



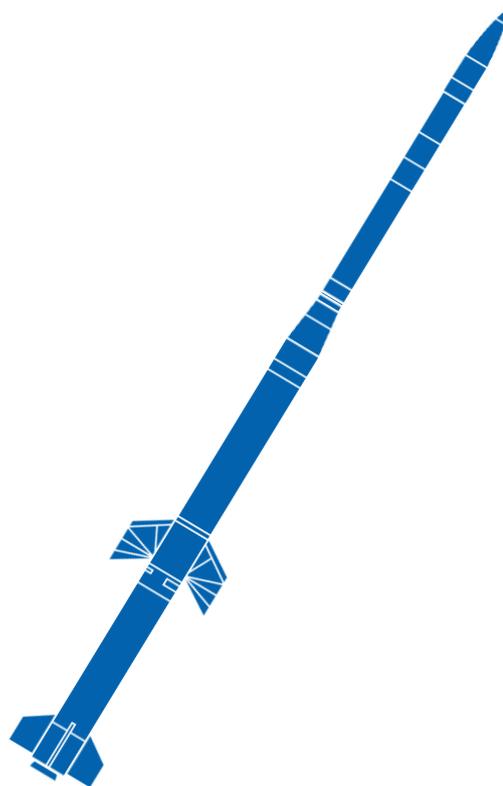
Degree project in Mathematical Statistics

Second cycle, 30 credits

Inverse Uncertainty Quantification for Sounding Rocket Dispersion

A Hierarchical Bayesian Approach using
Neural Network Meta Models

TOVE ÅGREN



Date: 26/12/2022

Division of Mathematical Statistics
Department of Mathematics
School of Engineering Sciences
KTH Royal Institute of Technology
Lindstedsvägen 25, 10044 Stockholm, Sweden

Host company: SSC - Swedish Space Corporation
Supervisor: Ludvig Sjöberg (SSC), Camilla Landén (KTH)
Examiner: Camilla Landén

Swedish title: Invers Osäkerhetskvantifiering för Sondraketer

Degree Project in Mathematical Statistics (30 credits)
M.Sc. in Applied and Computational Mathematics (120 credits)
Degree Programme in Engineering Physics (300 credits)

Abstract

Sounding rocket impact points are subject to dispersion due to uncertainties in simulation model parameters and perturbations of the rocket trajectory during flight. Estimating the area of dispersion assumes that associated model uncertainties and magnitude of perturbations have already been inferred. In this thesis, a method to inversely quantify uncertainty in rocket simulation models based on launch data is presented. We take on a probabilistic approach based on Bayesian hierarchical modeling, to address both epistemic and aleatory uncertainty while incorporating prior knowledge about the modeled system. Bayesian computational techniques, including Markov Chain Monte Carlo simulations and modular Bayesian analysis, are accounted for and employed in numerical case studies. Surrogate deep neural network models are shown to ease otherwise infeasible computational burden that posterior distribution exploration suffers from. Numerical experiments are carried out based on actual launch data from Esrange Space Center, serving as validation of the methodology and providing posterior distributions of the target dispersion parameters. The results imply almost certainly that the currently used dispersion parameters can be reduced, for all considered sources of uncertainty in the study. Updating said parameters accordingly yields a potential 20% decrease in theoretically estimated dispersion area, which is in good agreement with empirical observations.

Key Words: Uncertainty quantification, Bayesian inference, Rocket dispersion, Neural networks, Markov Chain Monte Carlo.

Sammanfattning

Sondraketers nedslagspunkter är föremål för spridning på grund av osäkerheter i simuleringsmodellens parametrar och störningar av raketbanan under flygning. Uppskattning av spridningsområdet förutsätter att modellosäkerheter och störningarnas magnitud redan är kända. I detta arbete presenteras en metod för att inverst kvantifiera osäkerhet i raketsimuleringsmodeller baserad på uppskjutningsdata. Med ett probabilistiskt tillvägagångssätt baserat på Bayesianisk hierarkisk modellering kan epistemisk och aleatorisk osäkerhet adresseras simultant, samtidigt som förkunskaper om det modellerade systemet kan införlivas. Bayesianiska beräkningstekniker, inklusive Markovkedje-Monte Carlo-simuleringar och modulär Bayesianisk analys, redogörs för och används i numeriska fallstudier. Surrogatmodeller i form av djupa neurala nätverk visas underlätta den beräkningsbörda som utforskning av a posteriori-fördelningar ofta innebär i praktiska tillämpningar. Numeriska experiment utförs baserat på uppskjutningsdata från Esrange Space Center, vilket validerar metodiken och tillhandahåller a posteriori-estimat av eftersökta spridningsparametrar. Resultaten implicerar nästan säkert att de för närvarande använda spridningsparametrarna kan reduceras, för alla studerade källor till osäkerhet. Uppdatering av nämnda parametrar ger en potentiell 20% minskning i teoretiskt uppskattad spridningsarea, vilket stämmer väl överens med empiriska observationer.

Nyckelord: Osäkerhetskvantifiering, Bayesianisk inferens, Raketspridning, Neurala nätverk, Markovkedje-Monte Carlo.

Acknowledgments

First, I would like to express my gratitude to my academic supervisor Camilla Landén for her diligent support, encouragement and continuous constructive feedback in the writing of this thesis. Secondly, the completion of this work was made possible by Ludvig Sjöberg, my industrial supervisor at SSC. His investment of time and dedication to this project, invaluable input and never-failing faith in my ideas and abilities have been indispensable. I also want to thank the Swedish Space Corporation for the opportunity to take on this exciting project. An especially sincere thanks is due to the entire Orbital Launch and Rocket Test division for the warmest of welcomes and letting me be a part of an inspiring and competent team. Further, I want to direct a word of appreciation to Pierre Nyquist at KTH for taking time to answer my questions and give advice regarding the mathematical details in this study, in particular on the intricacies of Bayesian data analysis. Finally, the brilliant rocket illustration is credited to Demitris Vincefi, to whom I also owe a dept of thanks for moral support and inspiration.

Contents

1	Introduction	1
1.1	Related Work	2
1.2	Objectives and Limitations	2
1.3	Report Outline	3
2	Sounding Rockets	4
2.1	Technical Definition	4
2.2	Launch Procedures	5
2.2.1	Predicting Impact Point	5
2.2.2	Rocket Launch Data	9
2.3	Sounding Rocket Dispersion	10
2.3.1	The 3σ Vehicle Dispersion Area	10
2.3.2	Theoretical Dispersion Estimation	11
2.3.3	Empirical Dispersion	14
3	Inverse Uncertainty Quantification (IUQ)	20
3.1	Introduction to IUQ	20
3.2	Bayesian Statistics	23
3.2.1	Important Distributions and Definitions	24
3.3	Bayesian IUQ	26
3.3.1	Hierarchical Bayesian Modeling	26
3.3.2	Probabilistic Modeling of Rocket Dispersion	27
4	Bayesian Computation	35
4.1	Markov Chain Monte Carlo	35
4.1.1	Introduction to Markov Chains	36
4.1.2	Metropolis-Hastings and Gibbs Sampler	37
4.1.3	Diagnostics	39
4.1.4	Limitations of Standard MCMC Methods	41
4.2	Modal Approximations	42

4.3	Modular Bayesian Analysis	44
5	Surrogate Modeling	46
5.1	Introduction to Neural Networks	46
5.1.1	Fully Connected Feed-Forward Neural Networks	47
5.1.2	Training a Neural Network	48
5.2	Surrogate Models of Rocket Simulator	49
5.2.1	Data Generation	49
5.2.2	Training and Evaluation Setup	50
5.2.3	Performance	51
6	Methods and Model Selection	53
6.1	Choosing Hyperparameters	53
6.1.1	Informative Priors on QoIs	53
6.1.2	Additive Noise	56
6.2	Variable Selection	57
6.2.1	Sensitivity Analysis of Forward Model	57
6.2.2	Model Specifications	58
6.3	Methods for Posterior Exploration	59
6.3.1	MCMC Algorithm	59
6.3.2	Method of Modularization	62
7	Numerical Results	64
7.1	Minimal Model	64
7.1.1	Convergence Diagnostics	65
7.1.2	Resulting Posteriors	66
7.2	Multidimensional Model	69
7.2.1	Full Bayesian MCMC sampling	69
7.2.2	Posterior Modes	70
7.2.3	Modular Analysis	73
7.3	Posterior Rocket Dispersion Estimates	76
7.3.1	Sensitivity to Hyperparameters	78
8	Conclusions and Outlook	80
8.1	Conclusions	80
8.2	Future Work	81
	Bibliography	83

A Supporting Material	88
A.1 Example Wind Profile	88

Chapter 1

Introduction

Models of physical systems are inevitably incomplete. Accepting this fact is easier once one knows just how much model based predictions can be expected to deviate from actual observations. Estimating uncertainty in computational models is an important part of validating the use of simulations in engineering practice. In particular, it becomes crucial in cases where prediction error is associated with risk. Launching sounding rockets is an excellent example of such an application. Sending vehicles without active guiding systems on ballistic trajectories to the upper part of the atmosphere requires careful considerations to safety. Characterizing the dispersive nature of the rocket ground impact point plays a critical role in this; safe operations necessitates defining an area that most certainly will contain the impact point. In industry, this region is referred to as the dispersion area. However, estimating named area presupposes that uncertainties in atmospheric conditions, rocket manufacturing and performance characteristics have already been quantified. Determining parametric uncertainties of this kind in a mathematical model often relies on estimates based on expert opinions, prior experience and/or conventions. As data becomes available from experiments, such estimates should preferably be revised considering that one has gained knowledge of the actual behavior of the system. However, updating model uncertainty based on observations is highly non-trivial. In fact, the main body of this work is dedicated to this task, in the setting of sounding rocket impact point dispersion.

1.1 Related Work

To this day and to the best knowledge of the writer, no established method exists to estimate uncertainty in sounding rocket models based on launch data. Nevertheless, the task fits well into a more general setting, namely as an Uncertainty Quantification (UQ) problem. The ever-growing field of scientific computing, in synergy with development of efficient hardware, allows for constructing and numerically solving ambitiously parameterized models of complex systems. In addition to the task of specifying the values of model parameters, a parallel field of research is concerned with determining the level of confidence, or uncertainty, in the same values. The field of UQ engages with ubiquitous questions of this character. Probabilistic approaches in UQ have received increasing attention in research, in particular under a Bayesian framework [1]. The latter has shown to be a successful and robust approach in engineering applications in a vast variety of fields such as nuclear physics [2, 3], geophysics [4], aerospace [5] and structural dynamics [6], to name a few.

1.2 Objectives and Limitations

The main objective of this thesis is to develop a statistical method to quantify uncertainties in sounding rocket simulation models. Specifically, we target inferring parameters governing the distributions of disturbances that cause rocket impact point dispersion. Throughout this work, these parameters of interest will be referred to as individual σ -values or simply dispersion parameters. To demonstrate and validate the method, we implement it for the specific example of commonly launched rockets from Esrange Space Center by the Swedish Space Corporation (SSC). Employing the novel approach, we compute updates of the currently used σ -values considering provided launch data. Method evaluation necessitates that available data is aggregated and represented appropriately, including estimations of the empirically observed impact dispersion. Thus, a condensed data analysis of the empirical dispersion of launched rockets at Esrange is entailed in the scope of this study.

The ultimate goal of revising σ -values is to improve the accuracy of dispersion estimates. In addition to circumventing having ad-hoc estimated parameters as basis of safety analysis, updates can potentially enable reducing the size of estimated dispersion area. Such an area reduction allows more

frequent launches as more rocket configurations will meet the safety requirements, thus increasing efficiency of the operations at SSC.

In the numerical study, we limit ourselves to two different types of rockets. In spite of this, we aim to present an approach with such a generality that results are applicable in a wider sense, i.e. to dispersion estimates of similar rockets. To avoid the curse of high-dimensionality, the demonstration is restricted to only consider the variables heavily dominating dispersion.

