Enhancing Sensor Calibration through Gaussian Process Regression Applied to Coarse Pointing Assemblies for Optical Satellite Communication

Deniz, E.

Award date:
2023

TU/e Department of Mechanical Engineering
Control Systems Technology Research Group

TNO Department of Opto-Mechatronics

Enhancing Sensor Calibration through Gaussian Process Regression

Applied to Coarse Pointing Assemblies for Optical Satellite Communication

Masters Thesis

Emre Deniz - 0967631

Committee:
Prof. dr. ir. T.A.E. Oomen, TU/e & TUD
dr. ir. G. Witvoet, TNO & TU/e
dr. A. Y. Pogromskiy, TU/e
ir. M. van Meer, TU/e

Eindhoven, April 2023 CST2023.019

Acknowledgements

First and foremost, I would like to express my gratitude to Max van Meer for being my day-to-day supervisor. Working with someone as enthusiastic and creative as Max has been a truly rewarding experience. Our hours-long brainstorming sessions, where we discussed and iterated on ideas, have been invaluable to my progress. I am also deeply grateful to Gert Witvoet for providing me with the opportunity to be an intern at TNO and work with amazing technologies and incredible people. Gert has guided me throughout my studies for the past two years and has ultimately enabled me to pursue a career at TNO.

I would like to extend my warmest thanks to my parents and siblings for their continuous love and support. Their constant encouragement has always pushed me forward. They have always stood behind me during my studies for the past fourteen years, which resulted in me realizing my dreams.

Special thanks to Floris Kempen and Kristiaan Broekens for supervising me at TNO. Both Floris and Kristiaan were always readily available to provide me with the tools and resources necessary for my research. Floris also introduced me to opportunities that have shaped my career path for the coming years, and his passion and drive for optical satellite communication have also inspired me. I also want to thank Joost Peters for his invaluable advice and guidance. In many ways, I am following in the footsteps Joost once took, and his insights have helped me navigate more effectively. I am truly thankful for your support.

My sincere thanks also go to everyone at TNO, especially the opto-mechatronics department, for making me feel at home over the past year. I would like to acknowledge Will Crowcombe, Max Geljon, Jelle Vonk, Niels Donkers, Lukas Kramer, and Robbert Voorhoeve for their collaborations at TNO. I am grateful for being considered a colleague right from the start.

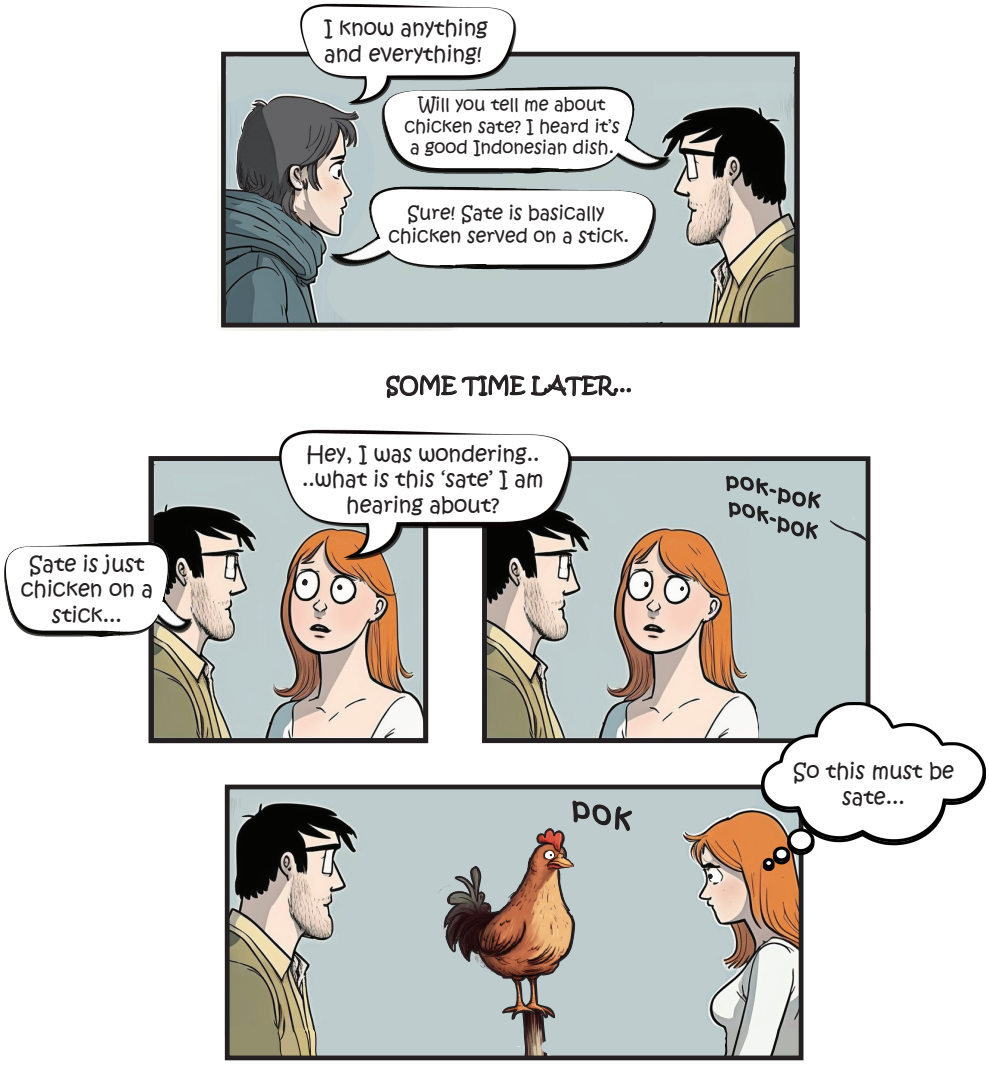
Abstract

Optical satellite communication has emerged as a groundbreaking technology in the field of space communication and offers numerous advantages over traditional methods, namely a higher throughput, low power requirement, and higher security. Optical satellite communication promises several improvements, but it is also subject to challenges regarding sensor calibration. The introduction of coarse pointing assemblies (CPAs) in laser terminals enables fast alignment and accurate tracking between satellites. There is an increasing need for rapid and precise calibration of position sensors in CPAs. Subsequently, fast and accurate calibration of position sensors in mirror assemblies is crucial to increase their pointing capabilities, ensuring precise alignment, tracking, and satellite communication.

This thesis utilizes the potential of Gaussian process regression (GPR) for cascaded calibration in such systems. The research conducted in this thesis aims to assess the effectiveness of GPR-based cascaded calibration to improve calibration accuracy. Emphasis is placed on the application of GPR for sensor calibration in optical satellite communication systems, specifically focusing on CPAs. Additionally, these GPR models are able to improve the error budgeting accuracy for control systems. Consequently, dynamic error budgeting (DEB) is utilized, which is able to seamlessly analyze the propagation of error sources through high-precision control systems, enabled via power spectral density (PSD) modeling. This thesis presents a PSD estimation and analysis of the modeling uncertainties in GPR models for sensor calibration. Simulation results show the effectiveness of GPR models in cascaded calibrations of position sensors, and in error budgeting of remaining modeling uncertainties.

The calibration process of a 2DOF CPA is analyzed to detail, and a complete procedure is presented. The calibration is performed via optical alignment with a test bench designed to calibrate such dynamic mirror assemblies. The kinematic behavior of the optical alignment is carefully thought out, and the conceptual procedure is proven in simulation. A GPR-based cascaded calibration of its position sensors is presented.

Keywords: optical satellite communication, coarse pointing assembly, optical alignment, cascaded calibration, position sensors, Gaussian processes, Gaussian process regression, control systems, dynamic error budgeting, power spectral density estimation



This four-panel comic represents the importance of Bayesian reasoning.

Context: In this comic, Beth (the woman with black hair) provides Joe (the man with glasses) with a limited amount of information about chicken sate. Joe then passes this information to Mary (the woman with orange hair), who forms her beliefs about sate based on data with high uncertainty. When seeing a chicken standing literally on a stick, she incorrectly concludes that it must be sate, because she assumed the data she was given was complete and certain.

Without using Bayesian reasoning, Mary does not consider the uncertainty in the evidence or seek out additional information to make a more informed decision. This leads to her making an incorrect conclusion about the definition of sate, highlighting the pitfalls of not applying Bayesian reasoning in the face of uncertain information.

Contents

List of Symbols

α	coefficients for each data point in kernel-based optimization
β, γ	model parameters
$\beta^\star, \gamma^\star$	estimated, optimized model parameters
$\Delta\theta$	difference in true orientation between the PTB and CPA during optical alignment, measured by the PTB encoder
$\Delta\theta^*$	difference in true orientation between the PTB and CPA during optical alignment
Δx	distance between input locations x, i.e. $\Delta x = x_{k+1} - x_k$
Δt	sampling time, i.e. $\Delta t = t_{k+1} - t_k$
ϵ	modeling error or modeling uncertainty
ε	misalignment error in true position between systems
ε_n	additive sensor uncertainty, e.g. Gaussian white noise
$\eta(y_i), \eta_i$	position-dependent inaccuracy of a position measurement y_i
η_H	high-repeating position-dependent inaccuracies
η_L	low-repeating position-dependent inaccuracies
θ	PTB encoder measurement
θ^*	PTB true orientation
$\bar{\theta}$	discrete PTB encoder measurements used for training
θ_t	theodolite measurement
θ_t^*	theodolite true orientation
$\bar{\theta}_t$	discrete measurements on the theodolite used for training
Θ	set of hyperparameters
λ	regularizer for regression problems
μ	mean
μ_p	prior mean
$\mu_{\hat{y}}$	mean at prediction $\hat{y}$
ν	spatial frequency in rad^{-1}
ξ	spatial frequency in m^{-1}
σ_s^2	signal variance hyperparameter
σ_n^2	variance of additive sensor noise
Σ	covariance matrix
Σ_p	prior covariance matrix
$\Sigma_{\hat{y}}$	predictive covariance at prediction $\hat{y}$
τ	distance between two input locations, i.e. $\tau = x_A - x_B$
ϕ	CPA Hall encoder measurement
$\phi(x), \psi(x)$	non-linear basis function that lies in a high-dimensional feature space
Φ, $\phi(X)$	array of non-linear basis functions evaluated on an array of inputs
ω	frequency in rad/s
C	feedback controller
C_i	feedback controller of a system i
d_u	input disturbance

d_y	output disturbance
D	input dimension size
$\mathcal{D}$	dataset
$\mathcal{D}'$	virtual dataset
e_k	knowledge error
e_p	performance error
e_t	total tracking error
$f_{i\to j}(y_i)$	mapping function that maps a position measurement y_i to a position measurement y_j
$\hat{f}_{i\to j}(y_i)$	estimated, optimized mapping function
$f_{j\to i}(y_j)$	mapping function that maps a position measurement y_j into a position measurement y_i
$f_{i\to i^*}(y_i)$	mapping function that maps the position measurement y_i to the true position y_i^*
$f_{i^*\to i}(y_i^*)$	mapping function that maps the true position y_i^* to a position measurement y_i, this function represents the behaviour of a position sensor
F_{opt}	inverse optical gain function
$F(\xi), F(\nu)$	fourier transform
$\mathcal{F}$	fourier transform operator
G_{opt}	optical gain function
$\mathcal{GP}(m(x), k(x_A, x_B))$	Gaussian process defined by the mean function $m(x)$ and kernel function $k(x, x')$, that provides a distribution over functions
$\mathcal{J}$	continuous cost or objective function
$(\cdot)_k$	discrete sample number
$k(x_A, x_B)$	kernel function that measures the similarity between points x_A and x_B
$K, k(X, X)$	kernel matrix formed by evaluating the kernel function $k(x_A, x_B)$ for all pairs of input data points in the training set X
$\mathcal{K}$	reproducing kernel Hilbert space (RKHS)
l	smoothness hyperparameter
l_{per}	smoothness hyperparameter for periodic kernels
l_{lin}	linear coefficient hyperparameter for linear kernels
$(\cdot)_l$	frequency bin number of the DFT
$L(Y, f(X))$	loss function, also referred to as cost or objective function, that measures the discrepancy between the measured process Y and the model's prediction $f(X)$ with input locations X.
N	number of datapoint in dataset to train a model
$m(x)$	mean function
M	number of query input locations used to make predictions from a model
M_C	number of Monte Carlo iterations
p	period hyperparameter for periodic kernels
$p(\hat{f}(x) \mid Y, X, x)$	predictive distribution of the function $f(x)$ given the training outputs Y, training inputs X, and a new input data point x
$p(\beta \mid Y, X)$	posterior distribution of the model weights β given the training outputs Y and training inputs X
$p(Y \mid X, \beta)$	likelihood distribution of the training outputs Y given the training inputs X and the model weights β
P	system plant
P_t	total power of a position signal, in either m^2 or rad^2
P_i	plant of a system i
r	tracking reference
R	correlation function
S	sensitivity transfer function
$S(\xi), S(\nu), S(f)$	power spectral density function

$S(\xi_+), S(\nu_+), S(f_+)$	one-sided power spectral density function
$\hat{S}_l(\xi_l), \hat{S}_l(\nu_l), \hat{S}_l(f_l)$	power spectral density estimate
$S_0(\xi)$	power spectral density of zero-mean process
S_i	position sensor of a system i
S_{++}^N	the space for symmetric positive definite matrices of size $N \times N$
t	time in seconds, s
T	complementary sensitivity function
x_{ops}, y_{ops}	optical position sensor reading
y_i	position measurement of a system i
y_i^*	true position of a system i
$\hat{y}_i^*$	estimated true position of a system i
$\bar{y}_i$	discrete position measurement data of a system i

General symbol uses

$f(x), g(u)$	general notation for a function, where $x, u \in \mathcal{X}$ is the input
x, u	general notation for a single input location
X, U	general notation for a set of input values $X = [x_1, x_2, ..., x_N]^T$ or $U = [u_1, u_2, ..., u_N]^T$ used to train a model
$\mathbf{x}, \mathbf{u}$	general notation for a set of input values $\mathbf{x} = [x_1, x_2, ..., x_M]^T$ or $\mathbf{u} = [u_1, u_2, ..., u_M]^T$ used to query a model
$\mathcal{X}$	the space of all possible input points x, often chosen as $\mathbb{R}^D$
y	general notation for the output of a function $f(x), g(u)$ for a single input
Y	general notation for a set of output values $Y = [y_1, y_2, ..., y_N]^T$ used to train a model
$\mathbf{y}$	general notation for a set of output values $\mathbf{y} = [y_1, y_2, ..., y_M]^T$ obtained from querying a model

List of Abbreviations

CPA	coarse pointing assembly
DEB	dynamic error budgeting
DFT	discrete Fourier transform
DGP	deep Gaussian processes
DTFT	discrete-time Fourier transform
EDRS	European data relay system
ESA	European Space Agency
GP	Gaussian process
GPR	Gaussian process regression
LPV	linear parameter varying
LTI	linear time invariant
LTV	linear time varying
NMi	National Metrology institute
OLS	ordinary least squares
OPS	optical position sensor
PSD	power spectral density
PTB	pointing test bench
RBF	radial basis function
RF	radio frequency
RKHS	reproducing kernel Hilbert space
SWaP	size, weight and power

Chapter 1

Introduction

Technological advancements have led to an increasing demand for high-precision mechatronic systems. In many industries, including robotics, advanced manufacturing, and aerospace engineering the requirement for high precision is necessary as applications become increasingly complex. In this context, proper sensor calibration is essential to achieve accurate measurements, which directly impacts the system's performance capabilities. Careful consideration is given to the sensor calibration process to ensure high performance for sophisticated mechatronic systems.

Accurate position measurements are a necessary requirement for the high-precision capabilities of mechatronic systems. Position sensors that are used to position mechatronic systems, e.g. via feedback control, are described by their precision, accuracy, and their uncertainty. The uncertainty of a sensor measurement is a measure of confidence, taking into account both precision and accuracy. Using the dartboard analogy in Figure 1.1 the differences between these terminologies are described. Sensor calibration in this context involves correcting for systematic errors that negatively impact the positioning capabilities of the corresponding mechatronic system. By mitigating sensor inaccuracies through calibration, the overall positioning capabilities of such systems are enhanced. Consequently, the sensor accuracy is improved, while the precision of the sensor is dictated mostly by the manufacturer.

Direct calibration of position sensors can be challenging for some mechatronic systems. In such cases, a multi-stage calibration process can be employed to indirectly calibrate a low-accuracy sensor using a highly accurate reference sensor, ultimately improving the positioning capabilities of the mechatronic system. The attention will be limited to position sensors that consistently exhibit position-dependent inaccuracies. Often these inaccuracies can be removed with offline calibrations, i.e. when the system is not yet in operation. These methods typically yield high-precision and accurate calibration models.

Additionally, modeling errors resulting from sensor calibration, affect the system's achievable performance. These modeling errors are included in error budgeting frameworks, to measure the impact on the achievable performance. In error budgeting frameworks, multiple error sources are identified and managed to ultimately optimize the system's performance, e.g. the European Space Agency (ESA) follows specific instructions to budget pointing errors [1]. Sensor inaccuracies are common error sources and play a critical role in determining the total error budget that is achievable. Validating the impact of these modeling errors and incorporating their contributions to the total error budget enables further enhancement of a system's performance capabilities.

One prominent industry where the need for high-precision mechatronic systems have become increasingly vital is the field of optical satellite communication. Optical satellite communication has emerged as a groundbreaking technology in the field of space communication, offering numerous advantages over traditional radio frequency (RF) methods. Optical satellite communication enables high throughput, requires low power, and has high security for optical wavelengths. In recent years ESA has created a sophisticated laser communication network known as the European data relay system (EDRS). This project aims to provide a fast and secure communication net-

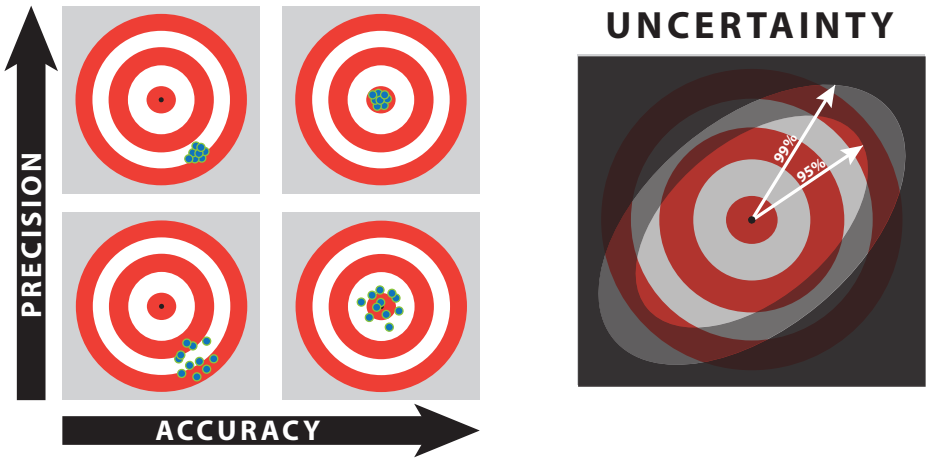


Figure 1.1: Dartboard analogy to visualize the difference between sensor precision, accuracy, and uncertainty. The bullseye of the dartboard depicts the true value, while the darts (●) are individual measurements of the sensor. Sensor uncertainty described the probability of a measurement being in a certain range.

work between satellites in orbit and satellite to ground stations. It consists of a large network of geostationary satellites equipped with laser communication terminals. In this regard, the laser communication terminals developed by TESAT played a pivotal role. Inter-satellite optical communication enabled a factor 10 higher transmission rates and increased security, while shortly after, laser links from space to ground were realized by mitigating atmospheric turbulence [2, 3].

Since optical beams are very narrow, it is essential to perform accurate acquisition and tracking of partnering terminals. Minor inaccuracies in beam-pointing can lead to significant performance losses. Accurate sensor calibration is crucial for ensuring stable and reliable acquisition and tracking in optical satellite communication systems.

1.1 Motivating example

The field of optical satellite communication is becoming more popular as technology advancements enabled accurate pointing of optical beams. The introduction of coarse pointing assemblies (CPA) in laser terminals enables fast alignment and accurate tracking between satellites [4]. The high demand for these components in laser terminals draws attention to the need for low size, weight, and power (SWaP) design [5]. Consequently, there is an increasing need for rapid and precise calibration of position sensors in CPAs.

A high-efficiency laser communication terminal that is currently being developed by TNO is shown in Figure 1.2. This terminal includes a 2DOF CPA which acts as the motivating example in this thesis. Calibration of this CPA using highly accurate measurement tools, e.g. a theodolite, is not directly available, since these high calibration tools are not suitable for dynamic testing. For this reason, a pointing test bench (PTB) has been developed by TNO in 2019 [6], which enables fast dynamic calibration and performance verification of coarse pointing assemblies. The PTB requires only a single static calibration using a theodolite and is able to calibrate CPAs afterward. This is an example where a multi-stage calibration is required to indirectly calibrate the CPA to a highly accurate measurement tool. This multi-stage calibration process is schematically visualized in Figure 1.3. The visualization illustrates a cascade of sensor calibrations, where each calibration stage affects the next.

Figure 1.2: Optical satellite communication is becoming an important solution to achieve safe and high-speed communication. This image showcases a high-efficiency miniature demonstrator for optical satellite communication that is being developed by TNO [7].

1.2 Research outline

In sensor calibration, a calibration model is created to eliminate systematic errors. Due to limited data, sensor reach, and other factors, the model uncertainty can be relatively high, and hence its reliability low. This thesis draws attention to cascaded calibration procedures for sensors, specifically addressing the calibration of position sensors in mechatronic systems. High-precision calibration of position sensors can also be performed by a National Metrology institute (NMi) [8], e.g. calibration methods using an AFM probe [9]. It is also common practice to perform sensor calibrations in-house, e.g. via the use of theodolites [10]. In a cascaded sensor calibration process, the uncertainty from one stage is carried forward and impacts the training data for all subsequent stages. By emphasizing the utilization of uncertainty information, resources can be effectively allocated, which enhances the modeling accuracy of the cascaded calibration process.

In [11], a variety of statistical sensor calibration methodologies are presented and applied to diverse models using both classical and Bayesian frameworks. Bayesian methods are particularly suitable for cascaded sensor calibration, as they offer a well-founded approach that is able to systematically integrate prior knowledge, and manage uncertainties at each calibration stage. In [12,13] Bayesian methods are applied to the calibration of Hall sensors, showing that combining sensor data with prior information enables accurate modeling with limited resources. Bayesian methods naturally extend to non-linear applications, where [14] presents a Bayesian inference-

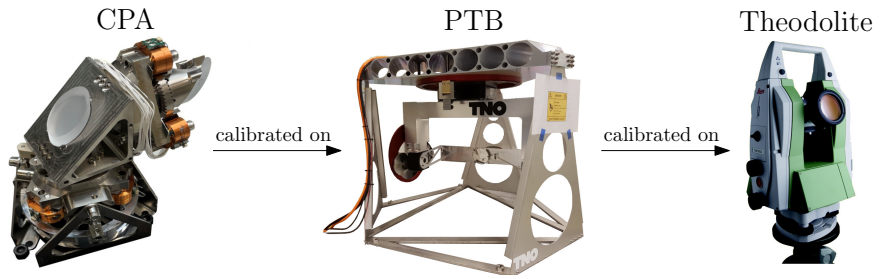


Figure 1.3: Overview of the calibration sequence for coarse pointing assemblies.

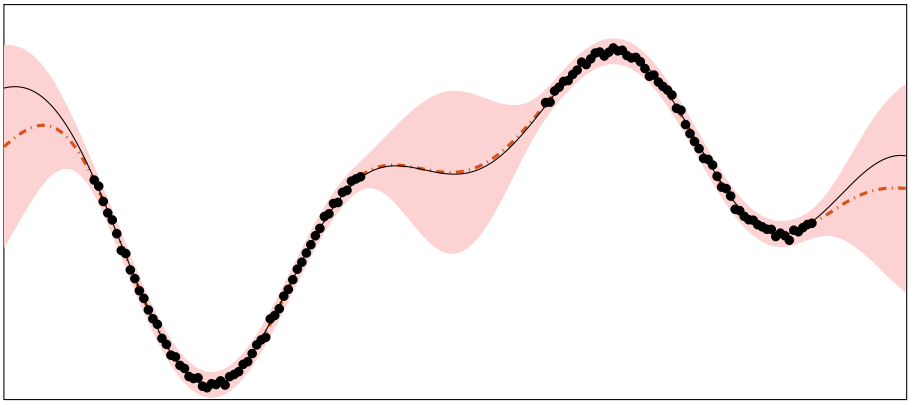


Figure 1.4: A visual illustration of Gaussian process regression.

based dynamic calibration method addressing sensor non-linearity and time drift, validated on both real and synthetic datasets.

Bayesian regression methods inherently regularize an optimization problem as it addresses overfitting by placing weights on the prior distribution parameters. Subsequently, these methods are popular modeling tools that are able to perform more accurate regularization in resource-limited environments compared to non-Bayesian techniques [15]. In [16, Chapter 4] the strengths of Bayesian frameworks are discussed regarding regularization, and showcases the powerful modeling tools these frameworks provide, e.g. empirical Bayes or maximum entropy principles.

Gaussian process regression (GPR) [17], visualized in Figure 1.4, is a non-parametric Bayesian regression method, which uses Gaussian processes (GPs) to model prior distributions. It enables flexible, fast, and accurate Bayesian modeling, and lately has become popular in the field of control systems [18–22]. This also applies to sensor calibration, where in [23] an efficient GP-based sensor calibration procedure is presented.

Although there are conclusive results for sequential calibrations using Bayesian frameworks [24], there is still a need for a comprehensive Bayesian framework regarding cascaded sensor calibrations. Sensor calibration modeling uncertainties, which are estimated in closed-form by Bayesian frameworks, are also not yet included in error budgeting for control systems. Comprehensive error budgeting techniques already exist and are widely used in industry for their quick and flexible design approaches, and can be combined with Bayesian sensor calibrations to create more accurate error budgets. This results in the following research questions (RQ) that will be covered in this thesis:

RQ1: What is the potential of a novel GPR framework in cascaded calibration of position sensors?

- Q1.1: What is the effect on the position estimation if multiple sensors are consecutively calibrated on each other?
- Q1.2: How can the variance of the position estimate be quantified using GP models, when considering a consecutive sequence of sensor calibrations?

RQ2: How can the uncertainty in GPR models, which are generated from (cascaded) sensor calibration, be incorporated into error budgeting frameworks to optimize for system performance?

- Q1.1: What methods can be used to obtain a power spectral density (PSD) model of the uncertainty in GPR models for (cascaded) sensor calibration?
- Q1.2: How can this model be utilized in error budgeting frameworks?

1.3 Overview of thesis

The aim of this thesis is to find a novel GPR framework for cascaded sensor calibrations. Additionally, these models can enable more accurate error budgeting for control systems by including modeling inaccuracies, which can enable more accurate control design. This thesis has two main contributions (C) followed up by a case study (CS) regarding cascaded sensor calibration for optical satellite applications:

C1: A GPR framework for cascaded calibration of position sensors is developed. The approach takes the model uncertainty of the first calibration step into account to arrive at a more accurate estimation in the subsequent calibration step. The effectiveness of the approach is demonstrated through Monte Carlo simulations on a reproducible case study, and it is shown that the developed calibration method yields significantly more accurate models of the sensor offsets than alternatives such as lookup tables. The results indicate that more accurate calibration of position sensors is possible with fewer resources.

C2: A PSD estimation and analysis framework for GPR modeling uncertainties is developed. This framework is able to critical power spectral information for error budgeting purposes. This framework also clarifies the role of the GP prior model and the training data in determining the spectral characteristics of the modeling uncertainty. Using this framework the inclusion of GPR modeling uncertainties in dynamic error budgeting frameworks is enabled.

CS: A case study on cascaded sensor calibration for optical satellite communication subsystems, utilizing contributions 1 and 2, is presented. Specifically, the motivating example provided in Section 1.1 is considered. In addition to contributions 1 and 2, the entire calibration procedure is thoroughly examined. This includes the optical alignment procedure between the CPA and PTB, which involves complex inverse kinematics.

This thesis is constructed as follows. First, in Chapter 2, a general framework is presented for the cascaded calibration of position sensors. This framework lays out a method for performing cascaded calibrations through GPR models. Second, in Chapter 3 a framework is presented for PSD estimation and analysis of GPR modeling uncertainties. The PSD characteristics of these calibration models impose an additional budget on the spectral requirements for the system. Afterward, by making use of the findings in previous chapters, Chapter 4 presents a case study regarding sensor calibration of CPAs in optical satellite communication terminals. In Chapter 5 a conclusion is drawn and an answer is provided for the research questions in this thesis. Finally in Chapter 6 recommendations are provided for future work.

Chapter 2

Cascaded Sensor Calibration using Bayesian Regression Models

The demand for high-precision mechatronic systems has been steadily increasing over the past years, due to rapid advancements in technologies e.g. in optical satellite communication [25]. To meet strict accuracy requirements, test beds are being developed to rapidly test optical subsystems and sometimes entire terminals [26]. The accuracy requirements can be met through sensor calibration on these sophisticated test beds. In this context, accuracy refers to the proximity of the position measurement to the actual position, in contrast to precision, which refers to the repeatability of the position measurements. The difference between these terminologies has been previously clarified through a dartboard analogy in Figure 1.1. Here, only position sensors that consistently exhibit position-dependent inaccuracies are considered.

Calibration via test beds is a prime example of cascaded sensor calibration. Essentially, this process reduces the inaccuracies of a position sensor in a mechatronic system by utilizing test bed calibration. Meanwhile, the position sensors on the test bed require calibration using high-accuracy calibration instruments. This was also evident in the motivating example, where the CPA is not directly accessible by high-precision theodolites, leading to the calibration procedure depicted in Figure 1.3. This chapter will describe the considered problem in broader terms. The calibration procedure boils down to the following order of position sensor calibrations: *1.* test bed sensor is calibrated on a high-accuracy calibration instrument, *2.* mechatronic system sensor is calibrated on the test bed sensor. Thus, this cascaded calibration process estimates a two-stage regression model, where the first-stage regression model provides training data for the second-stage regression model.

This chapter provides a formal problem description for cascaded calibration procedures. Using Bayesian regression methods, a closed-form expression is obtained for the uncertainty of a regression model. In multi-stage regression models, such as the previous example, uncertainty propagates through the regression models and affects the predictions. The uncertainty information in Bayesian regression methods, specifically GPR, is utilized to obtain more accurate corrections for the position-dependent inaccuracies that are present in the measurements of these sensors.

First, in Section 2.1 the concept of sensor calibration in the context of this thesis is clarified. Section 2.2 presents the problem description for a cascaded calibration procedure applied to position sensors, additionally, a multi-stage regression problem is presented. In Section 2.3 regularized regression techniques are related to a Bayesian perspective, which helps lay the foundation to GPR modeling for cascaded calibrations. In Section 2.4 a common example of a cascaded calibration is provided, for which the GPR method is utilized. Afterward, Section 2.5 provides Monte Carlo simulation results, which prove the effectiveness of said method for cascaded calibrations. Finally, the results are summarized and a conclusion is provided in Section 2.6.

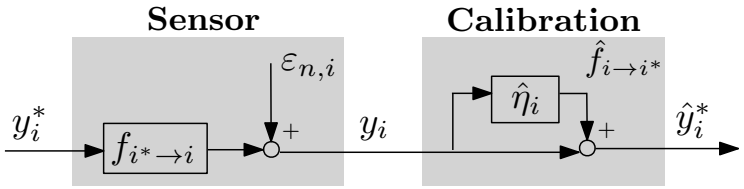


Figure 2.1: A graphical description of the definition for sensor calibration. The true position y_i^* is measured by a sensor with additive noise $\varepsilon_{n,i}$. The sensor measurement y_i is obtained by the mapping function $f_{i^*\to i}(y_i^*)$, which transforms the true position into a measurement with position-dependent inaccuracies. The position-dependent inaccuracies can be removed by means of calibration, i.e. by finding a function estimate $\hat{f}_{i\to i^*}(y_i)$ which maps the measurement back to an estimate of the true position.

2.1 Sensor calibration

Sensor inaccuracies can have multiple causes, e.g., manufacturing tolerances, mechanical wear, or imperfect assembly. Many types of deviations can exist between a sensor reading and the true value that should be measured [11], in this context only position-dependent inaccuracies are considered.

Consider a system denoted by i that is equipped with a position sensor known as S_i. The system's true position is denoted by $y_i^* \in \mathbb{R}$. The position sensor S_i of this system measures a position $y_i \in \mathbb{R}$, which contains position-dependent inaccuracies.

Definition 1 (Sensor calibration for position-dependent inaccuracies) *To calibrate a position sensor S_i is to find the function which describes a mapping of the position measurements y_i to the true position y_i^*, given by,*

$$y_i^* = f_{i\to i^*}(y_i), \tag{2.1}$$

where $f_{i\to i^*} : \mathbb{R} \to \mathbb{R}$ is a non-linear function, mapping a position measurement to the true position. This mapping serves to eliminate any systematic, position-dependent inaccuracies from the measurements. ■

A graphic description for sensor calibration as described by Definition 1 can be seen in Figure 2.1. The sensor measurement y_i is obtained by the mapping function $f_{i^*\to i}(y_i^*)$, which transforms the true position into a measurement with position-dependent inaccuracies. Note that $f_{i^*\to i}(y_i^*)$ is the inverse of the function $f_{i\to i^*}(y_i)$.

Assumption 1 (Invertability) *The function given by $y_i^* = f_{i\to i^*}(y_i)$ is assumed to be invertible, where the inverse is equal to*

$$(f_{i\to i^*}(y_i))^{-1} := f_{i^*\to i}(y_i^*) \quad \forall\ y_i, y_i^*. \tag{2.2}$$

■

After sensor calibration, an estimate of the true position can be found by

$$\hat{y}_i^* = \hat{f}_{i\to i^*}(y_i). \tag{2.3}$$

The sensor measurements are often close to the true value, thus for notational convenience, the function in (2.3) can be described by

$$\hat{f}_{i\to i^*}(y_i) = y_i + \hat{\eta}_i(y_i), \tag{2.4}$$

where $\hat{\eta}_i(y_i) \in \mathbb{R}$ depicts the estimated position-dependent inaccuracies. Temporal inaccuracies, such as sensor drift, are considered to be non-existent in this context.

Assumption 2 (No time-dependence) *Dependence of position measurements on time is assumed to average out, i.e.,* $f_{i\to i^*}(y_i, t) \approx f_{i\to i^*}(y_i)$*: the focus is restricted to position-dependent inaccuracies, and sensors are assumed to yield highly repeatable measurements during calibration.* ■

The sensor measurement y_i is corrupted by a zero-mean additive white Gaussian noise, given by $\varepsilon_{n,i}(t) \sim \mathcal{N}(0, \sigma^2_{n,i})$. The effect of zero-mean sensor noise is negligible.

Assumption 3 (Sensor noise is negligible) *Measurements* y_i *of* y_i^* *are corrupted by zero-mean additive Gaussian white noise* ε_i *with variance* $\sigma^2_{n,i}$*, which is assumed to be small with respect to position-dependent inaccuracies* $\eta_i(y_i)$*.* ■

Removing the position-dependent inaccuracies of sensor S_i requires knowledge of the true position of system i. However, the true position of system i can never be exactly known. Thus a reference system can be utilized by first aligning the reference system j with system i, such that their true position are equal. The measurement of sensor S_j can now be used to measure the inaccuracies of sensor S_i, given that sensor S_j is on average expected to be more accurate, i.e.

$$\begin{aligned} \mathbb{E}\left[\, y_j - y_j^* \,\right] < \mathbb{E}\left[\, y_i - y_i^* \,\right], \ \forall y_i, y_j \\ \mathbb{E}\left[\, \eta_j \,\right] < \mathbb{E}\left[\, \eta_i \,\right], \ \forall y_i, y_j \end{aligned} \tag{2.5}$$

where $\eta_i := \eta(y_i)$ and $\eta_j := \eta(y_j)$. In practice, when there are no other means of measurement, the reading of sensor S_j is considered to be the true position such that y_j serves as a proxy for y^*

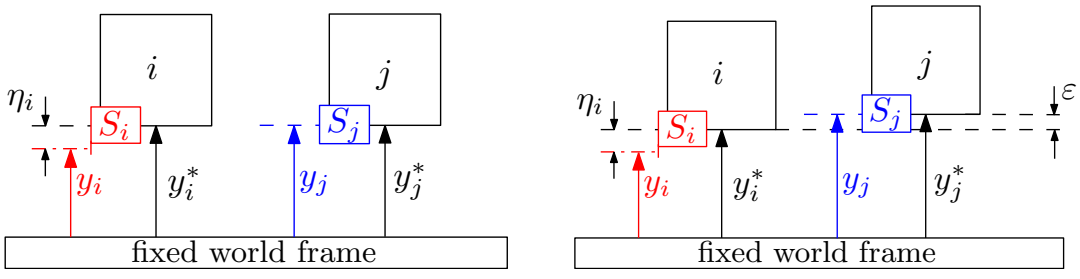
2.1.1 Sensor inaccuracies vs misalignment

Consider again a sensor S_i that measures the position of a system i, and similarly a more accurate sensor S_j that measures the position of system j, as depicted in Figure 2.2. To compare both sensor readings, systems i and j must be aligned such that their true positions are equal, i.e. $y_i^* = y_j^*$. However, aligning two sensors for means of calibration is not always perfect, which leads to a sensor misalignment denoted with ε.

Assumption 4 (Misalignment errors are negligible) *When a pair of sensor readings* (y_i, y_j) *is measured at a fixed point in time, it is assumed that misalignment errors are negligible w.r.t. sensor inaccuracies:*

$$|y_i^* - y_j^*| \ll |y_i^* - y_i|. \tag{2.6}$$

■



(a) aligned systems **(b)** misaligned systems (ε)

Figure 2.2: Schematic visualization of a single dimension calibration of sensor S_i. In this example, sensor S_i of system i is calibrated onto sensor S_j of system j. Sensor S_j is assumed to be measuring the true position of system i, i.e. the sensor inaccuracy of sensor S_j is equal to $\eta_j = y_j^* - y_j = 0$. In **(a)**, the case is presented when there is no misalignment between S_i and S_j, where $y_i^* = y_j^*$. In **(b)**, the case is presented when there is a misalignment in the true position ε between both sensors.

