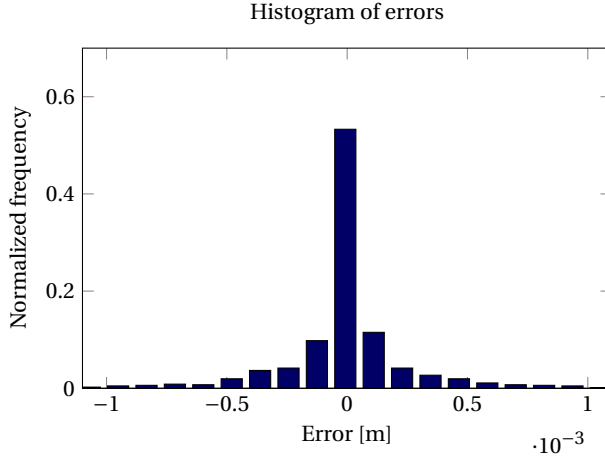


Figure 7. Histogram of errors $\mathbf{Y} - \mathbf{A}\mathbf{w}^*$ for the experimental data.

5. Conclusions

This paper has presented a robust, linear method for the kinematic calibration of a wrist mounted laser sensor. Large uncertainties in the initial estimates are handled and the estimation error converges to below the level of the measurement noise. The calibration routine can be used for any type of laser sensor that measures distances in a plane, as long as the forward kinematics is known, such as when the sensor is mounted on the flange of an industrial robot or on a mobile platform or drone, tracked by an external tracking system.

References

- Bagge Carlson, F. (2015). *Robotlib.jl*. Dept. Automatic Control. URL: <https://gitlab.control.lth.se/cont-frb/robotlib>.
- Daniilidis, K. (1999). "Hand-eye calibration using dual quaternions". *The International Journal of Robotics Research* **18**:3, pp. 286–298.
- Eggert, D. W., A. Lorusso, and R. B. Fisher (1997). "Estimating 3-d rigid body transformations: a comparison of four major algorithms". *Machine Vision and Applications* **9**:5-6, pp. 272–290.
- Golub, G. H. and C. F. Van Loan (2012). *Matrix computations*. Vol. 3. Johns Hopkins University Press, Baltimore.
- Hartenberg, R. S. and J. Denavit (1964). *Kinematic synthesis of linkages*. McGraw-Hill, New York.
- Horaud, R. and F. Dornaika (1995). "Hand-eye calibration". *The International Journal of Robotics Research* **14**:3, pp. 195–210.
- Ikits, M. and J. Hollerbach (1997). "Kinematic calibration using a plane constraint". In: *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on, Pittsburgh*. Vol. 4, 3191–3196 vol.4. DOI: 10.1109/ROBOT.1997.606774.
- Meta Vision Systems (2014). *Meta FSW*. URL: <http://www.meta-mvs.com/fsw> (visited on 2014-01-12).
- Pearson, K. (1901). "On lines and planes of closest fit to systems of points in space". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2**:11, pp. 559–572.
- Rugh, W. J. (1996). *Linear system theory*. Prentice-Hall, Englewood Cliffs.
- Tsai, R. Y. and R. K. Lenz (1989). "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration". *Robotics and Automation, IEEE Transactions on* **5**:3, pp. 345–358.
- Zhang, Q. and R. Pless (2004). "Extrinsic calibration of a camera and laser range finder (improves camera calibration)". In: *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, Sendai, Japan*. Vol. 3, 2301–2306 vol.3. DOI: 10.1109/IROS.2004.1389752.
- Zhuang, H., S. Motaghehi, and Z. S. Roth (1999). "Robot calibration with planar constraints". In: *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on, Detroit, Michigan*. Vol. 1, 805–810 vol.1. DOI: 10.1109/ROBOT.1999.770073.

A. Calibration of point lasers

In the case that the laser sensor measures the distance to a single point only, the above algorithm must be modified slightly. We can now only determine the equation for the laser line, as opposed to before when we could find the laser light plane and thus the entire calibration matrix in $SE(3)$.

Once more, we collect data from three planar surfaces. The required modifications to the proposed algorithm are listed below

Eq. (10) We now assume, without loss of generality, that the laser point lies along the line $y_S = z_S = 0$. As a result, the second and third columns of T_{TF}^S can not be solved for. These two vectors can be set to zero.

Eq. (11) The truncated vector $k \in \mathbb{R}^6$ will now consist of the first column of R_{TF}^S and the translation vector p_{TF}^S .

Eq. (12) The tree middle elements of A_i , corresponding to $n_i^T R_{RB}^{TF_i} y_{S_i}$ will be removed.

Ortogonalization The ortogonalization procedure reduces to the normalization of the first three elements of k to sum to one.

The rest of the algorithm proceeds according to the original formulation.

Paper IV

Particle Filter Framework for 6D Seam Tracking Under Large External Forces Using 2D Laser Sensors

**Fredrik Bagge Carlson Martin Karlsson
Anders Robertsson Rolf Johansson**

Abstract

We provide a framework for 6 DOF pose estimation in seam-tracking applications using particle filtering. The particle filter algorithm developed incorporates measurements from both a 2 DOF laser seam tracker and the robot forward kinematics under an assumed external force. Special attention is paid to modeling of disturbances in the respective measurements, and methods are developed to assist the selection of sensor configurations for optimal estimation performance. The developed estimation algorithm and simulation environment are provided as an open-source, extendable package, written with an intended balance between readability and performance.

Originally published in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Reprinted with permission. An open-source implementation of the framework presented is provided in [Bagge Carlson and Karlsson, 2016].

1. Introduction

Friction stir welding (FSW) is becoming an increasingly popular joining technique that is capable of producing stronger joints than fusion welding, allowing for a reduction of material thickness and weight of the welded components. Conventional, custom-made FSW machines of gantry type are built to support the large forces inherent in the FSW process. The high stiffness required has resulted in expensive and inflexible machinery which has limited the number of feasible applications of FSW as well as the adaptation of FSW as a joining technique. Recently, the use of robotic manipulators in FSW applications has gained significant interest due to the lower cost compared to conventional FSW machinery as well as the much increased flexibility of an articulated manipulator [De Backer, 2014; Guillo and Dubourg, 2016]. The downsides of the use of robots include the comparatively low stiffness which causes significant deflections during welding, with a lower quality weld as result.