1.3 Report Outline

The thesis is organized as follows: Chapter 2 acquaints the reader with the subject of sounding rockets in general and impact dispersion analysis in particular. In addition, we present statistical summaries of the launch data from Esrange. Next, in Chapter 3 the thesis objective is re-phrased into the framework of Bayesian Inverse Uncertainty Quantification. To this end, we give an overview of Bayesian statistics and its applicability to UQ problems. The probabilistic model of rocket dispersion derived in Section 3.3.2 is a particularly significant outcome of this thesis. Solving the same model involves specialized computational methods; Chapter 4 accounts for a selection of such. Accompanying standard approaches to full Bayesian inference, we present approximate and modular alternatives to remedy issues associated with the former. Chapter 5 introduces the use of surrogate models in UQ problems, specifically deep neural networks, to ease potentially impractical computational time complexity. In Chapter 6, we tailor the proposed framework to the specific case of rocket launches at Esrange by specifying variables, hyperparameters and designing algorithms. Numerical results obtained when employing the methods from preceding chapters are presented in Chapter 7, based on launch data and rocket simulation models provided by SSC. Chapter 8 concludes the report with a discussion of results, contributions and limitations of the study and future implications.

Chapter 2

Sounding Rockets

The first chapter is devoted to the main objects of interest, namely sounding rockets. A basic technical definition is given in Section 2.1, followed by a brief introduction to the operations carried out by the Swedish Space Corporation (SSC). Subsequent sections will introduce necessary general concepts and terminology concerning sounding rocket dispersion, and for the rockets launched from Esrange specifically.

2.1 Technical Definition

Herein, the term sounding rocket will be used in a general sense to encompass suborbital launch vehicles, with trajectories reaching space at upper atmospheric altitudes from 40 km up to 1500 km. The technical features and capabilities vary greatly over different vehicles; rockets have different propulsion and guiding systems, sizes, motors and consist of multiple stages, depending on the mission objectives. While there are ways to aerodynamically stabilize rockets during flight, e.g. spin stabilization using canted fins, the sounding rockets launched from Esrange are typically unguided. Thus, the only mode of control is to adjust the orientation of the launch rail prior to take-off.

Most sounding rockets carry a payload for the purpose of doing scientific research, including microgravity experiments in fields of biology, material sciences and physics, as well as atmospheric research [7]. The main advantages of using sounding rockets for scientific experiments include low costs and short lead times, as compared to orbital launch missions. Furthermore, sounding rockets cover a layer of the atmosphere otherwise difficult to ac-

cess, as the altitudes in which they can operate are just above the maximum height for balloon-borne experiments, but below the operational capabilities of Low Earth Orbit (LEO) satellites [8].

2.2 Launch Procedures

The Swedish Space Corporation (SSC) is a state-owned company that owns and operates Esrange Space Center, located outside Kiruna, Sweden. At Esrange, sounding rockets are regularly launched on suborbital trajectories for a variety of scientific purposes. Throughout this thesis, at the main focal point for examples and results are two different rocket types, called Improved Orion and VSB-30. Both rockets are unguided, carry a payload and are driven by solid-propellant. The former is a small, single-stage rocket while the latter consists of two stages. They are both spin stabilized via canted fins inducing a roll rate, acting to increase aerodynamical stability by diminishing effects of asymmetry in the rocket [9]. The VSB-30 is additionally stabilized via spin-up motors further increasing the roll rate. The properties of the VSB-30 rocket makes it exhibit a relatively large amount of *dispersion*. The term rocket dispersion is central for this thesis, and thoroughly discussed in Section 2.3. It refers to the phenomenon of statistical displacement from the *expected point of impact* due to unknown perturbations to the rocket during flight and uncertainties in launch conditions. Before further elaboration on rocket dispersion, the procedure of obtaining the expected point of impact is outlined in the following section.

2.2.1 Predicting Impact Point

The impact point (IP) prediction procedure can be summarized as follows. Prior to launch, nominal launch settings, i.e. elevation γ and azimuth angle δ of the launch rail defined as in Fig. 2.1, are chosen to obtain a desired impact point or aim point. For this setting, one assumes that no wind is affecting the rocket trajectory. To determine the nominal point of impact, the rocket trajectory is simulated using a numerical solver of the differential equations representing the dynamical system of the rocket and its environment. It is a six degrees of freedom (6DOF) simulation, and a call to the model code on a standard CPU takes between 0.5 and 1 second depending on rocket configuration.

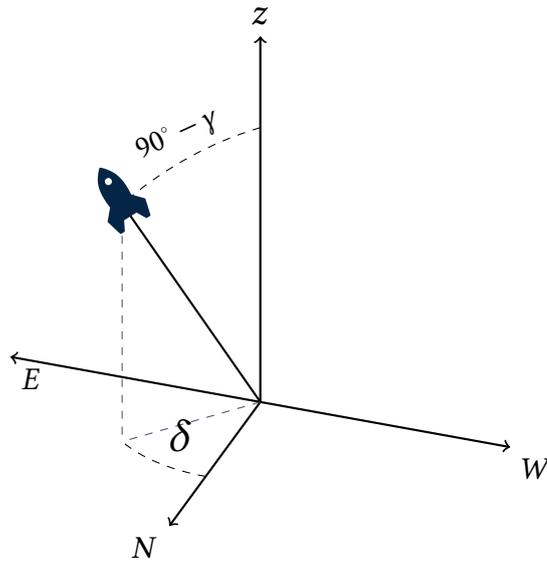


Figure 2.1: Definition of launch angles γ and δ .

Wind Weighting

The prevailing wind conditions during launch will cause the rocket to deviate from a nominal trajectory. Prior to launch, wind data is collected through multiple measurements from wind towers and atmospheric balloons. An average wind profile stretching from the ground to the top of the effective atmosphere is constructed from the aggregated data. Next, wind is compensated for with a technique referred to as wind weighting, with the objective of obtaining the desired aim point but in the presence of a non-trivial measured wind profile.

The wind response of an unguided rocket changes during the different phases of the flight trajectory. During the acceleration phase, the rocket will turn into the wind known as the weathervane effect, while a decelerating rocket will drift with the wind [10]. Furthermore, sounding rockets are more sensitive to winds in the lower parts of the atmosphere, due to the low rocket velocity and the high atmospheric density. Consequently, the wind in different altitude layers contribute in varying degrees to the total deflection from a hypothetical no-wind impact point. Wind weighting relies on constructing a so called *ballistic wind* (BW) with the same net effect on the impact displacement as the total measured wind profile. In order to do so, one determines the so called *wind weighting function* to obtain a relative measure

of wind caused displacement to a given altitude. Formally, let $\Delta W(z)$ denote impact displacement from a nominal, zero-wind trajectory due to a uniform wind from the ground level to a given altitude z [km]. Note that $\Delta W(0) = 0$ and

$$\lim_{z \rightarrow \infty} \Delta W(z) = \Delta W,$$

where ΔW denotes the total displacement from a uniform, constant wind throughout the entire atmosphere. In practice, one considers a practical upper limit h and sets $\Delta W = \Delta W(h)$ for e.g. $h = 40$ km. The wind weighting function is defined by

$$f : [0, h] \rightarrow [0, 1], \quad f(z) := \frac{\Delta W(z)}{\Delta W} \quad (2.1)$$

Now, via Eq. (2.1) one can compute the relative contribution to IP displacement originating from wind in the k :th layer of the altitude strata, located at $[z_{k-1}, z_k]$, $k = 1, \dots, K$. These relative contributions are referred to as *wind weighting coefficients*, defined by $\Delta f(z_k) := f(z_k) - f(z_{k-1})$.

The wind weighting coefficients serve as a basis when constructing the ballistic wind, a constant wind vector acting uniformly through the entire atmosphere that equates to the measured wind profile. As such, the ballistic wind is a vehicle specific weighted wind average. Denote the average wind in the k :th layer by w_k . The ballistic wind is obtained from the wind weighting coefficients via

$$w_{ballistic} = \sum_{k=1}^K \Delta f(z_k) w_k \quad (2.2)$$

To estimate the total impact displacement, the ballistic wind is multiplied with a vehicle specific Unit Wind Effect (UWE), quantifying the magnitude of the wind displacement vector due to a ballistic wind. Finally, one determines an updated set of launch settings in order to compensate for the estimated displacement. This can be done by iterating launch settings by a first order correction approach readily derived via geometric arguments. For details, see e.g. [11].

Example: Constructing a Wind Weighting Curve

To illustrate the method of wind weighting, we conduct a toy study based on a measured wind profile from a VSB-30 launch campaign from 1/12 2005. The exact wind profile is found in Appendix A.1. In addition, it will allow us to estimate the magnitude of the error introduced by the linearization in the

wind weighting procedure. First, a no wind trajectory is simulated to obtain a reference nominal impact point. Next, for each wind layer, a constant, uniform wind perturbation of 0.3 m/s is applied from 0 to z_k , $k = 1, \dots, K$ with $z_K = 40$ km. We log the displacement $\Delta W(z_k)$ from the nominal IP and use Eq. (2.1) to obtain the final wind weighting curve. Note that in Fig. 2.2, the function $1 - f(z)$ is plotted. For conclusiveness, results from perturbing either the east or north wind component is computed. The two resulting curves show very good, yet not perfect, resemblance. The small differences, small enough to barely be discerned in Fig. 2.2, are likely to stem from non-linear dependence on the launch settings. In the example, the launch azimuth angle is not aligned with north or east direction.

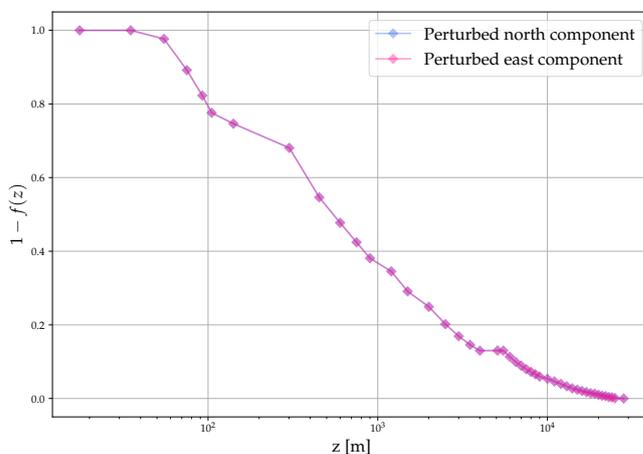


Figure 2.2: Wind weighting curve for VSB-30 toy example.

Finally, the resulting ballistic wind is computed via Eq. (2.2) and given in Eq. (2.3). We form an average from the results using the two different wind weighting curves in Fig. 2.2. The components of $w_{ballistic}$ are in north and east direction respectively, given in m/s. In this example, the underlying wind profile is dominated by a strong east component. As expected, we see that this results in a ballistic wind with a large resulting east component.

$$w_{ballistic} = [0.874, 3.46] \quad (2.3)$$

The approximation error introduced by equating the ballistic wind to the full wind profile is estimated by comparing the resulting respective impact points. The discrepancy in impact point denoted by ϵ is of the magnitude $\epsilon \sim$

0.1 km, corresponding to a perturbation of the ballistic wind of the order 10^{-2} m/s. As a small note, the results have some dependency on the target wind profile. A wind weighting curve based on perturbing the east component yields a more accurate end result when compared to the ground truth. In all, the results indicate that the wind weighting technique yields a satisfactorily accurate result in most applications, and that this relatively small model error can be neglected in cases where uncertainty in wind is of orders larger than 10^{-2} m/s.

2.2.2 Rocket Launch Data

As part of a launch campaign, data is collected prior and post to launch and summarized as the example entry in Table 2.1. Note that this example is not exhaustive, since variables and parameters not directly relevant for the purpose of this study have been excluded. In addition, the rocket type label contains all rocket specific parameters, e.g. characterizing engine and aerodynamic properties, used for simulating the trajectory.

Rocket type	Payload	Ballistic wind	Predicted IP	Actual IP	Launch setting
VSB-30	M kg	$w_{ballistic}$ m/s	$(\hat{r}, \hat{\alpha})$	(r, α)	(γ, δ)

Table 2.1: Launch data description.