Given Definition 1 and Assumptions 1-4, the sensor reading from pair (y_i, y_j) can be utilized to estimate a function $f_{i\to j}(y_i)$ that maps a measurement from S_i to a measurement from S_j.The modeling accuracy of the estimate $\hat{f}_{i\to j}(y_i)$ may be restricted due to the limited number of measurements that accurate reference sensors permit. Additionally, data points obtained from accurate reference sensors are often expensive, such as with theodolite-based calibrations [27, 28]. A common solution to negate high costs is by means of calibration via sophisticated test beds. This results in a cascaded calibration procedure described in the following section.

2.2 Problem description: cascaded calibration of sensors

This section provides the problem description regarding the cascaded calibration procedure of a position sensor. First in section 2.2.1, the concept of cascaded sensor calibration is introduced. Then, section 2.2.2 defines the objective of cascaded calibration, along with the corresponding cost function that requires optimization. Finally, in section 2.2.3, the optimization is presented as a multi-stage regression problem.

2.2.1 Cascaded calibration of sensors

A schematic representation of a cascaded calibration is shown in Figure 2.3, where the aim is to remove any position-dependent inaccuracies of Sensor 1. This type of cascaded calibration procedure sees S_n as the absolute truth, as this sensor measurement has highest accuracy. Moreover, the objective is to obtain $f_{1\to n}$, which describes a mapping from the measurements on the least accurate sensor S_1 to the most accurate sensor S_n. This can be done by means of many sequential sensor calibrations as outlined by Definition 1.

2.2.2 Objective function for cascaded calibrations

The cost function can be defined as the continuous integral of squared errors between the estimate $\hat{f}_{1\to n}(y_1)$ and the true function $f_{1\to n}(y_1)$. To optimize for this cost function a regression problem is defined. The goal is to obtain a regression model (see A.1) in the form of

$$\hat{y}_n = \hat{f}_{1\to n}(y_1, \beta), \tag{2.7}$$

where β are the model parameters. The cost function is given by

$$\mathcal{J} = \left(\frac{\int_a^b [\hat{f}_{1\to n}(y_1, \beta) - f_{1\to n}(y_1)]^2 \, dy_1}{b-a} \right)^{\frac{1}{2}}. \tag{2.8}$$

where the sensor measurement has range $y_1 \in [a, b]$. In practice, the true function $f_{1\to n}$ that is being modeled is not actually measurable. Only via a series of calibrations can a measurement of sensor S_1 be mapped to a measurement of a more accurate sensor, i.e. a cascaded calibration procedure is required.

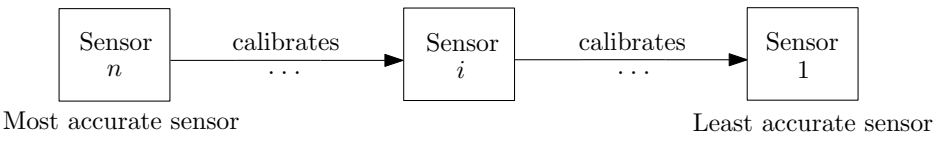


Figure 2.3: Schematic diagram of a cascaded calibration procedure. Several position sensors are being calibrated in sequential order. This means that any imperfections in the calibration of the more accurate sensors will affect the calibration and accuracy of the less accurate sensors down the chain.

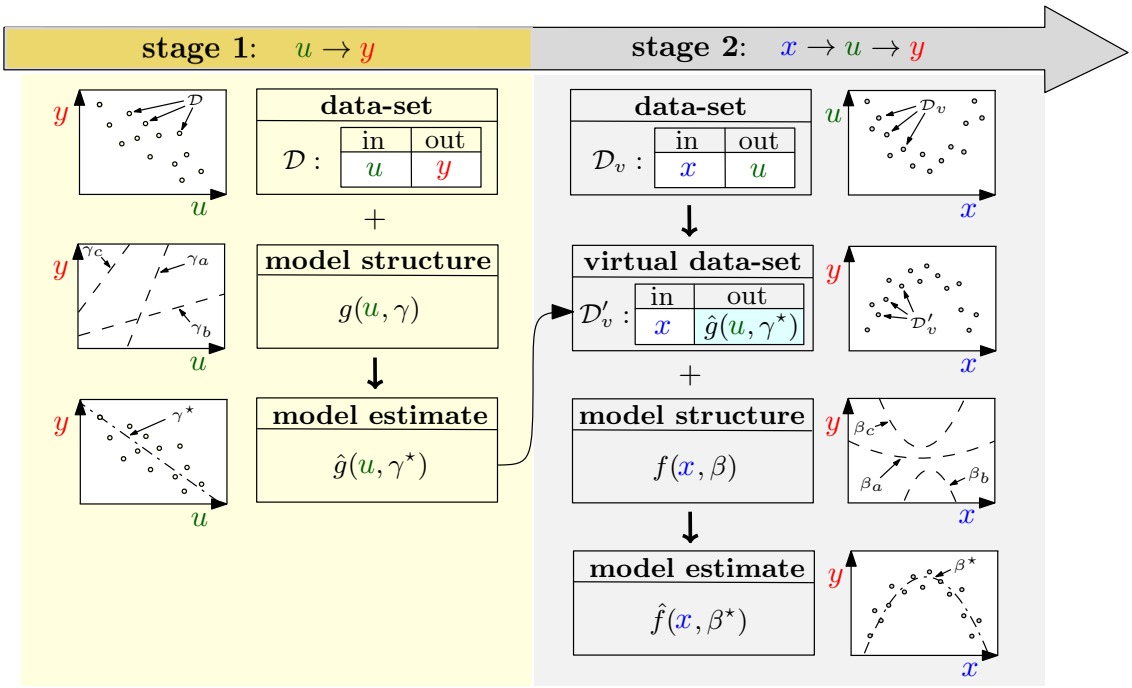


Figure 2.4: Simple schematic of a two-stage regression problem. The objective in a virtual regression problem ( ) is to learn a model estimate $\hat{f}(x, \beta^\star)$, where $\beta^\star$ are optimized model parameters. This model is learned via a virtual data-set $\mathcal{D}'_v$, i.e the dependent variable y is generated by another estimated model $\hat{y} = \hat{g}(u, \gamma^\star)$ that is found via a first-stage regression problem ( ).

2.2.3 Cascaded calibration through regression

The cascaded calibration problem can be defined as a multi-stage regression problem, where two different types of regressions can be identified: *(i)* a first-stage regression, where one sensor is calibrated on a highly accurate sensor that serves as a proxy for the true position, and *(ii)* a series of virtual regressions, given by Definition 2.

Definition 2 (Virtual regression) *In a virtual regression problem, the data set has dependent variables that are generated via an estimated model. In multi-stage regressions, every stage after the first can be characterized as a virtual regression. An example of a virtual regression problem can be seen in Figure 2.4. In literature, this type of regression is also referred to as instrumental variable regression or two-stage regression.* ■

A graphical description of the cascaded calibration procedure presented in Figure 2.3 can be posed as a multi-stage regression problem as schematically depicted in Figure 2.5. The two regression types that have been introduced above, are given by:

(i) The first-stage regression problem is the calibration of sensor S_{n-1} on sensor S_n. This is the only instance where measurements on sensor S_n can take place. Suppose the data-set obtained from measurements on both sensors is given by $\mathcal{D}_{n-1} = \{\bar{y}_{n-1,k}, \bar{y}_{n,k}\}_{k=1}^{N_{n-1}}$, where N_{n-1} is the number of available datapoints. An overline notation $\bar{y}_i$, implies measured data on sensor S_i. The objective is to solve

$$\beta^\star_{n-1} = \underset{\beta_{n-1}}{\operatorname{argmin}} \sum_{k=1}^{N_{n-1}} \left(\hat{f}_{n-1\to n}\left(\bar{y}_{n-1,k}, \beta_{n-1}\right) - f_{n-1\to n}\left(\bar{y}_{n-1,k}\right) \right)^2 + \lambda \left\| \beta_{n-1} \right\|^2, \qquad (2.9)$$

where $\beta^\star_{n-1}$ is the minimizer, and $\lambda \geq 0$ is a scaling parameter. The second term in (2.9) is for regularization, to prevent over- and underfitting of parameters.

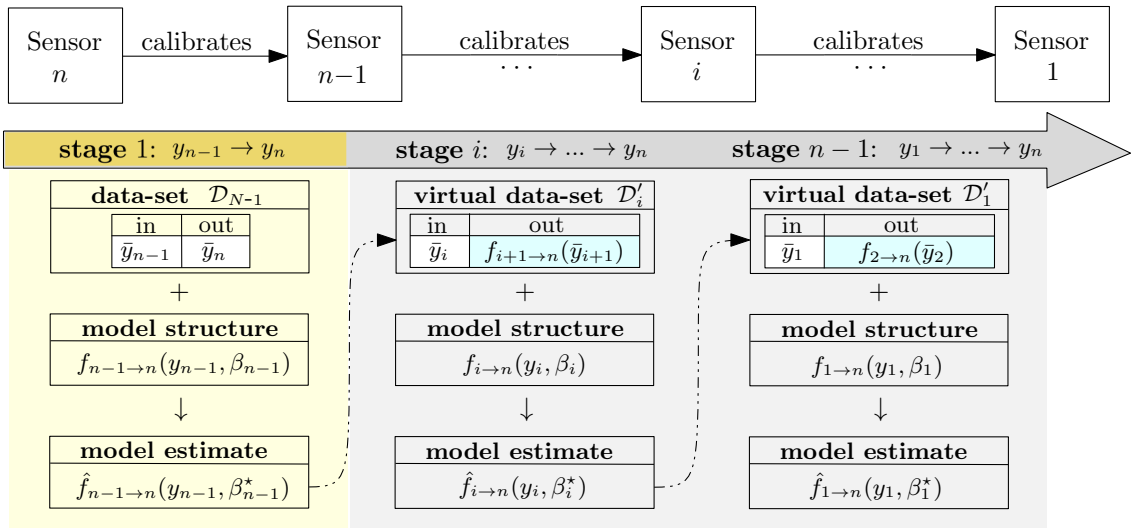


Figure 2.5: The cascaded calibration procedure is defined as a multi-stage regression problem, which consists of a first-stage regression problem ( ) and a series of virtual regression problems ( ).

(ii) Following the first-stage regression step, the remaining calibrations of sensors S_i, $i \in \{1, .., n-2\}$ require solving a virtual regression problem. Consider sensor S_i which is to be calibrated on sensor S_{i+1}. In the cascaded chain of calibrations, sensor S_{i+1} has already been calibrated, i.e. a function estimate $\hat{y}_n = \hat{f}_{i+1\to n}(y_{i+1})$ already exists. This means that the dataset given by $\mathcal{D}_i = \{\bar{y}_{i,k}, \bar{y}_{i+1,k}\}_{k=1}^{N_i}$, can be transformed into $\mathcal{D}'_i = \{\bar{y}_{i,k}, \hat{f}_{i+1\to n}(\bar{y}_{i+1,k})\}_{k=1}^{N_i}$, which is called the virtual dataset, where N_i is the number of available datapoints. The objective is to solve

$$\beta_i^\star = \underset{\beta_i}{\operatorname{argmin}} \sum_{k=1}^{N_i} \left(\hat{f}_{i\to n}\left(\bar{y}_{i,k}, \beta_i\right) - \hat{f}_{i+1\to n}\left(\bar{y}_{i+1,k}, \beta^\star_{i+1}\right) \right)^2 + \lambda \left\|\beta_i\right\|^2, \tag{2.10}$$

where $\beta_i^\star$ is the minimizer for the optimal model estimate given by $\hat{f}_{i\to n}\left(y_i, \beta_i^\star\right)$. This process has to be repeated for all sensors S_i in the cascaded calibration chain, such that the regression model given by (2.7) can be derived.

2.3 Bayesian interpretation of regularization

This section provides the most important results that relate well-known regression techniques and regularization methods to a Bayesian perspective. As kernel methods and regularization are not necessarily Bayesian in nature, this discussion will help in laying the foundation for GPR modeling, as kernel methods are a key component in Bayesian probability.

In the previous section, the problem definition and the corresponding regression problems that lead to an optimal model estimate were introduced. The non-linear model structure for which these nonlinear regression problems are solved, can be described by

$$f(x) = \phi(x)^T \beta, \tag{2.11}$$

where $x \in \mathcal{X} = \mathbb{R}^D$ and $\phi(x) \in \mathbb{R}^M$ is a nonlinear function that maps a D-dimensional input vector x into an M-dimensional (possibly infinite) feature space, i.e. $\phi : \mathbb{R}^D \to \mathbb{R}^M$. The corresponding model parameters are given by $\beta \in \mathbb{R}^M$.

Section 2.3.1 describes how to circumvent the high dimensional feature space $\mathbb{R}^M$ using kernels. Afterwards, section 2.3.2 establishes the link from kernel methods to Bayesian regression.

Section 2.3.3 explains the reason for the effectiveness of Bayesian regression methods for cascaded calibration. Finally, section 2.3.4 introduces the GPR method, which uses Bayesian regression to estimate predictions.

2.3.1 The representer theorem

The model in (2.11) is expressed as a linear combination of non-linear basis functions $\phi(x)$, also called features. This feature space is used to model a non-linear relationship between the input vector x and the output variable y. A commonly used set of basis functions are polynomial basis functions, where $\phi(x) = \begin{bmatrix} 1 & x & x^2 & \dots & x^{M-1} \end{bmatrix}$.

The nature of the non-linear relationship between the input variables and the output variable is often unknown or difficult to specify. In such cases, it is difficult to find an appropriate set of basis functions that captures the non-linear relationship accurately. Or it might be the case that the dimension of this feature space is large, which makes it computationally difficult to solve a regression problem. In such cases, kernel functions can be utilized, as they allow for the modeling of complex non-linear relationships without the need to specify or compute any basis functions. A formal definition for the kernel function is given by Definition 3.

Definition 3 (Kernel function) *Consider a basis function $\phi(x) \in \mathbb{R}^M$ with $x \in \mathcal{X}$ the input vector. The kernel function $k(x_A, x_B) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is given by the inner-product of the basis functions evaluated at two instances, i.e.*

$$k(x_A, x_B) = \phi(x_A)^T \phi(x_B). \tag{2.12}$$

This way the kernel function can measure the similarity between two data points x_A and x_B without directly computing the inner product of the feature vectors as defined in (2.12). ■

An intuitive approach to show how kernels can be used for non-linear regularized regression can be found in Appendix A.2. The representer theorem, given by Theorem 1, shows how an optimization problem can be solved using kernels. The solution is given by a linear combination of a kernel function and the corresponding coefficients. When the feature space lies in a higher dimension than the number of data points, the optimization can be solved more efficiently.

Theorem 1 (Representer theorem [29, 30]) *Consider a real-valued positive definite kernel $k(x_A, x_B) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, let $\mathcal{K}$ be its associated reproducing kernel Hilbert space (RKHS) [31], and $\mathcal{X}$ be non-empty. Given a data-set $\mathcal{D} = \{x_k, y_k\}_{k=1}^N$, where $x_k \in \mathcal{X} = \mathbb{R}^D$ and $y_k \in \mathbb{R}$, which describes the unknown non-linear process $y = f(x)$. The optimization to find minimizer $f^\star$ is given by*

$$\min_{f \in \mathcal{K}} \mathcal{J} = L(Y, f(X)) + \lambda \|f\|_{\mathcal{K}}^2, \tag{2.13}$$

where $L(Y, f(X))$ is an arbitrary loss function, with $X = \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix}^T \in \mathbb{R}^{N \times D}$ and $Y = \begin{bmatrix} y_1 & y_2 & \dots & y_N \end{bmatrix}^T \in \mathbb{R}^N$, and $\lambda \geq 0$ is a scaling parameter. The minimizer $f^\star$, at an arbitrary query point $x \in \mathbb{R}^D$, has the form

$$f^\star(x) = \sum_{k=1}^{N} \alpha_k k(x, x_k), \tag{2.14}$$

where $\alpha_i \in \mathbb{R}$. In matrix form, the minimizer is given by $f^\star(x) = k(x)^T \alpha$, where $k(x) := k(X, x) \in \mathbb{R}^N$ and $\alpha \in \mathbb{R}^N$. The proof can be found in [29]. ■

When solving a regression problem, the loss function $L(Y, f(X))$ in (2.13) is generally chosen to be the residual sum of squares given by $\sum_{i=1}^N (y_i - f(x_i))$. The minimizer resulting from this optimization problem is given by

$$f^\star(x) = k(x)^T \underbrace{(K + \lambda I_N)^{-1} Y}_{\alpha}, \tag{2.15}$$

where $I_N \in \mathbb{R}^{N\times N}$ the identity matrix and $K := k(X,X) \in \mathbb{R}^{N\times N}$ is the kernel matrix, with entries $K_{ij} = k(x_i, x_j)$.

2.3.2 Predictive distribution for Bayesian regression

Here the Bayesian treatment of regression for non-linear models is discussed. A generalized kernel method for regularized regression problems has been introduced in the previous section. In this section, equivalent regularized regression problems are viewed from a Bayesian perspective. The regression model that is found via Bayesian inference holds a similar form to that of (2.15), which resulted from the representer theorem given by Theorem 1. Specifically, Maximum a Posteriori (MAP) estimation is utilized to find the most likely values for the parameters that maximize the posterior distribution, given the observed data and prior beliefs.

Recall the non-linear model structure given by (2.11). The observed target value is given by

$$y = f(x) + \varepsilon_n, \tag{2.16}$$

where $y \in \mathbb{R}$ and $\varepsilon_n \sim \mathcal{N}(0,\Sigma)$ is an additive zero-mean Gaussian uncertainty to the measurement. Here, $\Sigma \in S^N_{++}$ is a symmetric positive definite covariance matrix. For further details on multivariate Gaussian distributions and its properties, see B.1.3. Using Baye's rule, prior beliefs over the weights can be updated given observations of the process. The updated distribution over the weights also referred to as the posterior distribution, is given by

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}, \qquad p(\beta|Y,X) = \frac{p(Y|X,\beta)p(\beta)}{p(Y|X)}, \tag{2.17}$$

where $Y = \begin{bmatrix} y_1 & y_2 & \dots & y_N \end{bmatrix}^T \in \mathbb{R}^N$ and $X = \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix}^T \in \mathbb{R}^{N\times D}$. The likelihood is given by

$$p(Y|X,\beta) \sim \mathcal{N}(\Phi^T\beta, \Sigma), \tag{2.18}$$

where $\Phi := \phi(X) = \begin{bmatrix} \phi(x_1) & \phi(x_2) & \dots & \phi(x_N) \end{bmatrix}^T \in \mathbb{R}^{N\times M}$. To express beliefs over the weights a prior can be specified over the parameters before looking at the observations. For simplicity, a zero mean Gaussian prior is put on the weights,

$$\beta \sim \mathcal{N}(0, \Sigma_p), \tag{2.19}$$

with $\Sigma_p \in \mathbb{R}^{N\times N}$ the prior covariance matrix. Using the definitions given by (2.18) and (2.19), the posterior distribution can be described by

$$p(\beta|X,Y) \sim \mathcal{N}(\beta^\star,\ \Phi\Sigma^{-1}\Phi^T + \Sigma_p^{-1}), \tag{2.20}$$

where $\beta^\star = \left(\Phi\Sigma^{-1}\Phi + \Sigma_p^{-1}\right)\Phi\Sigma^{-1}Y$, which is the MAP estimate. The predictive distribution for $f^\star(x) := \hat{f}(x)$ is computed by averaging the output of all possible model parameters w.r.t. the Gaussian posterior. Additionally, via mathematical reformulations (see A.3 for further elaboration), the predictive distribution is given by

$$\begin{aligned} p(\hat{f}(x)|X,Y,x) &\sim \mathcal{N}\left(\hat{f}(x), \text{cov}(\hat{f}(x))\right), \\ &\sim \mathcal{N}\left(k(x)^T(K+\Sigma)^{-1}Y,\ k(x,x) - k(x)^T\left(K+\Sigma\right)^{-1}k(x)\right), \end{aligned} \tag{2.21}$$

where the kernel function has form $k(x_A, x_B) = \psi(x_A)^T\psi(x_B)$, with $\psi(x) = (\Sigma_p)^{1/2}\phi(x)$, and the kernel matrix is defined by $K = \Phi^T\Sigma_p\Phi$. Additionally, an estimate for the uncertainty of the prediction is provided, i.e. a closed-form expression of the covariance $\text{cov}(\hat{f}(x))$ is estimated. Note that the mean of the predictive distribution $\hat{f}(x)$ in (2.21) has the same form as the minimizer $f^\star(x)$ given by (2.15) This states that in Bayesian regression, given a positive definite kernel function, the likelihood covariance acts as a regularizer. This property is formally stated in Lemma 1.

Lemma 1 (Regularizer for Bayesian regression) *Consider a finite set of observations* $\mathcal{D} = \{x_k, y_k\}_{i=1}^{N}$*, i.e. N is finite. Recall the regularizer term* $\lambda\|f\|_{\mathcal{K}}^2$ *in the regularized regression optimization given by* (2.13)*. Considering the likelihood has a Gaussian distribution, e.g.* $p(Y|X,w) \sim \mathcal{N}(\Phi^T w, \Sigma)$*, then the scaling parameter* λ *for Bayesian regression is given by the covariance matrix* Σ *of the likelihood.* ■

2.3.3 Bayesian inference for virtual regressions

Bayesian regression methods are an especially powerful tool when considering virtual regression problems, as defined by Definition 2. Section 2.2.3 states that cascaded calibration procedures consist of one regular regression problem and a total of $n-2$ virtual regression problems that follow. Bayesian regression methods can lead to more accurate optimization of a model estimate in case of virtual regression problems. This property has been stated formally in Conjecture 1.

In a two-stage regression, as visualized in Figure 2.4, the objective is to find a model estimate $\hat{f}(x)$. In the virtual regression, the finite dataset of the process is given in the form, $\mathcal{D}_v' = \{X, \hat{g}(U)\}$, where the dependent variable y, is estimated by a model $\hat{y} = \hat{g}(u)$.

The model estimate $\hat{g}(y)$ is learned via Bayesian regression. Given finite dataset $\mathcal{D} = \{U, Y\}$, the predictive distribution is given by,

$$p(\hat{g}(u)|\mathcal{D}, u) \sim (\hat{g}(u), \mathrm{cov}(\hat{g}(u))) \,. \tag{2.22}$$

The predictive distribution for the model estimate $\hat{f}(x)$ is also learned via Bayesian regression, through the prior and likelihood distribution, given by (2.21). However, in this case, the likelihood covariance Σ is evaluated from the predictive distribution of $\hat{g}(u)$ (2.22). This results in a precise estimate for the likelihood covariance,

$$\Sigma = \mathrm{cov}(\hat{g}(U)). \tag{2.23}$$

In essence, the two-stage regression problem can be solved accurately through Bayesian methods. The uncertainty of the first-stage regression is used as the regularizer for the second-stage virtual regression, which is supported by Lemma 1. This provides evidence that the uncertainty information of the first-stage regression problem can be utilized in the second-stage virtual regression, which is formally stated in Conjecture 1.

Conjecture 1 (Bayesian regression for virtual regression problems) *In a two-stage regression problem the second stage is referred to as the virtual regression. The uncertainty that propagates to the virtual regression can be utilized in the regression problem using the Bayesian framework. This property is supported by Lemma 1.* ■

Evidence for Conjecture 1 is provided by means of Monte Carlo simulations in section 2.5, which shows a more accurate model estimation in a cascaded calibration setting.

2.3.4 Gaussian process regression

Gaussian process regression (GPR) is a type of non-linear Bayesian regression method, where the prior is defined by a Gaussian process. Instead of expressing beliefs over the weights as done in (2.19), they are expressed over the function.

Definition 4 (Gaussian process [17]) *A Gaussian process (GP) is a collection of infinitely many random variables, e.g.* $\{f(x)|x \in \mathbb{R}^D\}$*, any finite number of which has a joint multivariate Gaussian distribution.* ■

A GP can be fully specified by its mean function $m(x)$ and kernel (or covariance function) $k(x_A, x_B)$ given by

$$f(x) \sim \mathcal{GP}(m(x), k(x_A, x_B)), \tag{2.24}$$

with

$$\begin{aligned} m(x) &= \mathbb{E}\left[f(x)\right], \\ k(x_A, x_B) &= \mathbb{E}\left[\left(f(x_A) - m(x_A)\right)\left(f(x_B) - m(x_B)\right)\right]. \end{aligned} \tag{2.25}$$

The choice for the mean and kernel functions directly affects the behavior of the prior distribution and, consequently, the resulting posterior- and predictive distributions. The previous section presented a predictive distribution (2.21) with a zero-mean prior distribution. Now the prior can be modeled by a GP as defined by (2.24). This form of Bayesian regression is referred to as *Guassian process regression*.

In GPR, given a data-set $\mathcal{D} = \{X, Y\}$ where the observed target values are given by (2.16), the predictive distribution of $f(x)$ for an arbitrary query input x is given by

$$\begin{aligned} p(\hat{f}(x)|X, Y, x) &\sim \mathcal{N}\left(\hat{f}(x), \mathrm{cov}(\hat{f}(x))\right), \quad \text{where} \\ \hat{f}(x) &:= \mathbb{E}\left[\hat{f}(x)\right] = m(x) + k(x)^T \left(K + \Sigma\right)^{-1} \left(Y - m(X)\right), \\ \mathrm{cov}(\hat{f}(x)) &= k(x, x) - k(x)^T \left(K + \Sigma\right)^{-1} k(x). \end{aligned} \tag{2.26}$$

In section 2.4 a GPR example is given for a cascaded calibration procedure. It also includes a demonstration on how to choose the mean and kernel function.

2.4 Cascaded sensor calibration via test beds

A common practice for calibrating low-accuracy position sensors in mechatronic systems involves a cascaded calibration procedure that utilizes sophisticated test beds. This section presents a problem description of such a cascaded calibration which involves a total of three position sensors: a low-accuracy position sensor in a mechatronic system, a test bed sensor, and a high-accuracy calibration instrument.

In section 2.4.1 a detailed cascaded calibration procedure is presented for a low-accuracy position sensor of a mechatronic system. In section 2.4.2 the objective function and the corresponding regression problems for this example are defined. Section 2.4.3 works out the GPR models, where-after in section 2.4.4 the model structure is defined. In section 2.4.5 a hyper-parameter optimization is presented for the GPR model. The GPR algorithm for cascaded calibration is presented in 2.4.6.

2.4.1 Calibration procedure

For the rapid calibration of low-cost sensors, often test beds are utilized. Only a single calibration is then required of the test bed, for which a highly accurate manual calibration instrument is required. A schematic example of this type of procedure is displayed in Figure 2.6, with three sensors. In this example, the mechatronic system has an angular position sensor, called S_1, that needs calibration. The test bed has sensor S_2, and the manual calibration instrument has sensor S_3. The objective is to obtain a model $\hat{f}_{1\to3}$ of $f_{1\to3}$, while $f_{1\to3}$ cannot be observed directly. This is because the manual calibration instrument cannot be directly aligned with the mechatronic system for calibration, due to two reasons:

1. it is not economically viable to calibrate the sensors of multiple mechatronic systems using a manual calibration instrument,

2. and some mechatronic systems may be built too compactly to allow physical access for manual calibration.

The indirect mapping of S_1 onto S_3 is visualized in Figure 2.7. This two-step cascaded calibration procedure is described in detail in Procedure 1. This cascaded calibration procedure is derived from the general problem description described in Section 2.2. The two-step procedure

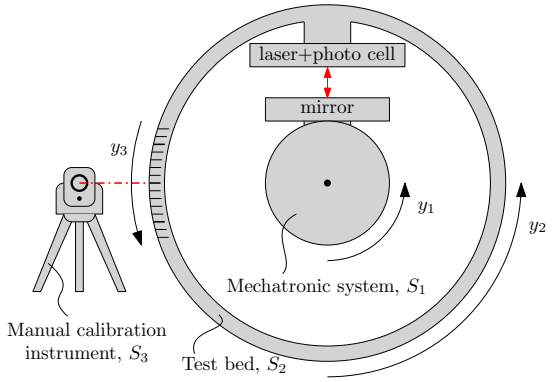


Figure 2.6: A mechatronic system with angular position sensor S_1 is optically linked for calibration with a test bed, with its own sensor S_2. The test bed itself is calibrated on a highly accurate manual measuring instrument S_3.

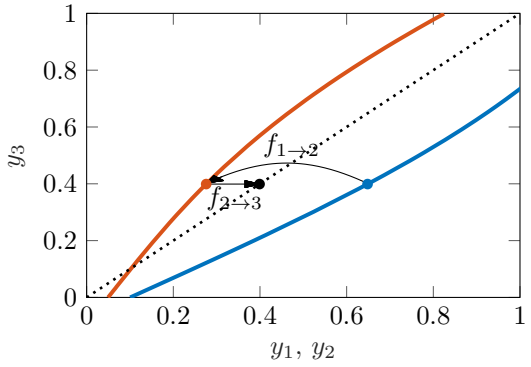


Figure 2.7: Sensor S_1 (—) is calibrated on a test best with sensor S_2 (—), which is in turn calibrated on S_3 (·····). This procedure constructs an indirect mapping from the less-accurate measurements y_1 to a highly accurate measurement y_3.

Procedure 1 Cascaded calibration of S_1 with n=3

1: Align systems 2 and 3 and obtain a data-set $\mathcal{D}_2 = \{\bar{y}_{2,k}, \bar{y}_{3,k}\}_{k=1}^{N_2}$. Use these observations to fit a function $\hat{y}_3 = \hat{f}_{2\to3}(y_2)$.
2: Align systems 1 and 2 and obtain a data-set $\mathcal{D}_1 = \{\bar{y}_{1,k}, \bar{y}_{2,k}\}_{k=1}^{N_1}$. Then construct virtual data-set $\mathcal{D}_1' = \{\bar{y}_{1,k}, \hat{f}_{2\to3}(\bar{y}_{2,k})\}_{k=1}^{N_1}$ using the model of Step 1, and use these virtual observations to fit a function $\hat{y}_3 = \hat{f}_{1\to3}(y_1)$.

consists of a first-stage regression and a single virtual regression. Since measurements of S_3 are labor-intensive and time-consuming, the first-stage regression model $\hat{f}_{2\to3}$ might be based on a limited amount of data ($N_2 \ll N_1$), and consequently, it may contain modeling errors. This potentially deteriorates the accuracy of $\hat{f}_{1\to3}$ w.r.t. the true $f_{1\to3}$.

The following section outlines how the construction of a model $\hat{f}_{1\to3}(y_1)$ through Steps 1-2 of Procedure 1 can be formally framed as a series of regression problems.

2.4.2 Objective function and cascaded regression problem

The objective function for this motivating example is given by the cost in (2.8), where n=3, i.e. there are in total three sensors in the cascaded calibration chain. The cost to be optimized is given by

$$\min_{\beta_1,\beta_2} \mathcal{J} = \left(\frac{\int_a^b [\hat{f}_{1\to3}(y_1, \beta_1) - f_{1\to3}(y_1)]^2 \, dy_1}{b-a} \right)^{\frac{1}{2}}. \tag{2.27}$$

where $y_1 \in [a, b]$. From Procedure 1 the cost function is subject to two regression problems, an first-stage regression and a following virtual regression, given by

$$\begin{aligned} \beta_2^\star &= \underset{\beta_2}{\operatorname{argmin}} \sum_{k=1}^{N_2} \left(\hat{f}_{2\to3}(\bar{y}_{2,k}, \beta_2) - f_{2\to3}(\bar{y}_{2,k}) \right)^2 + \lambda \|\beta_2\|^2, \\ \beta_1^\star &= \underset{\beta_1}{\operatorname{argmin}} \sum_{k=1}^{N_1} \left(\hat{f}_{1\to3}(\bar{y}_{1,k}, \beta_1) - \hat{f}_{2\to3}(\bar{y}_{2,k}, \beta_2^\star) \right)^2 + \lambda \|\beta_1\|^2. \end{aligned} \tag{2.28}$$

Gaussian process regression (GPR), introduced in section 2.3.4, will be used to solve for $\beta_1^\star$ and $\beta_2^\star$. GPR models are a powerful non-parametric tool that can model all kinds of processes,

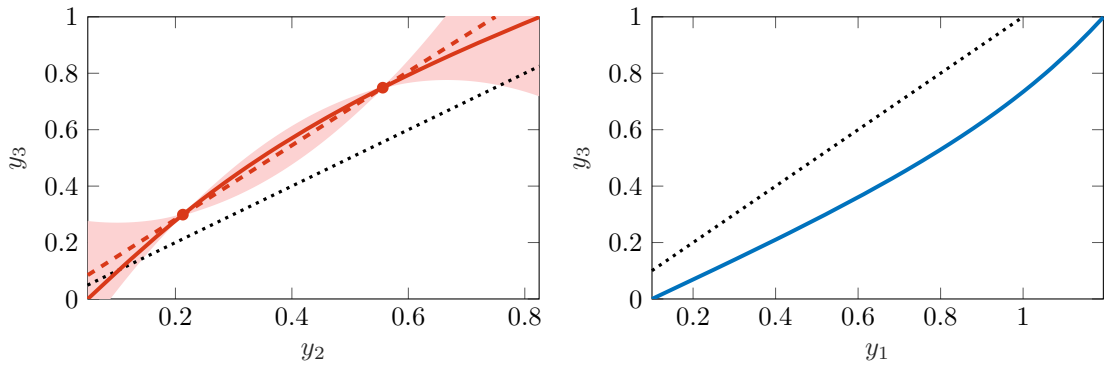


Figure 2.8: Here Step 1 of Procedure 1 is visualized, consider the position-dependent inaccuracies as introduced in figure 2.7, where the unknown mapping functions $f_{2\to3}$ (—) and $f_{1\to3}$ (—) are depicted. Given data-set $\mathcal{D}_2$ (●), a GPR model is generated which gives a predictive mean $\hat{f}_{2\to3}$ (- -) and predictive covariance $\text{cov}(\hat{f}_{2\to3})$ ( ).

including non-linear functions. Additionally GPR models are able to estimate an uncertainty measure over the predictions. The latter property, as stated by Conjecture 1, is a key characteristic that enables improved model estimation for cascaded calibrations.

2.4.3 Gaussian process regression for cascaded calibrations

Here a solution is presented for the regression problems, presented in Procedure 1 via GPR models. GPR models are able to estimate the uncertainty at the predictions, which makes them inherently strong for cascaded approaches.

Procedure 1 - Step 1: calibration of sensor S_2

The first-stage regression problem given by Step 1 of Procedure 1, is the calibration of sensor S_2 on sensor S_3. The function that maps the measurement y_2 to a measurement y_3 is given by $f_{2\to3}(y_2)$. The GP prior is defined as

$$f_{2\to3}(y_2) \sim \mathcal{GP}(m_2(y_2), k_2(y_{2,A}, y_{2,B})). \tag{2.29}$$

Given the dataset $\mathcal{D}_2 = \{\bar{y}_{2,k}, \bar{y}_{3,k}\}_{k=1}^{N_2}$, the likelihood is given by

$$p(\bar{Y}_3|\bar{Y}_2, \beta_2) \sim \mathcal{N}\left(f_{2\to3}(\bar{Y}_2, \beta_2), \Sigma_{\bar{Y}_3}\right), \tag{2.30}$$

where $\bar{Y}_i = \begin{bmatrix}\bar{y}_{i,1} & \bar{y}_{i,2} & \dots & \bar{y}_{i,N_2}\end{bmatrix}^T \in \mathbb{R}^{N_2\times N_2}$ and $\Sigma_{\bar{Y}_3} = \sigma_n^2 I$ the covariance matrix of the observations. The predictive distribution for any arbitrary measurements $\mathbf{y}_2 \in \mathbb{R}^{M_2}$ becomes

$$\begin{aligned} p(\hat{f}_{2\to3}(\mathbf{y}_2) \mid \bar{Y}_2, \bar{Y}_3) &\sim \mathcal{N}\left(\hat{f}_{2\to3}(\mathbf{y}_2),\ \text{cov}(\hat{f}_{2\to3}(\mathbf{y}_2))\right), \quad \text{where} \\ \hat{\mathbf{y}}_3 &:= \mathbb{E}\left[\hat{f}_{2\to3}(\mathbf{y}_2)\right] = m_2(\mathbf{y}_2) + k_2(\mathbf{y}_2)^T\left[K_2 + \Sigma_{\bar{Y}_3}\right]^{-1}(\bar{Y}_3 - m_2(\bar{Y}_2)), \\ \text{cov}(\hat{\mathbf{y}}_3) &:= \text{cov}(\hat{f}_{2\to3}(\mathbf{y}_2)) = k_2(\mathbf{y}_2, \mathbf{y}_2) - k_2(\mathbf{y}_2)^T\left[K_2 + \Sigma_{\bar{Y}_3}\right]^{-1} k_2(\mathbf{y}_2), \end{aligned} \tag{2.31}$$

where $k_2(\mathbf{y}_2) := k(\bar{Y}_2, \mathbf{y}_2) = k_2(\mathbf{y}_2, \bar{Y}_2)^T \in \mathbb{R}^{N_2\times M_2}$ and $K_2 := k_2(\bar{Y}_2, \bar{Y}_2) \in \mathbb{R}^{N_2\times N_2}$. The mean prediction $\hat{y}_3 := \mathbb{E}[\hat{f}_{2\to3}(y_2)]$ maps any given measurement y_2 into a measurement y_3. It is essential to notice that (2.31) provides an analytical expression for the uncertainty of our prediction, i.e. the covariance estimate $\text{cov}(\hat{f}_{2\to3}(y_2))$.

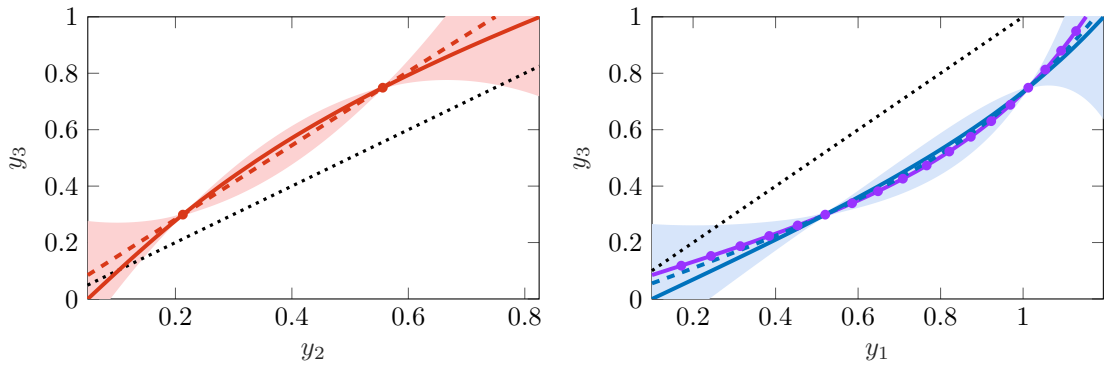


Figure 2.9: Here Step 2 of Procedure 1 is visualized. Given virtual dataset $\mathcal{D}_1'$ (●), a GPR model is generated which gives a predictive mean $\hat{f}_{1\to3}$ (- -) and predictive covariance $\text{cov}(\hat{f}_{1\to3})$ ( ). It is essential to notice that the high uncertainty of the first-stage calibration influences the predictive mean and covariance estimate of the following virtual regression.