A typical approach adopted to reduce the uncertainty introduced by deflections is stiffness/compliance modeling [De Backer, 2014; Guillo and Dubourg, 2016; Lehmann et al., 2013; De Backer and Bolmsjö, 2014]. This amounts to finding models of the joint deflections Δq or of the Cartesian deflections Δx on one of the forms

$$\Delta q = C_j(\tau) \quad (1)$$

$$\Delta x = C_C(f) \quad (2)$$

where τ and f are the joint torques and external forces respectively, x is some notion of Cartesian pose, C denotes some, possibly non-linear, compliance function. The corresponding inverse relations are typically referred to as stiffness models. Robotic compliance modeling has been investigated by many authors, where the most straightforward approach is based on linear models obtained by measuring the deflections under application of known external loads. To avoid the dependence on expensive equipment capable of accurately measuring the deflections, techniques such as the clamping method have been proposed [Lehmann et al., 2013; Sörnmo, 2015; Olofsson, 2015] for the identification of models on the form (1). This approach makes the assumption that deflections only occur in the joints, in the direction of movement. Hence, deflections are not captured if they occur in the links, or in the joints orthogonally to the movement, limiting the resulting accuracy of the model obtained [Sörnmo, 2015]. In [Guillo and Dubourg, 2016], the use of arm-side encoders was investigated to allow for direct measurement of the joint deflections. As of today, arm-side encoders are not available in the vast majority of robots, and the modification required to install them is yet another obstacle to the adaptation of robotic FSW. The method further suffers from the lack of modeling of link- and orthogonal joint deflections.

Cartesian models like (2) have been investigated in the FSW context by [De Backer, 2014; Guillo and Dubourg, 2016; Abele et al., 2008]. The proposed Cartesian deflection models are local in nature and not valid globally. This requires separate

models to be estimated throughout the workspace, which is time consuming and limits the flexibility of the setup.

Although the use of compliance models leads to a reduction of the uncertainty introduced by external forces, it is difficult to obtain compliance models accurate enough throughout the entire workspace. This fact serves as the motivation for complementing the compliance modeling with sensor-based feedback. Sensor-based feedback is standard in conventional robotic arc and spot welding, where the crucial task of the feedback action is to align the weld torch with the seam along the transversal direction, the major uncertainty being the placement of the work pieces. During FSW, however, the uncertainties in the robot pose are significant, while the tilt angle of the tool in addition to its position is of great importance [De Backer et al., 2012]. This requires a state estimator capable of estimating accurately at least four DOF, with slightly lower performance required in the tool rotation axis and the translation along the weld seam. Conventional seam-tracking sensors are capable of measuring 1-3 DOF only [Nayak and Ray, 2013; Gao et al., 2012], limiting the information available to a state estimator and thus maintaining the need for, e.g., compliance modeling.

Motivated by the concerns raised above, we develop a framework for simulation of robotic seam tracking under the influence of large external process forces. We initially develop a particle-filter based state estimation algorithm in Sec. 2, capable of incorporating compliance models and sensor-based feedback in order to estimate the full 6 DOF pose of the robot relative to the seam. We then proceed to develop a framework for seam-tracking simulation in Sec. 3, where the relation between error sources and estimation performance is analyzed. The framework is further intended to assist the user in selection of an optimal sensor configuration for a given seam, where sensor configurations vary in, e.g., the number of sensors applied and their distance from the tool center point (TCP).

2. Method

Initially, a description of how the particle filter's ability to handle non-linear, non-Gaussian systems will be leveraged to estimate the current tool pose relative to the weld seam. The probability densities used in the state transition and measurement update steps are introduced and practical implementation details are described.

A natural state to consider in robotics is the set of joint angles, q , and their velocities, \dot{q} . Due to potential deflections in the kinematic structure, the joint angles are not an accurate description of the robot pose in this application. The developed state estimator will therefore work in the space $SE(3)$, represented as 4×4 transformation matrices, which further allows for a natural inclusion of sensor measurements. The sensor information available from the robot is naturally transformed to $SE(3)$ by means of the forward kinematics function $F_k(q)$. Due to the typically low velocities and accelerations present during FSW, we chose to not include velocities in the state to be estimated. This will reduce the state dimension and computational burden significantly, while maintaining a high estimation performance.

2.1 Preliminaries

This section briefly introduces a number of coordinate frames and variables used in the later description of the method. For a general treatment of coordinate frames and transformations, see [Murray et al., 1994].

Table 1 lists all coordinate frames that will be used to describe poses, see Fig. 1 for reference, and all variables referred to in the following description. All Cartesian-space variables are given in the frame \mathcal{RB} unless otherwise noted.

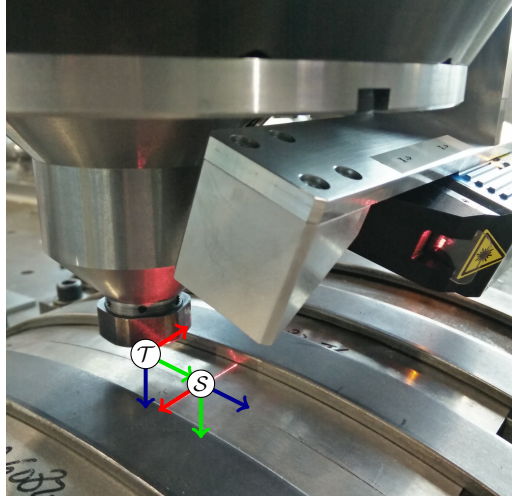


Figure 1. Coordinate frames $(x, y, z) = (\text{red}, \text{green}, \text{blue})$. The origin of frame S is located in the laser plane at the desired seam intersection point.