The impact point (IP) is commonly parameterized with polar coordinates with range r [km] from launch position and azimuth angle α [deg] with north defining $\alpha = 0$, illustrated in Fig. 2.3. Finally, define the *downrange* and *crossrange* directions as parallel and perpendicular to the vector $\vec{IP} - \vec{O}$.

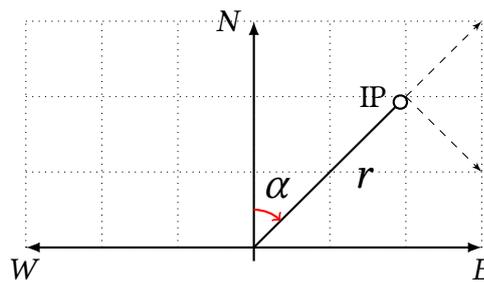


Figure 2.3: Conventional definition of IP parameterization. Downrange and crossrange directions are indicated with dashed arrows.

2.3 Sounding Rocket Dispersion

During flight, sounding rockets are subject to disturbances that will cause the trajectory to deviate from the nominal case. Disturbances can relate to wind measurement errors and gusts, off-sets in the launch rail and engine imperfections resulting in impact deflection. In addition, uncertainties in flight environment and rocket specific parameters can also result in a non-negligible discrepancy between predicted and actual IP. From a safety requirement perspective, it is crucial to estimate the size and shape of the *dispersion area*, defined in the following section.

2.3.1 The 3σ Vehicle Dispersion Area

The theoretical dispersion estimate must enclose the region within which there is a non-negligible probability of ground impact. In the continuation of this work, this estimate will be referred to as the 3σ vehicle dispersion area. Early practices aiming to estimate the dispersion area are based on a symmetric simplification where the probability of a given displacement is assumed to be identical in all directions. This will always yield a circular dispersion area. Furthermore, assuming that the statistical deviation of actual IP from the predicted IP follows a bivariate normal distribution, the probability that the 3σ dispersion area contains the IP is 0.989 [12].

Aerodynamical characteristics vary between different types of sounding rockets, as well as sensitivity to launch conditions and external disturbances. Before a vehicle can be approved for a launch campaign, a trajectory analysis to estimate the 3σ vehicle dispersion area must be conducted and assured to comply with the safety requirements of the launch site. At Esrange, sounding rockets must comply with a $3\sigma = 60$ km requirement. 60 km refers to the radius of the estimated dispersion disc [13]. Note that this is based on the 2-dimensional isoprobability assumption. It is up to the practitioner to generate the 3σ dispersion of a vehicle, which in turn requires identification of key perturbation factors that contribute to the dispersion of the vehicle and a statistical description of these. In most practices, this means that one estimates the individual 3σ -values of the key dispersion factors. Prescribing the σ -values is a way of quantifying the variability or uncertainty in the input variable or parameter causing dispersion. Assuming that the statistics of the input is well described by e.g. a normal distribution, σ can be interpreted

as a standard deviation [10].

In Table 2.2, an example of the identified dispersion factors for a two stage fin stabilized vehicle is presented, along with associated σ -values. The configuration of factors is not uniquely defined, but should be proposed and motivated by the practitioner prior to a launch campaign. Also note that some parameter values are given relative to the corresponding rocket specific quantity.

STAGE 1	3σ
Thrust	3%
Thrust alignment in pitch	0.1°
Thrust alignment in yaw	0.1°
Aerodynamic drag	20%
Fin alignment	0.1°
STAGE 2	
Thrust	3 %
Thrust alignment in pitch	0.1°
Thrust alignment in yaw	0.1°
Aerodynamic drag	20%
Ignition time	0.25 sec
Launch elevation	1°
Launch azimuth	4°
North wind	3 m/s
East wind	3 m/s

Table 2.2: Example of key dispersion factors for a two stage sounding rocket.

2.3.2 Theoretical Dispersion Estimation

Essentially two methods are used to estimate the dispersion area in industry today, accounted for in the following sections. In both methods, impact displacement from a simulated nominal unperturbed impact point in the center of the dispersion area is considered.

Residual Sum of Squares

Historically, the widely used method to estimate the 3σ dispersion area is a so called Residual Sum of Squares (RSS) procedure. It is also the proposed method by FAA guidelines [12], and the method utilized by SSC. In a RSS procedure, one considers the individual effect of each key dispersion factor separately, assuming independence. For each factor, a series of simulations are run when only perturbing the input under consideration by its 3σ -value, e.g. by adding or subtracting it from the nominal value. Consider the example of wind disturbance for illustrative purposes. Two simulations are run when the headwind speed is equal to ± 3 m/s respectively, tabulating the resulting impact deflections in downrange and crossrange directions. The square root of summed squared impact displacements following these perturbations yields the so called total 3σ dispersion area of the sounding rocket. More generally one can allow the dispersion area to be elliptic, such that the downrange and crossrange semiaxes are estimated separately. What follows is a mathematical formulation of the procedure.

Let a_i^+ denote the resulting displacement from the nominal IP in downrange, a , when applying a positive sign perturbation to the i :th variable with its individual 3σ -value. The sign superscript corresponds to the respective sign of the perturbation. Now, define σ_a^+ by

$$3\sigma_a^+ = \sqrt{\sum (a_i^+)^2} \quad (2.4)$$

The corresponding quantity when applying perturbations with a negative sign, denoted by σ_a^- follows analogously. By taking the average, we obtain the downrange semiaxis of the dispersion area

$$\sigma_a = \frac{\sigma_a^+ + \sigma_a^-}{2} \quad (2.5)$$

The equivalent holds for displacements in the crossrange direction, b , to obtain σ_b . Assuming isoprobability, we obtain a circular dispersion area by taking the average

$$\sigma = \frac{\sigma_a + \sigma_b}{2} \quad (2.6)$$

Monte Carlo Sampling

A more contemporary method is that of a large-sample Monte Carlo (MC) approach. The dispersion area is estimated by simulating a series of trajectories when perturbations to model inputs, identified to cause dispersion, are

drawn from their prescribed probability distributions. By the assumption that the dispersion can be well approximated by a bivariate normal distribution, dispersion estimates are provided by the sample mean and covariance matrix of the impact point displacements. We go back to the example of wind disturbance. Assuming a nominal zero wind speed in north and east directions, we model their statistics (independently) as normally distributed with $\sigma = 1$ m/s. In each simulation run, the perturbed wind speeds are sampled from standard normals [10].

As opposed to RSS, MC simulations account for all sources of dispersion simultaneously, and thus captures possible non-linear interactions between perturbed inputs. Potentially, this yields a more precise and realistic estimate, in particular for more exotic dispersion areas where the assumption of a circular shape is invalid. However, the computational time complexity of MC simulations is much greater than with RSS, as the former typically requires thousands of forward model calls compared to the order of tens for the latter [14].

Dispersion Estimate Example of a VSB-30 rocket

In this section, we use the dispersion factors presented in Table 2.2 to conduct a benchmark dispersion study of a VSB-30 rocket. We will come back to this example to evaluate the effect of updating the individual σ -values. The launch configuration for our hypothetical campaign is given in Table 2.3. We make a note that the dispersion estimation depends on the choice of nominal launch settings. A smaller launch elevation angle γ will cause the impact range to increase, which in general may cause an increase in sensitivity to perturbations thus a larger dispersion area.

Rocket type	Payload mass	Launch settings	No wind IP
"VSB-30"	396.3 kg	$\gamma = 87.7^\circ, \delta = 0^\circ$	$r = 78.142$ km, $\alpha = 357.71^\circ$

Table 2.3: Launch configuration for dispersion example.

For the Monte Carlo estimate, a sample of 10^4 simulations is used to estimate the total dispersion parameter σ , presented in Fig. 2.4.

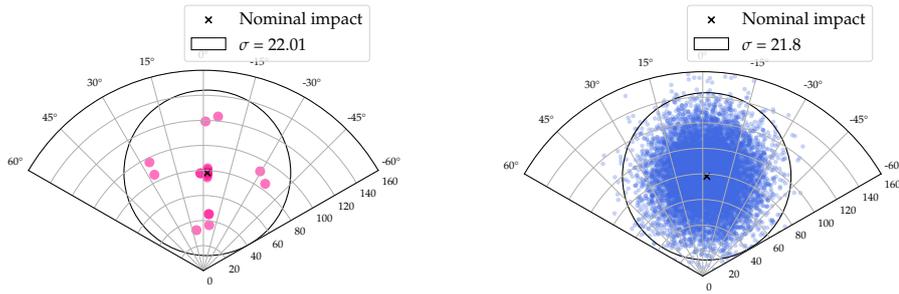


Figure 2.4: Comparison of theoretical dispersion estimation methods for a VSB-30 rocket. **Left:** RSS. **Right:** Monte Carlo.

The general trend, visible in Fig. 2.4, is that the RSS method seems to slightly overestimate the dispersion as compared to the Monte Carlo simulation in this particular example. For this "simple" type of dispersion, when the assumption of circular dispersion holds, the RSS procedure can be considered sufficient and preferable. The main advantage is the low computational cost, as it only requires a handful of model evaluations, compared to the tens of thousands for an MC estimate. For more complex dispersion estimates, a Monte Carlo procedure is likely to be more valid, capturing complex, non-linear phenomena.

2.3.3 Empirical Dispersion

As the size of the data set of rocket launches increases, so does the relevance of studying the empirically observed dispersion at Esrange and compare it to the theoretical estimates. One way to do so is by means of density estimation and point estimators of the empirical σ -value characterizing the dispersion area. Stepping back to a more general setting, it is a common problem in data modeling to fit an appropriate density to experimental data that is assumed to be generated from an unknown underlying distribution. Next, we present two approaches to this end, namely non-parametric and parametric density estimation.

Non-parametric Density Estimation

An advantage with non-parametric density estimation is that it does not require any assumptions on the form of the underlying distribution of the data. It is a useful tool for a first inspection and characterization of a data set; perhaps the simplest and most familiar form of such estimation is a histogram. Kernel density estimates is a popular non-parametric density estimator. For

data $\{y_i\}_{i=1}^n$ drawn from the density $p_Y(\cdot)$, we define the *kernel density estimator* \hat{p}_h of p_Y by

$$\hat{p}_h(y) = \frac{1}{nh} \sum_{j=1}^n k\left(\frac{y - y_j}{h}\right), \quad y \in Y, \quad h > 0, \quad (2.7)$$

where k is some kernel function, such that $\hat{p}_h(y)$ is indeed a density, and h is called a band or window width. One can interpret kernel density estimations as a smoothed histogram, where the choice of h governs the smoothness of the estimator. Larger bandwidth leads to a smoother density distribution, whereas a smaller bandwidth captures high-variance features. One natural class of kernel functions are (multivariate) probability density functions, Gaussian kernels being a common choice. Moreover, the non-parametric density estimation serves as a generative model from which one can sample to generate "new" data points from the fitted density [15].

Parametric Density Estimation

In parametric density estimation, one makes assumptions of what parametric family $P_\theta = \{p_\theta(\cdot), \theta \in \Theta\}$ the underlying distribution of the data belongs to. The task reduces to estimating the parameters $\hat{\theta}$ that fits the data "best". A standard approach is that of Maximum Likelihood Estimation (MLE), in which the parameters are taken as the maximizers of the likelihood of the data, or equivalently, minimizers of the negative log-likelihood [16]. Assuming independent observations $\{y_i\}_{i=1}^n$, the MLE estimate is obtained via

$$\hat{\theta}_{MLE} = \arg \max_{\theta \in \Theta} \prod_{i=1}^n p_\theta(y_i) = \arg \min_{\theta \in \Theta} \sum_{i=1}^n -\log(p_\theta(y_i)) \quad (2.8)$$

Empirical Dispersion Estimates from Esrance Data

In this section, we analyze the impact data from a selection of rocket types launched from Esrance in the course of the last decades.