Procedure 1 - Step 2: calibration of sensor S_1

Following the first-stage regression, a virtual regression is performed to obtain an estimate for the objective function $f_{1\to3}(y_1)$, that maps the measurement y_1 to a measurement y_3. The GP prior is defined as

$$f_{1\to3}(y_1) \sim \mathcal{GP}(m_1(y_1), k_1(y_{1,A}, y_{1,B})). \tag{2.32}$$

A data-set $\mathcal{D}_1 = \{\bar{y}_{1,k}, \bar{y}_{\underline{2},k}\}_{k=1}^{N_1}$ is measured, and transformed into a virtual data-set $\mathcal{D}_1' = \{\bar{y}_{1,k}, \hat{y}_{3,k}\}_{k=1}^{N_1}$, where $\hat{y}_{3,k} := \hat{f}_{2\to3}(\bar{y}_{\underline{2},k})$. The likelihood is given by

$$p(\hat{Y}_3|\bar{Y}_1, \beta_1) \sim \mathcal{N}\left(f_{1\to3}(\bar{Y}_1, \beta_1), \Sigma_{\hat{Y}_3}\right). \tag{2.33}$$

Recall from Lemma 1 and Conjecture 1 that for this virtual regression problem, the likelihood covariance $\Sigma_{\hat{Y}_3}$ is fully described by the model uncertainty $\text{cov}(\hat{f}_{2\to3}(\bar{Y}_{\underline{2}}))$, i.e

$$\begin{aligned} \hat{Y}_3 &:= [\hat{f}_{2\to3}(\bar{y}_{\underline{2},1}), \dots, \hat{f}_{2\to3}(\bar{y}_{\underline{2},N_1})]^\top, \\ \Sigma_{\hat{Y}_3} &:= \text{cov}(\hat{f}_{2\to3}(\bar{Y}_{\underline{2}})). \end{aligned} \tag{2.34}$$

The predictive distribution for any arbitrary measurements $\mathbf{y}_1 \in \mathbb{R}^{M_1}$ becomes

$$\begin{aligned} p(\hat{f}_{1\to3}(\mathbf{y}_1) \mid \bar{Y}_1, \bar{Y}_2) \sim \mathcal{N}\left(\hat{f}_{1\to3}(\mathbf{y}_1), \text{cov}(\hat{f}_{1\to3}(\mathbf{y}_1))\right), \quad \text{where} \\ \tilde{\mathbf{y}}_3 := \mathbb{E}\left[\hat{f}_{1\to3}(\mathbf{y}_1)\right] = m_1(\mathbf{y}_1) + k_1(\mathbf{y}_1)^T \left[K_1 + \Sigma_{\hat{Y}_3}\right]^{-1} (\hat{Y}_3 - m_1(\bar{Y}_1)), \\ \text{cov}(\tilde{\mathbf{y}}_3) := \text{cov}(\hat{f}_{1\to3}(\mathbf{y}_1)) = k_1(\mathbf{y}_1, \mathbf{y}_1) - k_1(\mathbf{y}_1)^T \left[K_1 + \Sigma_{\hat{Y}_3}\right]^{-1} k_1(\mathbf{y}_1), \end{aligned} \tag{2.35}$$

where $k_1(y_1) := k(\bar{Y}_1, \mathbf{y}_1) = k_1(\mathbf{y}_1, \bar{Y}_1)^T \in \mathbb{R}^{N_1 \times M_1}$ and $K_1 := k_1(\bar{Y}_1, \bar{Y}_1) \in \mathbb{R}^{N_1 \times N_1}$. The mean prediction $\hat{f}_{1\to3}(y_1) := \mathbb{E}[\hat{f}_{1\to3}(y_1)]$ maps any given measurements y_1 into a measurement y_3. Figure 2.9 shows that by choosing the regularizer as the likelihood covariance, the mean estimate diverges from the virtual observations and gets closer to the actual function $\hat{f}_{1\to3}(y_1)$.

2.4.4 Model structure for position sensor calibrations

In GPR, the likelihood (2.18) and a GP prior (2.24) describe the predictive distribution (2.26). A GP prior distribution has a mean function and kernel function, where the latter captures the underlying assumptions about the correlation and smoothness between different points in the input space.

Recall (2.4), which showed that a calibration function can be seen as the sum of a sensor measurement y_i and a position-dependent inaccuracy $\eta_i(y_i)$. As a result, the mean function can be chosen to be $m(y_i) = y_i$, while the kernel function describes the basis features of $\eta_i(y_i)$.

Although there are many kernels to choose from [17], in this chapter the attention is restricted to the squared exponential kernel (or radial basis function (RBF)), given by

$$k_i(y_{i,A}, y_{i,B}) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}(y_{i,A} - y_{i,B})^2\right), \tag{2.36}$$

where ℓ_i and $\sigma_{f,i}^2$ are the characteristic length scale and the magnitude of the prior variance, respectively. Where-as the likelihood covariance, when unknown, can also be parameterized with the noise variance $\sigma_{n,i}^2$. Parameters that are defined in the GP prior and the likelihood are called hyper-parameters, given by $\Theta_i = \{l_i, \sigma_{f,i}^2, \sigma_{n,i}^2\}$.

2.4.5 Hyper-parameter optimization

The hyper-parameters for GPR, which involves a GP prior and Gaussian likelihood, can be estimated by means of *empirical Bayes*. The key idea for empirical Bayes is to find these hyper-parameters by using the data itself. This can be done by evaluating the marginal likelihood (2.17) for different combinations of hyperparameters and selecting the set of hyperparameters that maximize the log marginal likelihood, e.g. given data-set $\mathcal{D}_i = \{\bar{Y}_i, \bar{Y}_j\}$ the optimization for Θ_i is given by

$$\log\, p(Y_j|Y_i, \Theta_i) = -\frac{1}{2}Y_j^\top \tilde{K}_i^{-1} Y_j - \frac{1}{2}\log\left|\tilde{K}_i\right| - \frac{N_i}{2}\log 2\pi, \tag{2.37}$$

where $\tilde{K}_i = K_i(Y_i, Y_i) + \Sigma_{Y_j}$. This expression can be maximized w.r.t. Θ_i using an optimization algorithm for non-convex problems. The log marginal likelihood (2.37) is dependent on the likelihood covariance Σ_{Y_j}.

2.4.6 Summary

In this section a common example has been presented, where low-accuracy position sensors S_1 are calibrated via a test bed S_2, which also requires a one-time calibration using a highly accurate calibration instrument S_3. This is referred to as cascaded calibration, where multiple sensor calibrations are performed sequentially. The three sensors given by S_1, S_2 and S_3 measure a position y_1, y_2 and y_3 respectively.

The objective is to find a function which maps the position measurement y_1 onto position measurement y_3. To solve for this cascaded problem, GPR models have been selected as a suitable method. Here, Algorithm 1 presents the sequence of steps for a cascaded calibration with three position sensors.

Algorithm 1 Cascaded calibration via Bayesian inference

Require: Data-sets $\mathcal{D}_1$, $\mathcal{D}_2$, test points $y_1 \in \mathbb{R}^{M_1}$.
1: Specify kernel functions k_1, k_2 with initial hyper-parameters $\Theta_{i,0}$, see Section 2.4.4.
2: Find optimal hyper-parameters $\Theta_2^\star$ by maximization of $\log\, p(\bar{Y}_3|\bar{Y}_2, \Theta_2)$, see (2.37).
3: Compute $\hat{Y}_3 = \mathbb{E}[\hat{f}_{2\to3}(\bar{Y}_2)] \in \mathbb{R}^{N_1}$ and $\mathrm{cov}(\hat{Y}_3) = \mathrm{cov}(\hat{f}_{2\to3}(\bar{Y}_2)) \in \mathbb{R}^{N_1 \times N_1}$ with (2.31).
4: Find optimal hyper-parameters $\Theta_1^\star$ by maximization of $\log\, p(\hat{Y}_3|\bar{Y}_1, \Theta_1)$, see (2.37).
5: Compute $\tilde{y}_3 = \mathbb{E}[\hat{f}_{1\to3}(y_1)] \in \mathbb{R}^{M_1}$ with (2.35).
6: **return** $\tilde{y}_3$

In the following section Monte Carlo simulations are presented, which show that for the purpose of cascaded calibrations, GPR models are highly capable of estimating and utilizing uncertainty information to achieve higher accuracy.

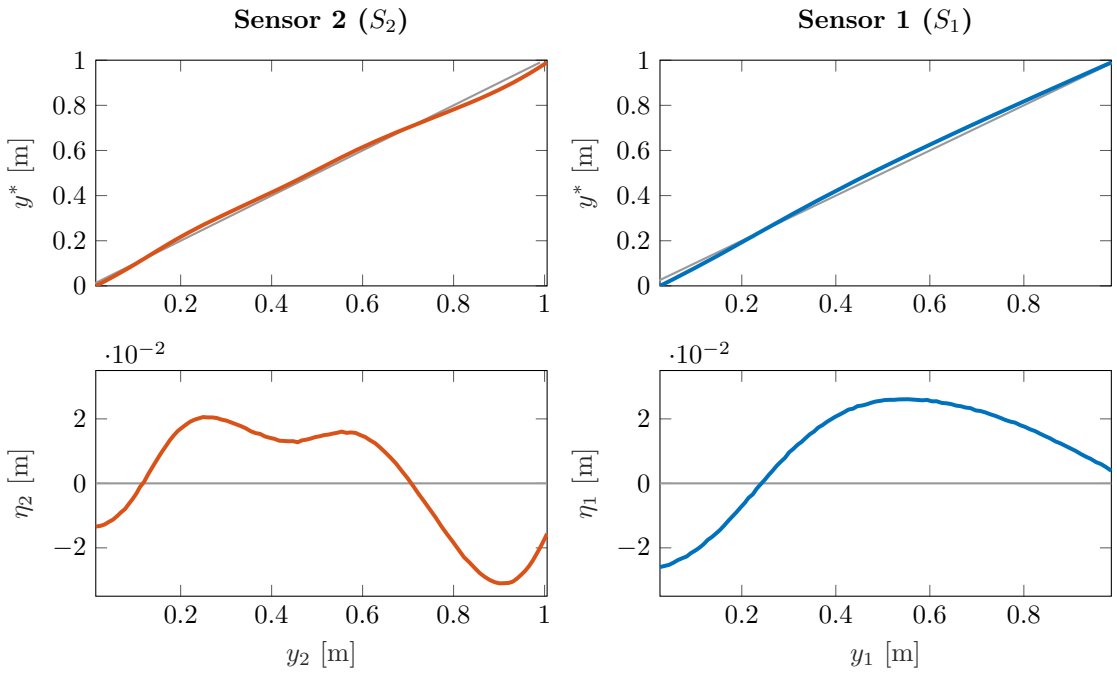


Figure 2.10: A sample set of functions from (2.38), with $f_{2\to 2^*}(y_2)$ (—) and $f_{1\to 1^*}(y_1)$ (—). The bottom two figures show the position-dependent inaccuracies $\eta_i(y_i) = y_i^* - y_i$, $i \in \{1, 2\}$.

2.5 Monte Carlo simulations

This section provides proof of the effectiveness of the developed cascaded calibration approach, demonstrated through Monte Carlo simulations. In section 2.5.1 the simulation set-up is given first, where-after in section 2.5.2 the results are presented.

2.5.1 Simulation set-up

Consider that each system in the cascaded calibration procedure is aligned at once and is positioned at a certain y^*, e.g. $y^* = y_1^* = y_2^* = y_3^*$. Each sensor S_i, $i \in \{1, 2, 3\}$, has measurement y_i, given by

$$\begin{aligned} y_3 &= y^* + \varepsilon, \\ y_2 &= y^* + \sum_{k=1}^{N_s} a_k \sin(\omega_{1,k} y^*) + b_k \cos(\omega_{1,k} y^*) + \varepsilon, \\ y_1 &= y^* + \sum_{k=1}^{N_s} c_k \sin(\omega_{2,k} y^*) + d_k \cos(\omega_{2,k} y^*) + \varepsilon, \end{aligned} \tag{2.38}$$

with $N_s = 10$ the amount of sines, $\varepsilon \sim \mathcal{N}(0, 10^{-8})$ an additive Gaussian white noise to the measurements, $a_k, b_k, c_k, d_k \sim \mathcal{N}(0, 10^{-4})$ the discrete Fourier coefficients, and $\omega_{i,k} \sim \mathcal{N}(0, 6)$ the frequencies. A sample set of functions from (2.38) can be seen in Figure 2.10.

The position range of interest is $y^* \in [0, 1]$ m. The position measurements are defined in the form, $f_{i^*\to i}(y_i)$, while the calibration functions that are being estimated have the form $\hat{f}_{i\to i^*}(y_i)$. Note that $f_{i\to i^*}(y_i)$ is the function inverse of $f_{i^*\to i}(y_i^*)$, which exists under Assumption 1. The inverse cannot be expressed in a closed-form expression, however, it was ensured that each sampled set of functions is valid.

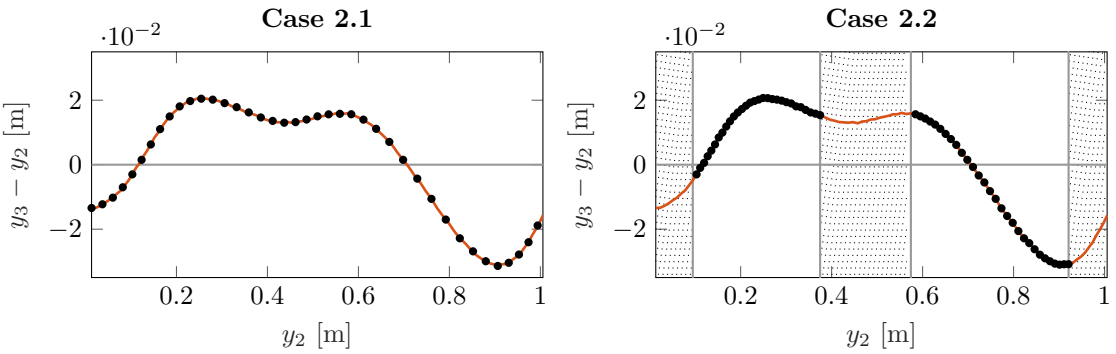


Figure 2.11: For the sample set of functions given in Figure 2.10, here the observations of $f_{2\to3}(y_2)$ (—), given by $\mathcal{D}_2 = \{y_{2,k}, y_{3,k}\}_{i=1}^{N_2}$ (•), have been plotted for both Case 2.1 and 2.2. In Case 2.1 (left plot), the data-set consists of an evenly spaced grid with $N_2 = 40$. In Case 2.2 (right plot), the data-set consists of a more dense grid, with $N_2 = 64$, but the edges and center of the data-set cannot be reached.

In order to prove the efficiency of GPR models in uncertainty-rich regression problems, Monte Carlo simulations will be utilized. At least $M_C = 5000$ different pairs of position measurements, as defined by (2.38), are generated. The two-step calibration procedure consists of a first-stage regression and a following virtual regression. The first-stage regression will be varied for two different cases:

Case 2.1: y_2 is observed for an equally spaced grid of $N_2 = 40$ values of y_3, to obtain data-set $\mathcal{D}_2$

Case 2.2: y_2 is observed for an equally spaced grid of 100 values of y_3, but then 10% of the data on either edge and 20% of the data in the center are removed, leading to $N_2 = 64$. Here, data-set $\mathcal{D}_2$ has limited reach. This represents a scenario where S_3 cannot measure the test bed everywhere, e.g., because it is physically obstructed.

Both cases have varying levels of uncertainty due to limitations in data reach, i.e. **Case 2.2** is considered to be uncertainty rich due to limitations in measurement data. Figure 2.11 shows the difference in data-set reach for both cases. The virtual regression problem that follows the first-stage regression problem does not have any limitations for the reach of its data-set, i.e. data-set $\mathcal{D}_1$ can be infinitely large.

Algorithm 1 will be utilized to obtain a model estimate $\hat{f}_{1\to3}(y_1)$ for all N iterations. The cost, given by (2.27), demonstrates the modeling accuracy of the method. In addition, Algorithm 1 will be compared to two alternative methods, given by:

Alternative 1: Algorithm 1 is followed, except for the virtual regression of $\hat{f}_{1\to3}(y_1)$ the likelihood covariance is assumed to be unknown. Instead, the variance of the model is parameterized by $\Sigma_{\hat{Y}_3} = \sigma_{n,3}^2 I$, where $\sigma_{n,3}^2$ is found by maximization of the marginal likelihood. Where-as for Algorithm 1 the likelihood covariance is known via $\text{cov}(\hat{f}_{2\to3}(y_2))$.

Alternative 2: A lookup table is made of $\hat{f}_{2\to3}$ using $\mathcal{D}_2$. Subsequently, a lookup table $\hat{f}_{1\to3}$ is made using virtual data-set $\mathcal{D}_1'$. In Case 2.1, cubic spline interpolation is used between entries in the lookup table, and in Case 2.2, linear interpolation is used, since cubic splines are not well suited for extrapolation.

The cost (2.27) will be evaluated for these alternative methods to compare to the efficiency of Algorithm 1. **Alternative 1** will be used to validate and prove Conjecture 1. In essence, **Alternative 1** neglects the added advantage of Bayesian regression for virtual regression problems, as described in Conjecture 1.

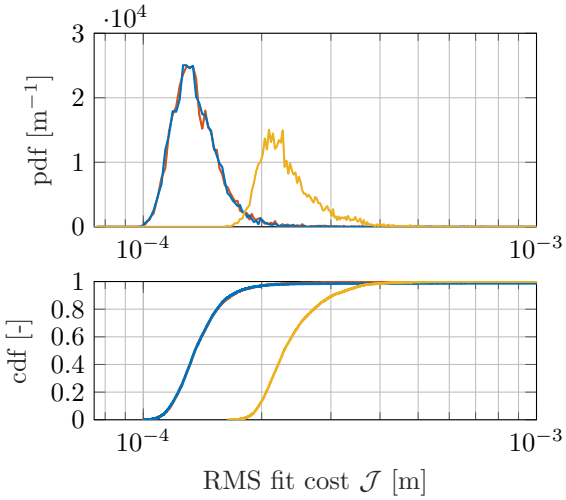


Figure 2.12: Top: Normalized empirical probability density functions; Bottom: Cumulative density functions of $\mathcal{J}$ (M_C=5000) for Case 2.1, with an equidistant grid of data points from S_3. Algorithm 1 (—) achieves an equally accurate fit $\hat{f}_{1\to3}$ as Alternative 1 (—), which disregards the variance of the first model. However, both methods significantly outperform Alternative 2 (—) which uses a lookup table with cubic spline interpolation.

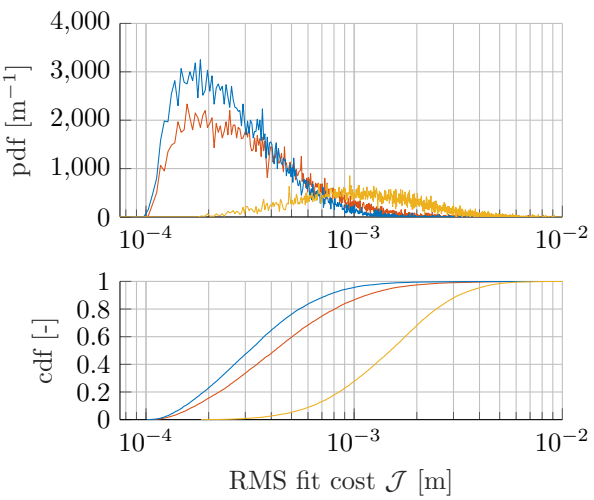


Figure 2.13: Top: Normalized empirical probability density functions; Bottom: Cumulative density functions of $\mathcal{J}$ (M_C=12000) for Case 2.2, where some sensor locations are not measured by S_3. Algorithm 1 (—) results in a noticeably more accurate fit $\hat{f}_{1\to3}$ compared to Alternative 1 (—), which ignores the variance of the first model. Both methods significantly outperform Alternative 2 (—) which employs a lookup table with linear interpolation.

2.5.2 Simulation results

The results of the Monte Carlo simulations for both Case 2.1 and 2.2 can be seen in Figures 2.12 and 2.13 respectively. For both cases, the empirical probability density and the cumulative density functions of the cost $\mathcal{J}$ in (2.27) have been determined. Here the empirical probability density function has been normalized to have a total unit area.

From the results of Case 2.1, no difference can be observed in performance between Algorithm 1 and Alternative 1, while Alternative 2 performs considerably worse. Here Algorithm 1 and Alternative 1 perform equally well, i.e., taking into account the model uncertainty of $\hat{f}_{2\to3}(y_2)$ does not lead to a more accurate model $\hat{f}_{1\to3}(y_1)$. Nevertheless, both Algorithm 1 and Alternative 1, which use Bayesian inference with kernel regularization, lead to a considerably better model estimate compared to lookup tables with cubic interpolation.

For Case 2.2, the reach of sensor S_3, and thus data-set $\mathcal{D}_2$, has been limited considerably. This introduces high uncertainty in the prediction of $\hat{f}_{2\to3}(y_2)$. As Algorithm 1 makes use of the uncertainty estimate from this prediction, supported by Lemma 1, it can lead to considerably better model estimates for $\hat{f}_{1\to3}(y_1)$ compared to Alternative 1. For large gaps in data-set reach, lookup-tables are out of the question, as they perform substantially worse.

2.6 Conclusion

This chapter has presented a Bayesian framework for cascaded sensor calibrations. Specifically, GPR models are utilized to efficiently model a multi-stage regression problem, which leads to more accurate model estimation. The uncertainty in multi-stage regression problems can be utilized in Bayesian frameworks, which shows more accurate modeling compared to traditional methods, e.g. lookup tables. The use of such frameworks is especially recommended given uncertainty-rich data-sets.

A common cascaded calibration example is presented: test-bed calibration of position sensors

in mechatronic systems. A simulation example in combination with Monte Carlo experiments, has provided evidence for more efficient model estimation using the novel Bayesian calibration framework.

Chapter 3

Budgeting of Bayesian Regression Modeling Uncertainties

Sensor calibration is efficiently performed using Bayesian regression methods. In the Bayesian framework, modeling uncertainties are estimated in closed-form and give great insight into the positioning capabilities of control systems. This chapter introduces a framework that aims to include these modeling uncertainties in existing error budgeting frameworks for control systems. The previous chapter introduced cascaded sensor calibrations and how these are efficiently performed using the Gaussian process regression (GPR) method. A GPR model consists of a Gaussian process (GP) prior model, which is a prior distribution over functions. Given a set of observations, the GP prior is updated to produce a predictive distribution (Section 2.3.4). Additionally, the predictive distribution estimates uncertainty at the predictions in closed-form. In this context, it is possible to define and estimate modeling uncertainties through Bayesian regression methods.

This chapter provides methods to obtain the PSD of said modeling uncertainties. It is also shown that the kernel function of a GP prior encodes much of the power spectral characteristics of these modeling uncertainties. This is highly valuable information, as the GP prior, before making observations or predictions, encodes power spectral characteristics of the modeling uncertainties. Using these techniques, the PSD of modeling uncertainties that result from sensor calibration is made available. The objective is to include the PSD of these modeling uncertainties in error budgeting frameworks, e.g. when designing a feedback controller that requires specific spectral characteristics, for instance, bandwidth limitations.

First in Section 3.1 an introduction is provided for error budgeting frameworks, which includes DEB. Additionally, a clear distinction is made between types of error contributions in a control system. Section 3.2 gives the formal definition for modeling uncertainties given predictive distributions. Afterward, 3.3 introduces methods to find the PSD of GPR models. Section 3.4 explains how these modeling uncertainties can be accounted for in error budgeting frameworks. In Section 3.5 a complete overview is provided of the frameworks presented in this thesis for GPR-based sensor calibration and error budgeting. Finally, in section 3.6 a conclusion is provided on the main results.

3.1 Error Budgeting frameworks

Error budgeting is a valuable approach for managing and estimating errors in a system, and is an iterative process that takes place at various stages of a system's development. Without available experimental data, error budgeting can be performed using models and prior knowledge. This preliminary analysis helps set expectations and design guidelines based on a theoretical understanding of the system. Alternatively, error budgeting can be performed after experimental data becomes available, which in most cases provides critical information about the system's capabilities, e.g. levels of sensor inaccuracies. This empirical data can be included in the error

Procedure 2 Dynamic Error Budgeting for linear control systems [34]

1: Design and model a control system conceptually, such that closed-loop transfer functions are defined properly.
2: Identify all significant error sources in the closed loop system and find their respective PSD models.
3: Define and observe the total error of the control system by simulating the system. The contribution of each error source to the total error can be analyzed.
4: Check if the total error budget is acceptable, if not, make changes to the system and go back to step 3.
5: Repeat steps 3-4 until the total error budget is in an acceptable range.

budgeting process, to create a more accurate estimate of the achievable system performance.

In this context, the DEB framework will be regarded as an essential framework, which allows for managing and budgeting errors in high-precision machines. This framework analyzes the propagation of error sources through a high-precision control systems, which is enabled via a PSD modeling of these error sources.

In a larger scope of error budgeting frameworks, the European Space Agency (ESA) Pointing Error Engineering handbooks [32] serve as an important reference. This handbook provides guidelines on how to create and manage well-organized error budgets for laser-pointing systems. Although this handbook serves as a guideline specifically for pointing systems, it can also be applied to other complex systems requiring high precision and careful budgeting. Certain aspects of these guidelines are out of the scope of this thesis, though a few key guidelines are emphasized. That is, a clear distinction is made between knowledge errors and performance errors, each playing a role in the total error budget.

In Section 3.1.1 the DEB framework is explained in more detail. Afterward, in Section 3.1.2 the distinction is made between performance error sources and knowledge error sources. The objective is to categorize modeling uncertainties that arise from sensor calibration. These modeling uncertainties are position-dependent, which can complicate error budgeting for these error sources and highlight the need for careful considerations.

3.1.1 Dynamic Error Budgeting

The DEB high-level design process is presented in the Procedure and describes the concept of systematic budgeting for certain performance specifications. For technical details and examples regarding DEB framework see the research conducted in [33] and [34]. The assumptions for this framework are as follows:

1. Assume the (sub-)systems can be described accurately with linear-time invariant models.
2. Error sources acting on the system must be stationary, their statistical properties are not allowed to change over time.
3. Stochastic error sources are only allowed, as signals with pure harmonic or dc-components give infinite peaks in their PSDs.

There is no underlying assumption on the type of distribution for the error sources. This thesis has continuously used Gaussian distributions and thus is suitable for this DEB framework.

3.1.2 Performance error vs knowledge error

A distinction is made between two main types of error sources in a control system, performance error and knowledge error sources. A control system uses data, obtained via sensors or via models, to position itself accurately for a given reference signal. In Figure 3.1 a schematic is shown that depicts different types of error sources and in which category these land. The performance error

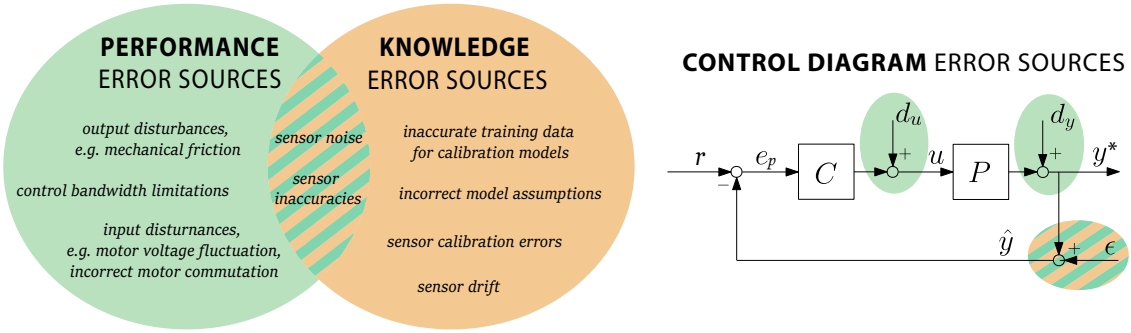


Figure 3.1: For control systems a distinction can be made between two main types of error sources, performance error sources and knowledge error sources. Performance errors and knowledge errors are correlated for some error sources, e.g. sensor noise can fall into both categories.

and knowledge error are denoted by e_p and e_k respectively. The sum of the two types of error constitutes the total error, denoted by e_t.

Performance error sources are sources that directly influence the control performance of the system. In other words, every error source that causes deviations between the reference signal and a measured signal is categorized as a performance error. One such example are external vibrations that act on a control system. In the control diagram, the performance error is given by

$$\begin{aligned} e_p(s) &= r(s) - \hat{y}(s), \\ &= Sr(s) - S\epsilon(s) - Sd_y(s) - PSd_u(s), \end{aligned} \tag{3.1}$$

where s is the Laplace variable, S is the sensitivity given by, $S = 1/(1+PC)$, and ϵ represents any errors in the sensor measurement. A knowledge error disturbs the system's understanding of a certain state but does not necessarily affect the system's output or performance. One example of a knowledge source is an incorrect assumption about the system dynamics. The knowledge error in the control diagram is given by

$$\begin{aligned} e_k(s) &= \hat{y}(s) - y^*(s), \\ &= \epsilon(s). \end{aligned} \tag{3.2}$$

Knowledge error sources like sensor noise can act as performance error sources. An intuitive example is when the control bandwidth of a system is placed at high frequencies. This results in the amplification of high-frequency sensor noise, which can deteriorate the overall tracking performance. Although sensor noise is defined as a knowledge error source, it can act as a performance error source, given poor control strategies. The total error is given by the sum of both errors types, which is given by

$$\begin{aligned} e_t(s) &= r(s) - y^*(s), \\ &= e_p(s) + e_k(s), \\ &= Sr(s) - \underbrace{(1-S)}_{T}\epsilon(s) - Sd_y(s) - PSd_u(s), \end{aligned} \tag{3.3}$$

where $(1-S) := T$ the complementary sensitivity. Modeling uncertainties that arise from sensor calibration are primarily categorized as knowledge errors, and are described in Section 3.2. These error sources are one of the contributors to the total error budget. Section 3.3 provides the necessary tools for estimating the PSD function of these modeling uncertainties, enabling their inclusion in the DEB framework.

3.2 Problem description: modeling uncertainties in Bayesian regression methods

In Bayesian regression methods, the predictive covariance represents the uncertainty in the predictions. To better understand this concept, an intuitive example is provided to illustrate the definition of modeling uncertainty and the possible modeling errors that can arise.

In Section 3.2.1 a definition is provided for modeling uncertainties in GPR methods. Afterward, in Section 3.2.2 an intuitive example is provided, which shows how possible modeling errors emerge from Bayesian regression models. Finally, in Section 3.2.3 a link is made to sensor calibrations and the effect of modeling uncertainties on the feedback control of a dynamical system.

3.2.1 Defining modeling uncertainties in Bayesian regression

The following discussion focuses on the modeling uncertainties in Bayesian regression methods and how to characterize them using the estimated predictive covariance. The predictive distribution resulting from Bayesian regression methods provides a mean and covariance estimate.

Instead of using the notations for sensor calibration as described in Definition 1, here for notational simplicity the process is given by $y = f(x) + \varepsilon_n : \mathbb{R}^D \to \mathbb{R}$. A GP prior model of a process $f(x)$ is given by

$$f(x) \sim \mathcal{GP}\left(m(x), k(x_A, x_B)\right). \tag{3.4}$$

Assume here that hyper-parameter optimization provides a GP model with a squared exponential kernel function $k(x_A, x_B)$ and a zero-mean function $m(x)$ (Section 2.4.5). In the absence of any observations, samples can be taken from the GP prior. The GP prior is evaluated on a query grid of points $\mathbf{x} \in \mathbb{R}^{M \times D}$, which gives a Gaussian distribution

$$p(f(\mathbf{x})) \sim \mathcal{N}\left(\mu_p, \Sigma_p\right), \tag{3.5}$$

where $\mu_p := m(\mathbf{x}) \in \mathbb{R}^M$ the prior mean, and $\Sigma_p := k(\mathbf{x}, \mathbf{x}) \in \mathbb{S}^M_{++}$ the prior covariance matrix.

Given the GP prior and observations, the predictive distribution (2.26) is derived. Consider a data-set given by $\mathcal{D} = \{X, Y\}$, where $X \in \mathbb{R}^{N \times D}$ and $Y \in \mathbb{R}^N$. Take the same set of query grid points $\mathbf{x}$, which results in the predictive distribution given by

$$p(\hat{f}(\mathbf{x})|X, Y, \mathbf{x}) \sim \mathcal{N}(\mu_{\hat{\mathbf{y}}}, \Sigma_{\hat{\mathbf{y}}}), \tag{3.6}$$

where $\mu_{\hat{\mathbf{y}}} := \mathbb{E}[\hat{f}(\mathbf{x})] \in \mathbb{R}^M$ the predictive mean, and $\Sigma_{\hat{\mathbf{y}}} := \mathrm{cov}(\hat{f}(\mathbf{x})) \in \mathbb{S}^M_{++}$ the predictive covariance matrix. The modeling uncertainty, which is encoded by the predictive covariance matrix $\Sigma_{\hat{\mathbf{y}}}$, describes possible deviations from the predictive mean. In other words, the predictive covariance indicates that possible modeling error exists within the predictive distribution. The definition for modeling uncertainty in GPR methods is formally stated in Definition 5.

Definition 5 (Modelling uncertainty in Gaussian process regression) *Gaussian process regression provides a predictive Gaussian probability distribution (2.21), which includes an estimate for the covariance $\Sigma_{\hat{\mathbf{y}}}$ at the predictions. The modeling uncertainty ϵ is described by the zero-mean predictive distribution, given by*

$$p(\epsilon(x)|X, Y, x) \sim \mathcal{N}(0, \Sigma_{\hat{\mathbf{y}}}). \tag{3.7}$$

■

Samples are drawn from this modeling uncertainty, with each sample being a possible deviation from the mean prediction, i.e. a modeling error. The following section describes this phenomenon with an intuitive example.

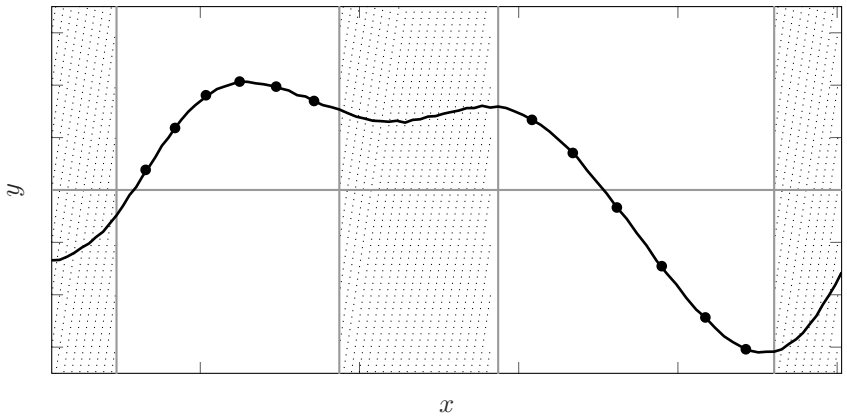


Figure 3.2: A process $y = f(x) + \varepsilon_n$ (—), with a finite amount of observations (•). This example has been taken from Case 2 in Section 2.5, where observations have been severely limited to a certain range.

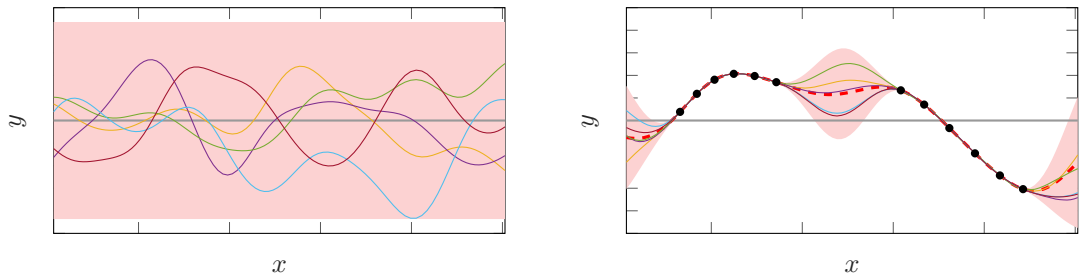
3.2.2 Sampling modelling errors from the predictive distribution

Possible modeling errors resulting from Bayesian regression are sampled by the modeling uncertainty, see Definition 5. To provide an intuitive understanding of how modeling errors can arise in Bayesian regression, a straightforward single-dimensional example is presented. A simple process, given by $y = f(x) + \varepsilon_n : \mathbb{R} \to \mathbb{R}$, has been plotted in Figure 3.2, with a finite amount of observations.

The prior distribution (3.5) and the predictive distribution (3.6) are visualized in Figure 3.3. Samples are drawn from the prior distribution to demonstrate that, in the absence of observations, the function space described by the GP prior encompasses all possible functions. Appendix A.4 demonstrates the procedure for sampling from multivariate Gaussian distributions by utilizing the Cholesky decomposition of the covariance matrix.

It is apparent that adding observations squeezes the uncertainty in the predictive distribution, i.e. the prior knowledge has been updated via Bayesian inference. However, due to the restricted and finite reach of the observations, uncertainty remains. Samples drawn from the predictive distribution deviate from the prior mean at locations with high uncertainty. These deviations are possible modeling errors that arise from Bayesian regression and are all encoded in the predictive covariance matrix, as described in Definition 5.