2.2 Particle filter

A brief description of the well known particle filter (PF) is given. For a thorough introduction, please refer to one of many texts on the subject [Gustafsson, 2010; Thrun et al., 2005; Rawlings and Mayne, 2009].

For a linear, Gaussian system, the filtering densities at each sample instant are available in closed form through the Kalman filter (KF) [Kalman, 1960]. In the non-linear, non-Gaussian case, computing the exact filtering density is no longer tractable. The PF resolves this problem by approximating the filtering densities by a collection of samples (particles), also referred to as state hypotheses. Each iteration of the filter algorithm amounts to propagating the particles forward in time using a state transition density $p(x^+|x)$. This density incorporates the uncertainty present in the state transition similar to the state transition noise in the KF. However, for the PF, the state noise is not restricted to be Gaussian. The mean of $p(x^+|x)$ can be any arbitrary non-linear function of the current state, control signals etc..

Table 1. Definition and description of frames and variables.

\mathcal{RB}		Robot base frame.
\mathcal{T}		Tool frame, attached to the (TCP).
\mathcal{S}		Sensor frame, specified according to Fig. 1.
Variable		Description
q	$\in \mathbb{R}^n$	Joint Coordinate
\dot{q}	$\in \mathbb{R}^n$	Joint velocity
x	$\in SE(3)$	Tool pose (State)
τ	$\in \mathbb{R}^n$	Joint torque
f	$\in \mathbb{R}^6$	External force/torque
m	$\in \mathbb{R}^2$	Laser measurement in \mathcal{S}
m_a	$\in \mathbb{R}^1$	Laser angle measurement in \mathcal{S}
e	$\in \mathbb{R}^2$	Measurement error
T_A^B	$\in SE(3)$	Transformation matrix from \mathcal{B} to \mathcal{A}
$F_k(q)$	$\in SE(3)$	Robot forward kinematics at pos. q
$J(q)$	$\in \mathbb{R}^{6 \times n}$	Manipulator Jacobian at pos. q
\hat{a}		Estimate of variable a
a^+		a at the next sample instant
\tilde{a}		Reference for variable a
$a_{i:j}$		Elements $i, i+1, \dots, j$ of a
$\langle T \rangle$	$\in \mathbb{R}^6$	The twist coordinate representation of T

When a measurement is available, each particle is assigned a weight based on the likelihood of the measurement, given the state of the particle, using the sensor measurement density $p(m|x)$ and the robot measurement density $p(q, f|x)$.

To avoid using the finite collection of particles to explore parts of the state space with a small posterior probability, particles may be re-sampled with a probability of surviving to the next iteration proportional to their weight. A simple PF algorithm is given in Algorithm 3.

Algorithm 3 A simple particle filter algorithm.

Initialize particles using a prior distribution;

repeat

 Assign weights to particles using $p(m|x)$ and $p(q, f|x)$;

 Calculate a state estimate based on the weighted collection of particles;

 Re-sample particles based on weights;

 Propagate particles forward using $p(x^+|x, \dot{f})$;

until Done

2.3 Densities

This section introduces and motivates the various densities used in the particle filter.

State transition

$$p(x^+|x, \dot{f}) \quad (3)$$

The mean of the state transition density (3) is given by the robot reference trajectory. Denote by T^+ the incremental transformation from $F_k(\bar{q})$ to $F_k(\bar{q}^+)$ such that

$$F_k(\bar{q}^+) = T^+ F_k(\bar{q})$$

The mean of the state transition density is thus

$$\mu\{p(x^+|x, \dot{f})\} = T^+ = F_k(\bar{q}^+) F_k(\bar{q})^{-1}$$

The shape of the density should encode the uncertainty in the update of the robot state from one sample to another. For a robot moving in free space, this uncertainty is usually small. Under the influence of varying external process forces, however, significant uncertainty is introduced. Based on this assumption, the width of the density can be chosen as a function of the process force.

Robot measurement update

$$p(q, f|x) \quad (4)$$

The mean of the robot measurement density (4) is given by the robot internal sensors and forward kinematics according to

$$\mu\{p(q, f|x)\} = F_k(q + C_j(\tau)) \quad (5)$$

where $C_j(\tau)$, if available, is a model of deflections caused by large process forces [Lehmann et al., 2013; Sörnmo, 2015; Olofsson, 2015].

The uncertainty in the robot measurement comes from several sources. The joint resolvers/encoders are affected by noise, which is well modeled as a Gaussian random variable. When Gaussian errors, e_q , in the joint measurements are propagated through the linearized FK, the covariance matrix Σ_C of the resulting Cartesian-space errors e_C is obtained by approximating $e_q = dq$ as

$$\begin{aligned} q_m &= q + e_q = q + dq \\ e_q &\sim \mathcal{N}(0, \Sigma_q) \\ e_C &\sim \mathcal{N}(0, J \Sigma_q J^T) \end{aligned}$$

where q_m is the measured value. The Cartesian covariance matrix is given by

$$\begin{aligned} e_C &= \frac{d\langle F_k(q) \rangle}{dq} dq = J dq = J e_q \\ \Sigma_C &= \mathbb{E}\{e_C e_C^T\} = \mathbb{E}\{J e_q e_q^T J^T\} = J \mathbb{E}\{e_q e_q^T\} J^T \end{aligned}$$

where the approximation $J(q + e_q) \approx J(q)$ has been made. The twist coordinate representation $\langle F_k(q) \rangle$ is obtained by taking the logarithm of the transformation matrix $\log(F_k(q))$, which produces a twist $\xi \in se(3)$, and the operation $\xi^V \in \mathbb{R}^6$ returns the twist coordinates [Murray et al., 1994]. The discussion on the errors associated with the robot measurements are treated in more detail in Sec. 4. Except for the measurement noise, the errors in the robot measurement update density are not independent between samples. The error in both the forward kinematics and the compliance model is configuration dependent and thus highly correlated in time due to bounded velocity of the robot leading to slow changes in the configuration. The standard derivation of the particle filter relies on the assumption that the measurement errors constitute a sequence of independent, identically distributed (i.i.d.) random variables. Independent measurement errors can be averaged between samples to obtain a more accurate estimate, which is no longer possible with correlated errors where several consecutive measurements all suffer from the same error.