First, we summarize the impact dispersion with point estimates. The empirical σ -values are estimated based on the assumption of a symmetric dispersion area. Again, denote downrange and crossrange displacement for the i :th launch by a_i and b_i for $i = 1, 2, \dots, n$, respectively. For a collection of

displacements $\{(a_i, b_i)\}_{i=1}^n$, the sample variances are estimated according to

$$\hat{\sigma}_a^2 = \frac{1}{n-1} \sum_{i=1}^n (a_i - \bar{a})^2, \quad \hat{\sigma}_b^2 = \frac{1}{n-1} \sum_{i=1}^n (b_i - \bar{b})^2, \quad (2.9)$$

where $\bar{a} = \frac{1}{n} \sum_{i=1}^n a_i$. The unbiased pooled variance estimate is obtained as the mean of the sample variances according to Eq. (2.10).

$$\hat{\sigma}^2 = \frac{\hat{\sigma}_a^2 + \hat{\sigma}_b^2}{2} \quad (2.10)$$

For the estimates presented herein, we consider two rocket types; the by now familiar VSB-30 and Improved Orion. Note that the latter exhibits a much smaller dispersion area. Impact point displacement from all launches projected onto the downrange and crossrange axes, along with the estimated empirical and theoretical 3σ -dispersion areas are plotted in Fig. 2.5. The σ -values are itemized in Table 2.4.

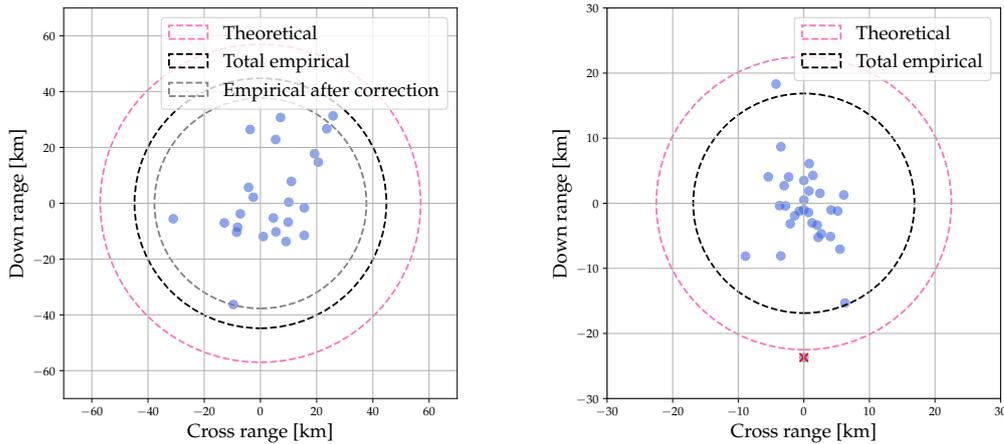


Figure 2.5: Visualization of dispersion with theoretical and empirical 3σ -values. **Left:** VSB-30. **Right:** Improved Orion. Non-nominal flight is marked with red.

Some additional comments on the data sets are in place. Following early launches, the launch rail used for the VSB-30 rocket was found not to be entirely straight. This is likely to have caused increased dispersion due to the unexpected misalignment when rockets left the rail. In 2014 the launcher's rail assembly was realigned and straightened. We present some additional

results when only considering launches after the realignment. Furthermore, for Improved Orion one non-nominal launch due to early separation of motor and payload, is contained in the data set. This data point will not be included in the following analysis.

Rocket type	Theoretical 3σ [km]	Empirical 3σ [km]
VSB-30	57	44.80
Improved Orion	22.5	16.94

Table 2.4: Theoretical dispersion of commonly launched rockets at Esrange.

As evident from Table 2.4, the empirical σ -values are significantly smaller than the theoretical estimates. As illustrated in Fig. 2.5, all impacts are well within the theoretical limit. This observation serves as an additional reinforcement for re-iterating over the theoretical dispersion estimation. In particular, it motivates the relevance of updating current beliefs on the magnitude of individual dispersion factor σ -values.

Moving on to density estimates, we proceed with a disclaimer regarding statistical significance due to the scarcity of data. As it is necessary to differentiate the analysis between rocket types, the size of the data sets shrink dramatically. In general, scarcity of data hampers the validity of some statistical methods and estimates. For example, there is a risk of overfitting the models to the available data, that is not necessarily representative for the actual data generating distribution [15].

First, results from non-parametric density estimation of the empirical dispersion are presented in Fig. 2.6. In particular, we do kernel density estimates with Gaussian kernels. The bandwidth is optimized via grid search cross-validation with regards to a standard Euclidan metric. For implementation details, refer to the documentation in Scikit-learn [17].

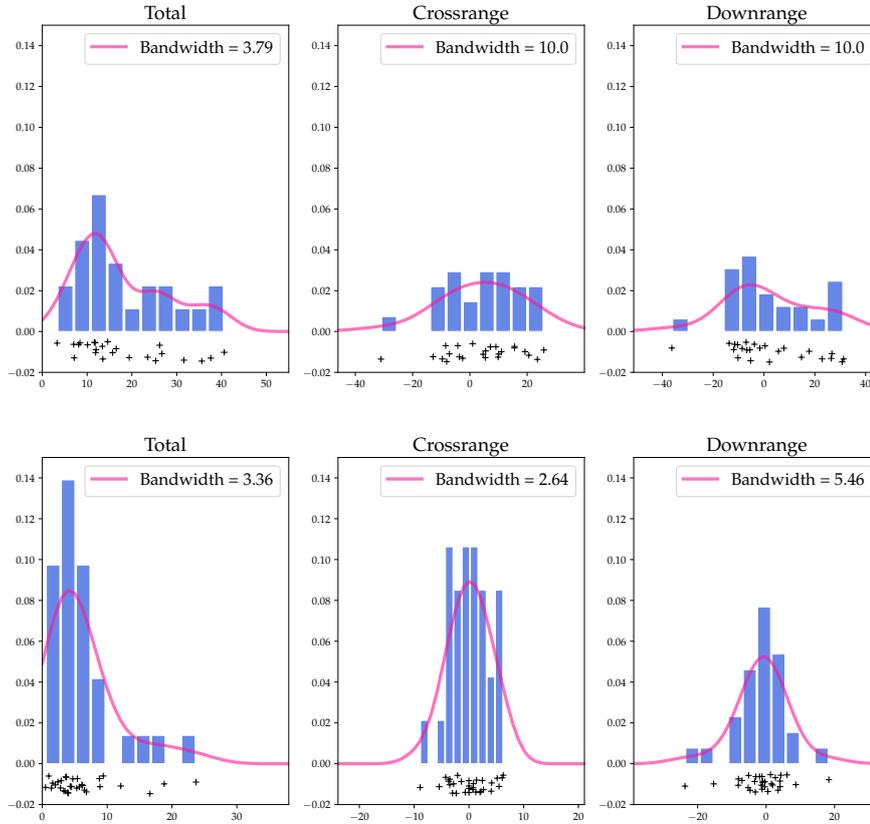


Figure 2.6: Kernel density estimation. **Top:** VSB-30. **Bottom:** Improved Orion.

Turning to parametric based density estimation, we fit the dispersion in crossrange and downrange assuming underlying normal distributions. This assumption implies that, theoretically, the total dispersion follows a Rayleigh distribution. Mathematically, for identically distributed normal random variables $A \sim N(0, \sigma^2)$ and $B \sim N(0, \sigma^2)$, it follows that the root square sum $\sqrt{A^2 + B^2} \sim \text{Rayleigh}(\sigma)$ [18]. Referring to this property, we fit a Rayleigh distribution to the total dispersion. The resulting distributions with fitted mean μ and standard deviation η are presented in Fig. 2.7. The distributions are fit based on MLE as implemented in SciPy. For details, refer to [19].

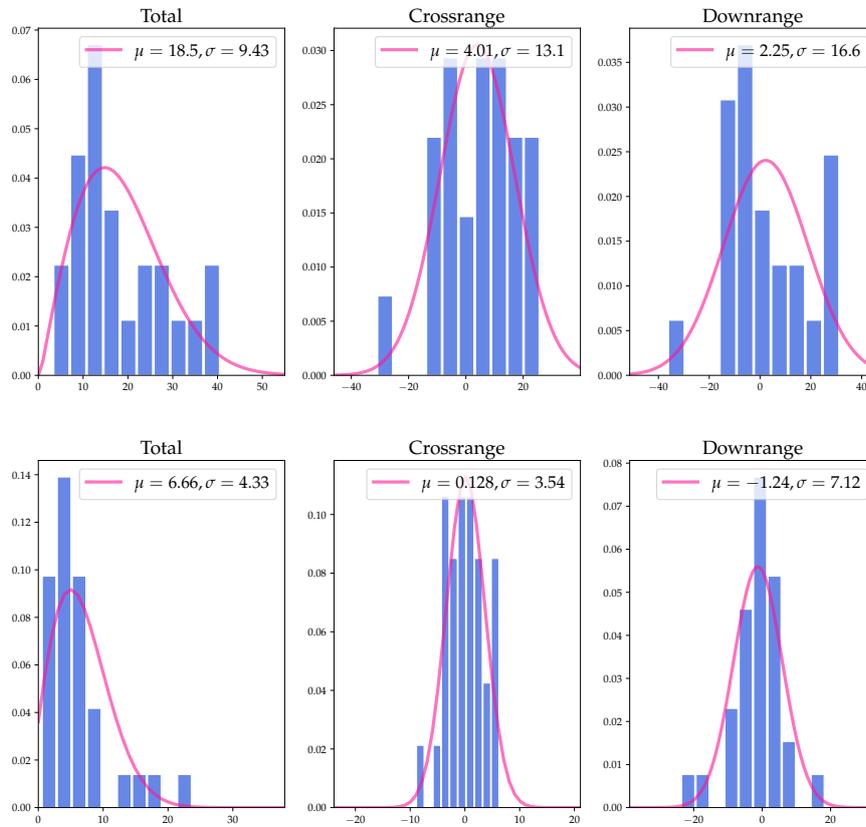


Figure 2.7: Parametric density estimation. **Top:** VSB-30. **Bottom:** Improved Orion.

Based on the distributions fitted in Fig. 2.7 and the theoretical σ -values in Table 2.4, we conclude this chapter by computing the probabilities of having an impact displacement smaller than the theoretical σ . Let y and \hat{y} denote actual and predicted impact point, respectively. The results are presented in Table 2.5. We see that the theoretical dispersion area indeed complies with the common definition to contain 98.9% of ground impact points [12].

Rocket type	$P(\ \hat{y} - y\ \leq \sigma)$	$P(\ \hat{y} - y\ \leq 3\sigma)$
VSB-30	0.56	0.999
Improved Orion	0.61	0.999

Table 2.5: Probability statements for rocket displacement being below the current theoretical σ dispersion radii.

Chapter 3

Inverse Uncertainty Quantification (IUQ)

In this chapter, we account for the mathematical setting in which we will approach the thesis main objective. We propose a novel method to estimating sounding rocket dispersion parameters, aiming to provide a data driven approach in order to avoid using values set by convention or ad hoc statements. To this end, we re-formulate the problem in the framework of Bayesian Inverse Uncertainty Quantification.

The first section introduces the key concepts of Uncertainty Quantification (UQ) in general and inverse UQ in particular. As we will see, the appealing features of Bayesian statistics fit naturally with many IUQ problems. To accustom the reader to this probabilistic approach, Section 3.2 serves as an overview providing definitions and tools that will come in handy for Bayesian IUQ. Finally, we introduce the methodology of Bayesian IUQ for the problem at hand by formulating a probabilistic model of rocket dispersion.

3.1 Introduction to IUQ

At its very broadest definition, Uncertainty Quantification (UQ) aims to investigate the sources and degree of discrepancy between reality and the mathematical models and simulations thereof. Thus, solving UQ problems is a highly multidisciplinary task; it often requires tools from statistics, applied mathematics, engineering and computer science. Even if a constructed model perfectly captures the structural behavior of a system, the accuracy

of predictions will still be limited by the knowledge of physical and model parameters. Lack of knowledge could mean, but is not limited to, uncertain parameter values or model inputs that are subject to a variability that is not precisely known [1].

Throughout this work, the following terminology will be used frequently. Examples are given in the context of launching sounding rockets.