In Figure 3.4 several samples have been drawn from the modeling uncertainty which presents possible modeling errors that might be present in the predictive distribution. The following sec-



(a) prior distribution $f(x) \sim \mathcal{N}(\mu_p, \Sigma_p)$

(b) posterior distribution $p(\hat{f}(x)|X, Y, x) \sim \mathcal{N}(\mu_{\hat{\mathbf{y}}}, \Sigma_{\hat{\mathbf{y}}})$

Figure 3.3: **(a)** The GP prior distribution given by (3.5) is sampled for the entire input range x for five different occasions, with prior covariance Σ_p ( ) for 99% confidence. With a finite amount of observations (•) the predictive distribution (3.6) is determined. **(b)** The predictive distribution, with mean $\mu_{\hat{\mathbf{y}}}$ (- -) covariance estimate $\Sigma_{\hat{\mathbf{y}}}$ ( ), is sampled five times.

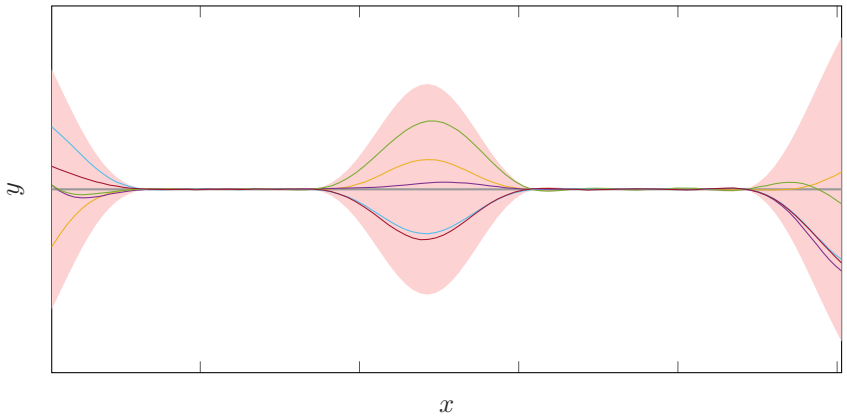


Figure 3.4: The modeling uncertainty $p(\epsilon(x)|X, Y, x) \sim \mathcal{N}(0, \Sigma_{\hat{\mathbf{y}}})$

tion examines the implications of modeling uncertainties in sensor calibration models, and the consequences these have on feedback control of a dynamical system.

3.2.3 Modeling uncertainties in sensor calibrations

The objective is to obtain the power spectrum of modeling uncertainties and to assess how these propagate in a feedback control system. This will allow for more accurate budgeting of modeling uncertainties for control design purposes.

Bayesian regression methods, such as GPR methods, have proven to be useful in the case of sensor calibration. Notably, in a cascaded calibration problem, these methods are believed to be superior to traditional approaches, as described in Conjecture 1.

Consider in Figure 3.5 a feedback control diagram of a dynamical system P_i, that has sensor S_i. The sensor S_i measures the true position y_i^* and returns a position measurement y_i, this process is depicted by $S_i(y_i^*) = f_{i^*\to i}(y_i^*)$. Using Bayesian regression methods, this sensor has been calibrated on a more accurate reference sensor S_j, which results in a model estimate given by $\hat{y}_j = \hat{f}_{i\to j}(y_i)$. A perfect model estimate would indicate that it perfectly describes the inverse of the sensor S_i, i.e. specifically $S_i(\hat{f}_{i\to j}(y_i)) = 1$. In reality, a model is never perfect and results in modeling uncertainties,

$$S_i(\hat{f}_{i\to j}(y_i)) = 1 + \epsilon(y_i) \tag{3.8}$$

where $\epsilon(y_i) := \epsilon_i$ is the modeling uncertainty that encodes imperfect modeling.

Section 3.3 provides the necessary tools to determine the PSD of modeling uncertainties. Following that, Section 3.4 explains how this PSD information can be employed for error budgeting purposes.

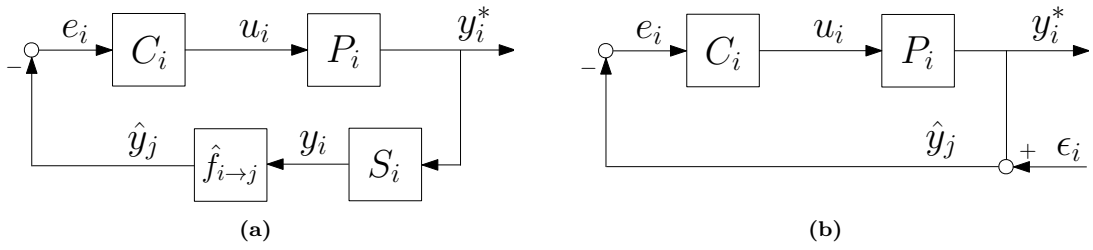


Figure 3.5: Control diagrams including sensor calibration models, where **(a)** shows sensor calibration being included in a control system and **(b)** shows a compact representation of calibration modeling uncertainties in control systems

3.3 Power spectral density for GPR models

A GPR model has a GP prior which is updated using observations, to obtain the predictive distribution. As shown in Definition 5 in the previous section, modeling uncertainties in Bayesian regression can be defined by utilizing the predictive covariance estimate. In this section, available methods are provided to perform a PSD analysis on modeling errors in GPR models.

The proposed framework for PSD analysis and estimation of GPR modeling uncertainties is given by two steps:

(i) The PSD model is obtained of the zero-mean GP prior

(ii) The PSD estimate is obtained of the modeling uncertainty distribution in (3.7).

The first step gives insight into the power spectral characteristics regarding the modeling uncertainties. In this step, the mean function of the GP prior can be set to zero as it does not affect the distribution of the modeling uncertainties. The distribution in (3.7) is fully dictated by the predictive covariance estimation and is thus invariant from the choice of the mean function.

The second step involves the estimation of a conditioned distribution, i.e. the modeling uncertainty which is conditioned on observations. There is no straightforward method to estimate a PSD model of a conditioned Gaussian distribution, thus a practical solution is provided.

In the following sections, the principles behind these methods will be explained in sufficient detail. Afterward, an example is given that showcases the two-step framework. First, in Section 3.3.1, the Wiener-Khinchin theorem is introduced as a means for calculating the PSD of a zero-mean GP prior model. This theorem is explained in detail, and its relevance to Gaussian processes is discussed within the section. Section 3.3.2 shows that the Wiener-Khinchin theorem does not hold for conditioned distributions, which are generally non-stationary. This section provides Monte Carlo methods that are able to estimate the PSD of conditioned distributions. Finally, Section 3.3.3 provides an example of a PSD analysis for a GPR model with a squared exponential kernel.

3.3.1 Power spectral density of Gaussian process priors

Here methods are provided to derive a closed-form expression for the power spectral density function for stationary stochastic processes. These methods can be used to determine the PSD function of a GP prior model. In a later example, it is shown that the GP prior provides valuable information regarding the power spectral characteristics of modeling uncertainties.

Before presenting these methods, it is worth noting Lemma 2, which demonstrates how one can determine whether the Fourier transform of a process exists. The Fourier transform of a stochastic process, such as a GP, is only defined under strict conditions.

Lemma 2 (Existence Fourier transform [35]) *Consider a signal that is defined by a function $f(x) : \mathcal{X} \to \mathbb{R}$. The input space is denoted by $\mathcal{X}$, i.e. $\mathcal{X} = \mathbb{R}^D$ with D the dimension of the input space. The Fourier transform for this multidimensional function is given by,*

$$F(\xi) = \mathcal{F}\{f(x)\}(\xi) \equiv \int_{\mathcal{X}} f(x) e^{-i2\pi\xi x} dx, \tag{3.9}$$

with $\xi \in \mathbb{R}$ the spatial frequency with unit m^{-1}. For this Fourier transform to exist, the function $f(x)$ is required to be Lebesgue integrable, given by

$$\int_{\mathcal{X}} |f(x)| dx < \infty. \tag{3.10}$$

■

The Fourier transform is typically defined for signals that are integrable over a finite interval, but the Fourier transform for infinite-duration stochastic processes, such as GPs, is not well-defined in this sense.

Consider a random process $Y(x)$, with $y(x)$ a realization of this random process. There may be some realizations of $y(x)$ which are not Lebesgue integrable, in which case the Fourier transform does not exist [36, Chapter 7]. Nonetheless, if the process $Y(x)$ is wide-sense stationary (WSS), and if the autocorrelation function is known, the PSD function can be calculated in closed form.

Definition 6 (Wide-sense stationary) *A Wide-sense stationary (WSS) stochastic process requires that its mean and correlation are independent of its argument [37, Chapter 3]. A continuous WSS stochastic process $f(x)$, has a mean μ and correlation function R given by,*

$$\begin{aligned} \mathbb{E}[f(x)] &= \mu, \\ \mathbb{E}[(f(x_A))(f(x_B))] &= R(\tau),\ \forall \tau, x_A, x_B \in \mathbb{R}, \end{aligned} \tag{3.11}$$

where $\tau = x_A - x_B$ is the difference between two position instances. ■

The PSD function for WSS stochastic processes is calculated by taking the Fourier transform of the correlation function. This property is formally defined in Theorem 2, and is known as the Wiener-Khinchin theorem. Note in this thesis, the term stationarity of a signal specifically denotes that the signal is assumed to be WSS.

Theorem 2 (Wiener-Khinchin theorem) *The power spectral density $S(\xi)$ of a WSS stochastic process $f(x)$, is the Fourier transform of the (auto)correlation function $R(\tau)$,*

$$R(\tau) \overset{\mathcal{F}}{\longleftrightarrow} S(\xi). \tag{3.12}$$

The proof is provided in [38]. ■

For a given continuous correlation function $R(\tau)$ the power spectral density function can be obtained given by,

$$S(\xi) := \int_{\infty}^{-\infty} R(\tau) e^{-i2\pi\xi\tau} d\tau. \tag{3.13}$$

For a discrete process $f_k(x_k)$ with discrete correlation sequence R_k the PSD function can be defined by the DTFT of the correlation sequence, given by

$$S(\xi) := \Delta x \sum_{k=-\infty}^{\infty} R_k e^{i2\pi\xi k\Delta x}, \tag{3.14}$$

where $k \in \mathbb{Z}$ are the sample points and $\Delta x = x_{k+1} - x_k$ is the distance between the sample points.

As an intuitive example, using Theorem 2, it can be shown that a white noise signal indeed has a flat power spectrum.

Example 1 (PSD of white noise signal) *Consider a discrete-time white noise temporal process $\varepsilon[k] := \varepsilon_k$ which has zero mean. This signal can be characterized as WSS, with the following mean and correlation,*

$$\begin{aligned} \mu_{\varepsilon,k} &= \mathbb{E}[\varepsilon_k] = 0\ \forall k, \\ R_{\varepsilon,k} &= \begin{cases} \sigma_n^2 \Delta x & \text{if} \quad k = 0 \\ 0 & \text{if} \quad k \neq 0 \end{cases}, \end{aligned} \tag{3.15}$$

where σ_n^2 is the variance. Using equation 3.14 the spectral density function is given by

$$S(\xi) = \sigma_n^2 \Delta x\ \forall \xi. \tag{3.16}$$

■

A GP, given by Definition 4, is an infinite-dimensional generalization of the multivariate normal distribution. Using the Wiener-Khinchin theorem from Theorem 2, the power spectral density function $S(\xi)$ of a stationary GP can be estimated.

A stationary kernel function of a GP, which is also known as the covariance function, can be expressed as $k(x_A, x_B) = k(\tau)$. For a zero-mean GP, the covariance function $k(\tau)$ and correlation function $R(\tau)$ become equal. This property is described in more detail in Appendix B.1.1. The following is then true for a stationary zero-mean process $f(x)$:

$$k(\tau) = R(\tau) = \mathbb{E}[f(x_A)f(x_B)]. \tag{3.17}$$

Under these conditions, the PSD function of a GP can be analytically defined in Corollary 1. Using this Corollary, the PSD function $S(\xi)$ of a GP prior model can be determined in closed-form.

Corollary 1 (PSD of a zero-mean stationary GP) *Consider a zero-mean stationary GP f(x), with stationary kernel function $k(\tau)$, the Wiener-Khinchin theorem in Theorem 2 states that the Fourier transform of the kernel gives the power spectral density $S_0(\xi)$ of the zero-mean process:*

$$S_0(\xi) := \mathcal{F}\{k(\tau)\} \tag{3.18}$$

■

3.3.2 Power spectral density of conditioned distributions

The PSD of conditioned Gaussian distributions, such as the modeling uncertainties (3.7), can be estimated by means of Monte Carlo estimation, which will be introduced here. An analytical method to obtain the PSD function of a stationary zero-mean GP is provided by Corollary 1. Here it is shown why it is not straightforward to obtain the spectrum of a conditioned Gaussian distribution. By conditioning a stationary Gaussian distribution it generally loses its stationary property.

Conditioned distributions are not generally stationary because they are conditioned on data and thus do not hold to the requirements for Corollary 1. To provide evidence for this statement, the property that a stationary covariance matrix is always symmetric Toeplitz is proven. This property is stated in Proposition 1.

Proposition 1 (Symmetric Toeplitz property for covariance matrices) *A covariance matrix is symmetric Toeplitz if the stochastic process holds to the WSS property, as defined in Definition 6.*

Proof: Samples are drawn from a zero-mean Gaussian process $y = f(x)$ with stationary kernel $k(\tau)$ given by $f(x) \sim \mathcal{GP}(0, k(\tau))$, for any input point $x \in \mathbb{R}^D$. Given a array of input points $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_M \end{bmatrix}^T \in \mathbb{R}^{M \times D}$, the covariance matrix can be defined by

$$k(\mathbf{x}, \mathbf{x}) = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ k(x_N, x_1) & \dots & \dots & k(x_N, x_N) \end{bmatrix}. \tag{3.19}$$

The kernel has stationary property $k(\tau) = k(x_A, x_B)$ and symmetric property $k(\tau) = k(-\tau)$. Using these properties the covariance matrix can be written as a function of τ, which results in a

symmetric Toeplitz matrix, given by

$$k(\mathbf{x},\mathbf{x}) = \begin{bmatrix} k(\tau_0) & k(\tau_1) & \dots & k(\tau_{N\text{-}1}) \\ k(\tau_1) & k(\tau_0) & \dots & k(\tau_{N\text{-}2}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\tau_{N\text{-}1}) & k(\tau_{N\text{-}2}) & \dots & k(\tau_0) \end{bmatrix}, \tag{3.20}$$

where $\tau_k = x_{0+k} - x_0$ *with* $k = 0, 1, .., N-1$ ■

Proposition 1 shows that the conditioned distributions, such as the predictive distribution, has a covariance matrix that is non-Toeplitz, which indicates a loss in stationary.

A number of methods exist in literature that attempts to determine a closed-form expression of a PSD estimate for non-stationary stochastic processes [35]. As these approaches tend to severely increase complexity, a practical approach is adopted instead. By means of Monte Carlo methods, the PSD of a non-stationary distributions can be estimated. Suppose there exists a process $y = f(x)$, with non-stationary distribution $f(x) \sim \mathcal{N}(\mu, \Sigma)$, where Σ is non-Toeplitz according to Proposition 1. From this non-stationary distribution, a total of M_C samples are drawn at query input points $\mathbf{x} \in \mathbb{R}^{M\times D}$. The query output vectors $\mathbf{y}_i \in \mathbb{R}^M$ are drawn independently and identically from the same Gaussian distribution with mean μ and covariance matrix Σ, i.e.

$$\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_{M_C} \sim \text{iid } \mathcal{N}(\mu, \Sigma), \tag{3.21}$$

where iid (independent and identically distributed) means that each vector is independent of the others and that they all have the same distribution. The Monte Carlo approach in Algorithm 2 computes the PSD estimates $\hat{S}_1, \hat{S}_2, .., \hat{S}_{M_C}$ for each output vector $\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_{M_C}$ via the discrete Fourier transform (DFT) and periodogram equations. The PSD estimate of the non-stationary Gaussian distribution is obtained by averaging over each Monte Carlo sample.

Algorithm 2 Monte Carlo PSD estimation of non-stationary Gaussian distributed signals

Require: non-stationary probability distribution of process $f(x) \sim \mathcal{N}(\mu, \Sigma)$, input grid points $\mathbf{x} \in \mathbb{R}^{M\times D}$, distance between grid points Δx, number of Monte Carlo iterations M_C.

1: Draw M samples at input points $\mathbf{x}$, to obtain $\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_{M_C} \sim \text{iid } \mathcal{N}(\mu, \Sigma)$.
2: For non-periodic processes $f(x)$, segment output vector $\mathbf{y}_i$ into K overlapping windows and apply windowing function at each window[1].
3: Obtain the DFT at each sample, using $F_{i,l} = \sum_{k=0}^{M-1} y_{i,k} e^{-i\frac{2\pi l}{M}k}$, where $i = 1, 2, .., M_C$ and l the frequency bin number, i.e. the DTFT is sampled at frequencies $\xi_l = \frac{l}{M\Delta x}$
4: Compute the periodogram, using $S_{i,l} = \frac{\Delta x}{M}|F_{i,l}|^2$, with $\frac{\Delta x}{M}$ the normalizing prefactor[2].
5: Compute the average over M_C samples, using $\hat{S}_l = \frac{1}{M_C}\sum_{i=1}^{M_C} S_{i,l}$.

More background information on the DTFT and DFT equations are shown in Appendix B.3. The DFT calculation in step 3 of Algorithm 2, samples at the frequencies ξ, which is the spatial frequency in m^{-1}. When dealing with rotational measurements the spatial frequency can also be expressed as ν, which has unit rad^{-1}.

Using the results presented in Corollary 1 and Algorithm 2 a complete PSD analysis for GPR modeling uncertainties is possible. An example is presented for a GPR model with a squared exponential kernel in the following section.

[1] Windowing functions are applied to the signal to prevent spectral leakage which is caused by a mismatch between the frequency components of the signal and the frequency resolution of the DFT. A common window function is the Hann window, which tapers both ends of each window, reducing the effects of spectral leakage.

[2] The normalizing prefactor $\frac{\Delta x}{N}$ normalizes the power spectral density $S_{i,k}$, which after scaling has unit $(\cdot)^2/m$

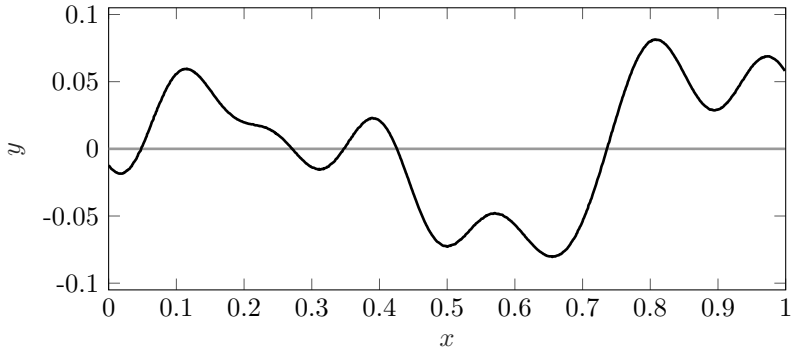


Figure 3.6: Non-linear single-dimensional process $y = f(x) + \varepsilon_n$ (—)

3.3.3 Example: squared exponential kernel

Here an example will be provided of a PSD analysis for GPR modeling uncertainties. Provided a non-linear single-dimensional process $y = f(x) + \varepsilon_n$ given in Figure 3.6, where $f(x) : [0, 1] \to \mathbb{R}$ and $\varepsilon_n \sim \mathcal{N}(0, \sigma_n^2 I)$. The variance of the additive noise is given by $\sigma_n^2 = 10^{-8}$. A GPR model is chosen with GP prior $f(x) \sim \mathcal{GP}(m(x), k(\tau))$ that has zero-mean $m(x)$ and has stationary squared exponential kernel $k(\tau)$ given by,

$$\begin{aligned} m(x) &= 0 \\ k(\tau) &= \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}(\tau)^2\right) \end{aligned} \tag{3.22}$$

where $\tau = x_A - x_B$, and kernel hyper-parameters is given by $\Theta = \{l, \sigma_f\} = \{5 \cdot 10^{-2}, 10^{-2}\}$. The GP prior is conditioned on observations to obtain a predictive distribution, consisting of a predictive mean and covariance. The estimate of the predictive covariance fully describes the modeling uncertainty distribution, as given by Definition 5. The observations are stored in dataset $\mathcal{D} = \{X, Y\}$, where $X \in [0, 1]^N$ and $Y \in \mathbb{R}^N$. The predictive distribution that is conditioned on N observations, can be evaluated on a grid of points $\mathbf{x} \in [0, 1]^M$ given by

$$p(\hat{f}_N(\mathbf{x})|X, Y, \mathbf{x}) \sim \mathcal{N}(\mu_{\hat{\mathbf{y}}}, \Sigma_{\hat{\mathbf{y}}}). \tag{3.23}$$

The modeling uncertainty at the predictions is described by the zero-mean predictive distribu-

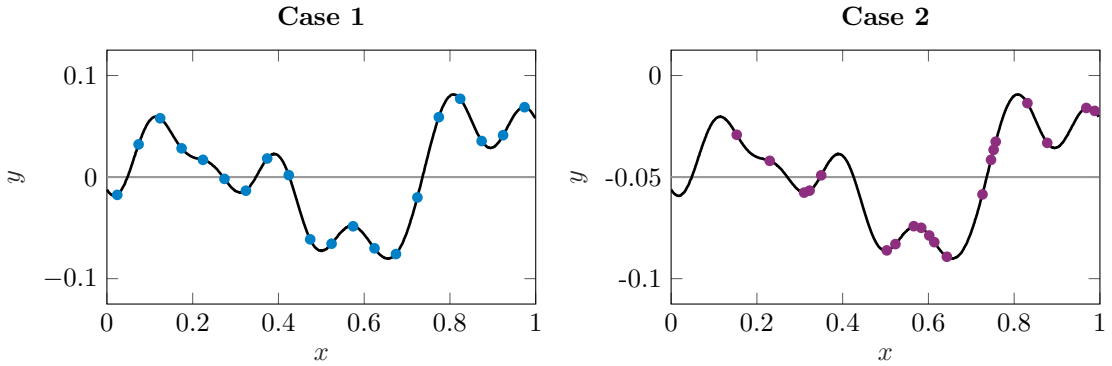
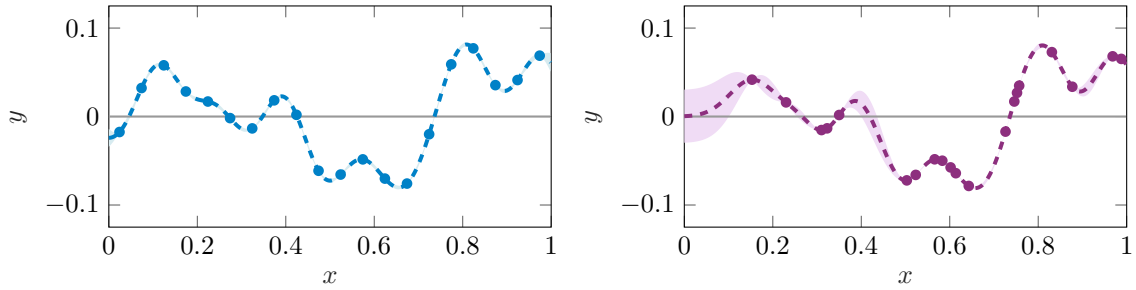
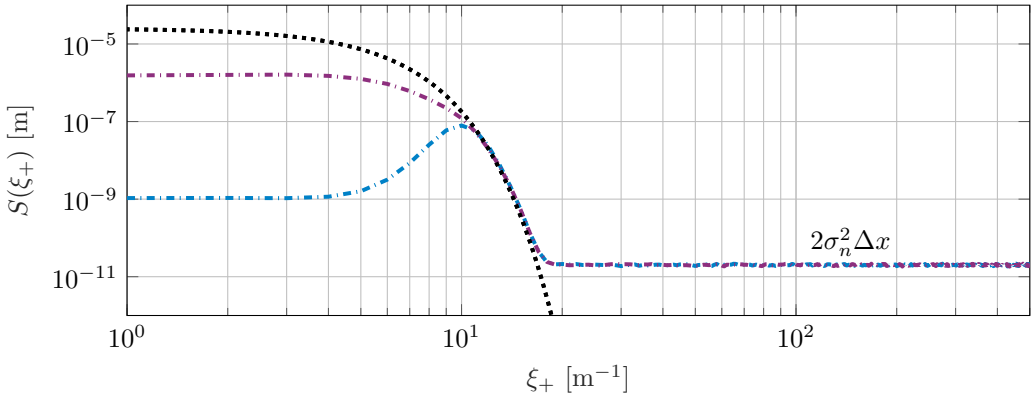


Figure 3.7: Given a total of 10 observations, the dataset $\mathcal{D}_1$ (•) and $\mathcal{D}_2$ (•) has been plotted for Case 3.1 and 3.2 respectively. While the observations in $\mathcal{D}_1$ are at equidistant input locations, the observations in $\mathcal{D}_2$ are drawn from a uniform distribution over the input space.



(a) predictive distribution (3.23) given observations $\mathcal{D}_1$ (●), with mean $\mu_{\hat{\mathbf{y}}}$ (- -) and covariance $\Sigma_{\hat{\mathbf{y}}}$ ( ).

(b) predictive distribution (3.23) given observations $\mathcal{D}_2$ (●), with mean $\mu_{\hat{\mathbf{y}}}$ (- -) and covariance $\Sigma_{\hat{\mathbf{y}}}$ ( ).



(c) the PSD estimates for Case 3.1, $\hat{S}_l(\xi_{l+})$ (-·-), and for Case 3.2, $\hat{S}_l(\xi_{l+})$ (-··-). Additionally, the PSD function of the zero-mean GP prior is plotted, denoted by $S_0(\xi_+)$ (·····)

Figure 3.8: Simulation results for N=20 observations: **(a)** predictive distribution is plotted for Case 3.1, **(b)** predictive distribution is plotted for Case 3.2, **(c)** the PSD estimates obtained from Monte Carlo simulations in Algorithm 2. This simulation example shows the importance of the placement of observations, which affects the level of uncertainty, and thus the PSD estimate.

tion, given by

$$p(\epsilon_N(\mathbf{x})|X, Y, \mathbf{x}) \sim \mathcal{N}(0, \Sigma_{\hat{\mathbf{y}}}). \tag{3.24}$$

The number of observations, and the placement of observations affect the predictive covariance $\Sigma_{\hat{\mathbf{y}}}$, and thus affect the modeling uncertainty distribution. To observe this relation, two cases are considered:

Case 3.1: Observation of process $y = f(x)$ are at equidistant input locations $X_1 \in [0, 1]^N$, resulting in dataset $\mathcal{D}_1 = \{X_1, Y_1\}$.

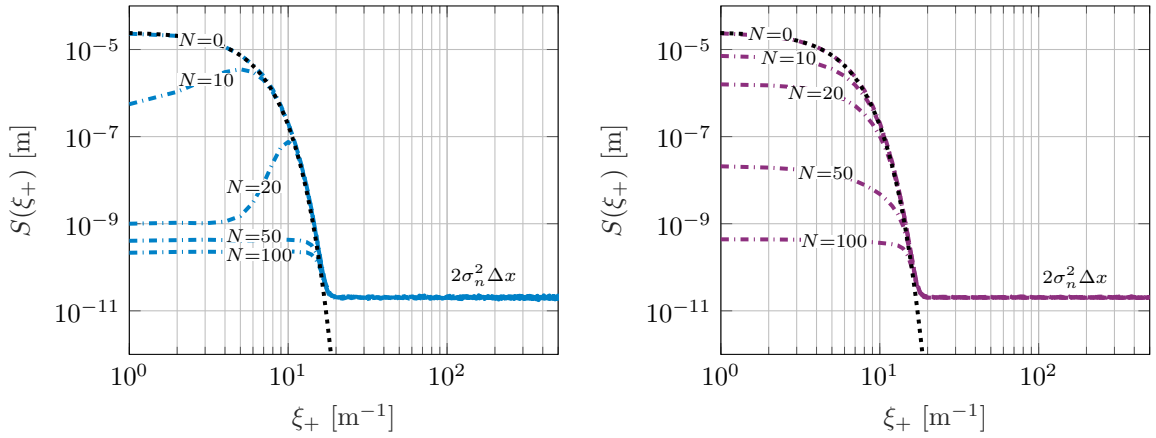
Case 3.2: Observations of process $y = f(x)$ are at random input locations $X_2 \in [0, 1]^N$, resulting in dataset $\mathcal{D}_2 = \{X_2, Y_2\}$. The input locations are sampled form a uniform distribution over the input space, i.e. the input locations $X_2 = [x_{2,1}\ x_{2,2}\ \dots\ x_{2,N}]^T$ are sampled from $x_{2,1}, x_{2,2}, .., x_{2,N} \sim$ idd $\mathcal{U}(0, 1)$, without replacement[3].

A PSD analysis is conducted for both cases, and the results are compared to evaluate whether the placement of observations has any significant impact.

Comparison Case 3.1 and Case 3.2 for N=20

Given N=20 observations the datasets for both cases are shown in Figure 3.7. A PSD analysis to measure the power spectrum of modeling uncertainties is performed in two steps.

[3]The term without replacement indicates that each draw from the uniform distribution can only be assigned once. This ensures that every input location $x_{2,1}, x_{2,2}, .., x_{2,N}$ is unique



(a) the PSD estimate $\hat{S}_l(\xi_{l+})$ (-·-) for varying amounts of observations, with equidistant input locations (Case 3.1).

(b) the PSD estimate $\hat{S}_l(\xi_{l+})$ (-·-) for varying amounts of observations, with random input locations (Case 3.2).

Figure 3.9: Simulation results for a varying number of observations: **(a)** the PSD estimate of the modeling uncertainty distribution is plotted for $N = 0, 10, 20, 50, 100$, **(b)** the PSD estimate of the modeling uncertainty is plotted for $N = 0, 10, 20, 50, 100$.

The first step is to obtain a closed-form expression of the PSD function zero-mean GP prior. The results obtained in Section 3.3.1, showed that using Corollary 1 the PSD function can be obtained by Fourier transforming the kernel function of the GP prior. The second step is to obtain a PSD estimate of the modeling uncertainty distribution. The results obtained in Section 3.3.2, showed that using Algorithm 2 a PSD estimate can be obtained of the modeling uncertainty given by (3.24).

First, the PSD model of the zero-mean GP prior is computed. The single-sided[4] PSD function of this prior is given by

$$S_0(\xi_+) = 2\mathcal{F}\{k(\tau)\} = 2\left(\sigma_f^2\sqrt{\pi/n}\; e^{-\pi^2\xi_+^2/n}\right), \tag{3.25}$$

with $n = (2l^2)^{-1}$. The derivation of the Fourier transform is provided in Appendix B.6.

Second, the PSD estimate of the modeling uncertainty is computed using Algorithm 2. For Algorithm 2 the input grid points are chosen to be $\mathbf{x} \in [0,1]^M$, with M=1000 and with Δx=1e-3. The average of M_C=1000 iterations results in the PSD estimate $\hat{S}_l(\xi_l)$ with frequency resolution $(M\Delta x)^{-1} = 1\text{m}^{-1}$. The single-sided PSD estimate is given by $\hat{S}_l(\xi_{l+}) = 2\hat{S}_l(\xi_l)$.

For the datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, given N=20 observations, a PSD analysis of the modeling uncertainty is performed and presented in Figure 3.8. From these results the following conclusions can be drawn:

- The PSD function $S_0(\xi_+)$ shows an identical roll-off frequency compared to the PSD estimate of the modeling uncertainty distribution $\hat{S}_l(\xi_{l+})$. This roll-off frequency is fully governed by the smoothness hyperparameter l of the kernel function.
- Case 3.1, which distributed the input locations at an equidistant grid, has substantially lower PSD values at low frequencies than Case 3.2, which randomly sampled the input locations.
- The noise floor for both Case 3.1 and Case 3.2 is due to the additive Gaussian white noise. The derivation of the PSD for additive Gaussian white noise is presented in Example 1. In Figure 3.8c one can verify that the noise floor is equal to $2\sigma_n^2\Delta x$, that is for a single-sided PSD.

[4] A single-sided PSD function represents the power spectral density of a signal only in the positive frequency range $\xi_+ \in [0, \infty)$. The total power is divided equally between the positive and negative frequency components, thus the single-sided PSD $S(\xi_+)$ is twice the two-sided PSD function $S(\xi)$, i.e. $S(\xi_+) = 2S(\xi)$

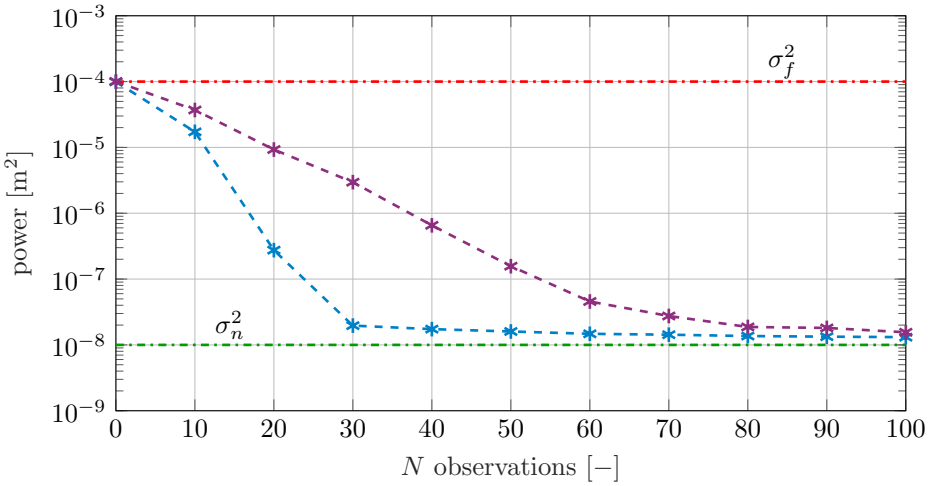


Figure 3.10: The total power P_t of the modeling uncertainty as a function of the number of observations, for both Case 3.1 (-∗-), and Case 3.2 (-∗-). The total power is always lower than the signal variance σ_f^2 of the GP prior (-·-), and always larger than the additive Gaussian noise variance σ_n^2 (-·-).

Comparison Case 3.1 and Case 3.2 for varying N

Here, Case 3.1 and Case 3.2 are compared for a varying number of observations N, that are available to condition the GP prior. The PSD function is estimated in an equivalent manner as done above, for the case where N=20.

The results are presented in Figure 3.9. Figures 3.9a and 3.9b show the PSD estimates for a varying number of observations. In Figure 3.10 the total power of the modeling uncertainty is compared as a function of the number of observations. From these results the following conclusions can be drawn:

- When no observations are available (i.e. $N = 0$) the PSD estimate $\hat{S}_l(\xi_{l+})$ obtained through Algorithm 2 equals the PSD function of the zero mean GP prior $S_0(\xi_+)$.
- Placing observations at equidistant input locations results in lower power P_t while requiring a smaller number of observations N.
- The total power of the modeling uncertainty at zero observations equals the signal variance σ_f^2. The total power converges towards the Gaussian white noise variance σ_n^2 as more observations are added.

The hyperparameters of the kernel are optimized using empirical Bayes, briefly introduced in Section 2.4.5. The results obtained for this simple example demonstrates that the hyperparameters for a squared exponential kernel influence the spectral characteristics. For instance, the smoothness parameter l affects the roll-off frequency, the signal variance σ_f^2 determines the maximum power, and the sensor noise variance σ_n^2 establishes the minimum achievable power.

The subsequent section will take a step back to examine how a PSD estimation of modeling uncertainties in sensor calibration can be utilized for error budgeting. Afterward, in Section 3.5, both frameworks presented in this thesis are revisited to provide visual aid that demonstrates the entire process of using GPR models for sensor calibration and error budgeting.

3.4 Error budgeting for modeling uncertainties in sensor calibration

Modeling uncertainties in sensor calibration are primarily categorized as knowledge errors. As seen in previous sections, Bayesian regression modeling uncertainties are depicted by a non-stationary

Procedure 3 Temporal PSD estimation for arbitrary motions $x(t)$

1: Obtain an arbitrary motion x(t) as a function of time.
2: Use this motion to evaluate modeling uncertainties $\epsilon(x)$ as a function of time, given by $\epsilon(x(t)) \rightarrow \epsilon(t)$
3: Use Algorithm 2 to estimate a temporal PSD $S(f)$, where input location grid $\mathbf{x} \in \mathbb{R}^{M \times D}$ is replaced by input time grid $\mathbf{t} \in \mathbb{R}^{M}$.

predictive distribution, i.e. the modeling uncertainty varies over the position. The PSD models $S(\xi)$ obtained in previous sections have spatial frequency ξ (or ν). This property complicates the inclusion of modeling uncertainties in the error budgeting frameworks like DEB, which require temporal PSD models $S(f)$ where the frequency component f has unit Hz.

Considering a non-stationary calibration error $\epsilon(x)$, one can define $x(t)$ and compute a PSD estimate. This can be done for two situations, when *(i)* $x(t)$ is linear w.r.t time and *(ii)* for any arbitrary motion $x(t)$ w.r.t time. For both situations the conversion from spatial to temporal PSD estimation is considered:

(i) For linear motions $x(t)$ w.r.t time, a spatial PSD can also be scaled linearly. Define a scaling parameter γ, which defines the linear relationship between the position x and time t, given by $x = \gamma t$. The spatial frequency can be transformed by the same scaling parameter to obtain the temporal frequency:

$$f = \gamma \xi. \tag{3.26}$$

Given this relation, a change of variables is used to transform a spatial PSD $S(\xi)$ into a temporal PSD $S(f)$, given by

$$S(\xi) \rightarrow S((1/\gamma)f). \tag{3.27}$$

(ii) For any arbitrary motion $x(t)$ w.r.t. time, the PSD cannot be scaled in a straightforward manner. A quick and practical solution is the transformation of a spatial PSD $S(\xi)$, to a temporal PSD using Procedure 3. Algorithm 2 will be utilized, but the spatial grid is now replaced with a temporal or time grid $\mathbf{t} \in \mathbb{R}^{M}$ with sampling time Δt.

In the context of sensor calibration, the input to my calibration model is an uncalibrated sensor measurement $y_i(t)$, so $x(t) \longrightarrow y_i(t)$. The position measurement of a dynamical system is often not perfectly linear, even in closed-loop situations. This implies that, for sensor calibration, the second situation is more common, where it is assumed that $y_i(t)$ is arbitrary with respect to time.

The temporal PSD estimate $\hat{S}_l(f_l)$, obtained for modeling uncertainties in sensor calibration, can now be utilized in DEB.