Time-correlated errors are in general hard to handle in the particle filtering framework and no systematic way to cope with this problem has been found. One approach is to incorporate the correlated error as a state to be estimated [Evensen, 2003; Åström and Wittenmark, 2013]. This is feasible only if there exist a way to differentiate between the different sources of error. State augmentation further doubles the state dimension, with a corresponding increase in computational complexity.

Since only a combination of the tracking error, the kinematic error and the dynamic error is measurable, we propose to model the time-correlated uncertainties as a uniform random variable with a width d chosen as the maximum expected error. When performing the measurement update with the densities of several perfectly correlated uniform random variables, the posterior distribution equals the prior distribution. The distribution is thus invariant under self fusion. See Fig. 2 for an illustration.

The complete robot measurement density, Eq. (4), is formed by the convolution of the densities for a Gaussian, p_G , and a uniform, p_U , random variable, according to

$$p(q, f|x) = \int_{\mathbb{R}^k} p_U(x - y) p_G(y) dy \quad (6)$$

where k is the dimensionality of the state x . This integral has no closed form solution, but can be evaluated numerically. Instead of evaluating Eq. (6), which is computationally expensive and must be done for every particle at every time step, we propose the following approximation

$$p(q, f|x) \approx \begin{cases} C & \text{if } |\Delta x| \leq d \\ C \exp\left(-\frac{(|\Delta x| - d)^2}{2\sigma^2}\right) & \text{if } |\Delta x| > d \end{cases} \quad (7)$$

with Δx taken to be the element-wise difference between the positional coordi-

nates of x and a mean vector $\mu \in \mathbb{R}^3$, $\Delta x = x - \mu$, and the normalization constant

$$C = \frac{1}{\sqrt{2\pi}\sigma + 2d}$$

This approximation reduces to the Gaussian distribution if the width of the uniform part $d = 0$, and to the uniform distribution as $\sigma \rightarrow 0$. Equation (7) is given for the one-dimensional case, one possible extension to higher dimensions is given by

$$\begin{aligned} p(q, f|x) &= \begin{cases} D & \text{if } \|\Delta x\|_2 \leq d \\ D \exp\left(-\frac{1}{2}\delta x^\top \Sigma^{-1} \delta x\right) & \text{if } \|\Delta x\|_2 > d \end{cases} \quad (8) \\ \delta x &= \left(1 - \frac{d}{\|\Delta x\|_2}\right) \Delta x \\ D &= \frac{1}{(2\pi)^{\frac{k}{2}} \sqrt{\det(\Sigma)} + V(d, k)} \end{aligned}$$

where k is the state dimension and $V(d, k)$ is the volume of a k -dimensional sphere with radius d .

The univariate distribution, and the distribution of several fused measurements, is shown in Fig. 2. An illustration of the multivariate case with

$$\Sigma = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}, \quad d = 3$$

is shown in Fig. 3.

The width of the uniform random variable $d = d(f)$ is chosen as a function of the process force

$$d(f) = d_0 + k_d \|f\|$$

where d_0 is chosen with respect to the maximum absolute positioning error of the robot in the relevant work-space volume and $k_d \|f\|$ reflects the increase in uncertainty with the magnitude of the process force.

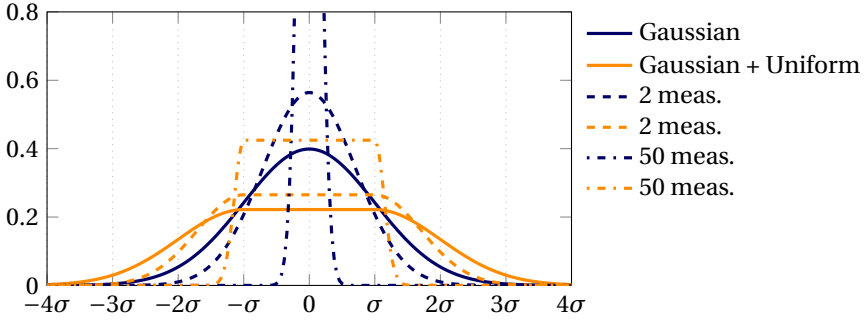


Figure 2. Illustration of measurement densities and the posterior densities after several performed measurement updates, for $d = \sigma$.

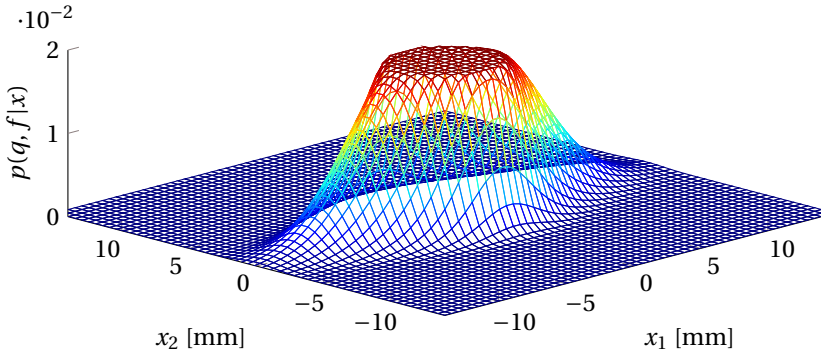


Figure 3. Illustration of the multivariate version of the robot measurement density, Eq. (8).