- **Inputs:** Model parameters, initial and environmental conditions, boundary conditions, exogenous forces; e.g. aerodynamic coefficients, rocket specific parameters, wind conditions and launcher settings.
- **Outputs or Responses:** Experimentally and/or numerically measurable quantities, i.e. sounding rocket impact point.
- **Quantities of Interest (QoI):** A problem specific quantity targeted for statistical inference. In the context of rocket dispersion, our QoIs are the σ -values characterizing the distribution of factors contributing to dispersion.

Given the wide definition of UQ, it umbrellas problems of different character. One certain type of UQ problems, lying at the core of this thesis, are *inverse problems*. As the name implies, problems of this type are the target of study in the field of Inverse Uncertainty Quantification (IUQ). In essence, IUQ is a process of characterizing model input uncertainty based on experimental data. By this, one hopes to mitigate potentially ad-hoc "expert opinions" or vaguely motivated conventions when estimating model uncertainties [20]. Once input uncertainties are quantified, they can be propagated through the forward model to obtain uncertainty predictions in the output. This is often required in order to support risk assessments and safety decisions and exactly what the theoretical rocket dispersion estimates amounts to.

We remain at the notion of inverse problems for a moment. As opposed to a *forward problem* where one makes inference about the response variable, an inverse problem targets the model input. Mathematically, given an observation $y \in Y$ and computational model mapping $\mathcal{M} : X \rightarrow Y$, we care to find and/or learn about $x \in X$ such that $\mathcal{M}(x) = y$. A simple illustration of the difference between how information flows in forward and inverse problems is given in Fig. 3.1. As indicated by the solid arrows, in inverse problems one goes in the direction from experimental observations and model outputs

towards model inputs. Forward problems in physical applications are well-posed in general, meaning that a unique solution that depend continuously on data exists. In contrast, inverse problems are known to suffer from ill-posedness [1]. The phenomenon stems from often having under-determined systems of equations due to lack of data contra model complexity as well as corrupted, noisy observations. This leads to deterministic methods often failing to produce meaningful results and the need for regularization to remedy non-uniqueness [21].

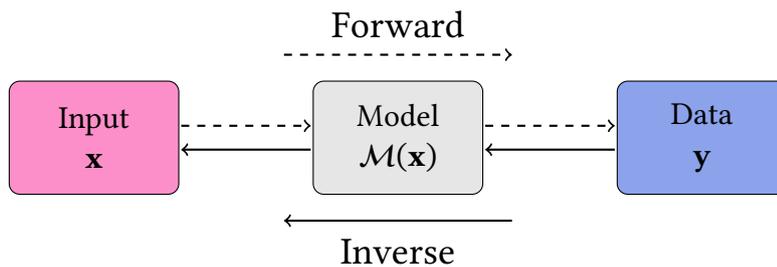


Figure 3.1: An illustration of different information flows in UQ problems. In inverse problems, output data is used to learn about the system inputs.

Epistemic Uncertainty and Aleatory Variability

It is common, and often convenient from a modeling perspective, to differentiate uncertainty into different characteristic categories. One way of doing so, albeit imprecise and subject of debate [22], is to make a distinction between *epistemic uncertainty* and *aleatory uncertainty* or sometimes *aleatory variability*. The former acts as a way of addressing lack of knowledge – exemplified as not knowing e.g. a physical constant to its true value. Epistemic uncertainty is reducible in the sense that gathering more evidence allows knowing the quantity of interest more exactly. For example, the aerodynamical drag coefficient of a rocket is considered to be a constant design parameter. However, the lack of knowledge in its exact value introduces uncertainty in the predicted point of impact. The term parametric is often used to refer to this specific type of epistemic uncertainty in constant parameters. With more data, one hopes to eventually know the parameter value more exactly. Further, another subdivision of epistemic uncertainty is of structural nature, considering doubts about the exactness of the system model as a whole. This topic is elaborated in section Section 3.3.2, where the case of model discrepancy is discussed.

In contrast, aleatory variability refers to uncertainty due to phenomena of variable nature in the system, e.g. a model input subject to some stochasticity. In the following, we will use aleatory uncertainty and variability interchangeably. This type of uncertainty is irreducible in the sense that more data will not reduce variability in future observations. However, uncertainties in e.g. hyperparameters governing the variability can be known to a larger or lesser degree of confidence. Consider for example the actual launcher setting, which is likely to differ from the prescribed setting with slight variations for every launch, assuming that there is no evident off-set. Consequently, we can expect to always have some uncertainty in this model input, no matter the number of launched rockets.

As a final note, this differentiation is by no means definite or unique. Often, it is not clear how to categorize uncertainties, leaving the decision completely in the hands of the modeler. For example, model inputs can potentially suffer from multiple different sources of uncertainty. The prevailing wind conditions during launch is an example of such. In addition to error in the measuring procedure, which can be considered epistemic uncertainty, the rocket is likely to be perturbed by additional wind gusts, an inherently variable phenomenon. One can argue that some uncertainty is indeed reducible, while some sources are irreducible, and the distinct categorization is insufficient. Nevertheless, an attempt to characterize model inputs indicates what the modeler can expect or hope for in terms of reducing total uncertainty in the system.

3.2 Bayesian Statistics

This section will serve as a gentle introduction to the main concepts and definitions in Bayesian data analysis. The fundamental way in which the Bayesian approach differs from the frequentist is how unknown or uncertain quantities are represented. From a frequentist perspective, quantities of interest are assumed to take one true, deterministic value, be it unknown. Under a Bayesian framework, we take on a probabilistic approach where unknown or uncertain quantities are represented as random variables, the distribution of which is the target of inference. As a consequence, probability statements about parameters or unobserved data make up for the main final conclusion in Bayesian data analysis.

The top-level workflow when making inference under a Bayesian framework in a general problem setting can be summarized in the following steps.

1. **Formulate a probabilistic model.**
2. **Condition on experimental data.**
3. **Evaluate fit of model and resulting posterior distributions.**

Results from evaluation typically imply the need to update or expand the model from step 1, yielding a re-iteration of the process until results are deemed satisfactory [23].

3.2.1 Important Distributions and Definitions

First, a small comment on notation is in place. If not stated otherwise, random variables (r.v.) will be denoted by capital letters whereas realizations of such will be in lower case. We almost exclusively consider continuous random variables, and will use the terms and notation for distribution and density interchangeably to refer to continuous probability density functions. Some abuse of notation will be allowed when indicating the distribution of a r.v. Specifically, we write $X \sim f(x)$ if X is distributed according to the density function f . Sometimes, if the distribution is known on a standard form, e.g. a Gaussian r.v. with mean μ and variance σ^2 , we will make use of the notation $X \sim \mathcal{N}(\mu, \sigma^2)$.

Likelihood

Consider a collection of observations, $\{y_i\}_{i=1}^N$, where each y_i is a realization of a random variable Y_i . Denote the full vector by $Y := (Y_1, \dots, Y_N)$. In the example of rocket dispersion, y_i is the observed impact point of the rocket from the i :th launch. Assume now that the distribution of Y is parameterized by the potentially unknown θ . In Bayesian analysis, we think of θ as a realization of the random vector Θ . The *likelihood* of the data quantifies the probability of the observations conditioned on a particular value of θ via the density

$$f_{Y|\Theta}(y|\theta) := f_{Y_1, Y_2, \dots, Y_N|\Theta=\theta}(y_1, y_2, \dots, y_N|\theta), \quad (3.1)$$

referred to as the *likelihood function*.

Prior Distribution

Representing a parameter as a random variable lets uncertainty in its exact

value be captured by specifying a probability distribution over it. We will refer to the distribution of Θ as a *prior distribution*, here given by the prior density $\pi_{\Theta}(\theta)$. Intuitively, if there is a large amount of uncertainty in θ , the prior density will be smeared out. On the other hand, resorting to the classical framework, stating that its true value is indeed θ_0 corresponds to specifying the prior, informally, as a point mass $\pi_{\Theta}(\theta) = \delta(\theta - \theta_0)$. Here δ is the Dirac delta function. With this definition in place, the conditional likelihood can be written in accordance with its general definition such that

$$f_{Y|\Theta}(y|\theta) = \frac{f_{Y,\Theta}(y, \theta)}{\pi_{\Theta}(\theta)} \quad (3.2)$$

Posterior Distribution

At the heart of Bayesian statistics lies Bayes' theorem. The first encounter with it is usually in the context of conditional probabilities of events. For events A and B , we can flip the conditional probabilities via the formula

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (3.3)$$

The corresponding identity for continuous random variables θ and Y is given by

$$\pi_{\Theta|Y}(\theta|y) = \frac{f_{Y|\Theta}(y|\theta)\pi_{\Theta}(\theta)}{f_Y(y)} = \frac{f_{Y,\Theta}(y|\theta)\pi_{\Theta}(\theta)}{\int f_{Y,\Theta}(y|\gamma)\pi_{\Theta}(\gamma)d\gamma} \quad (3.4)$$

In the Bayesian context, the left-hand side in Eq. (3.4) is referred to as the *posterior distribution* of θ given the data y . That is, by conditioning on the data, we can view the posterior as an update of our prior beliefs about θ , now also reflecting seeing the observations $Y = y$. In essence, this is how the Bayesian perspective is used for inference.

The posterior distribution is one of the main targets of computation, a task that more often than not poses severe challenges in real life applications. The marginal distribution of Y in Eq. (3.4) implies computing a potentially complex, or even intractable, integral. However, the emergence and development of specialized methods, Markov Chain Monte Carlo being an important class of algorithms [24, 25] and rapid increase in computational resources have paved the way for applying Bayesian statistics in many engineering applications [26]. The application to inverse uncertainty quantification will be elaborated in Section 3.3, and the computational aspects in Chapter 4.

Conjugate Priors

The task of specifying prior distributions is an important task for the modeler. One is faced with the trade-off of having a model capturing the complexity of the problem at hand and constructing a model that is tractable. Some choices of priors can make computations more feasible as they result in posterior distributions that can be expressed in a closed form. An important class of priors is so called conjugate priors. Suppose that the likelihood function is chosen such that it belongs to a parametric family $\mathcal{F} = \{f_{Y|\Theta}(\cdot|\theta), \theta \in \Theta\}$. We say that the family of priors Π is *conjugate* for \mathcal{F} if the posterior $\pi_{\Theta|Y}(\cdot|y)$ is in Π for all choices of prior $\pi_{\Theta} \in \Pi$ and y .

3.3 Bayesian IUQ

The Bayesian framework has proven to have appealing properties in challenging tasks of inverse problems in general [27] and IUQ in particular [6]. Today, many established techniques and approaches for reliable and robust inverse uncertainty quantification under a Bayesian framework have been established [3, 28]. Importantly, they apply when faced with ill-posed problems; for example with high-complexity models where experimental data is scarce and noisy, which is true in general in engineering and science applications. The probabilistic approach of Bayesian inference is useful in problems where uncertainty is omnipresent, i.e. appears in not only one but several levels of the system [29, 30]. One of the key attributes of the Bayesian framework is how prior knowledge of the system can be incorporated to mitigate issues of non-uniqueness and the unstable nature of solutions to inverse problems. Embedding all available information into the learning process, by means of updating a prior distribution to the posterior after conditioning on the observed data, acts as a way of regularization [31].

3.3.1 Hierarchical Bayesian Modeling

In this thesis, we propose a *multilevel approach* to probabilistic modeling of sounding rocket dispersion. The notion of multiple levels is crucial. It expands the classical inverse problem to include all deterministic and stochastic sub-models in the system, allowing to simultaneously address different types of uncertainties on multiple hierarchical levels [29]. Potential sub-models include the deterministic forward model, as well as stochastic components. The latter comprises prescription of prior input uncertainty and variability, as well as representation of forward model prediction errors. The full, as-

sembled model represents the entire system and all associated uncertainty therein.