3.5 A complete overview of sensor calibration through Gaussian process regression

In the previous chapter and this chapter, two distinct frameworks involving GPR models are introduced. This section aims to present a comprehensive overview of the relationships between these frameworks and their respective connections. The present chapter emphasizes estimating the PSD function, given specific Bayesian modeling uncertainties. The focus is on a calibration procedure and how the strength of GPR models extends beyond cascaded calibration to also encompass error budgeting of modeling uncertainties. A complete overview of sensor calibration through GPR is provided in Figure 3.11.

The general overview provided here uses for simplicity general notations for a process that is being estimated, i.e. the unknown non-linear (noisy) process $y = f(x) + \varepsilon_n$. When discussing sensor calibration specifically, refer to Definition 1, which states that the sensor measurement y_i of a system i can be calibrated to a more accurate sensor measurement y_j. This process can be

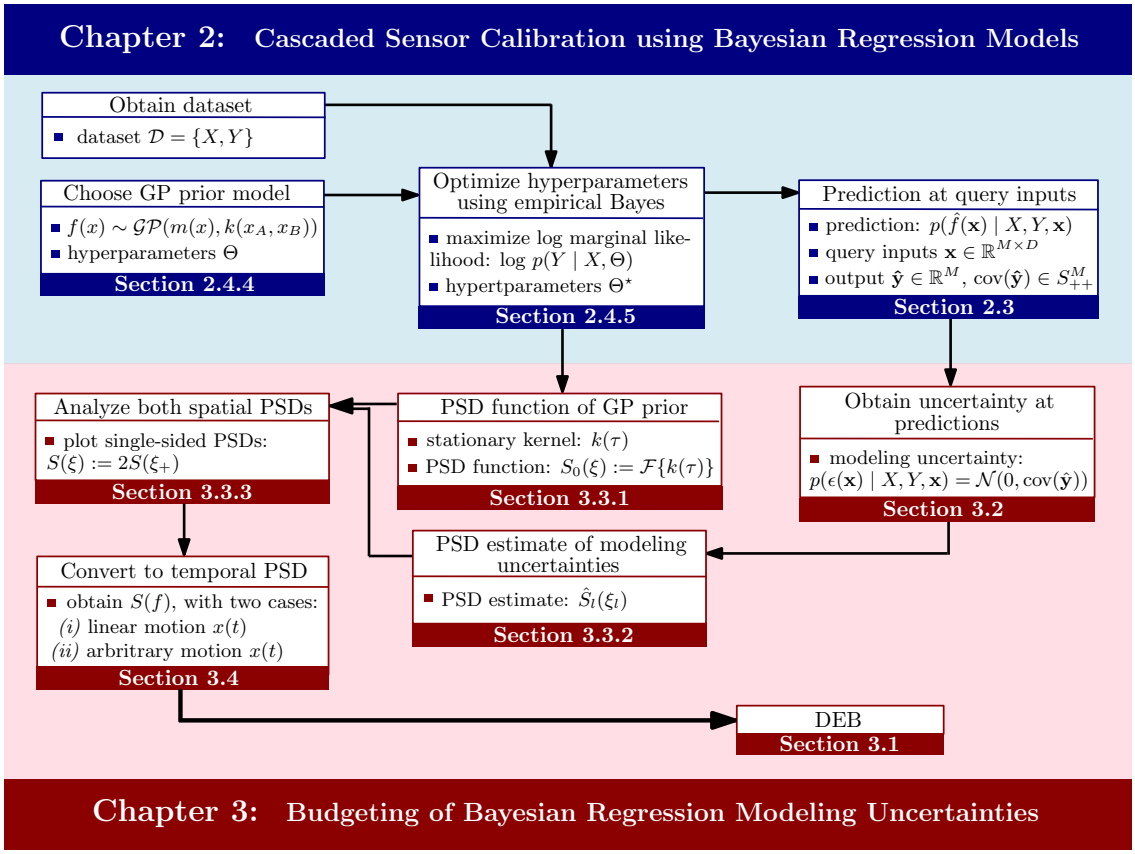


Figure 3.11: A complete overview of sensor calibration through GPR.

represented by a non-linear function defined as $y_j = f_{i\to j}(y_i) + \varepsilon_n$. So to apply this complete framework to sensor calibration, the following changes in notation are performed:

$$\begin{aligned}&\text{the function becomes } f(x) \longrightarrow f_{i\to j}(y_i),\\ &\text{the input becomes } x \longrightarrow y_i\\ &\text{and the output becomes } y \longrightarrow y_j.\end{aligned}$$

3.6 Conclusion

This chapter has demonstrated that modeling uncertainties in Bayesian regression can be represented by their PSD function. This framework can be applied to sensor calibration models and subsequently play a pivotal role in error budgeting frameworks.

A comprehensive framework has been introduced, which shows accurate analysis and estimation of the PSD function. This framework has been employed in a simple example, where PSD estimation and analysis are illustrated for a Gaussian Process Regression model with a squared exponential kernel.

This approach naturally extends to a dynamic error budgeting framework, capable of incorporating the PSD function into the total budget by examining how each PSD source propagates through a control system. While an in-depth analysis of this process has not been presented, it is evident that the framework outlined in this chapter yields accurate models that can be utilized in such error budgeting frameworks.

Chapter 4

Case Study: Cascaded Calibration of Coarse Pointing Assemblies

This chapter provides a cascaded calibration procedure for CPAs designed and manufactured by TNO. The cascaded calibration consists of a two-step calibration procedure: first a static calibration of the PTB using a theodolite which has highly accurate measurement, and second a dynamic calibration of CPAs using the PTB. The calibration procedure will make use of the cascaded calibration algorithm presented in Chapter 2. Given the structure of the expected calibration functions, appropriate kernels will be chosen that make up the Gaussian process prediction models.

Before the calibration of CPAs can take place, the alignment procedure of the CPA and the PTB is critical. Calibration by means of optical alignment between a test bench and a mechatronic system requires an optical sensor that manages the alignment. This requires additional steps in the calibration procedure.

In Section 4.1 the CPA and PTB both developed by TNO will be introduced. In Section 4.2 the calibration procedure will be discussed, where the optical alignment plays an important role. The optical alignment between the CPA and the PTB is discussed in Section 4.3. Afterward, in Section 4.4 a data-driven approach is provided to estimate the inverse optical kinematic behavior of the aligned optical systems. In Section 4.5, the detailed cascaded calibration procedure (Chapter 2) for CPA position sensors is presented, featuring a 2DOF approach. Finally, in Section 4.6 a conclusion is given on the results of this case study.

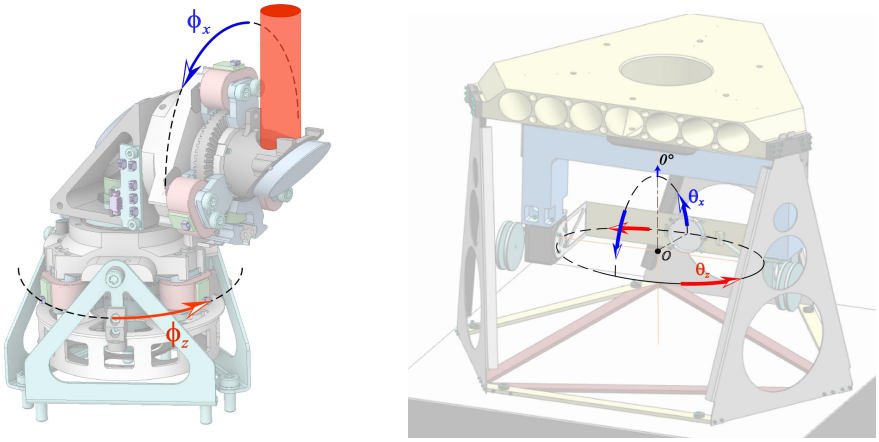
4.1 System characteristics

The calibration procedure of a CPA consists of three systems; a CPA, a PTB and a theodolite. This section provides key characteristics of these systems, that enable choosing suitable kernel functions. In this example, the assumption is made that the theodolite measurements are of absolute accuracy.

Coarse pointing assembly

First observe the CPA in Figure 4.9a. This 2 degrees of freedom CPA has a periscope type design, which enables a large field of regard. The rotation measurement around the azimuth axis is given by ϕ_z, and the rotation measurement around the elevation axis is given by ϕ_x. The CPA is able to reach a full hemisphere.

The rotation of these axes is performed by means of a switch reluctance motor, more detail can be found in Appendix C.1. The rotation axis contains a high-permeable magnetic rotor with an n amount of teeth. The magnetic flux variation over this rotor can be measured via Hall sensors and is used to estimate the angle of the rotor. More information on how the angle of the rotor is measured using a Hall encoder is explained in Appendix C.2. The measurement accuracy



(a) coarse pointing assembly (CPA) **(b)** pointing test bench (PTB)

Figure 4.1: Schematic drawings of the degrees of freedom for the optical systems. The azimuth rotation measurement for the CPA and PTB is given by ϕ_z and θ_z respectively. The elevation rotation measurement for the CPA and PTB is given by ϕ_x and θ_x respectively.

using Hall encoders is often relatively poor. This can be attributed to several factors, including incorrect tooth geometry and tooth pitch, improper lateral or vertical placement of Hall sensors, and magnetic hysteresis in rotor teeth.

The position dependent inaccuracies of the Hall encoder denoted by $\eta(\phi_i)$ with $i \in \{x, z\}$, can be distinguished in two categories:

- *low repeating position-dependent sensor inaccuracies* $\eta_L(\phi_i)$, which have periodicity given by, $\eta_L(\phi_i + 2\pi_i) = \eta(\phi_i)$
- *high repeating position-dependent sensor inaccuracies* $\eta_H(\phi_i)$, which are repeated over the teeth of the rotor. Given the rotor has n teeth, the periodicity can be defined by, $\eta_H(\phi_i + \frac{2\pi}{n}) = \eta_H(\phi_i)$. This is assuming the rotor has identical teeth. Appendix C.3 shows measurement data for measurement on an experimental single-dimensional CPA, where high repeating position-dependent sensor inaccuracies are observed.

The total sensor inaccuracy η_{ϕ_i} for a Hall encoder measurement ϕ_i, is described by,

$$\eta_{\phi_i} = \eta_H(\phi_i) + \eta_L(\phi_i, \phi_j) \quad \forall\, i, j \in \{x, z\} |\ i \neq j. \tag{4.1}$$

The significance of the expected structure for position-dependent inaccuracies is important when selecting an appropriate kernel function for GPR models. Additionally, it is worth noting that the periodicities in $\eta_i(\phi)$ can be seamlessly integrated into the kernel functions.

Pointing test bench

The schematic drawing of the PTB is given in figure 4.9b. It consists of two rotation stages, a large azimuth stage that measures a rotation θ_z and a smaller elevation stage that measures a rotation θ_x. In [6] it is shown that the calibration onto the theodolite was performed using a combination of a lookup table and a parametric model containing a sum of sines. There exists only low-rate position-dependent inaccuracies denoted with $\eta_L(\theta_i)$ with $i \in \{x, z\}$.

The total sensor inaccuracy η_{θ_i} for sensors measuring θ_i can be described by,

$$\eta_{\theta_i} = \eta_L(\theta_i, \theta_j) \quad \forall\, i, j \in \{x, z\} |\ i \neq j. \tag{4.2}$$

The design requirements and main specification of the PTB can be found in Appendix [6].

4.2 Calibration procedure

Recall the motivating example provided in the introduction, where Figure 1.3 showed an overview of the cascaded calibration procedure. Section 2.4 showed that a cascaded calibration via test bed can be performed via a two-stage regression problem, described in Procedure 1. As an extension to this procedure and the results obtained in previous chapters, this chapter introduces a 2DOF cascaded calibration procedure.

The three position sensors in this cascaded calibration are numbered, where the CPA Hall encoders are S_1, the PTB encoder are S_2 and the theodolite is S_3. The sensors measure the angular position in radians. The cascaded calibration of the CPA Hall encoders requires two calibration procedures and a single alignment procedure:

1. Static calibration of the pointing test bench onto a highly accurate theodolite. This calibration is expensive and is limited in the number of datapoints that can be measured. For a 2DOF calibration procedure, two separate processes are defined, each for one axis given by

$$\begin{aligned} \theta_{t,z} &= f_{2\to 3,z}(\theta), \\ \theta_{t,x} &= f_{2\to 3,x}(\theta), \end{aligned} \tag{4.3}$$

   where $\theta_{t,z} \in [0, 2\pi)$ and $\theta_{t,x} \in [0, 2\pi)$ are the theodolite measurements in azimuth and elevation respectively, $\theta = [\theta_z, \theta_x]^T \in \mathbb{R}^2$ is the PTB measurement in both dimensions. The mapping functions can be expressed as $f_{2\to 3,i}(\theta) = \theta_i + \eta_{\theta_i}$, $i \in \{z, x\}$, which can be recalled from (2.4). The position-dependent inaccuracies η_{θ_i} in the PTB z-stage and x-stage measurement is defined by (4.2).

   Obtain data-set $\mathcal{D}_{2,i} = \{\bar{\Theta},\ \bar{\Theta}_{t,i}\}$, where $\bar{\Theta} = [\bar{\theta}_1\ \bar{\theta}_2\ \dots\ \bar{\theta}_{N_2}]^T \in \mathbb{R}^{N_2\times 2}$ and $\bar{\Theta}_{t,i} \in \mathbb{R}^{N_2}$ are discrete measurements on the PTB and the theodolite respectively, used for training the GPR model. Finally, estimate both processes given by

$$\hat{\theta}_{t,i} = \hat{f}_{2\to 3,i}(\theta), \quad \text{for } i \in \{z, x\}. \tag{4.4}$$

2. Optical alignment of the CPA with the PTB, is discussed in Section 4.3. Additionally, data-driven estimation of the optical kinematics is provided in Section 4.4, which enables active optical alignment.

3. Dynamic calibration of the CPA onto an already calibrated PTB. Obtaining datapoints is inexpensive and can be done dynamically. The processes being estimated are given by

$$\begin{aligned} \theta_{t,z} &= f_{1\to 3,z}(\phi), \\ \theta_{t,x} &= f_{1\to 3,x}(\phi), \end{aligned} \tag{4.5}$$

   where $\phi = [\phi_z, \phi_x]^T \in \mathbb{R}^2$ is the Hall encoder measurement of the CPA orientation. The mapping functions can be expressed as $f_{1\to 3,i}(\phi) = \phi_i + \eta_{\phi_i}$, $i \in \{z, x\}$. The position-dependent inaccuracies η_{ϕ_i} of the Hall encoder in z-axis and x-axis is defined by (4.1). The CPA Hall encoders cannot be calibrated directly onto the theodolite measurements, which has been the main reason for developing a cascaded calibration procedure that can do this indirectly efficiently.

   Obtain data-set $\mathcal{D}_1 = \{\bar{\Phi},\ \underline{\bar{\Theta}}\}$, where $\bar{\Phi} = [\bar{\phi}_1\ \bar{\phi}_2\ \dots\ \bar{\phi}_{N_1}]^T \in \mathbb{R}^{N_1\times 2}$ are discrete measurements on the CPA and $\underline{\bar{\Theta}} \in \mathbb{R}^{N_1\times 2}$ are discrete measurements on the PTB. The estimated process given by (4.4) is used to transform data-set D_1 into two virtual data-sets given by $D'_{1,i} = \{\bar{\Phi},\ \hat{f}_{2\to 3,i}(\underline{\bar{\Theta}})\}$. Finally, using the virtual data-sets, estimate both processes given by

$$\hat{\theta}_{t,i} = \hat{f}_{1\to 3,i}(\phi), \quad \text{for } i \in \{z, x\}. \tag{4.6}$$

The cascaded calibration procedure, which consists of the first and third step of the entire calibration procedure, is given in Section 4.5. Before this, attention is put on the optical alignment between the CPA and PTB optical systems. This is necessary to obtain data-set $\mathcal{D}_1$, mentioned in third step of the calibration procedure.

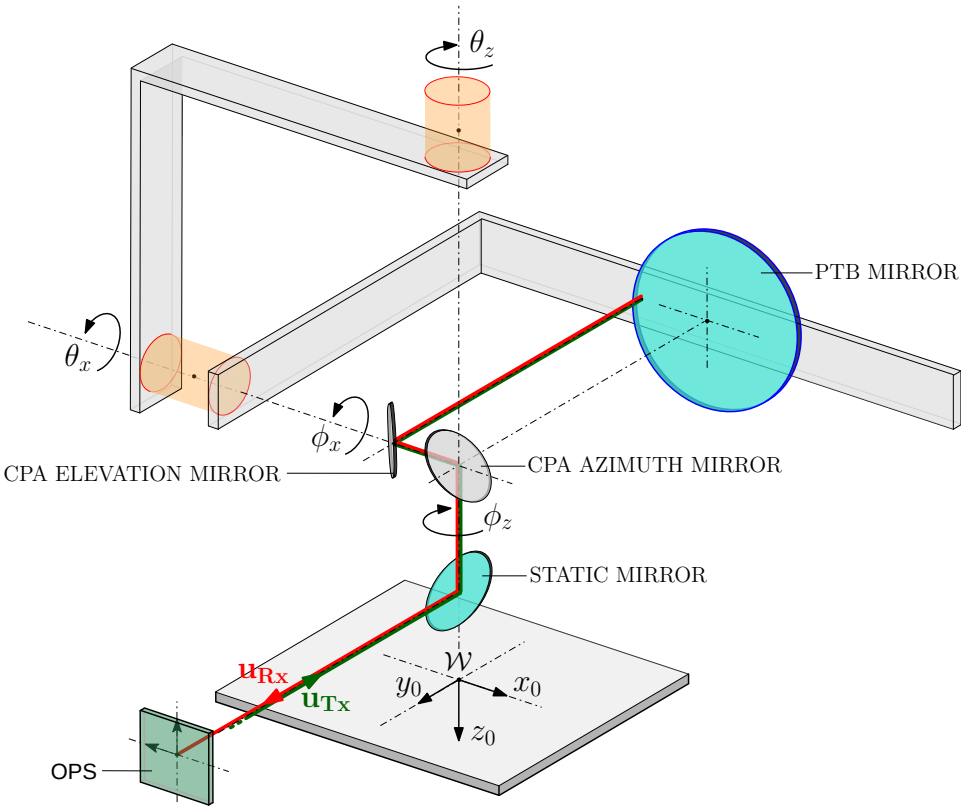


Figure 4.2: This is a schematic representation of the optical alignment between the CPA and PTB optical systems. The orientations in this schematic represents the initial orientation, where the angles are zero. The rotational degrees of freedom of the CPA, are given by ϕ_z and ϕ_x for the azimuth and elevation rotation respectively. The rotational degrees of freedom of the PTB are given by θ_z and θ_x for the azimuth and elevation rotation respectively. The difference in orientation between the optical systems is detected on the OPS via the receiving beam Rx.

4.3 Active optical alignment

The optical alignment between the CPA is governed by a laser source and an optical position sensor (OPS). The difference in orientation between the CPA and PTB can be evaluated using the OPS reading. Therefore, it is essential to convert the OPS reading into a meaningful delta orientation value, representing a difference between the true CPA angle and the true PTB angle. In this context, optical kinematics play a crucial role, and it is vital to determine them appropriately.

First, Section 4.3.1 describes the process of aligning the systems using an OPS. Second, in Section 4.3.2 more intuition is provided for the behavior of the optical kinematics. In Section 4.3.3 the invertibility of the optical kinematic model is analyzed. Finally in Section 4.3.4 two configurations are provided which enable active optical alignment between the CPA and PTB.

4.3.1 Optical alignment

The laser source transmits a laser-bundle (Tx) at an arbitrary location, the pointing direction is fixed and given by unit vector $\mathbf{u_{Tx}}$. The PSD receives a laser-bundle (Rx) at a unit direction given by $\mathbf{u_{Rx}}$. Figure 4.2 visualizes the optical path during alignment.

The OPS measures the precise location of the laser beam at the sensor's surface. The OPS returns the coordinates x_{ops} and y_{ops} of the light spot. A schematic visualization of the OPS sensor measuring the location of the laser beam spot is shown in Figure 4.3. An OPS reading at the origin, means that the CPA and PTB are aligned. In other words an OPS reading given

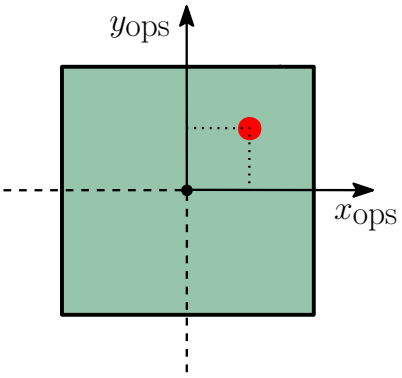


Figure 4.3: Schematic representation of the OPS surface, where the Rx-beam (●) lands.

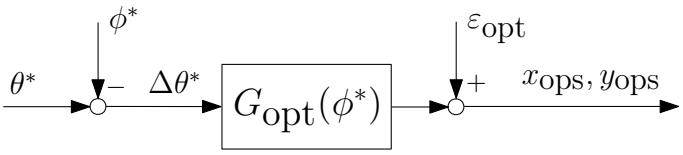


Figure 4.4: The optical gain function $G_{\text{opt}}(\phi^*)$ transforms a difference in true orientation $\Delta\theta^*$ to an OPS reading $x_{\text{ops}}, y_{\text{ops}}$.

by x_{ops}, $y_{\text{ops}} = 0$ implies that the orientation of the CPA and PTB are approximately equal in true position, i.e. $\phi^* \approx \theta^*$, where $\phi^* \in \mathbb{R}^2$ is the true orientation of the CPA, and $\theta^* \in \mathbb{R}^2$ is the true orientation of the PTB. Note that in the case of x_{ops}, $y_{\text{ops}} = 0$, the additive white noise in the OPS measurements and any disturbances in the optical path contribute to a misalignment in true position. This phenomenon is already explained in Figure 2.2 using a visual illustration for a single-dimensional example. The alignment between the CPA and PTB can be maintained by actively controlling the CPA or the PTB, such that the OPS reading stays at the origin.

The relationship between the OPS reading and the orientation of the optical systems can be described by the schematic shown in Figure 4.4. As the focus is on small orientation differences, the optical gain is considered to be linear in $\Delta\theta^*$. In this schematic, the optical gain function $G_{\text{opt}}(\phi^*) : \mathbb{R}^2 \to \mathbb{R}^2$ transform a difference in true orientation $\Delta\theta^* \in \mathbb{R}^2$ to the OPS reading $[x_{\text{ops}},\ y_{\text{ops}}]^T \in \mathbb{R}^2$, given by

$$\begin{bmatrix} x_{\text{ops}} \\ y_{\text{ops}} \end{bmatrix} = G_{\text{opt}}(\phi^*)\Delta\theta^* + \varepsilon_{\text{opt}}, \tag{4.7}$$

where $\Delta\theta^* = \theta^* - \phi^* \in \mathbb{R}^2$ and ε_{opt} describes the sensor noise and any disturbances in the optical path.

For active optical alignment, the objective is to obtain the inverse kinematic model. This can be achieved by first determining the forward kinematic model, linearizing this model, and then inverting the model to find the inverse kinematic behavior. The forward kinematics of the optical alignment are detailed in Appendix D. Although the forward kinematics are thoroughly explained, determining the inverse kinematics using the mentioned method can be challenging in practice. This is due to its time-consuming nature and the need to identify numerous geometrical constants to accurately model the forward kinematic model. Instead, this thesis presents a data-driven approach to obtain an estimate of the inverse kinematic model directly.

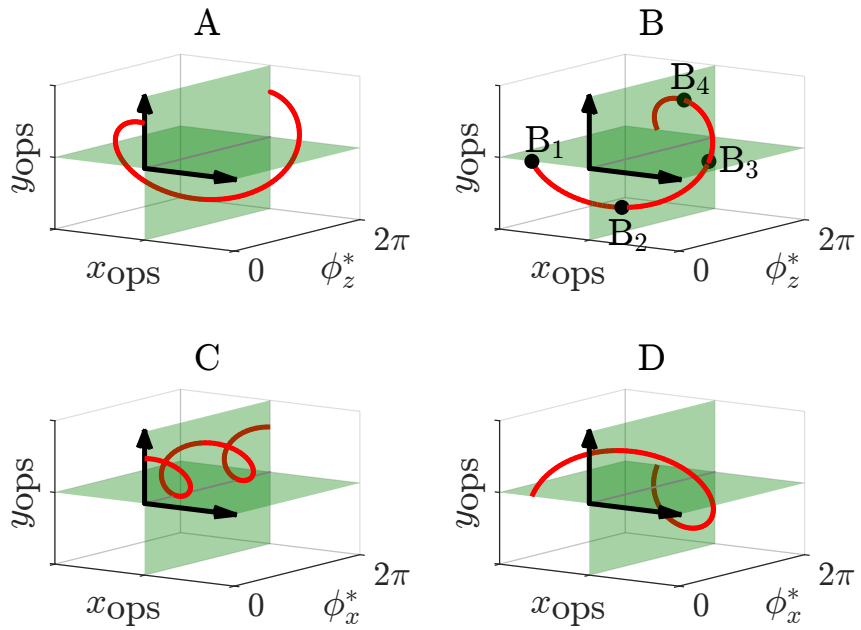


Figure 4.5: Here the OPS reading, given by x_{ops}, y_{ops} (—), is determined for each principle case A-D. These principle cases show that the optical kinematic behaviour differs significantly for varying orientations of the optical systems. For case B the position and orientation of the optical systems is visualized in Figure 4.6, at four different angles ϕ_z^* denoted with B_1-B_4. The OPS reading in principle case C shows two periods and crosses twice through the origin.

4.3.2 Optical kinematic behaviour

The optical kinematics is governed by the optical gain function that depends on the true CPA orientation ϕ^*. This section provides more insight into the behavior of the optical kinematics. In total, four principal cases will be considered, for which the forward kinematics model will be evaluated:

A: $\phi^* = \{(\phi_z^*,\ \phi_x^*) \mid 0 \leq \phi_z^* \leq 2\pi,\ \phi_x^* = 0\},\ \Delta\theta_z^* = 10^{-2},\ \Delta\theta_x^* = 0$

B: $\phi^* = \{(\phi_z^*,\ \phi_x^*) \mid 0 \leq \phi_z^* \leq 2\pi,\ \phi_x^* = 0\},\ \Delta\theta_z^* = 0,\ \Delta\theta_x^* = 10^{-2}$

C: $\phi^* = \{(\phi_z^*,\ \phi_x^*) \mid \phi_z = 0,\ 0 \leq \phi_x^* \leq 2\pi\},\ \Delta\theta_z^* = 10^{-2},\ \Delta\theta_x^* = 0$

D: $\phi^* = \{(\phi_z^*,\ \phi_x^*) \mid \phi_z^* = 0,\ 0 \leq \phi_x^* \leq 2\pi\},\ \Delta\theta_z^* = 0,\ \Delta\theta_x^* = 10^{-2}$

For each principal case, the OPS reading is plotted in Figure 4.5, showing four different circular motions. This demonstrates the effect of the optical gain function and its dependence on the true CPA orientation.

Each case shows a spiral pattern with a single period, except for case C. This is an exception and complicates the kinematics. At angles $\{\phi_x^*, \theta_x^*\} = \{1/2\pi, 1/2\pi\}$, a singular point is reached. In this specific orientation, the CPA elevation axis ϕ_x^* points towards zenith, i.e. it points directly upwards. Notably, in this orientation, any variations in $\Delta\theta_z^*$ do not result in changes to the OPS reading $x_{\text{psd}}, y_{\text{psd}}$. The same phenomenon is present at angles $\{\phi_x^*, \theta_x^*\} = \{3/2\pi, 3/2\pi\}$, i.e. when facing downwards.

For case B, the system orientations at four points have been schematically visualized in Figure 4.6, which provides more visual representation. Here, the PSD measurement makes a circular motion as the optical systems rotate, while the difference in orientation remains a positive constant, given by $\Delta\theta^* = [10^{-2},\ 0]$.

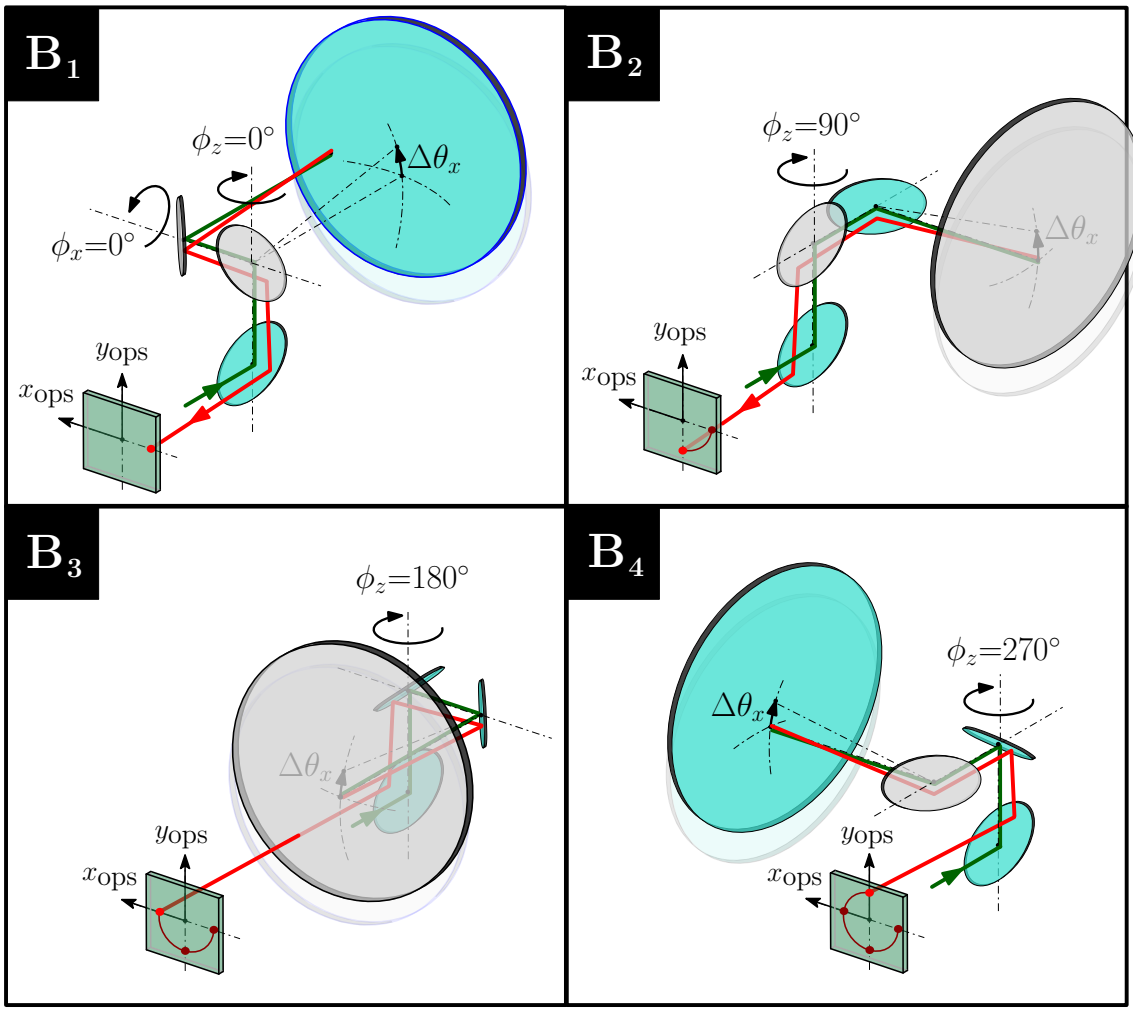


Figure 4.6: Four different orientations in case B, given by $\phi_z^* = \{0, 1/2\pi, \pi, 3/2\pi\} = \{0°, 90°, 180°, 270°\}$. The PSD reading makes a circular motion for a constant difference in elevation angle $\Delta\theta_x^*$ between the systems.

The inverse kinematics are required to enable active optical alignment between the optical systems. The following section will discuss the existence and uniqueness of the inverse optical kinematic model.

4.3.3 Invertability of the optical kinematics

Now that the optical forward kinematics have been defined, the existence of the optical inverse kinematics will be examined. It is assumed that an analytical expression for the forward kinematic model is not available, so the existence of the inverse kinematic model must be determined by acquiring data. First, the linearization against $\Delta\theta^*$ in the forward kinematics model is validated. At various orientations ϕ^*, data is made available that demonstrates a linear relationship between x_{ops} and y_{ops}. The optical gain function can be written as a square matrix given by

$$G_{\text{opt}}(\phi^*) = \begin{bmatrix} g_{11}(\phi^*) & g_{12}(\phi^*) \\ g_{21}(\phi^*) & g_{22}(\phi^*) \end{bmatrix}, \tag{4.8}$$

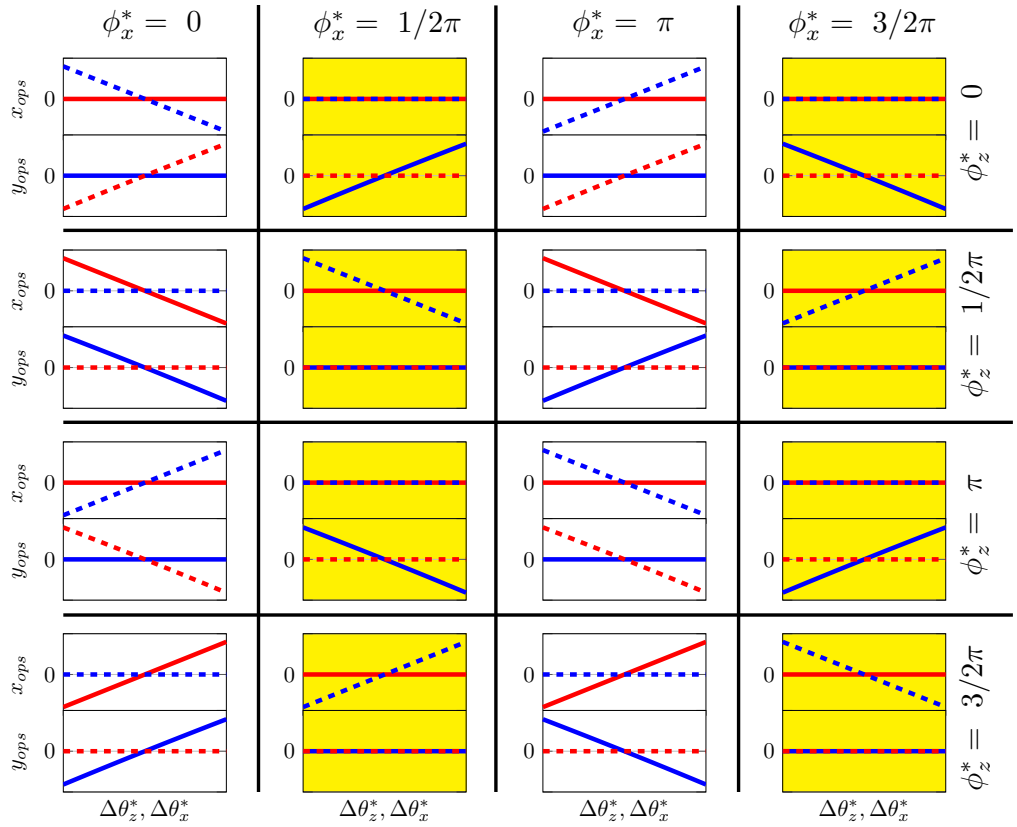


Figure 4.7: This image shows the linear behaviour of the optical gain function as a function of the CPA orientation for $\phi_x^*, \phi_z^* = \{0, 1/2\pi, \pi, 3/2\pi\}$, with $g_{11}(\phi^*)$ (—), $g_{12}(\phi^*)$ (- -), $g_{21}(\phi^*)$ (- -) and $g_{22}(\phi^*)$ (—). At the singular points $\phi_x^* = 1/2\pi$ and $\phi_x^* = 3/2\pi$, which are highlighted in the figure ( ), the optical gain function loses a single rank, and the matrix in (4.8) is thus singular. The highlighted graphs show that at least three out of four optical gains are zero, which proves the loss of rank.

which results in the optical kinematic model given by

$$\begin{aligned} x_{\text{ops}} &= g_{11}(\phi^*)\Delta\theta_z^* + g_{12}(\phi^*)\Delta\theta_x^*, \\ y_{\text{ops}} &= g_{21}(\phi^*)\Delta\theta_z^* + g_{22}(\phi^*)\Delta\theta_x^*. \end{aligned} \tag{4.9}$$

In Figure 4.7, it can be seen that at $\phi_x^* = 1/2\pi$ and $\phi_x^* = 3/2\pi$, the optical gain matrix in (4.8) is singular. This can be seen by observing that at a CPA orientation given by $\phi_z^* = 3/2, \phi_x^* = 1/2\pi$, three out of four optical gains g_{11}, g_{12}, g_{21} are zero. The singularity of a matrix can be proven by determining its determinant. For the optical gain matrix, the determinant is given by

$$\det\left(G_{\text{opt}}\right) = g_{11}(\phi^*)g_{22}(\phi^*) - g_{12}(\phi^*)g_{21}(\phi^*). \tag{4.10}$$

The determinant of this matrix at the singular point equals zero. To prove uniqueness for the inverse kinematic behavior, the determinant of G_{opt} needs to be non-zero for all ϕ^*. Figure 4.8 shows the determinant of the optical gain matrix as a function of the CPA orientation. It can indeed be observed that at the singular points, the determinant is equal to zero. The inverse optical kinematic model can be described by the inverse optical gain function, which is given by

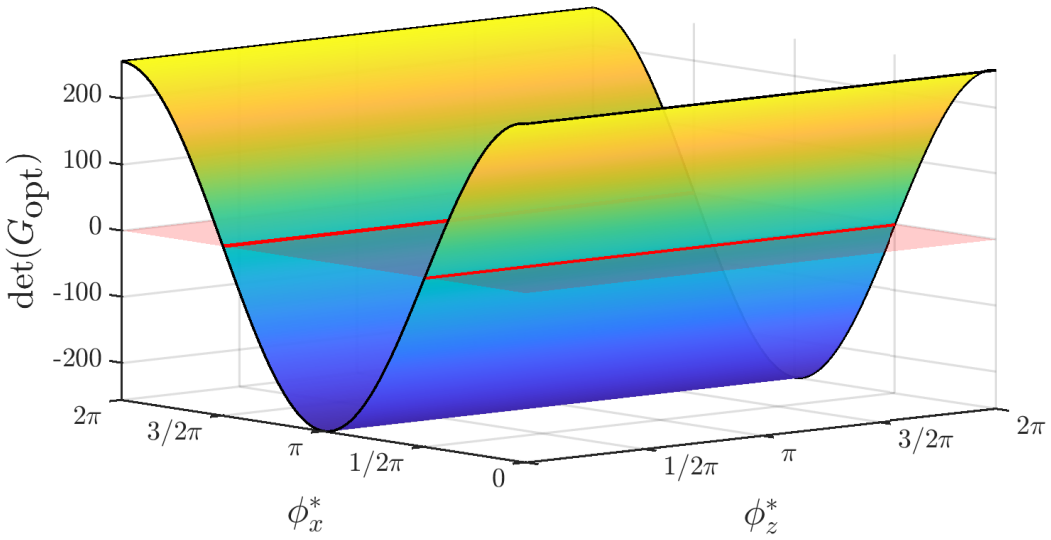


Figure 4.8: The determinant for the optical gain matrix as a function of the true CPA orientation ϕ^*. At the elevation angles $\phi_x^* = 1/2\pi$ and $\phi_x^* = 3/2\pi$ the determinant is zero (—).

$$
\begin{aligned}
F_{\text{opt}}(\phi^*) &= \begin{bmatrix} f_{11}(\phi^*) & f_{12}(\phi^*) \\ f_{21}(\phi^*) & f_{22}(\phi^*) \end{bmatrix} \\
&= \frac{1}{\det\left(G_{\text{opt}}\right)} \begin{bmatrix} g_{22}(\phi^*) & -g_{12}(\phi^*) \\ -g_{21}(\phi^*) & g_{11}(\phi^*) \end{bmatrix}
\end{aligned}
\tag{4.11}
$$

Here, $F_{\text{opt}}(\phi^*) : \mathbb{R}^2 \to \mathbb{R}^2$ is the inverse optical gain function that converts an OPS reading into a difference in orientation, given by

$$
\begin{aligned}
\Delta\theta_z^* &= f_{11}(\phi^*)x_{\text{ops}} + f_{12}(\phi^*)y_{\text{ops}} \\
\Delta\theta_x^* &= f_{21}(\phi^*)x_{\text{ops}} + f_{22}(\phi^*)y_{\text{ops}}
\end{aligned}
\tag{4.12}
$$

A unique inverse kinematic model holds everywhere except for the singular points where $\det\left(G_{\text{opt}}\right)$ is equal to zero.