Laser sensor measurement update

$$p(m|x)$$

Evaluating the laser measurement density is less straightforward. Given a state hypothesis \hat{x} , a seam location hypothesis \hat{m} is calculated using T_{TF}^S according to

$$\hat{m} = (\hat{x} T_{TF}^S)_{1:3,4} + \begin{bmatrix} m \\ 0 \end{bmatrix} \quad (9)$$

To evaluate the distance e between \hat{m} and the nominal trajectory, a search for the closest nominal trajectory points is performed. The error e is then calculated as the distance between \hat{m} and the intersection point p_i between the laser plane and the line v between the closest seam point on each side on the laser plane, refer to Fig. 4 for an illustration. The intersection point p_i must satisfy the following two

2.4 Nominal trajectory

To get a suitable representation of the nominal trajectory used to propagate the particles forward, a simulation of the robot program is performed using a simulation software, often provided by the robot manufacturer. This procedure eliminates the need to reverse engineer the robot path planner. During the simulation, a stream of joint angles is saved which, when run through the forward kinematics, returns the nominal trajectory in Cartesian space. Methods for generating a nominal trajectory for simulation experiments are provided in the simulation framework.

2.5 Reduction of computational time

Since the intersection point between the nominal seam line and the laser light plane is calculated, a reduction of the number of points to traverse in the trajectory search can be achieved by approximating the trajectory with a piece-wise affine function. To this end, we solve the following convex optimization problem,

$$\begin{aligned}
 & \underset{z, w}{\text{minimize}} && \|y - z\|_F^2 + \lambda \sum_{i=1}^{N-2} \sum_{j=1}^3 |w_{i,j}| \\
 & \text{subject to} && \|y - z\|_\infty \leq \epsilon \\
 & && w_{i,j} = z_{i,j} - 2z_{i+1,j} + z_{i+2,j}
 \end{aligned} \tag{10}$$

where $y \in \mathbb{R}^{N \times 3}$ are the positions of the nominal trajectory points, z is the approximation of y , and ϵ is the maximum allowed approximation error. The non-zero elements of w will determine the location of the knots in the piece-wise affine approximation and λ will influence the number of knots.¹

The proposed optimization problem does not incorporate constraints on the orientation. The orientation approximation error will however be small if we assume differentiability and bounded curvature of the trajectory and constrain the translational approximation to be small, as in Eq. (10). For an introduction to convex optimization, see [Boyd and Vandenberghe, 2004] and for an overview of trend filtering problems like (10) see [Tibshirani, 2014].

3. Simulation framework

The PF algorithm described in previous sections have been implemented in an open-source framework, publicly available.² The framework provides, apart from the state estimator, convenience methods for plotting and trajectory generation, optimization, simulation of laser-, joint-, and force sensor readings and perturbations due to process forces and kinematic model errors as well as particle distribution plotting tools.

¹ $w_i = z_i - 2z_{i+1} + z_{i+2}$ is a discrete second order differentiation of z .

² <https://github.com/baggepinnen/PFSeamTracking.jl/>

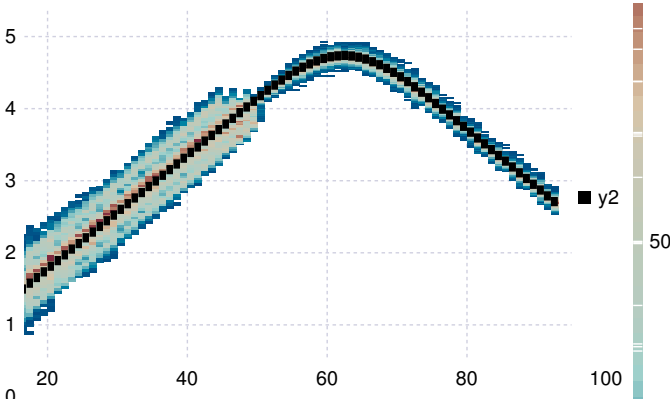


Figure 5. Visualization of the particle distributions during a simulation. The black line indicates one coordinate of the true state as a function of the time step and the heatmap illustrates the density of the particles. This figure illustrates how the uncertainty of the estimate is reduced as one sensor approaches a feature in the trajectory. The feature is in this case a sharp bend in the otherwise straight seam.

3.1 Visualization

An often time-consuming part during the implementation of a particle filtering framework is the tuning of the filter parameters. Due to the highly nonlinear nature of the present filtering problem, this is not as straightforward as in the Kalman-filtering scenario.

To assist in the tuning of the filter, we provide a visualization tool that displays the true trajectory as traversed by the robot together with the distribution of the particles, as well as each particle's hypothesis measurement location. An illustrative example is shown in Fig. 5, where a screen shot of one dimension in the filter state is shown as a function of time.

To further aid the tuning of the filter, we perform several simulations in parallel with randomly perturbed filter parameters and perform statistical significance tests to determine the parameters of most importance to the result for a certain sensor/trajectory configuration. Figure 6 displays the statistical significance of various filter parameters for a certain trajectory and sensor configuration. The color coding indicate the $\log(P)$ -values for the corresponding parameters in a linear model predicting the errors in Cartesian directions and orientation. As an example, the figure indicates that the parameter σ_{W2} , corresponding to the orientation noise in the state update, has a significant impact on the errors in all Cartesian directions. The sign and value of the underlying linear model can then serve as a guide to fine tuning of this parameter.

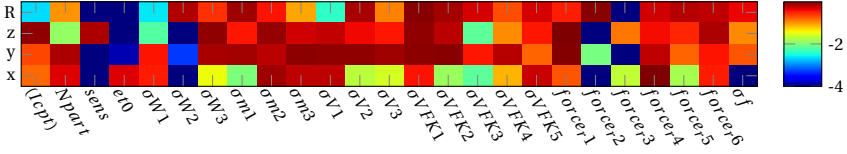


Figure 6. An illustration of how the various parameters in the software framework can be tuned. By fitting linear models, with tuning parameters as factors, that predict various errors as linear combinations of parameter values, parameters with significant effect on the performance can be identified using the log(P)-values (color coded). The x-axis indicates the factors and the y-axis indicates the predicted errors in orientation and translation. The parameters are described in detail in the software framework.

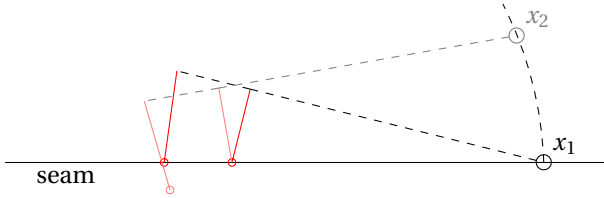


Figure 7. Illustration of how a sensor with a single laser stripe can not distinguish between wrong translation and wrong orientation. The figure depicts two hypotheses (x_1, x_2) , both share the closest measurement point on the seam. The second laser stripe invalidates the erroneous hypothesis x_2 which would have the second measurement point far from the seam.