In the context of sounding rocket dispersion estimates, the ultimate QoIs are the so called σ -values. By now, they should be familiar to the reader as the parameters governing the epistemic uncertainty and aleatory variability in model inputs used for predicting the ground impact point. Thus, we are not necessarily interested in learning the model inputs, but the parameters governing their distributions. Classical Bayesian inversion fails to suffice to this end [4], thus we resort to a hierarchical expansion. In the full system under consideration, there is uncertainty in both fixed parameters **and** model input that take on variable, launch specific values. Recalling the different types of uncertainty discussed in Section 3.1, an approach that can simultaneously address epistemic and aleatory uncertainty is desired.

The hierarchical modeling approach allows a natural way to address issues arising when the data has a clear hierarchical structure. By prescribing population distributions to e.g. a set of variables, one enforces a structural dependency among the model parameters. This way, one reduces the imminent risk of over-fitting, associated with non-hierarchical models that in general requires many parameters to fit large, complex data sets [23].

3.3.2 Probabilistic Modeling of Rocket Dispersion

Now that the mathematical framework is in place, this section is devoted to re-formulating the problem of rocket dispersion parameter estimation as a Bayesian IUQ problem. As the first step in any Bayesian analysis, we begin by setting up a probabilistic model for the problem at hand. This amounts to determining a joint probability distribution for all observable and unobservable quantities, accounting for our knowledge and beliefs about the underlying system and how data is acquired.

Minimal Model

For a first minimal model, we will allow the simplifying assumption of disregarding uncertainty and variability in all model inputs but wind. All other model inputs are considered deterministic and perfectly known, all comprised in the input vector \mathbf{d} . These inputs can be variables taking on launch specific values, i.e. for N launches we have \mathbf{d}^i , $i = 1, 2 \dots, N$, or constant parameters. The choice of focusing on wind conditions is motivated by (1) The

high model output sensitivity to this variable, but also by (2) The novel way in which the probabilistic model will be formulated, compared to classical Bayesian inversion.

In addition, practitioners in industry today use different values on the wind specific σ -value. Observations by experts at SSC have indicated that there is reason to believe that wind measurements are indeed more exact today than what the currently used σ -value at SSC implies. Ideally, the minimal model will serve as an instructive introduction to the proposed method and a benchmark to expand from. For illustrative purposes, we begin by deriving the model when only considering a single observation $y \in \mathbb{R}^2$.

Let $W = (N, E)^T$ denote the wind vector, where N and E are continuous random variables representing wind speed in north and east direction during the launch under consideration. We will use lowercase letters $w = (n, e)^T$ to denote realizations of W . Let $\mathcal{M} : (w, \mathbf{d}) \rightarrow \mathcal{M}(w, \mathbf{d})$ denote the deterministic forward model mapping. Specifically, the forward model is a 6-DOF numerical simulation of the rocket trajectory, providing the predicted point of impact prior to launch. To account for noisy observations and imperfections in model output, we introduce additive noise ε with density $f_\varepsilon(\varepsilon; \Sigma_\varepsilon)$. A more thorough discussion on how to handle model discrepancy is given in Section 3.3.2. The statistical model, also referred to as the *updating formula*, becomes

$$y = \mathcal{M}(w, \mathbf{d}) + \varepsilon, \quad (3.5)$$

To simultaneously account for variability in wind and represent the uncertainty in the same variability, we impose a so called structural prior on W , with unknown dispersion hyperparameter Θ . That is, we prescribe the conditional prior distribution of W denoted by $f_{W|\Theta}$. Then, we step into hierarchical modeling by putting a prior π_Θ on θ .

As previously explained, wind conditions naturally take on varying launch specific values. A naive approach is to set the location of the structural prior as the experimentally measured value, μ_W say, treating it as a known hyperparameter. However, this is a problem if we want to enforce hyperparameters to apply over all different launches so that $f_{W|\theta}$ serves as a common *population distribution* over the wind conditions. In here we propose yet another representation, effectively equivalent. The wind variable is to split into

a deterministic and stochastic part such that

$$W = d_W + \tilde{W}, \quad (3.6)$$

where the deterministic vector $d_W = (d_{W,1}, d_{W,2})^T$ comprises the at launch measured (ballistic) wind speeds in north and east directions, respectively, and $\tilde{W} := (\tilde{N}, \tilde{E})^T$ represents the unknown perturbations. This formulation yields the modified updating formula

$$y = \mathcal{M}(\tilde{w}, \mathbf{d}) + \varepsilon, \quad \varepsilon \sim f_\varepsilon(\varepsilon; \Sigma_\varepsilon) \quad (3.7)$$

We take a normal prior for $\tilde{W}|\Theta = \theta$ centered at $\mathbf{0}$ and proceed with the assumption that the wind speed in the two directions are conditionally independent given $\Theta = \theta$ and share the same variance $0 < \theta < \infty$. Mathematically, we have

$$\tilde{W}|\Theta = \theta \sim N(0, \Sigma_{\tilde{W}}; \theta) \quad \Sigma_{\tilde{W}} = \begin{bmatrix} \theta & 0 \\ 0 & \theta \end{bmatrix} \quad (3.8)$$

and thus

$$\tilde{N}|\Theta = \theta \sim N(0, \theta) \quad \tilde{E}|\Theta = \theta \sim N(0, \theta). \quad (3.9)$$

With the modified formulation as in Eq. (3.7), the full probabilistic model, with a slight abuse of notation, is presented in Eqs. (3.10) to (3.12) for any choice of hyperprior π_Θ ,

$$Y|\tilde{W} = \tilde{w} \sim f_\varepsilon(y - \mathcal{M}(\tilde{w}, \mathbf{d}); \Sigma_\varepsilon) \quad (3.10)$$

$$\tilde{W}|\Theta \sim f_{\tilde{W}|\Theta}(\tilde{w}|\theta) = f_{\tilde{E}|\Theta}(\tilde{e}|\theta)f_{\tilde{N}|\Theta}(\tilde{n}|\theta) \quad (3.11)$$

$$\Theta \sim \pi_\Theta(\theta) \quad (3.12)$$

In Eq. (3.11), we have used the conditional independence of wind directions.

Now, following the outlined workflow for Bayesian inference, experimental data is assimilated by conditioning on the observation y . Bayes' theorem yields the joint posterior of W and Θ up to a normalizing constant on a general form as

$$\begin{aligned} \pi(\tilde{w}, \theta|y) &\propto f_{Y|\tilde{W}, \Theta}(y|\tilde{w}, \theta)f_{\tilde{W}, \Theta}(\tilde{w}, \theta) \\ &= f_\varepsilon(y - \mathcal{M}(\tilde{w}, \mathbf{d}); \Sigma_\varepsilon)f_{\tilde{W}|\Theta}(\tilde{w}, \theta)\pi_\Theta(\theta) \end{aligned} \quad (3.13)$$

It is a common and widely recognized choice to model noise as Gaussian such that ε is distributed as $\mathcal{N}(\mathbf{0}, \Sigma_\varepsilon)$. Assuming independence and equal variance in the components of ε , the noise covariance matrix is given by

$$\Sigma_\varepsilon = \begin{bmatrix} \sigma_\varepsilon^2 & 0 \\ 0 & \sigma_\varepsilon^2 \end{bmatrix} \quad (3.14)$$

Note that the modeler has a choice whether to treat the hyperparameter σ_ε^2 as known or impose a prior $\pi_{\sigma_\varepsilon^2}$ also on this parameter, adding further hierarchical structure and complexity to the probabilistic model. At this stage we assume the former, and from Eq. (3.13) obtain the posterior

$$\begin{aligned} \pi(\tilde{w}, \theta | y) &\propto \exp\left(-\frac{1}{2\sigma_\varepsilon^2} \|y - \mathcal{M}(\tilde{w}, d)\|^2\right) f_{w|\Theta}(w|\theta) \pi_\Theta(\theta) \\ &\propto \exp\left(-\frac{1}{2\sigma_\varepsilon^2} \|y - \mathcal{M}(\tilde{w}, d)\|^2\right) \frac{1}{\theta} \exp\left(-\frac{1}{2\theta} \|\tilde{w}\|^2\right) \pi_\Theta(\theta) \\ &= \frac{1}{\theta} \exp\left[-\frac{1}{2} \left(\frac{1}{\sigma_\varepsilon^2} \|y - \mathcal{M}(\tilde{w}, d)\|^2 + \frac{1}{\theta} \|\tilde{w}\|^2\right)\right] \pi_\Theta(\theta), \end{aligned} \quad (3.15)$$

where $\|\cdot\|$ denotes Euclidan distance. For any choice of prior $\pi_\Theta(\theta)$ the conditional posteriors for \tilde{W} and Θ , respectively are given by

$$\pi(\tilde{w} | y, \theta) \propto \exp\left[-\frac{1}{2} \left(\frac{1}{\sigma_\varepsilon^2} \|y - \mathcal{M}(\tilde{w}, d)\|^2 + \frac{1}{\theta} \|\tilde{w}\|^2\right)\right] \quad (3.16)$$

$$\pi(\theta | y, \tilde{w}) \propto \frac{1}{\theta} \exp\left(-\frac{1}{2\theta} \|\tilde{w}\|^2\right) \pi_\Theta(\theta) \quad (3.17)$$

What remains is to choose a prior for Θ . A commonly considered choice of prior for variances in hierarchical models are inverse-gamma distributions, due to its conditional conjugate properties and potential non-informativeness [32]. Looking closer at Eq. (3.17) one can note that an Inverse Gamma prior parameterized as

$$X \sim IG(\alpha, \beta) \implies f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-(\alpha+1)} e^{-\frac{\beta}{x}} \quad x > 0, \quad (3.18)$$

is a conditional conjugate prior as this choice yields a conditional posterior for θ as

$$\begin{aligned} \pi(\theta | y, \tilde{w}) &\propto \frac{1}{\theta} \exp\left(-\frac{1}{2\theta} \|\tilde{w}\|^2\right) \frac{1}{\theta^{\alpha+1}} \exp\left(-\frac{\beta}{\theta}\right) \\ &= \frac{1}{\theta^{\alpha+2}} \exp\left[-\left(\frac{\|\tilde{w}\|^2}{2} + \beta\right) \frac{1}{\theta}\right] \end{aligned} \quad (3.19)$$

From Eq. (3.19) we identify parameters seeing that also the conditional posterior is an Inverse Gamma such that

$$\Theta|Y = y, \tilde{W} = \tilde{w} \sim IG\left(\alpha + 1, \beta + \frac{\|\tilde{w}\|^2}{2}\right) \quad (3.20)$$

Expanded Model

When attributing all impact point displacement to wind uncertainty only, the resulting estimate of σ_{wind} is likely to be very conservative. In order to obtain less conservative and more realistic results, the model in the preceding section is expanded in the following, by attributing output dispersion to multiple sources of uncertainties. By the principles of Bayesian IUQ, this amounts to representing the corresponding model inputs as stochastic variables.