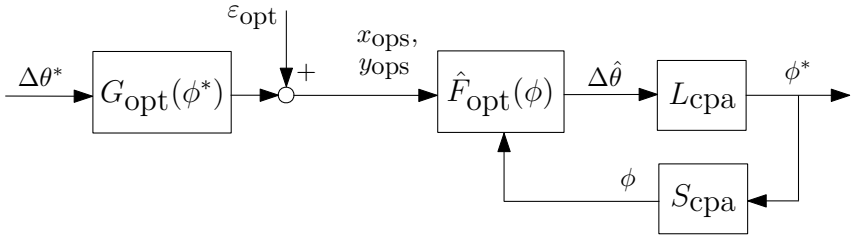
4.3.4 Controlling the optical alignment

The optical alignment can be actively maintained by adjusting the orientation of the optical systems such that the beam spot on the PSD surface stays at the center, i.e. $[x_{\text{ops}}, y_{\text{ops}}]^T = [0,\ 0]^T$ or $[\Delta\theta_z^*, \Delta\theta_x^*]^T = [0,\ 0]^T$.

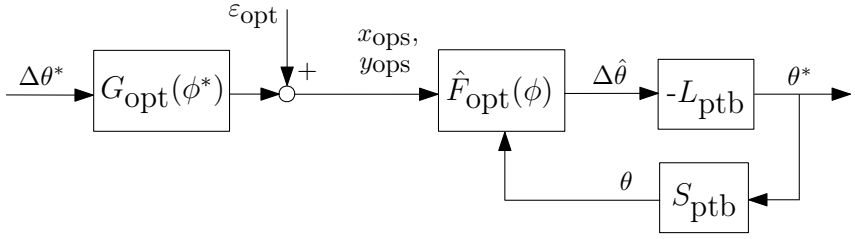
The previous section demonstrated that the forward optical kinematic model is a function of the true orientation of the CPA. The inverse kinematic model was also introduced in (4.12), which transforms an OPS reading x_{ops}, y_{ops}, into an estimation of the difference in true orientation $\Delta\theta^*$. In practice, onboard sensors on the CPA and PTB are used to obtain ϕ and θ, which contain position-dependent inaccuracies η_ϕ and η_θ as described in Section 4.1.

There are two configurations in order to actively stabilize the optical alignment:

- PTB: slave, CPA: master.



(a) PTB: slave, CPA: master



(b) PTB: master, CPA: slave

Figure 4.9: Two configurations to stabilize the optical alignment. In **(a)** the PTB is actively maintaining the optical alignment, where L_{ptb} and S_{ptb} the PTB open-loop transfer function and the PTB sensor respectively. In **(b)** the CPA is actively maintaining the optical alignment, with L_{cpa} and S_{cpa} the CPA open-loop transfer function and the CPA sensor respectively.

- PTB: master, CPA: slave,

The *slave* actively maintains the optical alignment, while the *master* can move freely. The feedback control diagram for both configurations is provided in Figure 4.9. The expected speed and acceleration are anticipated to be lower on the larger and heavier PTB compared to the CPA. Thus, the second configuration is deemed more reasonable for optical alignment. These configurations demonstrate that an estimate of the inverse kinematic model $\hat{F}(\phi)$ is used, which transforms an OPS reading $x_{\text{ops}}, y_{\text{ops}}$ into an estimate of the difference in true orientation $\Delta\hat{\theta}$.

The estimation of the optical inverse kinematics is crucial for active optical alignment. Optical alignment ensures that the true orientation of both optical systems is equal, enabling sensor calibration. This thesis focuses on the estimation of the optical inverse kinematics using GPR models. Although many other estimation methods exist, GPR modeling is employed in this work to provide more intuition into the multi-dimensional capabilities and varying model structures for GPR modeling.

4.4 Data-driven estimation of the inverse optical kinematics

In this section, the data-driven estimation of the optical kinematics is presented for the optical setup shown in Figure 4.2. A regression problem is defined and solved using multi-dimensional GPR. The data is generated via the forward kinematic model provided in Appendix D. It is assumed that the geometric parameters defined in this kinematic model are unknown. To simplify the regression problem, it is also assumed that the sensor measurements of the optical systems do not contain any position-dependent inaccuracies, i.e. $\phi \approx \phi^*$ and $\theta \approx \theta^*$.

First in Section 4.4.1 a procedure is described to obtain a data-set. In Section 4.4.2 the objective function and the regression problems are defined. Afterward, the GPR prediction models are defined in Section 4.4.3. These GPR models have a four-dimensional input space, and thus the model structure is crucial for accurate modeling of this process. The GPR model structure and the choice for the kernel function are provided in Section 4.4.4. Section 4.4.5 explains the

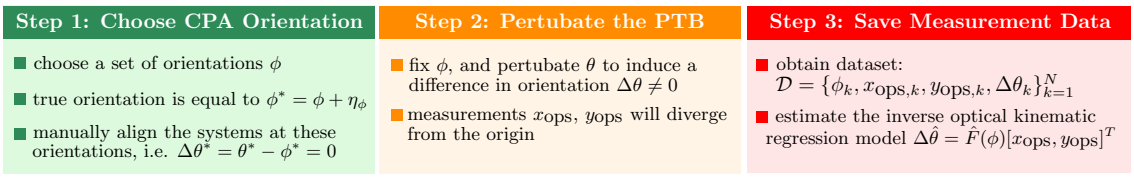


Figure 4.10: A three-step procedure to obtain a data-set used to estimate a regression model for the inverse optical kinematics.

procedure for hyperparameter optimization for multi-dimensional processes. Finally in Section 4.4.6 simulation results are provided for the estimation of the inverse optical kinematics.

4.4.1 Obtaining measurement data for optical kinematic model estimation

This section briefly describes the procedure for obtaining a data-set that will be used to find the inverse kinematic model. The procedure consists of three steps and is schematically visualized in Figure 4.10. The following steps are required:

1. Select the CPA orientations for data acquisition. At these locations, manually align the PTB with the CPA by finding the angles ϕ and θ at which the OPS reads zero values. By aligning the systems, it is ensured that the true orientation of the CPA and PTB is equal, i.e., $\Delta\theta^* = \theta^* - \phi^* = 0$.
2. Perturb the PTB to induce a difference in orientation $\Delta\theta$ between the systems.
3. Save the measurement data to a data-set $\mathcal{D} = \{\phi_k, x_{\text{ops},k}, y_{\text{ops},k}, \Delta\theta_k\}_{k=1}^N$ and estimate a regression model $\hat{F}_{\text{opt}}(\phi)$. This regression model can be used to actively align the optical systems as shown in Figure 4.9.

The effects of sensor inaccuracies on the estimation of the inverse kinematic model will not be discussed in this section. It is assumed for this example that the sensor measurements contain no inaccuracies by stating $\eta_\phi, \eta_\theta = 0$, or equivalently by stating $\phi \approx \phi^*$ and $\theta \approx \theta^*$. The following section introduces the regression problem, where the data-set obtained through this procedure can be utilized.

4.4.2 Regression problem for optical kinematics

The respective x_{psd} and y_{psd} locations on the PSD surface, can be parameterized by the CPA orientation ϕ and the difference in orientation $\Delta\theta$. The optical kinematic model is described by

$$\begin{aligned} \Delta\theta_z &= F_{\text{opt},z}(\phi, x_{\text{psd}}, y_{\text{psd}}), \\ \Delta\theta_x &= F_{\text{opt},x}(\phi, x_{\text{psd}}, y_{\text{psd}}). \end{aligned} \tag{4.13}$$

Given data-set $\mathcal{D}_i = \{z_k, \Delta\theta_{i,k}\}_{k=1}^{N_i}$, where $z_k = \{\phi_k, x_{\text{psd},k}, y_{\text{psd},k}\}$, where $i \in \{z, x\}$, both functions in (4.13) can be estimated. To find a model estimate $\Delta\hat{\theta}_i = \hat{F}_{\text{opt},i}(z_k), \forall i \in \{z, x\}$, regression problems can be defined given by

$$\begin{aligned} \beta_z^\star &= \underset{\beta_z}{\text{argmin}} \sum_{k=1}^{N_z} \left(\hat{F}_{\text{opt},z}(z_k, \beta_z) - F_{\text{opt},z}(z_k) \right)^2 + \lambda \left\| \beta_z \right\|^2, \\ \beta_x^\star &= \underset{\beta_x}{\text{argmin}} \sum_{k=1}^{N_x} \left(\hat{F}_{\text{opt},x}(z_k, \beta_x) - F_{\text{opt},x}(z_k) \right)^2 + \lambda \left\| \beta_y \right\|^2, \end{aligned} \tag{4.14}$$

where $\beta_z^\star$ and $\beta_x^\star$ are the optimal model parameters.

4.4.3 Gaussian process regression for optical kinematic estimation

Here the GPR predictive equations are provided for the inverse kinematic model. The inverse kinematic model is given by two processes given by $\Delta\theta_i = F_{\text{opt},i}(z)$, $i \in \{z, x\}$ in (4.13), where the GP prior of the inverse kinematic process is given by,

$$F_{\text{opt},i}(z) \sim \mathcal{GP}(m(z), k_2(z_A, z_B)). \tag{4.15}$$

Given the data-set $\mathcal{D} = \{\bar{z}_k, \Delta\bar{\theta}_k\}_{k=1}^N$, where $\bar{z}_k = \{\bar{\phi}_k, \bar{x}_{\text{psd},k}, \bar{y}_{\text{psd},k}\}$, the likelihood is given by

$$p(\Delta\bar{\Theta}_i|\bar{Z}, \beta_i) \sim \mathcal{N}\left(F_{\text{opt},i}(\bar{Z}, \beta_i), \Sigma_{\bar{\Theta}_i}\right), \tag{4.16}$$

where $\bar{Z} = \begin{bmatrix}\bar{z}_1 & \bar{z}_2 & \dots & \bar{z}_N\end{bmatrix}^T \in \mathbb{R}^{N\times 4}$ and $\Sigma_{\Delta\bar{\Theta}_i} = \sigma_n^2 I$ the covariance matrix of the observations. The predictive distribution for any arbitrary $\mathbf{z} \in \mathbb{R}^M$ becomes

$$\begin{aligned} p(\hat{F}_{\text{opt},i}(\mathbf{z}) \mid \bar{Z}, \Delta\bar{\Theta}_i) &\sim \mathcal{N}\left(\hat{F}_{\text{opt},i}(\mathbf{z}),\ \text{cov}(\hat{F}_{\text{opt},i}(\mathbf{z}))\right), \quad \text{where} \\ \boldsymbol{\Delta\hat{\theta}}_i &:= \mathbb{E}\left[\hat{F}_{\text{opt},i}(\mathbf{z})\right] = m(\mathbf{z}) + k_i(\mathbf{z})^T\left[K_i + \Sigma_{\bar{\Theta}_i}\right]^{-1}(\Delta\bar{\Theta}_i - m(\bar{Z})), \\ \text{cov}(\boldsymbol{\Delta\hat{\theta}}_i) &:= \text{cov}(\hat{F}_{\text{opt},i}(\mathbf{z})) = k_i(\mathbf{z},\mathbf{z}) - k_i(\mathbf{z})^T\left[K_i + \Sigma_{\bar{\Theta}_i}\right]^{-1} k_i(\mathbf{z}), \end{aligned} \tag{4.17}$$

4.4.4 Model structure for optical kinematic estimation

The model structure for multi-dimensional model structures is rather straightforward. Each input dimension can be addressed by its own kernel function and corresponding hyperparameters. There are thus many ways of constructing a kernel function for multi-dimensional processes. In Section 2.4.4 a model structure for a one-dimensional GPR model has been provided.

The inverse optical kinematic process as given by (4.13) has in total four input dimensions. For each input dimension, a kernel function can be specified, and these can be combined in order to describe the relationship and interaction between the multi-dimensional input and the output. By carefully selecting and combining kernel functions, one can construct a GP prior model that captures the underlying structure of the data, accounting for additional properties such as linearity, periodicity, and smoothness [39].

In the four-dimensional input space of the inverse optical kinematic process, two different kernels are employed to model the structure of the function $F_{\text{opt},i}$. Specifically, a periodic kernel for the first two input dimensions and a linear kernel for the third and fourth input dimensions. A mask is applied to isolate the relevant dimensions for each kernel, by defining

$$\begin{aligned} m_{\text{per}}(z) &= \begin{bmatrix} z_1, z_2 \end{bmatrix} = \begin{bmatrix} \phi_z, \phi_x \end{bmatrix}, \\ m_{\text{lin}}(z) &= \begin{bmatrix} z_3, z_4 \end{bmatrix} = \begin{bmatrix} x_{\text{psd}}, y_{\text{psd}} \end{bmatrix}. \end{aligned} \tag{4.18}$$

This kernel construction allows us to capture the unique characteristics of each input dimension, such as periodicity in the first two dimensions and linear relationships in the third and fourth dimensions. The combined kernel function is then defined as the product of the individual kernels applied to their respective masked inputs, given by

$$k_i(z_A, z_B) = k_{\text{per}}(m_{\text{per}}(z_A), m_{\text{per}}(z_B))k_{\text{lin}}(m_{\text{lin}}(z_A), m_{\text{lin}}(z_B)) \tag{4.19}$$

where the kernels $k_{\text{per}}(z_A, z_B)$ are a periodic kernel and $k_{\text{lin}}(z_A, z_B)$ a linear kernel. These kernels are defined by

$$\begin{aligned} k_{\text{per}}(m_{\text{per}}(z_A), m_{\text{per}}(z_B)) &:= \prod_{d=1}^{2} \sigma_d^2 \exp\left(-\frac{2}{l_{\text{per},d}^2}\sin\left(\pi\frac{|m_{\text{per},d}(z_A) - m_{\text{per},d}(z_B)|}{p}\right)\right) \\ &:= \sigma_f^2 \exp\left(-2\sum_{d=1}^{2}\frac{1}{l_{\text{per},d}^2}\sin\left(\pi\frac{|m_{\text{per},d}(z_A) - m_{\text{per},d}(z_B)|}{p}\right)\right), \\ k_{\text{lin}}(m_{\text{lin}}(z_A), m_{\text{lin}}(z_B)) &:= m_{\text{lin}}(z_A)^T P\ m_{\text{lin}}(z_B), \end{aligned} \tag{4.20}$$

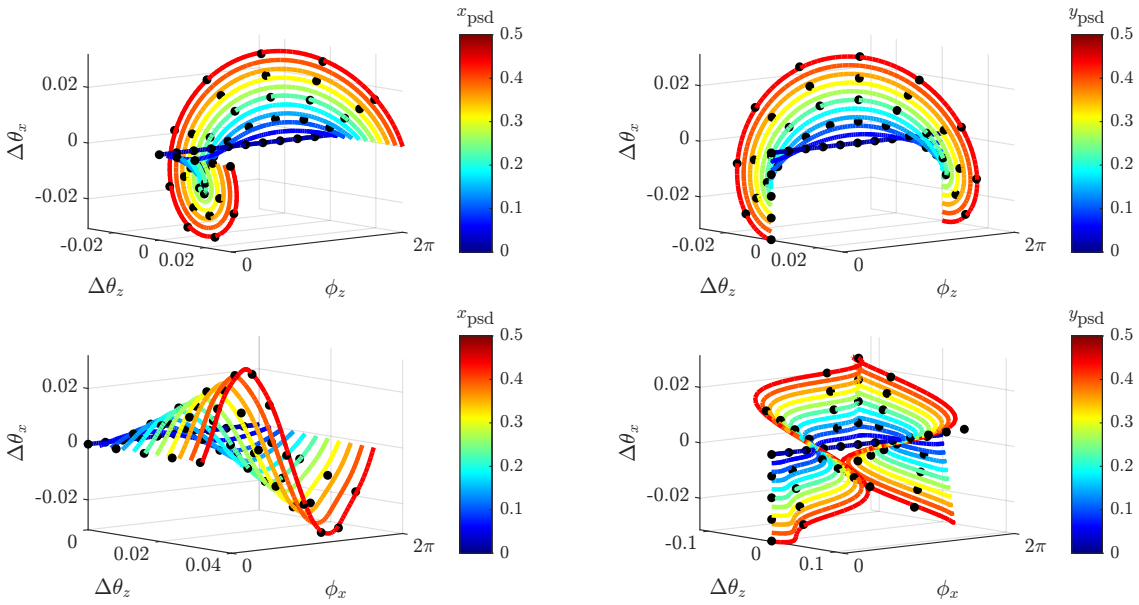


Figure 4.11: The estimated inverse kinematic model queried at several points. The true inverse kinematic model is also visualized (•).

where $P = \text{diag}(l^2_{\text{lin},1}, l^2_{\text{lin},2})$, insinuating that the interaction between the linear components in $m_{\text{lin}}(z)$ is assumed to be negligible. The periodic kernel has in total four hyperparameters, given by the signal variance σ_f^2, the period p, and smoothness parameters $l^2_{\text{per},1}$, $l^2_{\text{per},2}$.

4.4.5 Hyperparameter optimization

The hyperparameter optimization for multi-dimensional GPs is similar to one-dimensional processes which are already provided in Section 2.4.5. However, the computational effort of the optimization increases drastically for larger kernel functions. The hyperparameters for the inverse kinematic model are given by $\Theta_i = \{l^2_{\text{lin},1}, l^2_{\text{lin},2}, l^2_{\text{per},1}, l^2_{\text{per},2}, \sigma_f^2\}$. The period p is fixed at 2π.

4.4.6 Simulation results

Here simulation results are presented for the estimation of the inverse optical kinematics. As previously stated the forward kinematic model is used to generate data. The geometrical parameters that govern the behavior of the optical kinematic model are assumed to be unknown. The predictive distribution in (4.17) is evaluated at a grid of input locations $\mathbf{z} \in \mathbb{R}^{M \times D}$ and visualized in Figure 4.11. Similar to Figure 4.5 which visualized the forward kinematics in four separate cases, here a similar approach is taken. Although, here, the OPS measurements are also varied, which shows how the spiral pattern increases or decreases in size.

It is important to note that the GPR model does not take into account the singularities at elevation angles $\phi_x = 1/2\pi$ and $\phi = 3/2\pi$. This causes the GPR model to perform poor predictions near these points.

4.5 Cascaded calibration of Hall encoders

The cascaded calibration procedure has been thoroughly explained in Section 4.2. Here the GPR-based cascaded calibration of the CPA Hall encoder is provided. This is assuming the second step in this procedure, which involves the optical alignment of the PTB with the CPA, is already accomplished.

First, in 4.5.1, the objective function and regression problems are defined for a 2-DOF calibration procedure of Hall encoders. Subsequently, in Section 4.5.2, the GPR models are introduced. The model structure for these GPR models, including the choice for kernel and mean function, is discussed in Section 4.5.3. Finally, in Section 4.5.4, simulation results of a calibration procedure are presented. The generated sensor inaccuracies are based on real data obtained from experimental CPAs and the PTB calibration provided in [6].

4.5.1 Cascaded regression problem for coarse pointing assemblies

The 2DOF calibration procedure of the CPA Hall encoders naturally extends from the 1DOF example in Section 2.4.2. The cost to be optimized can be defined separately for both degrees of freedom, given by

$$\min_{\beta_{1,i},\beta_{2,i}} \mathcal{J}_i = \left(\frac{\int_0^{2\pi} \left(\hat{f}_{1\to 3,i}\left(\phi, \beta_{1,i}\right) - f_{1\to 3,i}(\phi) \right)^2 \, d\phi}{2\pi} \right)^{\frac{1}{2}}, \quad \forall i \in \{z, x\}. \tag{4.21}$$

For each degree of freedom the cost function $\mathcal{J}_i$ is subject to two regression problems, given by

$$\begin{aligned} \beta_{2,i}^{\star} &= \underset{\beta_{2,i}}{\operatorname{argmin}} \sum_{k=1}^{N_{2,i}} \left(\hat{f}_{2\to 3,i}\left(\bar{\theta}_k, \beta_{2,i}\right) - f_{2\to 3,i}\left(\bar{\theta}_k\right) \right)^2 + \lambda \left\| \beta_{2,i} \right\|^2, \\ \beta_{1,i}^{\star} &= \underset{\beta_{1,i}}{\operatorname{argmin}} \sum_{k=1}^{N_{1,i}} \left(\hat{f}_{1\to 3,i}\left(\bar{\phi}_k, \beta_{1,i}\right) - \hat{f}_{2\to 3,i}\left(\underline{\bar{\theta}}_k, \beta_{2,i}^{\star}\right) \right)^2 + \lambda \left\| \beta_{1,i} \right\|^2, \end{aligned} \tag{4.22}$$

The overline in $\bar{\theta}$ signifies that it represents discrete PTB measurement data utilized for calibrating the PTB sensors. The additional underline in $\underline{\bar{\theta}}$ serves to differentiate that the PTB measurement data is acquired specifically for the calibration of the Hall encoder.

4.5.2 Gaussian process regression for 2DOF CPAs

The regression problems discussed in previous sections are addressed through GPR models, which extend the 1DOF example presented in Section 2.4.3. Initially, a first-stage regression problem is tackled to calibrate the PTB sensors using theodolite measurements. Assuming optical alignment between the PTB and CPA has already been established, the process continues with a virtual regression that calibrates the CPA Hall encoders.

Calibration of the PTB sensors

The calibration of the PTB sensors serves as a first-stage regression. The processes being estimated are given by (4.3). The GP priors are defined as

$$f_{2\to 3,i}(\theta) \sim \mathcal{GP}(m_{2,i}(\theta), k_{2,i}(\theta_A, \theta_B)), \; i \in \{z, x\}. \tag{4.23}$$

Given the data-set $\mathcal{D}_{2,i} = \{\bar{\Theta}, \bar{\Theta}_{t,i}\}$, the likelihood is given by

$$p(\bar{\Theta}_{t,i} | \bar{\Theta}, \beta_{2,i}) \sim \mathcal{N}\left(f_{2\to 3,i}(\bar{\Theta}, \beta_{2,i}), \Sigma_{\bar{\Theta}_{t,i}} \right), \tag{4.24}$$

where $\Sigma_{\bar{\Theta}_{t,i}} \in S^{N_2}_{++}$, the covariance matrix of the theodolite observations. The predictive distribution for any arbitrary measurements $\boldsymbol{\theta} \in \mathbb{R}^{M_2 \times 2}$ becomes

$$
\begin{aligned}
p(\hat{f}_{2\to3,i}(\boldsymbol{\theta}) \mid \bar{\Theta}_{t,i}|\bar{\Theta}) \sim \mathcal{N}\left(\hat{f}_{2\to3,i}(\boldsymbol{\theta}),\ \mathrm{cov}(\hat{f}_{2\to3,i}(\boldsymbol{\theta}))\right), \quad \text{where} \\
\hat{\boldsymbol{\theta}}_t := \mathbb{E}\left[\hat{f}_{2\to3,i}\boldsymbol{\theta})\right] = m_{2,i}(\boldsymbol{\theta}) + k_{2,i}(\boldsymbol{\theta})^T \left[K_{2,i} + \Sigma_{\bar{\Theta}_{t,i}}\right]^{-1} (\bar{\Theta}_{t,i} - m_{2,i}(\bar{\Theta})), \\
\mathrm{cov}(\hat{\boldsymbol{\theta}}_t) := \mathrm{cov}(\hat{f}_{2\to3,i}(\boldsymbol{\theta})) = k_{2,i}(\boldsymbol{\theta},\boldsymbol{\theta}) - k_{2,i}(\boldsymbol{\theta})^T \left[K_{2,i} + \Sigma_{\bar{\Theta}_{t,i}}\right]^{-1} k_{2,i}(\boldsymbol{\theta}),
\end{aligned}
\tag{4.25}
$$

where $k_{2,i}(\boldsymbol{\theta}) := k_{2,i}(\bar{\Theta},\boldsymbol{\theta}) = k_{2,i}(\boldsymbol{\theta},\bar{\Theta})^T \in \mathbb{R}^{N_2\times M_2}$ and $K_{2,i} := k_{2,i}(\bar{\Theta},\bar{\Theta}) \in \mathbb{R}^{N_2\times N_2}$. The mean prediction $\hat{\boldsymbol{\theta}}_{t,i} := \mathbb{E}[\hat{f}_{2\to3,i}(\boldsymbol{\theta})]$ maps any given measurements θ into an estimate of the measurement $\theta_{t,i}$. The uncertainty of the predictions, i.e. the covariance estimate $\mathrm{cov}(\hat{f}_{2\to3,i}(\boldsymbol{\theta}))$, will be utilized in the following virtual regression problem regarding the CPA Hall encoder calibration.

Calibration of the CPA Hall encoders

The processes that we are trying to estimate for the CPA hall encoder are given by (4.5). The GP prior is defined as

$$
f_{1\to3,i}(\phi) \sim \mathcal{GP}(m_{1,i}(\phi), k_{1,i}(\phi_A,\phi_B)). \tag{4.26}
$$

A data-set $\mathcal{D}_1 = \{\bar{\Phi}, \underline{\bar{\Theta}}\}$ is measured, and transformed into a virtual data-sets $D'_{1,i} = \{\bar{\Phi},\ \hat{\Theta}_{t,i}\}$, where $\hat{\Theta}_{t,i} := \hat{f}_{2\to3,i}(\underline{\bar{\Theta}})$. The likelihood is given by

$$
p(\hat{\Theta}_{t,i}|\bar{\Phi},\beta_{1,i}) \sim \mathcal{N}\left(f_{1\to3,i}(\bar{\Phi},\beta_{1,i}), \Sigma_{\hat{\Theta}_{t,i}}\right). \tag{4.27}
$$

The likelihood covariance $\Sigma_{\hat{\Theta}_{t,i}}$ is fully described by the model uncertainty $\mathrm{cov}(\hat{f}_{2\to3,i}(\underline{\bar{\Theta}}))$, i.e

$$
\begin{aligned}
\hat{\Theta}_{t,i} &:= [\hat{f}_{2\to3,i}(\underline{\bar{\theta}}_1), \ldots, \hat{f}_{2\to3,i}(\underline{\bar{\theta}}_{N_1})]^\top, \\
\Sigma_{\hat{\Theta}_{t,i}} &:= \mathrm{cov}(\hat{f}_{2\to3,i}(\underline{\bar{\Theta}})).
\end{aligned}
\tag{4.28}
$$

The predictive distribution for any arbitrary measurements $\boldsymbol{\phi} \in \mathbb{R}^{M_1\times 2}$ becomes

$$
\begin{aligned}
p(\hat{f}_{1\to3,i}(\boldsymbol{\phi}) \mid \bar{\Phi}, \underline{\bar{\Theta}}) \sim \mathcal{N}\left(\hat{f}_{1\to3,i}(\boldsymbol{\phi}), \mathrm{cov}(\hat{f}_{1\to3,i}(\boldsymbol{\phi}))\right), \quad \text{where} \\
\tilde{\boldsymbol{\theta}}_{t,i} := \mathbb{E}\left[\hat{f}_{1\to3,i}(\boldsymbol{\phi})\right] = m_{1,i}(\boldsymbol{\phi}) + k_{1,i}(\boldsymbol{\phi})^T \left[K_{1,i} + \Sigma_{\hat{\Theta}_{t,i}}\right]^{-1} (\hat{\Theta}_{t,i} - m_{1,i}(\bar{\Phi})), \\
\mathrm{cov}(\tilde{\boldsymbol{\theta}}_{t,i}) := \mathrm{cov}(\hat{f}_{1\to3,i}(\boldsymbol{\phi})) = k_{1,i}(\boldsymbol{\phi},\boldsymbol{\phi}) - k_{1,i}(\boldsymbol{\phi})^T \left[K_{1,i} + \Sigma_{\hat{\Theta}_{t,i}}\right]^{-1} k_{1,i}(\boldsymbol{\phi}),
\end{aligned}
\tag{4.29}
$$

where $k_{1,i}(\boldsymbol{\phi}) := k_{1,i}(\bar{\Phi},\boldsymbol{\phi}) = k_{1,i}(\boldsymbol{\phi},\bar{\Phi})^T \in \mathbb{R}^{N_1\times M_1}$ and $K_{1,i} := k_{1,i}(\bar{\Phi},\bar{\Phi}) \in \mathbb{R}^{N_1\times N_1}$. The mean prediction $\hat{f}_{1\to3,i}(\phi) := \mathbb{E}[\hat{f}_{1\to3,i}(\phi)]$ maps any given measurements ϕ into an estimate of the measurement $\theta_{t,i}$. The following sections discuss the choices for appropriate kernels and mean functions given the information in Section 4.1 regarding the system characteristics.

4.5.3 Model structure and hyperparameter optimization

The PTB and CPA system characteristics in Section 4.1 offer valuable insights into the expected sensor inaccuracies. This information enables the selection of an appropriate kernel capable of accurately modeling these inaccuracies.

Section 2.4.4 presents the idea that, for sensor calibration purposes, the mean function can be chosen as the input of the GPR model. The mean functions in the cascaded calibration model are given by

$$\begin{aligned} m_{2,i}(\theta) &= \theta_i, \\ m_{1,i}(\phi) &= \phi_i. \end{aligned} \tag{4.30}$$

The kernels then assumes full responsibility for modeling the behavior of the sensor inaccuracies. The PTB system characteristics reveal that the inaccuracies exhibit periodicity over an entire rotation around the axis. These errors, referred to as low-repeating errors η_L, can fully describe the PTB position-dependent inaccuracies for both stages η_{θ_i}, as given by (4.2). The kernel functions $k_{2,i}$ for the process described by (4.3) are expressed as

$$\begin{aligned} k_{2,i}(\theta) &= k_{\text{per},i}(\theta) \\ &= \sigma_f^2 \exp\left(-2\sum_{j=1}^{2}\frac{1}{l_{\text{per},j}^2}\sin\left(\pi\frac{|\theta_{j,A}-\theta_{j,B}|}{p}\right)\right), \quad \text{with } p = 2\pi, \end{aligned} \tag{4.31}$$

where l_j^2 is the smoothness parameter for $j \in \{z, x\}$, and σ_f^2 is the signal variance. The periodicity of the periodic kernel remains fixed at 2π.

The CPA system characteristics reveal that the Hall encoder relies on flux variation over the teeth to determine the rotor position. This introduces high-repeating position-dependent sensor inaccuracies, as indicated in (4.5). The kernel functions are chosen as a product of two kernels: one for high-repeating inaccuracies and one for low-repeating inaccuracies. These kernels are expressed as

$$\begin{aligned} k_{1,i}(\phi) &= k_{\text{per},H,i}(\phi) * k_{\text{per},L,i}(\phi) \\ &= \sigma_{f,H}^2 \exp\left(\frac{-2}{l_{\text{per},H,i}^2}\sin\left(\pi\frac{|\phi_{i,A}-\phi_{i,B}|}{p_H}\right)\right) \cdot \\ &\qquad \sigma_{f,L}^2 \exp\left(-2\sum_{j=1}^{2}\frac{1}{l_{\text{per},L,j}^2}\sin\left(\pi\frac{|\phi_{j,A}-\phi_{j,B}|}{p}\right)\right), \end{aligned} \tag{4.32}$$

with the low-repeating and high-repeating periods fixed at $p = 2\pi$ and $p_H = 2\pi/n$, where n represents the number of teeth on the CPA rotor. High-repeating position-dependent inaccuracies are present only on the axis being calibrated. To offer more insight, consider the process given by $\theta_{t,z} = f_{1\to3,z}(\phi)$, which takes both ϕ_z and ϕ_x as inputs. This process can be described by $f_{1\to3,z}(\phi) = \phi_z + \eta_{\phi_z}$. From (4.1), the inaccuracy of the CPA Hall encoder in the z-axis is given by $\eta_{\phi_z} = \eta_H(\phi_z) + \eta_L(\phi)$, where the high-repeating inaccuracies only consider ϕ_z.

Equivalent to Sections 2.4.5 and 4.4.5, the hyperparameters for each kernel $k_{1,z}$, $k_{1,x}$, $k_{2,z}$ and $k_{2,x}$ can be optimized through the maximization of its corresponding log marginal likelihood.

4.5.4 Simulation results

The GPR algorithm for cascaded calibration of the 2DOF CPA is tested in simulation. As stated previously, this simulation is based in experimental results obtained on a prototype 1DOF CPA in Appendix C. The position-dependent sensor inaccuracies as described by Section 4.2 are randomly generated. The position-dependent inaccuracies for the PTB sensors are given by

$$\eta_{\theta_i}(\theta_z,\theta_x) = \sum_{k=1}^{N_{2,L}} \left(\frac{c_{L,i,k}}{k^3}\sin(k\phi_z) + \frac{d_{L,i,k}}{k^3} cos(k\phi_z) \right) \sum_{k=1}^{N_{2,L}} \left(\frac{c_{L,i,k}}{k^3}\sin(k\phi_x) + \frac{d_{L,i,k}}{k^3} cos(k\phi_x) \right). \tag{4.33}$$

The position-dependent inaccuracies for the CPA Hall encoder are defined by,

$$\begin{aligned}\eta_{\phi_i}(\phi_z,\phi_x) = &\sum_{k=1}^{N_{1,H}} \left(a_{H,i,k}\ \sin(k\cdot n\cdot\phi_i) + b_{H,i,k}\ \cos(k\cdot n\cdot\phi_i)\right) + \\ &\sum_{k=1}^{N_{1,L}} \left(\frac{a_{L,i,k}}{k^3}\ \sin(\phi_z) + \frac{b_{L,i,k}}{k^3}\ \cos(\phi_z) \right) \sum_{k=1}^{N_{1,L}} \left(\frac{a_{L,i,k}}{k^3}\ \sin(\phi_x) + \frac{b_{L,i,k}}{k^3}\ \cos(\phi_x) \right) + \eta_{\theta_i}.\end{aligned} \tag{4.34}$$

The constants used to generate the PTB inaccuracies in (4.33) are given by:

- $a_{L,z}, a_{L,x}, b_{L,z}, b_{L,x} \sim \mathcal{N}(0, 2.5\cdot 10^{-3})$,
- $N_{2,L} = 3$.

Samples for η_{θ_z} and η_{θ_x}, that are generated from these constants, are visualized in Figure 4.12. Similarly the constants for the CPA Hall encoder inaccuracies in (4.34) are given by:

- $a_{H,z}, a_{H,x}, b_{H,z}, b_{H,x} \sim \mathcal{N}(0, 10^{-4})$,
- $a_{L,z}, a_{L,x}, b_{L,z}, b_{L,x} \sim \mathcal{N}(0, 2.5\cdot 10^{-3})$,
- $N_{1,L} = 1,\ N_{1,H} = 3$,
- the number of teeth $n = 4$.

The samples η_{θ_z} and η_{θ_x}, that are generated from these constants, are visualized in Figure 4.13. The theodolite, and PTB sensor, CPA Hall encoder measurements are given by

$$\begin{aligned}\theta_{t,i} &= \theta^*_{t,i} - \varepsilon_n,\\ \theta_i &= \theta^*_i - \eta_{\theta_i},\\ \phi_i &= \phi^*_i - \eta_{\phi_i},\end{aligned} \tag{4.35}$$

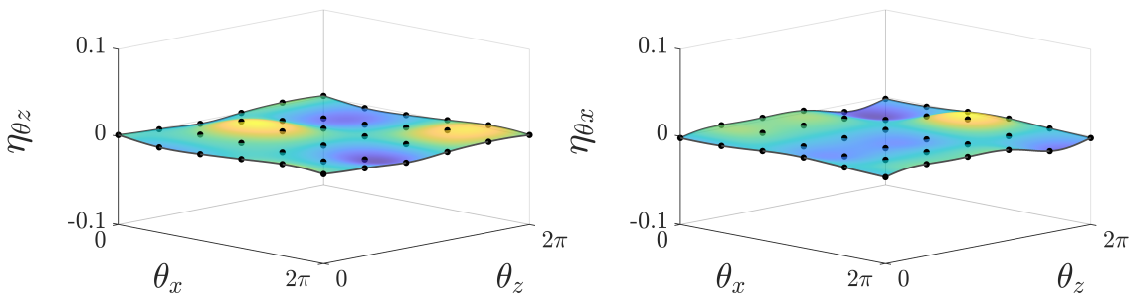


Figure 4.12: The position-dependent inaccuracies of the PTB sensors in both z-stage and x-stage, θ_z and θ_x respectively. The data-set that is obtained to calibrate the PTB is visualized and given by $\mathcal{D}_{2,z}$ (•, left) and $\mathcal{D}_{2,x}$ (•, right).