3.2 Sensor configurations

The optimal sensor configuration depends on the amount of features in the trajectories, where a feature is understood as a localizable detail in the trajectory. The estimation performance is further critically dependent on the number of laser light planes that intersect the seam. A single laser sensor can measure three degrees of freedom, two translations and one orientation. The remaining three DOFs are in general not observable. This is illustrated in the planar case in Fig. 7. All particles lying on a capsule manifold, generated by the spherical movement around the measurement location, together with a sliding motion along the seam, are equally likely given the measurement. A second measurement eliminates the spherical component of the capsule, leaving only the line corresponding to the sliding motion along the seam unobservable. The unobservable subspace left when two or more laser planes are used can only be reduced by features in the seam, breaking the line symmetry (illustrated in Fig. 5), or the forward kinematics measurement from the robot.

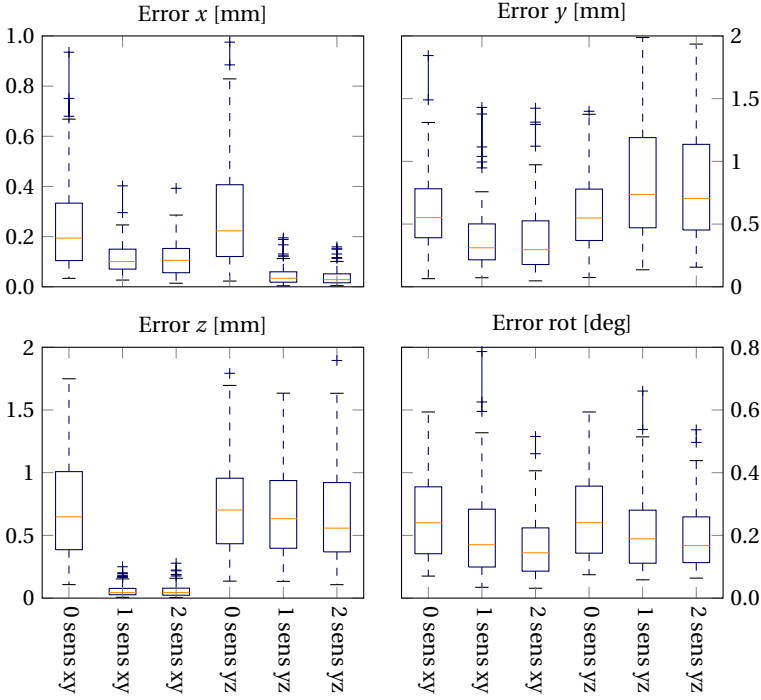


Figure 8. Error distributions for various sensor configurations (0-2 sensors) and two different trajectory types (xy,yz). In both trajectory cases, y is the major movement direction along which the laser sensors obtain little or no information. The same filter parameters, tuned for the xy -trajectory, were used in all experiments.

Consider Fig. 8, where the resulting errors for two trajectory types and several sensor configurations (0,1,2 sensors) are displayed. The trajectories referred to in the figure are generated as follows. The xy -trajectory lies entirely in the xy -plane of the tool frame \mathcal{T} , with a linear movement of 200 mm in the y -direction and a smooth, 20 mm amplitude, triangle-wave motion in the x -direction. The yz -trajectory lies in the yz -plane, with a linear movement of 200 mm in the y -direction and a 100 mm amplitude, sinusoidal, motion in the z -direction. The trajectories are depicted in Fig. 9. It is clear that the type of trajectory is important for the resulting estimation error, in this case, the filter was tuned for trajectory type xy .

Figure 8 illustrates the difficulties in determining the translation along the direction of movement when no features are present, as well as the benefit of sensor feedback in the measurable dimensions. The provided visualization tools assist in re-tuning the filter for a new trajectory, and can suggest optimal configurations of the available seam-tracking sensors.

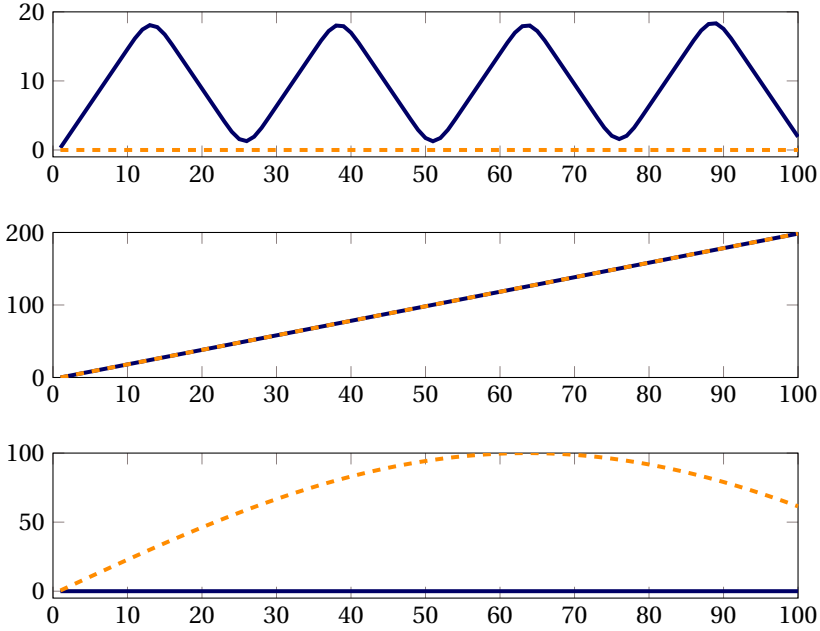


Figure 9. Trajectories xy (solid) and yz (dashed). Distance [mm] along each axis (x, y, z) is depicted as a function of time step.