Let $\mathbf{x} \in \mathbb{R}^m$ denote forward model input perturbations that take on launch specific values, subject to a variability that is unknown. In the context of the minimal model, the wind disturbance in east direction \tilde{E} now corresponds to one of the components of the random vector \mathbf{X} that \mathbf{x} is a realization of. Furthermore, we increase the dimensionality of our hyperparameter space such that $\boldsymbol{\theta} \in \mathbb{R}_+^k$. Every component θ_j of $\boldsymbol{\theta}$ governs the dispersion in a block X_j of \mathbf{X} ; equivalently to when we had $\theta \in \mathbb{R}_+$ parameterizing the distribution of the two dimensional wind disturbance vector \tilde{W} . We denote the dimension of the j :th block of \mathbf{X} by m_j . Again, we put a structural prior $f_{\mathbf{X}|\boldsymbol{\theta}}$ on \mathbf{X} conditioned on $\boldsymbol{\Theta}$ to incorporate our prior beliefs on the variability in \mathbf{x} . As before, let \mathbf{d} denote deterministic inputs. The full posterior can be readily derived via Eq. (3.7) and Bayes' rule to a general model of arbitrary finite dimension. Up to a normalizing constant, the joint posterior of $(\mathbf{X}, \boldsymbol{\Theta})$ given the observation \mathbf{y} is given by

$$\begin{aligned} \pi(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) &\propto f_{Y|\mathbf{X}, \boldsymbol{\Theta}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) f_{\mathbf{X}|\boldsymbol{\Theta}}(\mathbf{x}, \boldsymbol{\theta}) \\ &= f_\varepsilon(\mathbf{y} - \mathcal{M}(\mathbf{x}, \mathbf{d}); \Sigma_\varepsilon) f_{\mathbf{X}|\boldsymbol{\Theta}}(\mathbf{x}|\boldsymbol{\theta}) \pi_{\boldsymbol{\Theta}}(\boldsymbol{\theta}). \end{aligned} \quad (3.21)$$

By assuming conditional independence between the blocks X_j of \mathbf{X} given $\boldsymbol{\Theta}$ and independence of the individual hyperparameters Θ_j , the prior $f_{\mathbf{X}|\boldsymbol{\Theta}}$ factorizes and Eq. (3.21) takes the form

$$\pi(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) \propto f_\varepsilon(\mathbf{y} - \mathcal{M}(\mathbf{x}, \mathbf{d}); \Sigma_\varepsilon) \prod_{j=1}^k f_{X_j|\Theta_j}(x_j|\theta_j) \pi_{\Theta_j}(\theta_j) \quad (3.22)$$

Finally, we construct the posterior when considering all available data. For multiple observations, in our case launches, we will differentiate between different data points with a superscript. For N observations, i.e. launches, the statistical model becomes

$$\mathbf{y}^i = \mathcal{M}(\mathbf{x}^i, \mathbf{d}^i) + \boldsymbol{\varepsilon}^i, \quad \boldsymbol{\varepsilon}^i \sim f_{\boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon}; \Sigma_{\boldsymbol{\varepsilon}^i}), \quad i = 1, \dots, N \quad (3.23)$$

To simplify notation, let a set $\{\mathbf{z}^i\}_{i=1}^N$ be denoted by $\langle \mathbf{z}^i \rangle$. Assuming independent observations, the derived posterior in Eq. (3.21) can readily be extended to include the N datapoints. Furthermore, we will assume homoscedasticity such that $\Sigma_{\boldsymbol{\varepsilon}^i} = \Sigma_{\boldsymbol{\varepsilon}}$ for all i . The joint posterior of $\langle \mathbf{X}^i \rangle$ and $\boldsymbol{\Theta}$ given all observations, up to a normalizing constant becomes

$$\pi(\langle \mathbf{x}^i \rangle, \boldsymbol{\theta} | \langle \mathbf{y}^i \rangle) \propto \left(\prod_{i=1}^N f_{\boldsymbol{\varepsilon}}(\mathbf{y}^i - \mathcal{M}(\mathbf{x}^i, \mathbf{d}^i); \Sigma_{\boldsymbol{\varepsilon}}) f_{\mathbf{X}^i | \boldsymbol{\Theta}}(\mathbf{x}^i | \boldsymbol{\theta}) \right) \pi_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \quad (3.24)$$

Recall that in the end, $\boldsymbol{\theta}$ is the QoI. The marginal posterior distribution for $\boldsymbol{\theta}$ is obtained by integrating out the nuisance variables.

$$\pi(\boldsymbol{\theta} | \langle \mathbf{y}^i \rangle) = \int \pi(\langle \mathbf{x}^i \rangle, \boldsymbol{\theta} | \langle \mathbf{y}^i \rangle) d\langle \mathbf{x}^i \rangle \quad (3.25)$$

From now on, for a vector $\mathbf{z} \in \mathbb{R}^k$ or set $\langle \mathbf{z}^i \rangle$ we will use a negative subscript to denote the exclusion of a component such that $\mathbf{z}_{-j} = [z_1, \dots, z_{j-1}, z_{j+1}, \dots, z_k]$. For conclusiveness, note that the conditional posteriors of X_j^i , $\langle X_j^i \rangle$ and $\boldsymbol{\Theta}_j$ up to normalizing constants are given by

$$\pi(x_j^i | \langle \mathbf{y}^i \rangle, \boldsymbol{\theta}, \langle \mathbf{x}_{-j}^i \rangle) \propto f_{\boldsymbol{\varepsilon}}(\mathbf{y}^i - \mathcal{M}(\mathbf{x}^i, \mathbf{d}^i); \Sigma_{\boldsymbol{\varepsilon}}) f_{X_j | \boldsymbol{\Theta}_j}(x_j^i | \boldsymbol{\theta}_j) \quad (3.26)$$

$$\pi(\langle x_j^i \rangle | \langle \mathbf{y}^i \rangle, \boldsymbol{\theta}, \langle \mathbf{x}_{-j}^i \rangle) \propto \left(\prod_{i=1}^N f_{\boldsymbol{\varepsilon}}(\mathbf{y}^i - \mathcal{M}(\mathbf{x}^i, \mathbf{d}^i); \Sigma_{\boldsymbol{\varepsilon}}) f_{X_j | \boldsymbol{\Theta}_j}(x_j^i | \boldsymbol{\theta}_j) \right) \quad (3.27)$$

$$\pi(\boldsymbol{\theta}_j | \langle \mathbf{y}^i \rangle, \boldsymbol{\theta}_{-j}, \langle \mathbf{x}^i \rangle) \propto \left(\prod_{i=1}^N f_{X_j | \boldsymbol{\Theta}_j}(x_j^i | \boldsymbol{\theta}_j) \right) \pi_{\boldsymbol{\Theta}_j}(\boldsymbol{\theta}_j) \quad (3.28)$$

As previously, we impose a Gaussian structural prior such that $\mathbf{X} | \boldsymbol{\Theta} = \boldsymbol{\theta}$ is distributed as $\mathcal{N}(\mathbf{0}, \Sigma_{\boldsymbol{\theta}})$, where the covariance matrix is on block diagonal form,

$$(\Sigma_{\boldsymbol{\theta}}) = \begin{bmatrix} \Sigma_{\theta_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \Sigma_{\theta_k} \end{bmatrix} \quad (3.29)$$

where each block is a diagonal matrix constructed such that

$$(\Sigma_{\theta_j})_{rs} = \begin{cases} \theta_j, & \text{if } r = s \\ 0, & \text{if } r \neq s \end{cases}, \quad 1 \leq r, s \leq m_j \quad (3.30)$$

For the dispersion parameters, take $\Theta_j \sim IG(\alpha_j, \beta_j)$, assuming independence between Θ_i and Θ_j for all i and j , $1 \leq i, j \leq k$. Inserting the choice of priors in Eq. (3.28) yields the conditional posterior for Θ_j ,

$$\begin{aligned} \pi(\theta_j | \langle \mathbf{y}^i \rangle, \boldsymbol{\theta}_{-j}, \langle \mathbf{x}^i \rangle) &\propto \left(\prod_{i=1}^N f_{x_j | \theta_j}(x_j^i | \theta_j) \right) \pi_{\Theta_j}(\theta_j) \\ &\propto \left(\prod_{i=1}^N \frac{1}{\theta_j^{m_j/2}} \exp \left[-\frac{1}{2\theta_j} \|x_j^i\|^2 \right] \right) \frac{1}{\theta_j^{\alpha_j+1}} \exp \left[-\frac{\beta_j}{\theta_j} \right] \\ &= \frac{1}{\theta_j^{\frac{m_j N}{2} + \alpha_j + 1}} \exp \left[-\frac{1}{\theta_j} \left(\frac{1}{2} \sum_{i=1}^N \|x_j^i\|^2 + \beta_j \right) \right] \end{aligned} \quad (3.31)$$

Identifying parameters in Eq. (3.31), find yet again that the conditional posteriors follow an Inverse Gamma distribution. In particular, noticing that the posterior θ_j 's are independent implying $\pi(\theta_j | \langle \mathbf{y}^i \rangle, \boldsymbol{\theta}_{-j}, \langle \mathbf{x}^i \rangle) \equiv \pi(\theta_j | \langle \mathbf{y}^i \rangle, \langle \mathbf{x}^i \rangle)$, we have

$$\Theta_j | \langle \mathbf{Y}^i \rangle = \langle \mathbf{y}^i \rangle, \langle \mathbf{X}_j^i \rangle = \langle \mathbf{x}_j^i \rangle \sim IG \left(\frac{m_j N}{2} + \alpha_j, \frac{1}{2} \sum_{i=1}^N \|x_j^i\|^2 + \beta_j \right). \quad (3.32)$$

This concludes the formulation of the probabilistic model(s) of rocket dispersion and derivation of the relevant posterior distributions given launch data. It is worth recognizing that this result is a corner stone of this thesis. Recall that the last step in Bayesian data analysis entails evaluating the implications of these distributions. In order to do so, we can not disregard actually computing the posteriors. The entirety of Chapter 4 is dedicated to ways of undertaking this often challenging task.

Model Discrepancy

The question of model discrepancy is highly non-trivial yet important to discuss. Clearly, structural model errors are not explicitly known. The easiest way of treatment is to simply ignore them, which in principle implies an enforcement of equality between real life and the computational model. This assumption might be valid in cases when the impact of model errors are

known to be negligible or small compared to the impact of parameter uncertainties. However, there is a risk of over-fitting, as the parameter estimation will act to match available data [20].

One way to address model discrepancy is to include it in the additive noise in the statistical model. This is the route that we have taken in this study, by the introduction of the Gaussian noise variable ε in Eq. (3.5). Efficiently this implies that we have assumed no structural biases in our forward model. A more elaborate approach is to include an additional variable, unknown discrepancy term $\delta(\mathbf{d})$ such that the model becomes

$$\mathbf{y} = \mathcal{M}(\mathbf{x}, \mathbf{d}) + \delta(\mathbf{d}) + \varepsilon. \quad (3.33)$$

Including δ in the Bayesian model requires a prior model $\pi(\delta(\cdot))$, and a full solution will give a posterior also over δ . In general, the more advanced the method to account for model discrepancy, the higher the complexity of the resulting model. Thus, in the end it is a trade-off between an accurate representation of reality and computational complexity. For future development of the model proposed in this thesis, it could be of interest to include a systematic model discrepancy term like δ . This could enable discovering off-sets in e.g. launcher settings or similar. At this point, we will ignore it in favor of model tractability.

Chapter 4

Bayesian Computation

The full solution to a Bayesian problem revolves around the posterior density. Unfortunately, with non-trivial high-dimensional models, the posterior is seldom known in an explicit closed form. Analytic solutions are a rare occurrence in practice, as complicated or even intractable integrals appear increasingly with model complexity and dimension. In order to do inference, one has to resort to specialized methods that only require the target density up to a normalizing constant. Herein we account for some of the commonly employed techniques in advanced Bayesian computation and how they can be utilized for rocket dispersion parameter estimation. After accounting for probabilistic sampling via Markov Chain Monte Carlo methods in Section 4.1, we elaborate on the use of modal estimations to summarize the posterior density in Section 4.2. Finally, in Section 4.3 we introduce the concept of modularization in Bayesian computation, serving as an alternative when full Bayesian analysis is impractical or even infeasible.

4.1 Markov Chain Monte Carlo

A common approach to Bayesian computation is to do probabilistic sampling of the posterior. Markov Chain Monte Carlo (MCMC) sampling (or Markov Chain simulation) is a historically pivotal method to this end [33]. In the following sections we account for the principles of the Metropolis-Hastings (M-H) algorithm, which serves as the basis for a majority of MCMC-techniques. Lastly, we account for convergence diagnostics of MCMC-methods and discuss the specific challenges that arise from hierarchical, high-dimensional models.

4.1.1 Introduction to Markov Chains

Without diving too deep into the mathematical details of Markov chains, we give a top-level overview of concepts necessary for the purpose of understanding and evaluating MCMC-methods. For a more thorough description, we refer to e.g. [34].

The Random Walk

An important example of a Markov Chain is the *random walk* process. It is also an essential component for the M-H algorithm. The sequence of random variables $\{Z_n\}_{n=0}$ is called a random walk if its evolution satisfies

$$Z_{n+1} = Z_n + \rho_n, \quad (4.1)$$

where ρ_n is a r.v. independent of $\{Z_n\}_{n=0}$. In addition, if the distribution of ρ_n is centered symmetrically around zero, we say that the random walk is symmetric.