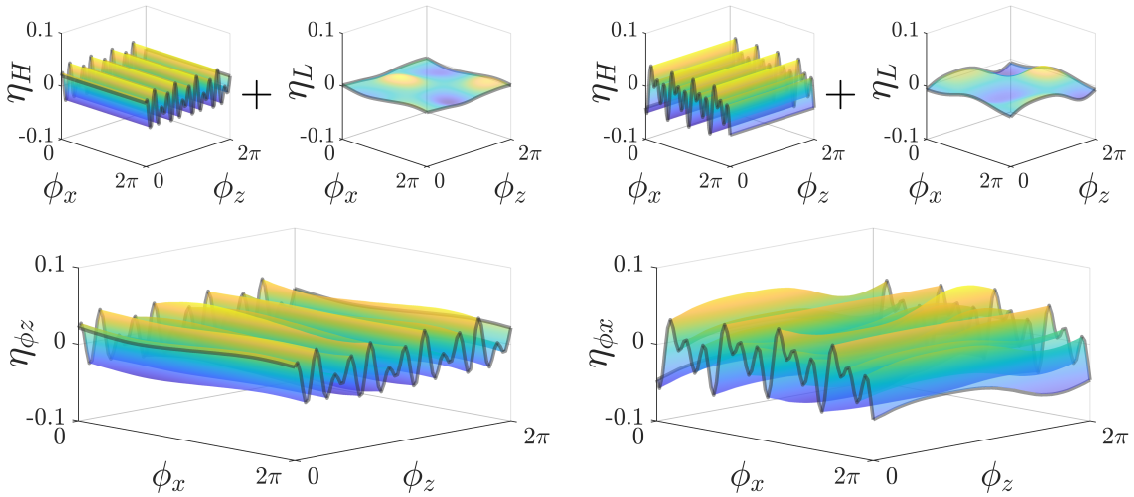


Figure 4.13: The position-dependent inaccuracies of the CPA Hall encoder in both azimuth and elevation axis, ϕ_z and ϕ_x respectively. The virtual data-sets $\mathcal{D}'_{1,z}$ and $\mathcal{D}'_{1,z}$ will be used to find a regression model. Note that, due to the size of the data-sets, they have not been visualized in the respective figure.

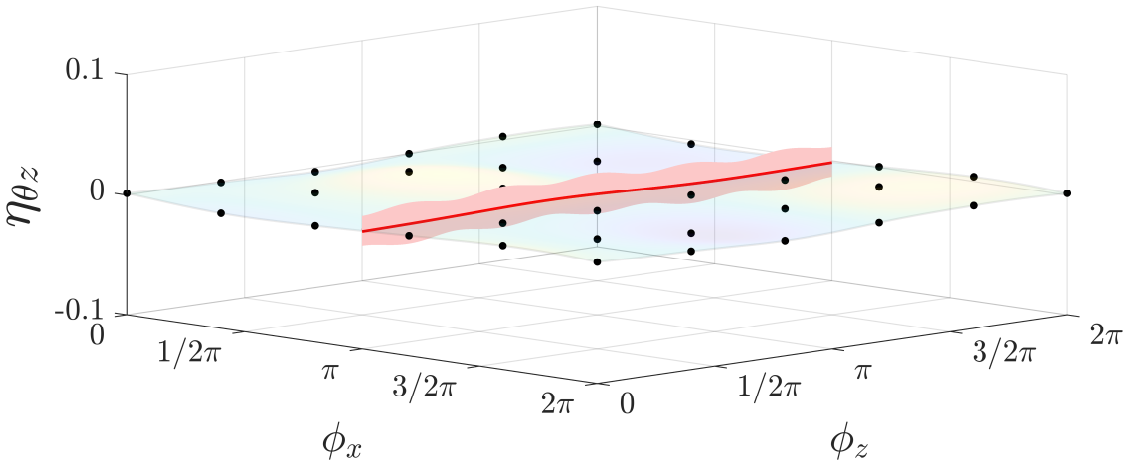
where $i = \{z, x\}$, and it is assumed the systems are perfectly aligned, i.e. the true orientations are equal $\theta_t^* = \theta^* = \phi^*$ The additive sensor noise is given by $\sigma_n^2 \sim \mathcal{N}(0, 10^{-6})$, which originates from the theodolite measurements.

Given datasets $\mathcal{D}_{2,z}$, $\mathcal{D}_{2,z}$, $\mathcal{D}'_{1,z}$, and $\mathcal{D}'_{1,z}$, the cascaded GPR model in Section 4.5.2 is employed for Bayesian inference. The first-stage regression, due to its limited quantity of data, exhibits high uncertainty in its predictions. This uncertainty is attributed to the expensive and time-consuming nature of calibration using the theodolite. The data-sets for calibration of the PTB sensors consist of a have size $N_2 = 25$, while the virtual data-set for calibration of the CPA Hall encoder is given by $N_1 = 6400$. For simplicity, the data-sets for z- and x-orientation have been chosen to be equal. The GPR models, for the z-axis only, in (4.25) and (4.29) are queried for two different inputs, given by

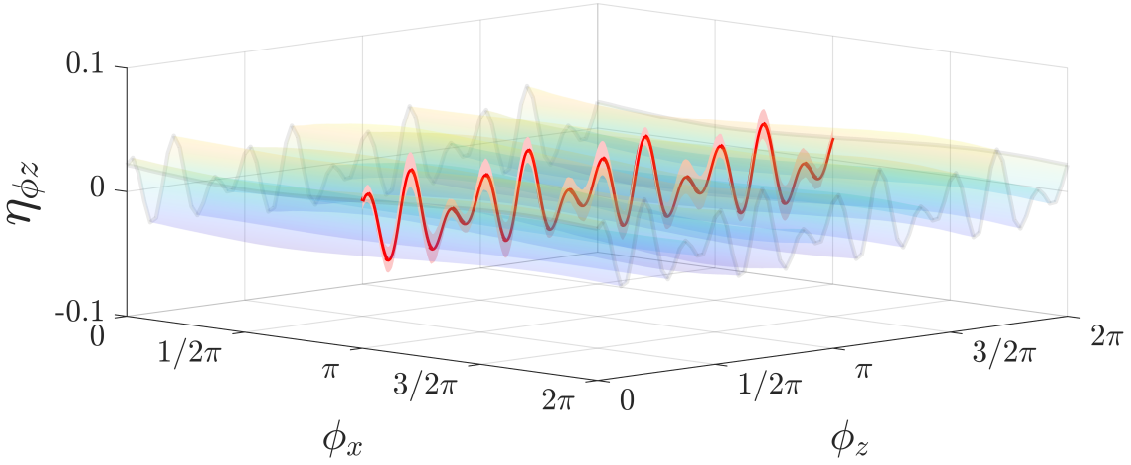
1. a pure motion in z-axis, or azimuth rotation,
2. and a diagonal motion through equal steps in both z-axis and x-axis, or azimuth- and elevation rotation.

The difference in the estimated covariance of the predicted sensor inaccuracies between these two types of motion is attributed to the scarce amount of data points for the first-stage regression problem. The dataset for the first-stage regression $\mathcal{D}_{2,z}$ contains a limited number of data points. This illustrates that the prediction uncertainty is high when the observations in the dataset are far apart. The predictions of the sensor inaccuracies in z-axis for a pure motion in z-axis, is shown in Figure 4.14. The predictions of the sensor inaccuracies in z-axis for a diagonal motion in both z-axis and x-axis is shown in Figure 4.15.

This example shows the strength of cascaded calibration through GPR models, which are able to seamlessly estimate the propagated uncertainty through each calibration stage. The following sections inspects the modeling uncertainty and utilizes the tools for PSD estimation of GPR models, which are provided in Chapter 3.

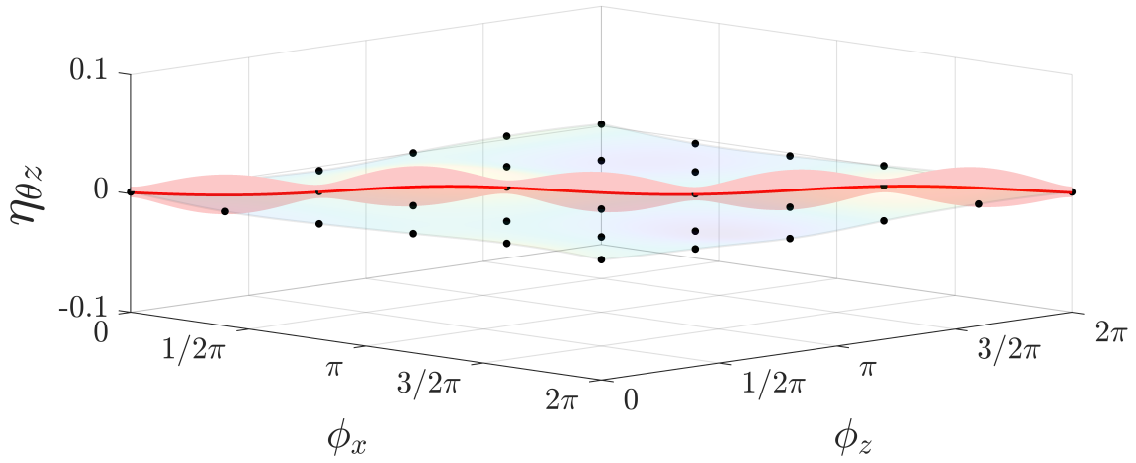


(a) The predicted sensor inaccuracy in the PTB z-stage $\hat{\eta}_{\theta_z} = \hat{\theta}_{t,z} - \theta_z$ (—, 99% interval) is shown. Through a pure motion in z-axis, at a constant elevation angle $\theta_x = \pi$, the predictions are at a large distance from the training data (•), which increases the uncertainty, given by $\text{cov}(\hat{\theta}_{t,z})$ ( ), of the predictions and increases the total uncertainty.

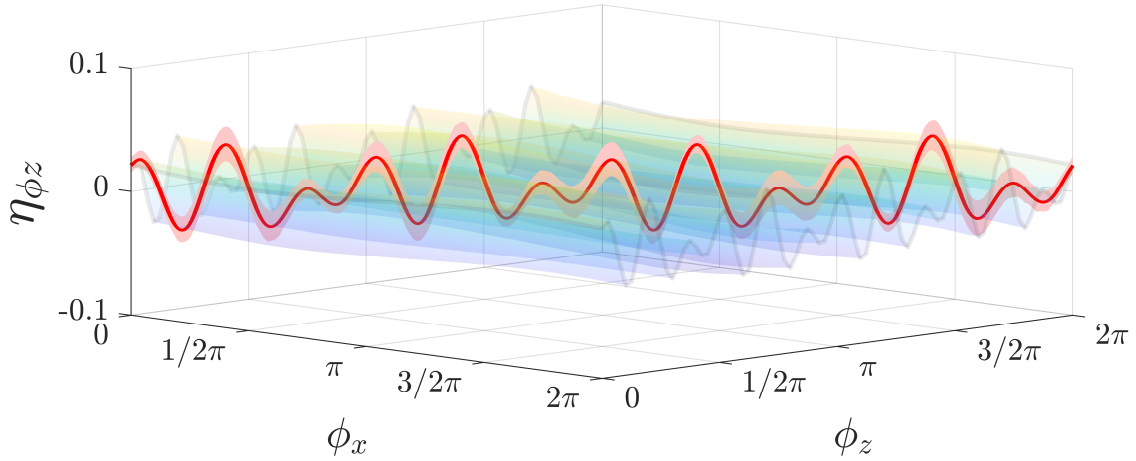


(b) The predicted sensor inaccuracy in the CPA z-axis, represented as $\hat{\eta}_{\phi_z} = \tilde{\theta}_{t,z} - \phi_z$ (—), is displayed. Although the training data-set $\mathcal{D}_{1,z}$ has a sufficient number of data points, the uncertainty from the first-stage regression of the PTB calibration model affects the prediction in the subsequent virtual regression. This is because the uncertainty from the initial stage propagates through the cascaded model, influencing the overall prediction uncertainty $\text{cov}(\tilde{\theta}_{t,z})$ (, 99% interval).

Figure 4.14: Predictions using the cascaded GPR model for the z-axis, given a constant motion in z-axis. (a) The estimated sensor inaccuracy of the PTB z-axis $\hat{\eta}_{\theta_z}$, and (b) the estimated sensor inaccuracy in the CPA Hall encoder $\hat{\eta}_{\phi_z}$ is plotted



(a) The predicted sensor inaccuracy in the PTB z-stage $\hat{\eta}_{\theta_z} = \hat{\theta}_{t,z} - \theta_z$ (—) is shown. Through a diagonal motion, the predictions align precisely with the observed training data, which reduces the uncertainty, given by $\text{cov}(\hat{\theta}_{t,z})$ (, 99% interval), of the predictions and decreases the total uncertainty.



(b) The predicted sensor inaccuracy in the CPA z-axis, represented as $\hat{\eta}_{\phi_z} = \tilde{\theta}_{t,z} - \phi_z$ (—), is displayed. The uncertainty from the initial stage propagates through the cascaded model, influencing the overall prediction uncertainty $\text{cov}(\tilde{\theta}_{t,z})$ (, 99% interval).

Figure 4.15: Predictions using the cascaded GPR model for the z-axis, given a diagonal motion in both z- and x-axis. (a) The estimated sensor inaccuracy of the PTB z-axis $\hat{\eta}_{\theta_z}$, and (b) the estimated sensor inaccuracy in the CPA Hall encoder $\hat{\eta}_{\phi_z}$ is plotted.

4.6 Conclusion

This chapter presents a case study for the cascaded calibration of CPAs in optical satellite communication. The complete calibration procedure thoroughly explained. This includes an active optical alignment procedure between the CPA and PTB optical systems. The existence and behavior of the forward and inverse kinematic models are analyzed to detail. In addition, a GPR-based approach is taken to model the inverse kinematics.

The cascaded calibration procedure is utilized for the calibration of the CPA Hall encoder. The GPR-based cascaded calibration of a sensor is introduced in Chapter 2, which provides core details regarding its efficiency and background information. This chapter shows that this procedure naturally extends to a 2DOF calibration procedure. The GPR model obtained in the cascaded calibration procedure is suitable for DEB via the framework presented in Chapter 3.

Chapter 5

Conclusion

This thesis shows the successful implementation of GPR models in position sensor calibration, which is able to provide accurate cascaded calibration models, and enables budgeting of modeling uncertainties in error budgeting frameworks. Both contributions are formulated in separate frameworks that are applied to the calibration of a position sensor. A complete calibration procedure is presented for a CPA in optical satellite communication.

Chapter 2 demonstrates the effectiveness of a GPR framework for a cascaded calibration of position sensors. This cascaded calibration procedure is described as a multi-stage regression problem. Using GPR models the uncertainty that propagates through the multi-stage procedure is modeled and included in the optimization of the regression problems. Evidence is provided via Monte Carlo simulations that this method as compared to traditional methods provides a more accurate estimation of the regression model. This evidence supports Conjecture 1, which states Bayesian methods enable efficient and accurate modeling of multi-stage regression problems by utilizing the uncertainty information that is carried from one stage to all the subsequent stages.

Chapter 3 demonstrates that modeling uncertainties in Bayesian regression, in particular in GPR, can be represented by their PSD function. A framework for PSD estimation and analysis is introduced, with a potential extension to dynamic error budgeting, that enables error budgeting using PSD models of stochastic disturbances. In the context of the thesis, the PSD of modeling uncertainties in position sensor calibrations can naturally be included in dynamic error budgeting frameworks.

This framework first determines the PSD of a GP prior by utilizing Corollary 1. This PSD function provides the spectral characteristic of the modeling uncertainties such as the roll-off frequency, which is governed by the smoothness hyperparameter of the prior kernel. Second, the PSD function of the modeling uncertainty is determined by utilizing Algorithm 2. This is based on a rather pragmatic Monte Carlo approach to estimating the PSD function of conditioned distributions.

In Chapter 4 a case study is presented for the cascaded calibration of CPAs in optical satellite communication. This case study presents the use of both frameworks in a complex 2-DOF example. Attention is put on the optical alignment procedure for the calibration of these systems. To enable sensor calibration via optical alignment, the inverse optical kinematic model is estimated via GPR methods. The inverse kinematic model features a four-dimensional input, which greatly increases the complexity of the GPR models.

Chapter 6

Discussion and Recommendations

In this chapter, recommendations for future research and improvements are presented based on the findings and insights gained throughout this thesis. First, the potential GPR methods in cascaded sensor calibration is discussed and the importance of comparing GPR with other Bayesian methods is highlighted. Second, the limitations of the predictive covariance estimate in expressing uncertainty is mentioned, and a suggestion is made to explore deep Gaussian processes to better handle uncertainty propagation. In the third section, the significance of understanding the influence of datapoint placement on the spectral characteristics of the PSD function is emphasized, and research into optimization based on spectral information is suggested. Lastly, the challenges of increasing dimensionality in GPR models is discussed and recommendations are provided to decrease the computational effort.

Gaussian process regression for cascaded calibrations

This thesis demonstrates the potential of GPR models for cascaded sensor calibration and its benefits in error budgeting. The primary focus has been on showcasing the advantages of Bayesian methods in general, and GPR specifically. To further strengthen these findings, it is crucial to compare the results obtained in this thesis with other existing Bayesian methods and frameworks. Future research should explore alternative Bayesian regression techniques, evaluate their performance in comparison with GPR, and investigate different model structures, kernel functions, and inference types. Conducting experimental validation may provide deeper insights into sensor calibration using Bayesian approaches.

Evidence supporting Conjecture 1 is provided through Monte Carlo simulations, without establishing a mathematical proof. Mathematical proof of the improved estimation via cascaded GPR models would reinforce the findings of this thesis and validate the results for a broader range of applications.

Limited expressiveness of the covariance estimate

This thesis has shown various examples of cascaded calibrations with a total of three position sensors. Although this is carefully worked out, the expressiveness of the predictive covariance is limited to only three sensors. The reason for this is explained by looking at the covariance estimation of the predictive distribution, which represents the uncertainty of the predictions. Notice that this predictive covariance only represents the uncertainty of the function values, and does not contain information regarding the observation uncertainties. It is common practice to sum the predictive covariance with the observation covariance. However, a requirement for this practice is that the observation covariance be stationary, e.g. additive Gaussian white noise. In a cascaded calibration setting the observation uncertainty after the first stage is in general non-stationary, which is supported by Proposition 1.

Moreover, deep Gaussian processes (DGP) [40] may provide more reliable uncertainty estimates due to its inherent ability to handle uncertainty propagation across layers. The cascaded calibration procedure provided in this thesis is chaining separate GPR models. Although DGPs require greater computational effort they may bring more seamless modeling of the uncertainty that propagates in a cascaded calibration procedure.

Efficient exploration based on spectral information

The PSD estimation and analysis framework enables flexible estimation and analysis of PSDs for modeling uncertainties in sensor calibration. Chapter 3 has shown that selecting the placement of datapoints has a great influence on the power spectral characteristics of the PSD function. It is recommended to find a better understanding of the placement of data points and the influence it has on the spectral characteristics of the PSD function. This would enable smart exploration based on spectral information, rather than using uncertainty information in Bayesian Optimization [41]. This approach will allow for more efficient exploration strategies in sensor calibration and lead to more refined error budgets. There is already an extensive amount of research available to be able to find a closed-form expression of the PSD estimate [35].

The PSD function can nonetheless be used in DEB frameworks to enable more accurate error budgeting. These budgeting frameworks assume linear time-invariant (LTI) systems, which in many cases, may not be the case. Future research should address the complexities introduced by linear parameter varying (LPV) or linear time-varying (LTV) systems, such as the CPA SRM motor [42].

Curse of dimensional in Gaussian Process regression

The case study presented in this thesis highlights the challenges of increasing dimensionalities, also referred to as the curse of dimensionality. This phenomenon plays a significant role in the optimizations shown in the case study. Estimating the uncertainty for the inverse optical kinematic model is not required, and there is no strict modeling accuracy requirement. This provides a solid reason to opt for more pragmatic parametric modeling techniques, such as ordinary least squares (OLS), eliminating the need for regularization by ensuring low but sufficient model complexity.

A cascaded sensor calibration procedure relies on the accurate estimation of uncertainty and aims to achieve the highest possible level of modeling accuracy. Future work can decrease computational effort by utilizing more efficient GPR approaches, such as sparse GP methods [43].

To better understand the effect of an additional dimension, let's observe the computational effort of a GPR model. The computational effort is defined by the effort required for training the GPR model on N input locations and for predictions on M query input locations. The computational effort for training is dictated by inverting the matrix in (2.26), which requires $\mathcal{O}(N^3)$ operations. Once this inverse is determined, predictions can be made, which require $\mathcal{O}(N^2M)$ for the covariance estimate and $\mathcal{O}(NM)$ for the mean estimate. The number of data points N increases quadratically with the size of input dimension D, i.e. the required number of data points increases to N^D and thus the required amount of operations for training increases to $\mathcal{O}(N^{3D})$.

Bibliography

[1] Massimo Casasco, Gonzalo Saavedra, Sven Weikert, Jochen Eggert, Marc Hirth, Thomas Ott, and Haifeng Su. Pointing Error Budgeting for High Pointing Accuracy Mission using the Pointing Error Engineering Tool. August 2013. 1

[2] M. Gregory, F. Heine, H. Kämpfner, R. Meyer, R. Fields, and C. Lunde. TESAT laser communication terminal performance results on 5.6Gbit coherent inter satellite and satellite to ground links. 10565(October):37, 2017. 2

[3] Zoran Sodnik and Marc Sans. Extending EDRS to Laser Communication from Space to Ground. *International Conference on Space Optical Systems and Applications*, 12:9–14, 2012. 2

[4] Fu rui Zhang, Jun-feng Han, and Ping Ruan. Beam pointing analysis and a novel coarse pointing assembly design in space laser communication. *Optik*, 189:130–147, July 2019. 2

[5] Eric D. Miller, Michael DeSpenza, Ilya Gavrilyuk, Graham Nelson, Brent Erickson, Britney Edwards, Ethan Davis, and Tony Truscott. A prototype coarse pointing mechanism for laser communication. *https://doi.org/10.1117/12.2264086*, 10096:200–211, feb 2017. 2

[6] M. Dresscher, J. D. Human, G. Witvoet, N. Van Der Heiden, R. Den Breeje, S. Kuiper, E. C. Fritz, C. W. Korevaar, N. C.J. Van Der Valk, T. J. De Lange, R. Saathof, N. Doelman, W. E. Crowcombe, C. M. Duque, and H. De Man. Key challenges and results in the design of cubesat laser terminals, optical heads and coarse pointing assemblies. *2019 IEEE International Conference on Space Optical Systems and Applications, ICSOS 2019*, 2019. 2, 42, 54

[7] TNO-led Dutch consortium to develop HemiCAT laser communication terminal, howpublished = `https://www.tno.nl/en/newsroom/2023/03/tno-led-dutch-consortium-develop-hemicat/`, note = Accessed: 2023-04-01. 3

[8] Leslie Pendrill. Euramet: European association of national metrology institutes. *NCSLI Measure*, 4:40–44, 12 2009. 3

[9] Richard Koops, M Veghel, Gerard Kotte, and Charl Moolman. Calibration strategies for scanning probe metrology. *Measurement Science and Technology*, 18:390, 01 2007. 3

[10] Ibrahim A. Sultan and John G. Wager. Simplified theodolite calibration for robot metrology. *Advanced Robotics*, 16(7):653–671, 2002. 3

[11] Anas Alhashimi. *Statistical Sensor Calibration Algorithms*. PhD thesis, May 2018. 3, 8

[12] Moritz Berger, Christian Schott, and Oliver Paul. Bayesian Sensor Calibration. *IEEE Sensors Journal*, PP:1–1, October 2022. 3

[13] Moritz Berger, Christian Schott, and Oliver Paul. Bayesian Sensor Calibration of a CMOS-Integrated Hall Sensor Against Thermomechanical Cross-Sensitivities. PP:1–1, April 2023. 3

[14] Soumyabrata Talukder, Souvik Kundu, and Ratnesh Kumar. *Dynamic Calibration of Nonlinear Sensors with Time-Drifts and Delays by Bayesian Inference*. August 2022. 3

[15] Christian Robert, Jean-Michel Marin, Mohammed El Anbari, and Gilles Celeux. Regularization in Regression: Comparing Bayesian and Frequentist Methods in a Poorly Informative Situation. *Bayesian Analysis*, 7, June 2012. 4

[16] Gianluigi Pillonetto, Tianshi Chen, Alessandro Chiuso, Giuseppe Nicolao, and Lennart Ljung. Bayesian Interpretation of Regularization. pages 95–134. May 2022. 4

[17] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006. OCLC: ocm61285753. 4, 15, 20

[18] Max van Meer, Gert Witvoet, and Tom Oomen. Optimal Commutation for Switched Reluctance Motors using Gaussian Process Regression. *IFAC-PapersOnLine*, 55:302–307, November 2022. 4

[19] Max van Meer, Maurice Poot, Jim Portegies, and Tom Oomen. Gaussian Process based Feedforward Control for Nonlinear Systems with Flexible Tasks: With Application to a Printer with Friction. *IFAC-PapersOnLine*, 55:241–246, November 2022. 4

[20] Noud Mooren, Gert Witvoet, and Tom Oomen. Gaussian process repetitive control: Beyond periodic internal models through kernels. *Automatica*, 140:110273, June 2022. 4

[21] Maurice Poot, Jim Portegies, Noud Mooren, Max van Haren, Max van Meer, and Tom Oomen. Gaussian Processes for Advanced Motion Control *. pages 3355–3355, June 2022. 4

[22] Max van Haren, Maurice Poot, Jim Portegies, and Tom Oomen. Position-Dependent Snap Feedforward: A Gaussian Process Framework. pages 4778–4783, June 2022. 4

[23] Zongyu Geng, Feng Yang, Xi Chen, and Nianqiang Wu. Gaussian Process Based Modeling and Experimental Design for Sensor Calibration in Drifting Environments. *Sensors and Actuators B: Chemical*, 216, April 2015. 4

[24] Chen Jiang, Zhen hu, Yixuan Liu, Zissimos Mourelatos, David Gorsich, and Paramsothy Jayakumar. A sequential calibration and validation framework for model uncertainty quantification and reduction. *Computer Methods in Applied Mechanics and Engineering*, 368:113172, August 2020. 4

[25] Rudolf Saathof, Will Crowcombe, Stefan Kuiper, Nick van der Valk, Federico Pettazzi, Dorus de Lange, Peter Kerkhof, Martijn van Riel, Harry de Man, Niel Truyens, and Ivan Ferrario. Optical satellite communication space terminal technology at TNO. In *International Conference on Space Optics — ICSO 2018*, volume 11180, pages 213–222. SPIE, July 2019. 7

[26] Remi Dumas and Bernard Laurent. System test bed for demonstration of the optical space communications feasibility. In *Free-Space Laser Communication Technologies II*, volume 1218, pages 398–411. SPIE, July 1990. 7

[27] Morris R. Driels and Uday S. Pathre. Robot calibration using an automatic theodolite. *The International Journal of Advanced Manufacturing Technology*, 9(2):114–125, March 1994. 10

[28] Ibrahim A. Sultan and John G. Wager. Simplified theodolite calibration for robot metrology. *Advanced Robotics*, 16(7):653–671, January 2002. 10

[29] Bernhard Olkopf, Ralf Herbrich, Alex Smola, and Robert Williamson. A Generalized Representer Theorem. *Computational Learning Theory*, 42, June 2000. 13

[30] Y Yu, H Chen, D Schuurmans, and Csaba Szepesvári. Characterizing the Representer Theorem. June 2013. 13

[31] Vern I. Paulsen and Mrinal Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2016. 13

[32] T. Ott, T. Benoit, P. Braembussche, and W. Fichter. ESA Pointing Error Engineering Handbook. In *8th International ESA Conference on Guidance, Navigation & Control Systems*, Karlovy Vary, CZ, June 2011. 26

[33] Leon Jabben and Jan van Eijk. Dynamic error budgeting. *Mikroniek, Proffesional Journal on Precision Engineering*, 51(2):5–12, 2011. 26

[34] Wouter Monkhorst. Dynamic error budgeting, a design approach. Master's thesis, 2004. 26

[35] Felipe Tobar. Bayesian nonparametric spectral estimation. 9 2018. 31, 34, 64

[36] Gordana Jovanovic-Dolecek. *Random signals and processes primer with MATLAB*. Springer, New York, 2013. OCLC: ocn812404795. 32

[37] Paul van den Hof and Arjan den Dekker. Signal Analysis and Estimation. 2010. 32

[38] Leon Cohen. Generalization of the Wiener-Khinchin Theorem. *Signal Processing Letters, IEEE*, 5:292–294, December 1998. 32

[39] David Kristjanson Duvenaud. *Automatic Model Construction with Gaussian Processes*. Phd thesis, University of Cambridge, June 2014. 52

[40] Thang D Bui, José Miguel Hernández-Lobato, Daniel Hernández-Lobato, Yingzhen Li, and Richard E Turner. Deep Gaussian Processes for Regression using Approximate Expectation Propagation. 64

[41] Apoorv Agnihotri and Nipun Batra. Exploring bayesian optimization. *Distill*, 5, 05 2020. 64

[42] Lukas Kramer, Joost Peters, Robbert Voorhoeve, Gert Witvoet, and Stefan Kuiper. Novel motorization axis for a coarse pointing assembly in optical communication systems. *IFAC-PapersOnLine*, 53(2):8419–8425, 2020. 64, 80

[43] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A Unifying View of Sparse Approximate Gaussian Process Regression. *The Journal of Machine Learning Research*, 6:1939–1959, December 2005. 64

[44] Trevor Hastie, Robert Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer series in statistics. Springer, New York, NY, 2nd ed edition, 2009. 69

[45] Ha Quang Minh, Partha Niyogi, and Yuan Yao. Mercer's Theorem, Feature Maps, and Smoothing. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Gábor Lugosi, and Hans Ulrich Simon, editors, *Learning Theory*, volume 4005, pages 154–168. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. 71

Appendix A

Additional Material for Regression and Gaussian processes

A.1 The regression model

A regression model is defined by $y = f(x, \alpha) + e$, where y is the dependent variable, x represents independent variables, α are unknown model parameters, and e are error terms. The goal of regression is to estimate the underlying relationship between y and x, enabling predictions and creating more understanding regarding the process $f(x)$.

The function $f(x, \alpha)$ models the relationship between variables, and can be either parametric or non-parametric, which enables more flexible modeling. Error terms e can be additive white noise in the process, or any behaviour in the process that is not modelled. Estimation of α is achieved using optimization techniques like gradient descent, linear regression, or Bayesian methods.

A.2 Regression methods and regularization

In this section an overview is given on the general framework for regression and regularization. Consider a discrete dataset given by $\mathcal{D} = \{x_k, y_k\}_{k=1}^{N}$, with N the amount of datapoints. To find model $f(x) : \mathcal{X} \to \mathbb{R}$, which fits to the measured data y, there are many types of regression methods.

In section A.2.1 the basic properties of linear regression using least squares is given. In sections A.2.2 and A.2.3 regularized regression for linear and non-linear models are presented respectively. In section A.2.4 kernel ridge regression is introduced, which combines the ridge regression method with the kernel trick to more efficiently model nonlinear functions.

A.2.1 Least square linear regression

The most well known regression method is least squares linear regression [44]. For linear regression, the model can be described by

$$f(x) = x^T w, \tag{A.1}$$

where $w \in \mathbb{R}$ the model parameters (or weights). Assume that the index set $\mathcal{X}$, which describes the set for all inputs, can be generalized to $\mathbb{R}^D$, where D is the dimension of the input. To find the weights that fit the datapoints the best the cost for linear regression is given by

$$J[w] = \sum_{i=1}^{N} \left(y - x^T w\right)^2. \tag{A.2}$$

For which the optimal weights can be derived, given by

$$w^* = \left(X^T X\right)^{-1} X^T Y, \tag{A.3}$$

where $X = [x_1\ x_2\ \dots\ x_N]^T \in \mathbb{R}^{N\times D}$ and $Y = [y_1\ y_2\ \dots y_N]^T \in \mathbb{R}^N$. In most cases, when there are many correlated variables in a linear regression model, the model parameters will be poorly determined, i.e. we experience over-fitting or under-fitting. In order to prevent over- and under-fitting model parameters, regularization can be used.

A.2.2 Regularized regression for linear models

A prominent example for regularized least squares regression is called ridge regression, where we introduce an additional cost for the model parameters. The cost for ridge regression is given by

$$J[w] = \sum_{i=1}^{N} \left(y - x^T w\right)^2 + \lambda \left\|w\right\|^2. \tag{A.4}$$

Here the first term is a data-fit term, as in (A.2), also referred to as the loss function, which assesses the quality of the fit given the observed data y. The second term is the regularizer, which puts a cost on the parameter values. The parameter λ can be used to scale between the two terms. The solution for the optimal w^* can be found by setting the differential of equation A.4 to zero. This results in the weight given by

$$\begin{aligned} w^* &= \left(X^T X + \lambda I_N\right)^{-1} X^T Y, \\ &= X^T \left(X X^T + \lambda I_N\right)^{-1} Y, \end{aligned} \tag{A.5}$$

where $I_N \in \mathbb{R}^{N\times N}$ the identity matrix. A prediction for regularized regression regarding non-linear model structures, can be easily derived using the solution given by (A.5). For any testpoint x, we can predict the function value using the optimal weights,

$$\hat{y} = x^T w^* = x^T X^T \left(X X^T + \lambda I_N\right)^{-1} Y, \tag{A.6}$$

let $\alpha = \left(X X^T + \lambda I_N\right)^{-1} Y$, then

$$\begin{aligned} \hat{y} &= x^T X^T \alpha, \\ &= x^T \sum_{i=1}^{N} \alpha_i x_i = \sum_{i=1}^{N} \alpha_i \cdot x^T x_i. \end{aligned} \tag{A.7}$$

Regularized regression for any type of non-linear model are derived in Section A.2.3 from the results obtained here.

A.2.3 Regularized regression for non-linear models

The non-linear model structure for $f(x)$ is given by

$$f(x) = \phi(x)^T w, \tag{A.8}$$

where $\phi(x) \in \mathbb{R}^N$ is a non-linear basis function. The introduction of ϕ which maps a D-dimensional input vector x into an M-dimensional feature map, i.e. $\phi : \mathbb{R}^D \to \mathbb{R}^M$, enables more complex modeling features. The solution for the weights can be found by replacing x with $\phi(x)$ and X with $\Phi := \phi(X) = [\phi(x_1)\ \phi(x_2)\ \dots\ \phi(x_N)]^T \in \mathbb{R}^{N\times M}$ in (A.4) and (A.5). The optimal weight are given by,

$$w^* = \Phi^T (\Phi\Phi^T + \lambda I_N)^{-1} Y, \tag{A.9}$$

where

$$\Phi\Phi^T = \begin{bmatrix} \phi(x_1)^T\phi(x_1) & \phi(x_1)^T\phi(x_2) & \dots & \phi(x_1)^T\phi(x_n) \\ \phi(x_2)^T\phi(x_1) & \phi(x_2)^T\phi(x_2) & \dots & \phi(x_2)^T\phi(x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(x_n)^T\phi(x_1) & \phi(x_n)^T\phi(x_2) & \dots & \phi(x_n)^T\phi(x_n) \end{bmatrix}. \tag{A.10}$$

The term $\Phi\Phi^T$ can be replaced by a kernel matrix K, also referred to as Gramm matrix, where each element is determined by $K_{ij} = k(x_i, x_j)$. This matrix is symmetric positive semi-definite, i.e. $K \in S_+^N = \{K \in \mathbb{R}^{N\times N} : K = K^T, x^T Kx \geq 0 \forall x \in \mathbb{R}^N | x \neq 0\}$. This enables us to describe every dot product of an unknown, possibly infinitely many basis functions, in terms of kernels. This is also referred to as the kernel trick and will be explained in more detail in Section A.2.4.

A.2.4 Kernel ridge regression

In the previous sections, it has been demonstrated that optimal weights can be solved for when the model is described by a non-linear kernel function. However, consider a high-dimensional feature space, i.e., many measurement points N. It can also be the case that the correct choice for basis functions is not known.

Using Mercer's theorem [45], the inner product of two vectors in some feature space can be replaced without the need to visit it. In other words, there is no need to know $\phi(x)$, only that it exists. The kernel can be described by a function $k(x_i, x_j) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. This phenomenon is also known as the kernel trick, which is a very efficient way of transforming data into higher dimensions. The kernel can be defined as the in-product between two feature vectors, i.e.

$$k(x_i, x_j) = \phi(x_i)^T\phi(x_j). \tag{A.11}$$

For a test input x, the prediction of the function value is given by the model in (A.8) and the optimal parameters given by (A.9). Recall from Section A.2.3 that the kernel matrix equals $K = \Phi\Phi^T$. The prediction can thus be written as

$$\hat{y} = \phi(x)^T\Phi^T(K + \lambda I_N)^{-1}Y. \tag{A.12}$$

Now let $\alpha = (K + \lambda I_N)^{-1}Y$,

$$\begin{aligned} \hat{y} &= \phi(x)^T\Phi^T\alpha, \\ &= \sum_{i=1}^{N} \alpha_i \cdot \phi(x)^T\phi(x_i). \end{aligned} \tag{A.13}$$

By substituting (A.11) in (A.13), an estimate of the function value using only the kernel function can be obtained, $\hat{y} = \sum_{i=1}^{N} \alpha_i \cdot k(x, x_i)$.

The aim is to find a suitable kernel function and its parameters that fit the observed data. A prominent example is a squared exponential kernel, which is introduced in Section 2.4.4. This kernel has an infinite number of dimensions in its feature space, as it can be expanded by the Taylor series. In essence, it is possible to operate in the original feature space without computing in the higher dimensional space, which is potentially infinitely large.