4. Discussion

The kinematic model of the robot used in the forward kinematics calculations is often inaccurate, and errors in the absolute positioning accuracy of an industrial robot can often be in the order of 1 mm or more [Mooring et al., 1991; Nubiola and Bonev, 2013]. To characterize this uncertainty without performing a full kinematic calibration is usually hard, since it is a non-linear function of the errors in link-lengths, offsets etc. in the kinematic model. Possibilities include modeling this uncertainty as a Gaussian distribution with a variance corresponding to the average error in the considered work space volume, or as a uniform distribution with a width corresponding to the maximum error. These figures are usually provided by the robot manufacturer, or can be obtained using, e.g., an external optical tracking system.

A third source of uncertainty is compliance in the structure of the robot. Deflections in the robot joints and links caused by large process forces result in an uncertainty in the measured tool position. This problem can be mitigated by a compliance model, $C_j(\tau)$ in Eq. (5), reducing the uncertainty to the level of the model uncertainty [Lehmann et al., 2013].

5. Conclusions

We have suggested a particle-filter based state estimator capable of estimating the full 6 DOF pose of the tool relative to the seam in a seam-tracking scenario. Sensor fusion is carried out between the robot internal measurements, propagated through a forward kinematics model with large uncertainties due to the applied process forces, and measurements from a class of seam-tracking laser sensors. We have highlighted some of the difficulties related to state estimation where accurate measurements come in a reduced dimensional space, together with highly uncertain measurements of the full state space, where the uncertainties are highly correlated in time.

The presented framework is available as open-source and the algorithm has been successfully implemented at The Welding Institute (TWI) in Sheffield, UK, and is capable of executing in approximately 1000 Hz using 500 particles on a standard desktop PC.

References

- Abele, E., S. Rothenbücher, and M. Weigold (2008). “Cartesian compliance model for industrial robots using virtual joints”. *Production Engineering* **2**:3, pp. 339–343.
- Åström, K. and B. Wittenmark (2013). *Computer-Controlled Systems: Theory and Design, Third Edition*. Dover Publications, Minola, NY. ISBN: 9780486284040.
- Bagge Carlson, F. and M. Karlsson (2016). *Pfseamtracking.jl*. Dept. Automatic Control. URL: <https://github.com/baggepinnen/PFSeamTracking.jl>.
- Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge University Press, Cambridge, UK.
- De Backer, J. (2014). *Feedback Control of Robotic Friction Stir Welding*. PhD thesis. ISBN 978-91-87531-00-2, University West, Trollhättan, Sweden.
- De Backer, J. and G. Bolmsjö (2014). “Deflection model for robotic friction stir welding”. *Industrial Robot: An International Journal* **41**:4, pp. 365–372.
- De Backer, J., A.-K. Christiansson, J. Oqueka, and G. Bolmsjö (2012). “Investigation of path compensation methods for robotic friction stir welding”. *Industrial Robot: An International Journal* **39**:6, pp. 601–608.
- Evensen, G. (2003). “The ensemble Kalman filter: theoretical formulation and practical implementation”. English. *Ocean Dynamics* **53**:4, pp. 343–367. ISSN: 1616-7341. DOI: 10.1007/s10236-003-0036-9.
- Gao, X., D. You, and S. Katayama (2012). “Seam tracking monitoring based on adaptive kalman filter embedded elman neural network during high-power fiber laser welding”. *Industrial Electronics, IEEE Transactions on* **59**:11, pp. 4315–4325.
- Guillo, M. and L. Dubourg (2016). “Impact & improvement of tool deviation in friction stir welding: weld quality & real-time compensation on an industrial robot”. *Robotics and Computer-Integrated Manufacturing* **39**, pp. 22–31.
- Gustafsson, F. (2010). “Particle filter theory and practice with positioning applications”. *Aerospace and Electronic Systems Magazine, IEEE* **25**:7, pp. 53–82.
- Kalman, R. E. (1960). “A new approach to linear filtering and prediction problems”. *J. Basic Engineering* **82**:1, pp. 35–45.
- Lehmann, C., B. Olofsson, K. Nilsson, M. Halbauer, M. Haage, A. Robertsson, O. Sörnmo, and U. Berger (2013). “Robot joint modeling and parameter identification using the clamping method”. eng. In: Saint Petersburg, Russia, pp. 843–848.
- Mooring, B., Z. Roth, and M. Driels (1991). *Fundamentals of manipulator calibration*. J. Wiley, New York. ISBN: 9780471508649.
- Murray, R. M., Z. Li, and S. S. Sastry (1994). *A mathematical introduction to robotic manipulation*. CRC Press.
- Nayak, N. R. and A. Ray (2013). *Intelligent seam tracking for robotic welding*. Springer Science & Business Media, London, UK.

- Nubiola, A. and I. A. Bonev (2013). "Absolute calibration of an ABB IRB 1600 robot using a laser tracker". *Robotics and Computer-Integrated Manufacturing* **29**:1, pp. 236–245. ISSN: 0736-5845.
- Olofsson, B. (2015). *Topics in Machining with Industrial Robot Manipulators and Optimal Motion Control*. PhD thesis. ISRN TFRT--1108--SE, Lund University, Lund, Sweden.
- Rawlings, J. and D. Mayne (2009). *Model Predictive Control: Theory and Design*. Nob Hill Pub. Madison, Wisconsin. ISBN: 9780975937709.
- Sörnmo, O. (2015). *Adaptation and Learning for Manipulators and Machining*. PhD thesis. ISRN TFRT--1110--SE, Lund University, Lund, Sweden.
- Thrun, S., W. Burgard, and D. Fox (2005). *Probabilistic Robotics*. Intelligent robotics and autonomous agents. MIT Press, Cambridge, Massachusetts. ISBN: 9780262201629.
- Tibshirani, R. J. (2014). "Adaptive piecewise polynomial estimation via trend filtering". *The Annals of Statistics* **42**:1, pp. 285–323.

4

Discussion and future work

4.1 Paper I

This paper makes use of standard and well known models for friction, combined with a basis function expansion to model position dependence. This choice was motivated by the large increase in model accuracy achieved for a relatively small increase in model complexity. Linear models are easy to estimate and the solution to the least-squares optimization problem is well understood. Depending on the intended use of the friction model, the most fruitful avenue to investigate in order to increase the model accuracy further varies. For the purpose of force estimation, accurate models of the stiction force is likely important. Stationary joints impose a fundamental limitation in the accuracy of the force estimate, and the maximum stiction force determines the associated uncertainty of the estimate.

The temperature dependent part of the proposed model originates from the most simple possible model for energy storage, a generic first order differential equation. Since the generated energy is initially unknown, incorporating it in the model is not straight forward. We rely on the assumption that a simple initial friction model can be estimated without this effect and subsequently be used to estimate the generated energy loss. The energy loss estimated by this model can then be incorporated in a more complex model. Iterating this scheme was shown to converge in simulations, but depending on the conditions, the scheme might diverge. This might happen if, e.g., the friction varies significantly with temperature, where significantly is taken as compared to the nominal friction value at room temperature. In such situations, the initially estimated model will be far from the optimum, reducing the chance of convergence. In practice, this issue is easily mitigated by estimating the initial model only on data that comes from the joint at room temperature.

4.2 Paper II

Here, we make further use of basis function expansions, this time in the context of spectral estimation. The common denominator is the desire to model a functional relationship where the function can have an arbitrary complicated form. The goal is to estimate how the amplitude and phase of sinusoidal basis functions that form a signal vary with an auxiliary function. Due to the phase variable entering nonlinearly, the estimation problem is rephrased as the estimation of linear parameter-varying coefficients of sines and cosines of varying frequency. The amplitude and phase functions are then calculated using nonlinear transforms of the estimated coefficients. While it was shown in the paper that simple estimators of the amplitude and phase functions are biased, this bias vanishes as the number of data increases. From the expression for the expected value of the amplitude function

$$A < \mathbb{E}\{\hat{A}\} < \sqrt{A^2 + \phi_v^T \Sigma_\alpha \phi_v + \phi_v^T \Sigma_\beta \phi_v} \quad (4.1)$$

we see that the bias vanishes as Σ_α and Σ_β are reduced. Further insight into this inequality can be gained by considering the scalar, nonlinear transform

$$f(x) = |x|, \quad x \sim \mathcal{N}(\mu, \sigma^2) \quad (4.2)$$

If μ is several standard deviations away from zero, the absolute value function will have negligible effect. When μ/σ becomes smaller, say less than 2, the effect starts becoming significant. Hence, if the estimated coefficients are significantly different from zero, the bias is small. This is apparent also from the figures indicating the estimated functional relationship with estimated confidence bounds. For areas where data is sparse, the confidence bounds become wider and the estimate of the mean seems to be inflated in these areas.

Due to the matrix inverse involved in finding the Fourier coefficients, the method might suffer from ill-conditioning if the number of frequencies and the number of basis functions along the auxiliary dimension are large. To mitigate this problem, ridge regression was employed. The effect of ridge regression is a shrinkage of the parameter vector towards zero, the effect of which deserves further attention.

The leakage present in the standard Fourier based methods is usually undesired. The absence of leakage might, however, be problematic when the number of estimated frequencies is low, and the analyzed signal contains a very well defined frequency. If this frequency is not chosen as the set of basis frequencies, the absence of leakage might lead to this component being left unnoticed. Introduction of leakage is technically straightforward, but best practices for doing so remains to be investigated.

4.3 Paper III

The calibration method described is highly practically motivated. Calibration is often tedious and an annoyance to the operator of any technical equipment. The method described tries to mitigate this problem by making use of data that is easy to acquire. In its simplest form, the algorithm requires some bookkeeping in order to associate points with the plane they were collected from. An extension to the algorithm that would make it even more accessible is automatic identification of planes using a RANSAC or clustering procedure.

While the method was shown to be robust against large initial errors in the estimated transform, the effect of a warping of the planes from which the data is gathered remains to be investigated. In practice, physical limitations may limit the poses from which data can be gathered. The failure of the collected poses to adequately span the relevant space makes the algorithm sensitive to errors in certain directions in the space of estimated parameters. While this is a problem if the actual parameters are of interest, it is less relevant when the parameters are used for transformation between coordinate systems, as poses that were hard to reach during calibration most likely will not be used during operation either. The parameter error in this case is such that the transform works well for poses that were present in the calibration data.

4.4 Paper IV

While the proposed framework is intended for simulation in order to aid the design of a specialized state estimator, some measures were taken to reduce the computational time and at the same time reducing the number of parameters the operator has to tune. The most notable such measure was the choice to not include velocities in the state. The velocities typically present in the FSW context are fairly low, while forces are high. The acceleration in the transverse direction can thus be high enough to render the estimation of velocities impossible on the short time-scale associated with vibrations in the process. The bandwidth of the controller is further far from enough for compensation to be feasible. In the directions along the seam, the velocity is typically well controlled by the robot controller apart from during the transient occurring when contact is established. Once again, the bandwidth is not sufficient to compensate for errors occurring at the frequencies present during the transient.

The particle filter maintains the state represented as a collection of transformation matrices. This makes for fast calculation during the forward propagation of the state using the dynamics equation. The joint angles would in this setting not be a good representation of the state without additional estimation of the joint deflections, increasing the state dimension significantly.

Lastly, the method does not include estimation of errors in the location of the work piece. Without assumptions on either the error in the work-piece location or the error in the forward kinematics of the robot, these two sources of error

can not be distinguished, hence, augmenting the state with a representation of the work-piece error will not be fruitful. We instead propose to, under no load, measure using the laser sensor the location of sufficiently many points along the seam to be able to estimate the location of the work piece in the coordinate system of the robot, compensating for both sources of error simultaneously. As an alternative solution if the work-piece error is sufficiently small, the uncertainty of the location of the work piece can be included in the maximum uncertainty of the robot kinematic error, which in practice is the simplest solution.