A.3 Derivation of predictive distribution for Bayesian regression

Here the derivation of the predictive distribution in (2.21). The posterior distribution of the weighs is given by,

$$p(\beta, \Phi, Y) \sim \mathcal{N}\left(\beta^\star, \Phi\Sigma^{-1}\Phi^T + \Sigma_p^{-1}\right), \tag{A.14}$$

where $\beta^\star = (\Phi\Sigma^{-1}\Phi^T + \Sigma_p^{-1})^{-1}\Phi\Sigma^{-1}Y$. Define a new variable A, which is given by

$$A = \Phi\Sigma^{-1}\Phi^T + \Sigma_p^{-1}, \tag{A.15}$$

then the predictive distribution can be obtained by averaging the output of all possible models with respect to the Gaussian posterior,

$$\begin{aligned} p(\hat{f}(x) \mid \Phi, Y, \phi) &= \int p(\hat{f}(x) \mid \phi, \beta) p(\beta \mid \Phi, Y) d\beta, \\ &= \mathcal{N}\left(\phi^T\beta^\star, \phi^T\left(\Phi\Sigma^{-1}\Phi^T + \Sigma_p^{-1}\right)\phi\right), \end{aligned} \tag{A.16}$$

where $\phi(x) := \phi$, and $\phi(X) = \Phi$. Using the new variable A in (A.15), the predictive distribution can be rewritten to,

$$p(\hat{f}(x) \mid \Phi, Y, \phi)\mathcal{N}\left(\phi^T A^{-1}\Phi\Sigma^{-1}Y, \phi^T A^{-1}\phi\right). \tag{A.17}$$

Now define a new variable, $K = \Phi^T\Sigma_p\Phi$. The relation between A and K can be described as follow,

$$\begin{aligned} A\Sigma_p\Phi &= \left(\Phi\Sigma^{-1}\Phi^T + \Sigma_p^{-1}\right)\Sigma_p\Phi, \\ &= \Phi\Sigma^{-1}\Phi^T\Sigma_p\Phi + \Phi, \\ &= \Phi\Sigma^{-1}\left(\Phi^T\Sigma_p\Phi + \Sigma\right), \\ &= \Phi\Sigma^{-1}\left(K + \Sigma\right). \end{aligned} \tag{A.18}$$

By substituting $A\Sigma_p\Phi$ with $\Phi\Sigma^{-1}(K + \Sigma)$ in (A.17), the predictive distribution in (2.21) is obtained.

A.4 Sampling from Gaussian distributions

Given a multivariate Gaussian distribution with mean vector $\mu \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$, the goal is to draw random samples from this Gaussian distribution. The probability density function of this distribution is given by (B.3).

Samples can be drawn from this distribution using the Cholesky decomposition. The Cholesky decomposition of the covariance matrix Σ obtains a lower triangular matrix $L \in \mathbb{R}^{n\times n}$, given by

$$\Sigma = LL^T. \tag{A.19}$$

Generate a vector $z = [z_1, z_2, ..., z_n]^T \in \mathbb{R}^n$ of independent standard normal random variables. Each component z_i is drawn from a univariate standard normal distribution given by $z_i \sim \mathcal{N}(0, 1)$ Transform z using the Cholesky factor L and the mean vector μ, which gives

$$x = \mu + Lz. \tag{A.20}$$

The resulting vector $x \in \mathbb{R}^n$ is a random sample drawn from the corresponding multivariate Gaussian distribution.

Appendix B

Gaussian and Fourier Transform Properties

B.1 Stochastic signal properties

First, the correlation -and covariance functions of stochastic processes will be described. Afterward, the basic properties of the (multivariate) Gaussian distribution.

B.1.1 Correlation and Covariance

Consider a WSS stochastic process $f(x)$, the mean and covariance function for this process have been given in (3.11). The correlation and covariance function partially describe the time domain behavior of the underlying stochastic process. They will be used later to derive the spectral (frequency domain) properties of the process. The correlation function R of $f(x)$ is given by,

$$\mathbb{E}[f(x_A)f(x_B)] = R(\tau) \ \forall \tau, x_A, x_B \in \mathbb{R}. \tag{B.1}$$

For processes with zero mean, $\mu = 0$, the covariance function in (3.11) equals the correlation function.

B.1.2 Gaussian distribution

A normal distribution, also referred to as the Gaussian distribution, is heavily used in statistics to describe independent random variables. The function, which describes the continuous probability distribution of a variable $f(x)$, is given by its mean μ, and its standard deviation σ, given by

$$p(f; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(f-\mu)^2/(2\sigma^2)}. \tag{B.2}$$

The shape of this distribution is a bell curve, where the mean gives the center of this bell curve and the standard deviation the amount of stretch of this curve. We can say that a variable, f, has a normal distribution by writing $f \sim \mathcal{N}(\mu, \sigma)$, where $\sigma = k(0) = \mathbb{E}\left[(x-\mu)^2\right]$

B.1.3 Multivariate Gaussian distribution

The multivariate normal distribution or the multivariate Gaussian distribution is a generalization of the single-dimension normal distribution to higher dimensions. Suppose we are looking at a random vector-valued variable given by $\mathbf{f} \in \mathbb{R}^n$, with n the size of the vector. The random vector $\mathbf{f}$ has a multivariate Gaussian distribution, with a mean vector $\mu \in \mathbb{R}^n$ and a symmetric positive

definite covariance matrix $\Sigma \in \mathbb{S}^n_{++} \subseteq \mathbb{R}^{n\times n}$. The multivariate Gaussian distribution is given by

$$p(\mathbf{f};\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{f}-\mu)^T\Sigma^{-1}(\mathbf{f}-\mu)\right), \tag{B.3}$$

where,

$$\begin{aligned}\mu &= \mathbb{E}\left[\mathbf{f}\right],\\ \Sigma &= \mathbb{E}\left[(\mathbf{f}-\mu)(\mathbf{f}-\mu)^T\right],\\ \mathbb{S}^n_{++} &= \{\Sigma \in \mathbb{R}^{n\times n} : \Sigma = \Sigma^T, x^T\Sigma x > 0\ \forall x \in \mathbb{R}^n |\ x \neq 0\}.\end{aligned} \tag{B.4}$$

B.2 Fourier transform

The Fourier transform decomposes a signal into its frequency components. The Fourier transform can show that any signal can be re-written as a sum of sinusoides,

$$F(\omega) := \mathcal{F}\{f(x)\} = \int_{-\infty}^{\infty} f(x)e^{-i\omega x}dx, \tag{B.5}$$

where the Fourier transform is function of ω: $\mathcal{F}\{f(x)\}(\omega)$, with ω the frequency in *rad/second*. The inverse Fourier transform is given by

$$f(x) := \mathcal{F}^{-1}\{\mathcal{F}\{f(x)\}\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega x}d\omega. \tag{B.6}$$

Note that the $\frac{1}{2\pi}$ term is added, so that the equation $f(x) = \mathcal{F}^{-1}\{\mathcal{F}\{f(x)\}\}$ holds.

B.3 Discrete Fourier Transform

The discrete-time variant of the Fourier transform is called the Discrete-time Fourier transform (DTFT). The DTFT for a finite amount of discrete samples is given by equation:

$$F_d(\omega_0) := \sum_{k=0}^{N-1} f_k e^{-i\omega_0 k}. \tag{B.7}$$

where ω_0 is the normalized frequency in rad/sample, f_k is the discrete time signal, with $k \in 0,..,N\text{-}1$. The periodicity of the normalized version of the DTFT is is 2π. The DTFT transforms a finite number of sampled points into a function $F_d(\omega)$ which has a continuous input ω_0. When including the sampling interval, the normalized frequency can be transformed using

$$w_0 = 2\pi f T \tag{B.8}$$

where f is the frequency in Hz and T is the sampling interval in seconds. Using this relation, the DTFT can be written as a function of frequency, given by

$$F_d(f) := \sum_{k=0}^{N-1} f_k e^{-i2\pi f k T}, \tag{B.9}$$

In numerical analyses discrete samples are taken from the DTFT, which results in the discrete Fourier transform (DFT). This prevents taking an infinite amount of computations to obtain the Fourier transform of a finite amount of spatial or temporal domain samples. The DTFT is sampled at frequencies $\omega_k = \frac{2\pi}{N}l$ which gives the DFT,

$$F_{d,l} := \sum_{k=0}^{N-1} f_k e^{-i\frac{2\pi l}{N}k}, \tag{B.10}$$

where $F_{d,l}$ and f_k are points in frequency and time respectively. The finite points in frequency domain $F_{d,l}$ are now a weighted combination of the finite points in time.

B.4 Properties of the Fourier transform

In this section, several properties of the Fourier transform that are relevant to this research are provided. These include differentiation, multiplication by the input, input scaling, and input shift properties. There are many more properties of the Fourier transform that are not discussed here.

B.4.1 Differentiation

The inverse Fourier transform, recovers a function from its Fourier transform. The inverse Fourier transform of a function $\mathcal{F}\{f(x)\}$ is given by

$$f(x) = \mathcal{F}^{-1}\{\mathcal{F}\{f(x)\}\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathcal{F}(\omega)e^{i\omega x}d\omega. \tag{B.11}$$

We can now observe the effects of taking the derivative of function $f(x)$, which is given by $\frac{d}{dx}f(x)$. The inverse Fourier transform of $\frac{d}{dx}f(x)$ can be written as

$$\begin{aligned}\frac{d}{dx}f(x) &= \frac{d}{dx}\left(\frac{1}{2\pi}\int_{-\infty}^{\infty}\mathcal{F}(\omega)e^{i\omega x}d\omega\right) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\mathcal{F}(\omega)\frac{d}{dx}\left(e^{i\omega x}\right)d\omega, \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} i\omega\mathcal{F}(\omega)e^{i\omega x}d\omega = \mathcal{F}^{-1}\{i\omega\mathcal{F}\{f(x)\}\}.\end{aligned} \tag{B.12}$$

This property shows that taking the Fourier transform of a functions derivative $\frac{d}{dx}f(x)$, is the same as taking the Fourier transform of the regular function $f(x)$ and multiplying the Fourier transform by a $i\omega$ term. Repeated differentiation will result in an increase of the power of the $i\omega$ term, and can be written as

$$\frac{d^n}{dx^n}f(x) = \mathcal{F}^{-1}\{(i\omega)^n\{f(x)\}\}. \tag{B.13}$$

B.4.2 Multiplication by the input

Suppose the input to our function $f(x)$ denotes time or position. The Fourier transform of a function given by x times the function itself $xf(x)$ is equal to the derivative of the Fourier transform,

$$\begin{aligned}\mathcal{F}'\{f(x)\} &= \frac{d}{d\omega}\left(\int_{-\infty}^{\infty} f(x)e^{-i\omega x}dx\right) = \int_{-\infty}^{\infty}\frac{d}{d\omega}\left(f(x)e^{-i\omega x}\right)dx = \int_{-\infty}^{\infty} -ixf(x)e^{-i\omega x}dx \\ &= \mathcal{F}\{-ixf(x)\}.\end{aligned} \tag{B.14}$$

Note that using the Leibniz integral rule, the derivative with respect to ω can be placed inside the integral. by multiplying both sides with the imaginary number i, the following property holds

$$\begin{aligned}i\mathcal{F}'\{f(x)\} &= \int_{-\infty}^{\infty} xf(x)e^{-i\omega x}dx, \\ &= \mathcal{F}\{xf(x)\}.\end{aligned} \tag{B.15}$$

B.4.3 Input-scaling

Scaling the input of the function $f(x)$ with a factor n will result in a compression or expansion of the Fourier transform. This can be logically explained by assuming the function $f(x)$ represents a periodic signal with x representing time. If time, denoted by x, is scaled by a factor n, which results in a function $f(nx)$, the time for the periodic signal to complete an entire period will also be scaled by n. We can essentially make time go faster (when $n > 1$) or slower (when $n < 1$). The Fourier transform of a function where the input is scaled $f(nx)$, is given by

$$\begin{aligned}
\mathcal{F}\{f(nx)\} &= \int_{-\infty}^{\infty} f(nx)e^{-i\omega x}dx, \\
\text{do a substitution of variables, let } t = nx, \text{ which results in, } x &= \frac{t}{n},\ dx = \frac{dt}{n}, \\
\mathcal{F}\{f(nx)\} &= \int_{-\infty}^{\infty} f(t)e^{-i\omega\frac{t}{n}}dt/n = \frac{1}{n}\int_{-\infty}^{\infty} f(t)e^{-i\frac{\omega}{n}t}dt, \\
&= \frac{1}{n}\mathcal{F}\{f(x)\}\left(\frac{\omega}{n}\right).
\end{aligned} \tag{B.16}$$

B.4.4 Input-shift

As with the input-scaling, suppose the function $f(x)$ is a periodic signal where x represent time. By delaying a function in time with a value given by m, the function becomes $f(x-m)$. The Fourier transform of a delayed function corresponds to a multiplication with a complex exponential given by $e^{-i\omega m}$. This is proven by taking the Fourier transform of $f(x-m)$,

$$\begin{aligned}
\mathcal{F}\{f(x-m)\} &= \int_{-\infty}^{\infty} f(x-m)e^{-i\omega x}dt, \\
\text{do a substitution of variables, let } t = x-m, \text{ which results in, } x &= t+m,\ dx = dt, \\
\mathcal{F}\{f(x-m)\} &= \int_{-\infty}^{\infty} f(t)e^{-i\omega(t+m)}dt = e^{-i\omega m}\int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt, \\
&= e^{-i\omega m}\mathcal{F}\{f(x)\}.
\end{aligned} \tag{B.17}$$

An interesting observation is the fact that the term $e^{-i\omega m}$ is equal to the frequency response function of a time-delay $H(i\omega) = e^{-i\omega T_d}$, which shifts the phase by $-\omega T_d$ radians, where T_d is the time-delay in seconds.

B.5 Gaussian integral

A Gaussian integral, also referred to as the Euler-Possion integral, is the integral of the function, $f(x) = e^{-x^2}$, over the entire real line. Note that the distribution of a normal (or Gaussian) distributed variable is of the same function type, hence the importance of this integral. The integral is given by,

$$\int_{-\infty}^{\infty} e^{-x^2}dx = \sqrt{\pi}. \tag{B.18}$$

An intuitive approach to understand why the integral contains π, can be shown by performing two steps. The first step is to take the square of the gaussian integral, which becomes a double integration in Cartesian coordinates. The second step is to transform the double integral to polar coordinates,

$$\left(\int_{-\infty}^{\infty} e^{-x^2}dx\right)^2 = \iint_{\mathbb{R}^2} e^{-(x^2+y^2)}dxdy = \int_0^{2\pi}\int_0^{\infty} re^{-r^2}\ drd\theta = 2\pi\int_0^{\infty} re^{-r^2}dr. \tag{B.19}$$

By a substitution of variables, $u = -r^2$ and $du = -2rdr$, the integral is solved,

$$2\pi\int_0^{-\infty} -\frac{1}{2}e^s ds = -\pi\int_0^{-\infty} e^s ds = -\pi(e^{-\infty} - e^0) = \pi. \tag{B.20}$$

The integration of the Gaussian integral squared results in π, hence the Gaussian integral should result in its root $\sqrt{\pi}$.

B.6 Power spectral density function for squared exponential kernels

The properties of the Fourier transform and the Gaussian distribution provided in this Appendix are used to determine the PSD function of the squared exponential kernel.

First the Fourier transform of a Gaussian function is obtained. The Gaussian function is given by

$$f(x) = e^{-x^2}, \tag{B.21}$$

with the Fourier transform is given by

$$\mathcal{F}\{f(x)\} \int_{-\infty}^{\infty} f(x)e^{-ikx}dx. \tag{B.22}$$

The derivative of a Gaussian function is equal to

$$\begin{aligned} \frac{df(x)}{dx} &= -2xe^{-x^2}, \\ &= -2xf(x), \end{aligned} \tag{B.23}$$

which is also Gaussian. Now using the properties for the Fourier transform in (B.11) and (B.15) with the Gaussian function in (B.21) and its derivative (B.23) are used to derive

$$i\omega\mathcal{F}\{f(x)\} = -2i\frac{d}{d\omega}\mathcal{F}\{f(x)\}. \tag{B.24}$$

A unique solution for this ordinary differential equation is given by

$$\mathcal{F}\{f(x)\} = c \cdot e^{w^2/4}, \tag{B.25}$$

where c is determine by plugging in $\omega = 0$, which results in the Gaussian integral in Section B.5. The final solution for the Fourier transform for a Gaussian function is given by

$$\mathcal{F}\{f(x)\} = \sqrt{\pi}e^{-\omega^2/4}, \tag{B.26}$$

which is also Gaussian. Now the input-scaling and input shift properties can be used to determine the Fourier transform of the squared exponential function, which has a shift in input with μ and the input scaled by $(2l^2)^{1/2}$.

Appendix C

Coarse Pointing Assmbly

This chapter includes information regarding a single DOF experimental CPA. The working principle of this experimental CPA is almost identical to that of the 2DOF CPA which is presented in this thesis. The results obtained in this chapter are taken from the preparation phase report.

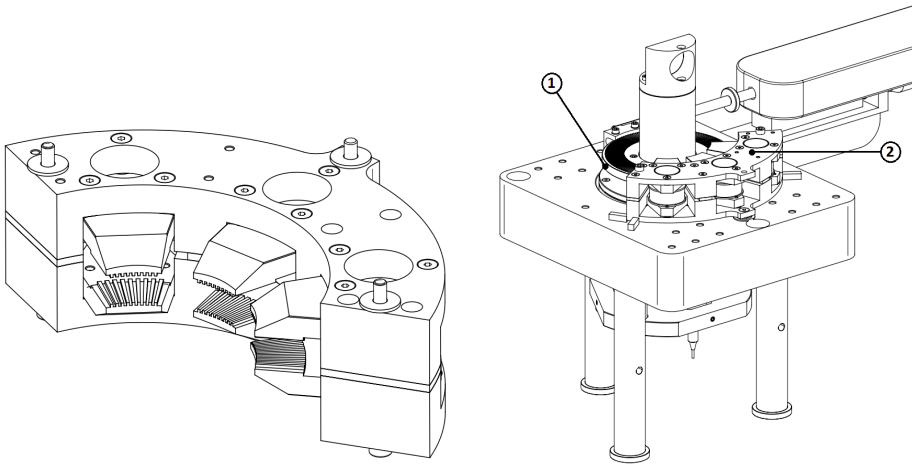
Section C.1 explain the principle behind the switch reluctance motor (SRM). Afterward in Section C.2 the Hall encoder principle is explained. Finally in Section C.3 experimental results are shown that explain the repetition of position-dependent inaccuracies in the Hall encoder.

C.1 Switch reluctance motor

Compared to DC motors, SRM motors deliver power to windings in the stator instead of the rotor. The motor principle consist of three phases which have been separated by 120°. By distributing the torque over each motor phase, continuous torque can be achieved. A schematic of the stator be seen in Figure C.1a. The stator and rotor of the single-axis CPA, which are highlighted in Figure C.1b, have been molded from a high-permeable magnetic material. Each stator contains a total of 9 individual teeth, which act as individual poles that are excited simultaneously. The CPA rotor has 131 teeth in total, thus a single tooth pitch equals a rotation of $\frac{2\pi}{131}$ rad. The motor torque of a single phase is a non-linear function of the angular position and quadratic current, given by,

$$T_c = i_c^2 \frac{1}{2} \frac{\partial L}{\partial \psi} = i_c^2 G(\psi), \tag{C.1}$$

where $c \in \{1, 2, 3\}$ is the coil number, i_c is the coil current and L is the inductance of the motor phase, determined by the tooth geometry, coil size, magnetic material. We assume there is no magnetic saturation present. A well thought commutation strategy is required for smooth transitions between the generated torque through each motor phase.



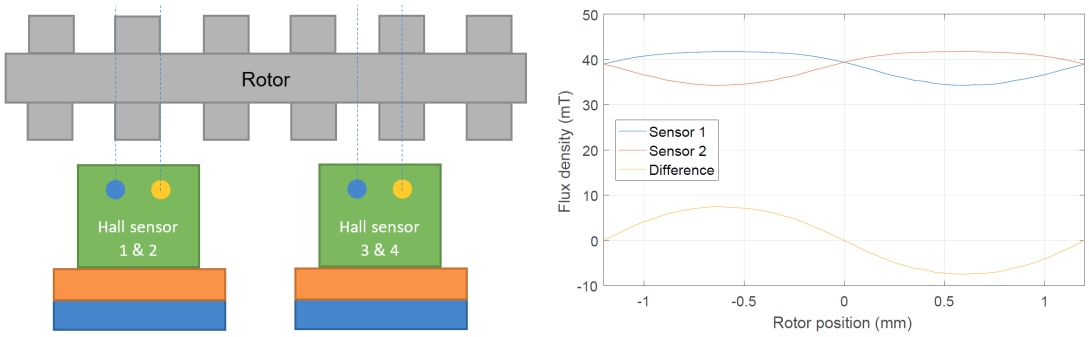
(a) (b)

Figure C.1: a) Motor assembly with three separate stators (b) Assembly of the single-axis CPA breadboard, where (1) is the rotor with 131 individual teeth and (2) the motor assembly.

C.2 Hall encoder principle

The actuation rotor of the CPA has several teeth that are also used to determine its angular position. The CPA contains magnetic Hall sensors that form a Hall encoder. Each sensor contains a pair of Hall plates which measure the flux density. The sensors are able to readout the differential flux density using two Hall plates, resulting in signals d_i with $i \in \{1, 2\}$. A schematic drawing of the working principle can be seen in Figure C.2a. The measured flux density of a single Hall sensor, with Hall plates numbered 1 and 2, has been plotted in Figure C.2b. The difference of both flux densities is an approximate sine function, as shown in Figure C.2b. A second Hall sensor, with Hall plates numbered 3 and 4, results in an approximate cosine signal. The Hall sensor measurements are communicated through an SPI interface.

Initial measurements have been performed on the Hall encoder of the single-axis CPA. The single-axis CPA mirror has been given a constant velocity trajectory of one whole rotation. The signals d_1 and d_2 have been plotted in Figure C.3. This data can also be presented in a Lissajous figure, where the cosine signal (d_2) has been plotted on the x-axis and the sine signal (d_1) has been plotted on the y-axis. The Lissajous figure shows many rotations, one for each tooth pitch. The angle between the x-axis and the vector drawn, is the relative motor phase $\psi \in \mathcal{S}$, with



(a) (b)

Figure C.2: (a) Working principle magnetic Hall sensor (b) Fem simulation Hall encoder [42]

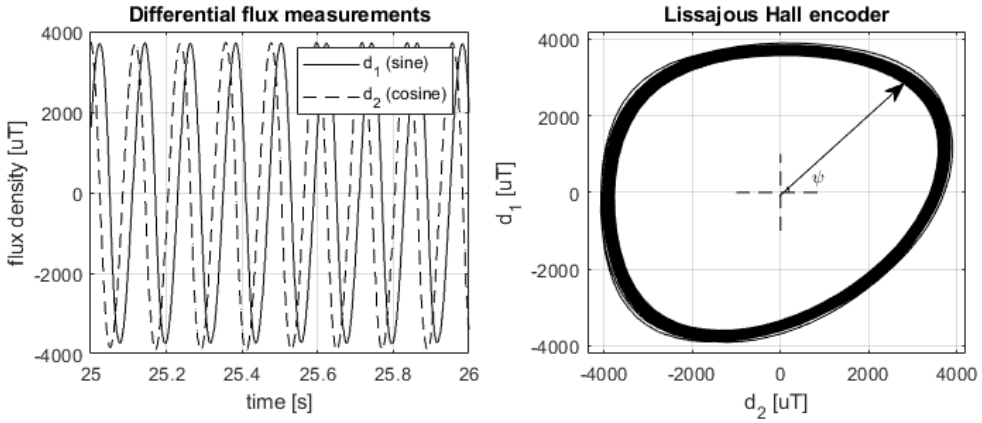


Figure C.3: Hall sensor measurements, showing the differential flux density measurements.

$\mathcal{S} = \{\psi | -\pi \leq \psi \leq \pi\}$. The relative motor phase is determined using,

$$\psi = \text{atan2}(d_1, d_2). \tag{C.2}$$

To obtain the absolute rotor angle, the relative motor phase ψ has to be unwrapped. This is accomplished using the equation given by,

$$\phi = \frac{\texttt{unwrap}(\psi)}{N}, \tag{C.3}$$

where $\phi \in \mathbb{R}$ is the rotor angle and N is the amount of teeth on the rotor. The function unwrap is available in Matlab, which unwraps phase jumps greater than π.

C.3 Repetition of errors

The Lissajous figure of the CPA Hall encoder has already been plotted in Figure C.3. Each tooth period in this figure is equal to a single tooth period on the CPA rotor. The thickness of the band in the Lissajous figure indicates differences between the teeth of the individual rotor. The spread between each tooth period has been visualized in Figure C.4. To plot this figure, data has been obtained from the Hall encoder $\psi(k)$ and the Renishaw encoder which measures the absolute rotor angle $\phi_{\text{ren}}(k)$. The absolute rotor angle is then converted to the relative motor phase using (C.3), in order to obtain $\psi_{\text{ren}}(k)$. Data has been obtained for an entire rotation of the CPA rotor, at discrete time steps k. For simplicity purposes the notation is shortened to ψ and ψ_{ren}.

The x-axis shows the Hall encoder motor phase ψ_{ren}, while the y-axis shows the difference between the Renishaw encoder and the Hall encoder, i.e. the measured position-dependent inaccuracies of the Hall encoder η_ψ. It shows that the angle obtained on each tooth period is varying. However, with this information we can assume a single mapping function taken for each tooth period to be sufficient.

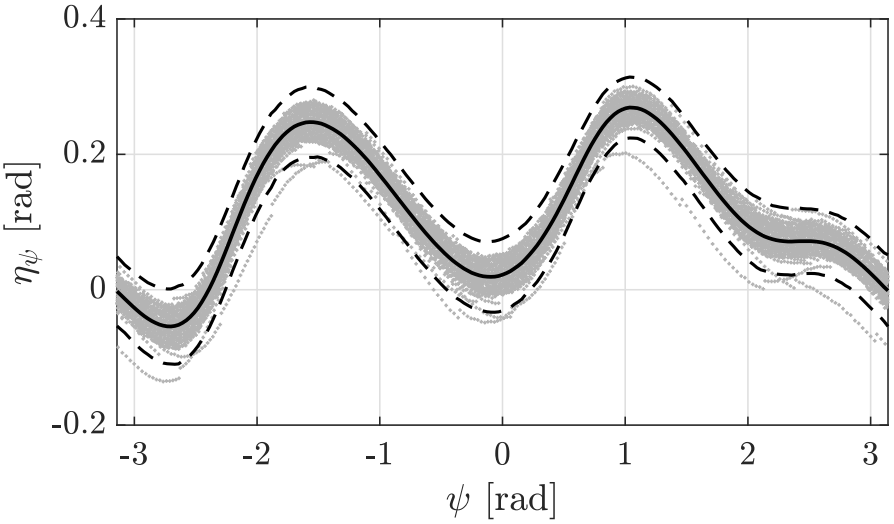


Figure C.4: The x-axis shows the relative motor phase of the Hall encoder ψ, while the y-axis shows the difference between ψ_{ren} and ψ, given by η_ψ. The spread in data indicates differences between each tooth period. The mean (—) is visualized and the confidence interval (- -, 99%)

Appendix D

Optical Forward Kinematics

This chapter provides derivation of the forward kinematic model $G_{\text{opt}}(\phi)$. First, in Section D.1 the kinematic chains of the CPA and PTB is given. Then in the plane-line intersection equations are provided D.2, where-after in Section D.3 the procedure and derivation of the forward kinematic model are provided.

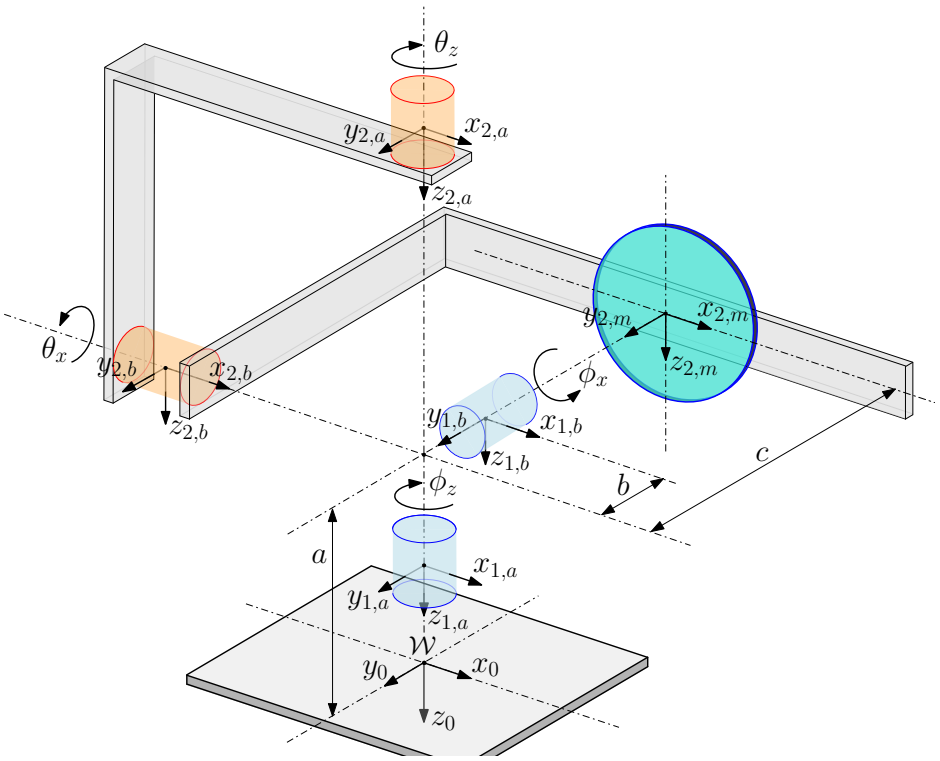


Figure D.1: Kinematic diagram of the CPA ( ) and PTB ( ) degrees of motion. The CPA has azimuth rotation ϕ_z and elevation rotation ϕ_x. The PTB has z-stage motion θ_z and x-stage motion θ_x.

D.1 System kinematics

The kinematics of the optical systems can be seen in Figure D.1. The CPA which has two rotational degrees of motion is depicted in blue, while the larger PTB system has two rotational stages, depicted in red. For every rotation degree of freedom a set of basis vectors $\{x_{(\cdot)}, y_{(\cdot)}, z_{(\cdot)}\}$

is defined. These basis vectors have their own position and orientation with respect to the world frame $\mathcal{W} := \{x_0, y_0, z_0\}$. Transformation matrices are defined that transform local coordinates to world coordinates.

The local coordinates of the CPA elevation mirror are given by $\{x_{1,b}, y_{1,b}, z_{1,b}\}$. The position and orientation of the elevation mirror with respect to the world frame are described by

$$\begin{aligned} T^0_{1,b} &= T_{\phi_z} T_{\phi_x}, \\ &= \begin{bmatrix} R^0_{1,b} & m^0_{1,b} \\ 0_3 & 1 \end{bmatrix}, \end{aligned} \tag{D.1}$$

where $m^0_{1,b}$ and $R^0_{1,b}$ are the position and of the orientation of the elevation mirror with respect to the world frame $\mathcal{W}$ respectively. The transformation matrix T_{ϕ_z} represents the rotation around the azimuth axis, and T_{ϕ_x} represents the rotation around the elevation axis. These transformation matrices in the kinematic chain are given by

$$T_{\phi_z} = \begin{bmatrix} C(\phi_z) & -S(\phi_z) & 0 & 0 \\ S(\phi_z) & C(\phi_z) & 0 & 0 \\ 0 & 0 & 1 & -a \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{\phi_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\phi_x) & -S(\phi_x) & -b \\ 0 & S(\phi_x) & C(\phi_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{D.2}$$

where $C(\phi) =: \cos(\phi)$, $S(\phi) =: \sin(\phi)$, a is the distance of the azimuth mirror w.r.t to the world frame, b is the distance between the elevation mirror and the azimuth mirror.

The center of the PTB mirror is given by the local coordinate $\{x_{2,m}, y_{2,m}, z_{2,m}\}$. The position and orientation of the PTB mirror with respect to the world frame are given by the transformation

$$\begin{aligned} T^0_{2,m} &= T_{\theta_z} T_{\theta_x} T_m, \\ &= \begin{bmatrix} R^0_{2,m} & m^0_{2,m} \\ 0_3 & 1 \end{bmatrix}, \end{aligned} \tag{D.3}$$

where $m^0_{2,m}$ and $R^0_{2,m}$ are the position and of the orientation of the PTB mirror w.r.t. the world frame $\mathcal{W}$ respectively. The transformation matrix T_{θ_z} represents the rotation around the z-stage, T_{θ_x} represents the rotation around the x-stage and T_m is the position offset of the PTB mirror with respect to the rotation stages. The individual transformation matrices in the kinematic chain are given by

$$T_{\theta_z} = \begin{bmatrix} C(\theta_z) & -S(\theta_z) & 0 & 0 \\ S(\theta_z) & C(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{\theta_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\theta_x) & -S(\theta_x) & 0 \\ 0 & S(\theta_x) & C(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_m = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{D.4}$$

where c is the location of the PTB mirror along the direction orthogonal to both the x-axis and z-axis, see Figure D.1.

D.2 Plane-line Intersections

The algebraic steps and calculation of the intersection point between a line and a plane is presented here. A plane in vector notation is given by,

$$(p - p_0)n = 0, \tag{D.5}$$

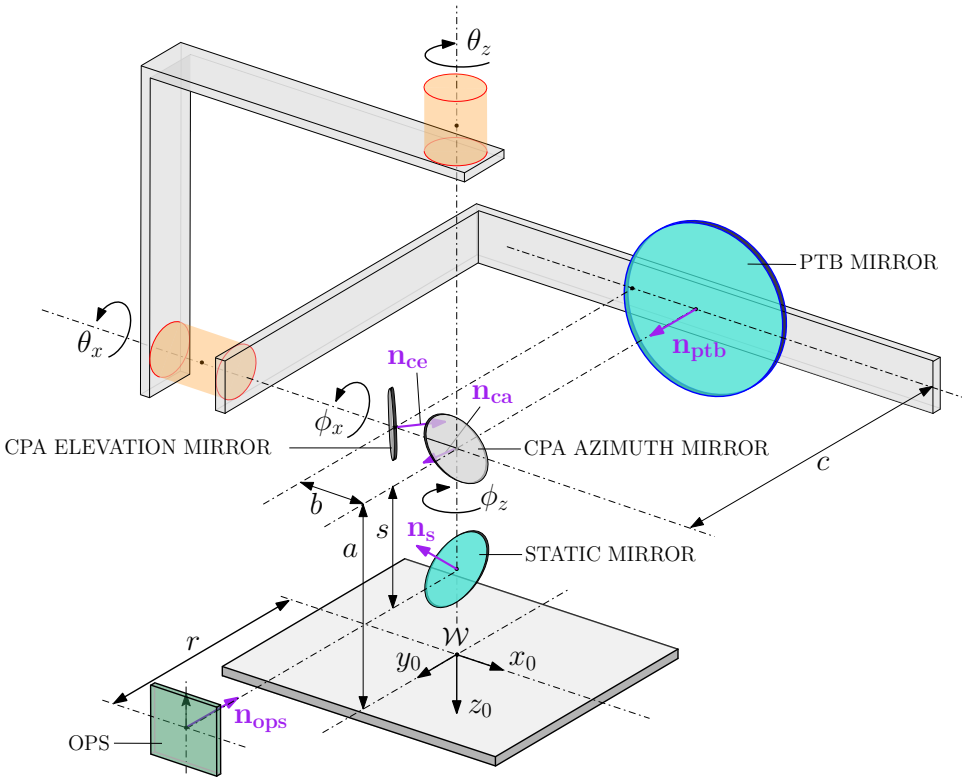


Figure D.2: The mirrors of the optical systems are visualized for their respective zero orientation.

where n is the normal vector to that plane, p_0 is any point vector on that plane, and p is any arbitrary point vector on that plane. A line in vector notation is given by

$$p(d) = l_0 + dl, \tag{D.6}$$

where l_0 is a point on the line expressed as a position vector, $d \in \mathbb{R}$ is the distance on the line and l is the direction of the line.

Now (D.6) is substituted in (D.5) to solve for d, which results in

$$d = \frac{(p_0 - l_0) \cdot n}{l \cdot n}. \tag{D.7}$$

The distance d at which an intersection takes place is a function of the plane normal n, any point vector on the plane p_0, and a line vector that starts at l_0 and has direction l. Use $p(d)$ in (D.6) to determine the line vector at which the intersection takes place.

D.3 Optical kinematics

The optical kinematics consists of a series of plane-line intersections, which determine the point and angle of incidence[1] as a function of the mirror normal. The mirror normal vectors $\mathbf{n}_{(\cdot)}$ have been depicted in Figure D.2, where the azimuth axis is positioned at ϕ_z=−90 degrees. These mirror normals are defined in their respective local frame, here each and every normal vector is provided with respect to the world frame:

[1] For a flat mirror the point/angle of incidence and point/angle of reflection is equal.

1. The surface normal for the PTB is given by $\mathbf{n_{ptb}} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$. Subsequently, the orientation of the surface normal can be described with respect to the world frame given by

$$\mathbf{n_{ptb,0}} = R^0_{2,m}(\theta_x, \theta_z)\mathbf{n_{ptb}}, \tag{D.8}$$

which is dependent fully on the orientation of PTB.

2. The surface normal for the C3PA elevation mirror is given by $\mathbf{n_{ce}} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$. Subsequently, the orientation of the surface normal can be described with respect to the world frame given by

$$\mathbf{n_{ce,0}} = R^0_{1,b}(\phi_x, \phi_z)\mathbf{n_{ce}}, \tag{D.9}$$

which is dependent on the orientation of the azimuth ϕ_z and the elevation angle ϕ_x of the C3PA.

3. The surface normal for the C3PA azimuth mirror is given by $\mathbf{n_{ca}} = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$. Subsequently, the orientation of the surface normal can be described with respect to the world frame given by

$$\mathbf{n_{ca,0}} = R^0_{1,a}(\phi_z)\mathbf{n_{ca}}, \tag{D.10}$$

which is dependent on the orientation of the azimuth angle ϕ_z only.

4. The surface normal of the static mirror is given by $\mathbf{n_s} = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}$. Subsequently, the orientation of the surface normal can be described with respect to the world frame given by

$$\mathbf{n_{s,0}} = \mathbf{n_s}, \tag{D.11}$$

which is independent of the orientation of the C3PA.

5. The surface normal of the OPS is fixed and is given by

$$\mathbf{n_{ops,0}} = \mathbf{n_{ops}} = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix} \tag{D.12}$$

In order to trace the laser ray that is responsible for aligning the systems, plane-line intersect equations are used. The ray is traced from the laser source (Tx), back to the PSD (Rx). Using this procedure the OPS the exact intersection between the OPS sensor plane and the Rx-beam is calculated.