Transition Kernel

Informally, the Markov chain is a special kind of stochastic process, the evolution of which depends solely on the current state of the process. In general, one can construct a Markov chain through a starting point Z_0 and defining a *transition kernel* κ . The transition kernel, in the continuous case, is a conditional probability density determining the probability of transitioning from the current state to the next. Allowing some abuse of notation, the chain evolves according to $Z_{n+1} \sim \kappa(Z_{n+1}|Z_n)$.

Stationary Distribution

Markov chains can possess a certain stability property, namely the existence of a unique *stationary probability distribution* π . Stationary is meant in the sense that if $Z_n \sim \pi$, it implies $Z_{n+1} \sim \pi$. This property is crucial in Markov chain simulation. In fact, the fundamental idea of MCMC-methods is to construct a Markov chain with the target distribution as its unique stationary distribution.

Finally, we state a sufficient, but not necessary condition in order to check that a distribution π is a stationary distribution associated with a transition

kernel κ , namely the *detailed balance condition*. We say, that detailed balance with respect to π is fulfilled for a transition kernel κ if it holds that

$$\kappa(x|y)\pi(y) = \kappa(y|x)\pi(x) \quad (4.2)$$

for every (x, y) . The interpretation of Eq. (4.2) is that there is an equal probability of being in the state y and transitioning to x , as it is being in x and going to y [33].

4.1.2 Metropolis-Hastings and Gibbs Sampler

Standard Metropolis-Hastings

Originally presented in [24], the Metropolis-Hastings algorithm is archetypical for Markov chain simulations. It is an iterative sampling algorithm, leveraging the limiting behavior of a cleverly constructed Markov chain. An appealing feature is that the constructed transition kernel only depends on the ratio of the target density evaluated in different states, thus only requires unscaled densities. This explains its frequent appearance in Bayesian computation, when the posterior is known only up to a normalizing constant.

Fundamentally, the algorithm consists of two components; a *proposal kernel* (sometimes referred to as a jumping distribution) p and an acceptance rule α . The idea is to construct a random walk by generating samples from $p(\cdot|Z_n)$ given the current state Z_n , but only accept candidates with probability α in order to adjust convergence to the desired target distribution. The construction of the acceptance probability α defines the M-H algorithm. For a target density π and proposal kernel p , the acceptance probability is defined as

$$\alpha(Z^*|Z_n) := 1 \wedge \frac{\pi(Z^*)p(Z_n|Z^*)}{\pi(Z_n)p(Z^*|Z_n)} \quad (4.3)$$

with the notation $a \wedge b := \min(a, b)$. One can show that this results in a modified transition kernel that fulfills a global detailed condition with regards to the target density π .

Algorithm 1 Metropolis-Hastings algorithm

```

Initialize  $Z_0$ 
for  $n = 0, \dots, N - 1$  do
  Draw  $Z^* \sim p(Z^*|Z_n)$ 
  Accept  $Z_{n+1} \leftarrow Z^*$  with probability  $\alpha(Z^*|Z_n)$ 
  Else, stay in current state  $Z_{n+1} \leftarrow Z_n$ 
end for

```

The resulting sequence $\{Z_n\}_{n=1}^N$ consists of correlated samples, approximately from the target density π [33].

Different types of proposals yield differently behaving Markov chains. A common choice, yielding the symmetric Random Walk Metropolis-Hastings (RWMH) makes use of a certain type of proposal, namely with the characteristic structure $p(Z^*|Z_n) = \tilde{p}(\|Z^* - Z_n\|)$ for some probability density function \tilde{p} (in the continuous case). The symmetry of the proposal reduces Eq. (4.3) to only depend on the target density such that

$$\alpha(Z^*|Z_n) = 1 \wedge \frac{\pi(Z^*)}{\pi(Z_n)} \quad (4.4)$$

The Gibbs Sampler

Although often considered a separate algorithm, the Gibbs sampler can be interpreted as a special case of the M-H algorithm with yet another proposal distribution, see for example [23]. The Gibbs sampler is a popular choice of algorithm for high-dimensional target distributions. It is based on the assumption that the target vector Z can be divided into sub-blocks (Z_1, \dots, Z_k) , where the conditional distributions of the blocks are easy to sample from. For each block, we aim to sample from the conditional distribution given all other blocks. In a Gibbs sweep, we traverse through the blocks sampling from $\pi(Z_j|Z_{-j})$ for $i = 1, 2, \dots, k$. In many statistical applications some, or all of, these conditional distributions are known in closed form thus allow direct sampling. This is due to the fact that many models, including hierarchical, are constructed using a sequence of conditional distributions [23].

Algorithm 2 The Gibbs sampler

```

Initialize  $\mathbf{Z}^0$ 
for  $l = 0, \dots, L - 1$  do
  for  $j = 1, \dots, k$  do
    draw  $Z_1^{l+1} \sim \pi(Z_1^{l+1} | Z_2^l, \dots, Z_k^l)$ 
    draw  $Z_2^{l+1} \sim \pi(Z_2^{l+1} | Z_1^{l+1}, Z_3^l, \dots, Z_k^l)$ 
     $\vdots$ 
    draw  $Z_k^{l+1} \sim \pi(Z_k^{l+1} | Z_1^{l+1}, \dots, Z_{k-1}^{l+1})$ 
  end for
end for

```

In Algorithm 2, the blocks are sampled from in a fixed order. However, it is also possible to do e.g. random scans, where the order of the blocks are randomized in every iteration [35].

Hybrid Sampling Schemes

A more general class of MCMC routines hybridizes the standard M-H algorithm and Gibbs sampler. The target vector is still divided into blocks, and whenever possible one samples directly from the conditional distributions. However, if only an unscaled block conditional is available, one proceeds with an M-H steps for this block specifically. We do not show here that such samplers fulfill a detailed balance condition, but refer to for example [33].

4.1.3 Diagnostics

Monitoring and assuring convergence to the target distribution is crucial in Markov Chain simulations. The notions of *stationarity* and *mixing* are important in evaluating MCMC performance. Loosely speaking, the term mixing refers to how fast the Markov chain converges to its stationary distribution. Moreover, ensuring that chains have reached a stationary distribution at all is required to have convergence.

The iterative nature of the sampling introduces potential difficulties to address. Firstly, the samples in MC simulations are correlated by nature. This makes inferences potentially less precise compared to when completely independent samples are used. Thus, checking the level of correlation is an important part of diagnostics. One way of doing so is by looking at the *autocorrelation function* (ACF) within sequences. A slowly decreasing ACF im-

plies high correlation within the samples. In addition, it is possible to derive an *efficient sample size*, as an estimation of how many independent samples the correlated sequence would correspond to. Secondly, the samples should be representative draws from the target distribution, which presupposes that the simulation has been run for a long enough time in order to prevent early phases of the sequence to influence the final estimates. Discarding the first samples in a sequence is a common procedure in Monte Carlo simulations, referred to as a *warm up* period, or *burn-in*. Primarily, this is a measure to avoid the influence of the starting point. The necessary length of warm up depends on the specific problem, but in general, the slower the mixing, the slower the chain reaches its stationary distribution and the longer the burn-in. It also depends on how close the initialization is to the target distribution. A conservative measure is to always disregard the first half of the sequence, but this is not always necessary.

Visual inspection of trace plot of simulated parameters of interest is a qualitative way of assessing convergence. Although imprecise and inefficient with a large number of parameters, it can give an indication of pathological behavior of a chain by how the parameter space is explored. For example, it yields a way to determine if the chain has reached stationarity or not. It is also useful to study the evolution of distributional summaries like sample mean and/or standard deviation of QoIs.

Gelman-Rubin Convergence Diagnostics

A more sophisticated and quantitative way of assessing convergence, as proposed by Gelman and Rubin [5], is to simulate multiple, independent chains from dispersed starting points. Ideally, the starting distribution from which initial points are drawn should resemble the target distribution, but overly dispersed and cover the support of the distribution. Firstly, the use of multiple sequences lets us evaluate the influence of the starting point. This can for example reveal problems related to multi-modal distributions where chains get stuck in local modes for a long time. We can also visually inspect if all simulated chains have mixed; this is required to have convergence. Secondly, it lets us study within- and between sequence variance. For S simulated sequences of length L , one can estimate the between-sequence variance B and within-sequence variance W for a scalar QoI θ , respectively by

$$B = \frac{L}{S-1} \sum_{s=1}^S (\bar{\theta}_s - \bar{\theta})^2, \quad W = \frac{1}{S} \sum_{s=1}^S \hat{\sigma}_s^2, \quad (4.5)$$

where

$$\bar{\theta}_s := \frac{1}{L} \sum_{l=1}^L \theta_{ls}, \quad \bar{\theta} := \frac{1}{S} \sum_{s=1}^S \bar{\theta}_s, \quad \hat{\sigma}_s^2 := \frac{1}{L-1} \sum_{l=1}^L (\theta_{ls} - \bar{\theta}_s)^2 \quad (4.6)$$

W and B can be used to calculate an estimate of the marginal variance of the quantity of interest denoted by σ_θ^2 via

$$\hat{\sigma}_\theta^2 = \frac{S-1}{S}W + \frac{1}{S}B \quad (4.7)$$

One can show that the estimate in Eq. (4.7) will overshoot the actual variance, assuming that initial points have been drawn from an overdispersed starting distribution, but unbiased under stationarity or as $S \rightarrow \infty$. The estimation W on the other hand, underestimates the target variance for finite S . This stems from the fact that a finite sample has not yet explored the entire support of the target distribution and subsequently, will show less variance. As the number of samples increase, the expectation of W will approach σ_θ^2 . Now, define the *potential scale reduction factor* \hat{R} for each simulated parameter of interest by

$$\hat{R} := \sqrt{\frac{\hat{\sigma}_\theta^2}{W}} \quad (4.8)$$

By the above reasoning, for finite S , it follows that $\hat{R} > 1$. The interpretation of \hat{R} is that large values indicate that the scale/variance of the current distribution estimate is larger than the target one, and continuing the simulation is likely to improve the quality of the sample. As S increases, one expects $\hat{R} \rightarrow 1$. A criteria of convergence is for \hat{R} to be close to 1 for all quantities of interest; a conventional limit is 1.1 [5].

4.1.4 Limitations of Standard MCMC Methods

As discussed in Section 4.1.3, Markov chain samples are correlated by construction. In addition, a hierarchical structure inherently introduces even more correlation. In fact, in multilevel modeling we are deliberately prescribing dependence among the model parameters. The degree of correlation affects mixing of the chains, and play a crucial role in the speed of convergence and sample quality in all MCMC methods. Improving efficiency often translates to the task of reducing sequence auto-correlation. In general this requires highly optimized, advanced sampling schemes due to insufficiencies of the simple standard M-H algorithm. Furthermore, tuning the proposal distribution(s) in a M-H step to have reasonable acceptance rates is difficult, as

is proper initialization. Hamiltonian Monte Carlo (HMC) is a common advanced sampler that has proven itself capable of reducing autocorrelation and significantly improve sampling efficiency. The method leverages results from Hamiltonian mechanics; by introducing auxiliary momentum variables one mitigates issues with slow exploration of the state space stemming from the diffusive random walk properties of M-H sampling [36].

Yet another potentially severe issue of iterative simulations in general, and MCMC in particular, is multi-modality. To obtain a representative and large enough sample from the full posterior we need the chain to traverse the entire support to explore the global properties of the distribution. However, when strong multimodality is present, there is a tendency for the sampler to get stuck in local modes. If modes or high density regions of the posteriors are separated by regions of very low posterior density, extremely long simulations are required as candidates proposed by M-H kernels are almost always discarded based on the acceptance probability. Parallel tempering is a way to remedy issues with sampling from multimodal distributions. In short, tempering acts to rescale the target distribution with a temperature variable T , so that modes are flattened, which in turn facilitates more efficient state space exploration. This is accomplished by embedding the target distribution into augmented state spaces with increased temperatures. For a more detailed description, we refer to [37].

High dimensionality often aggravates all aforementioned difficulties. In addition, increase in model size is associated with more computationally costly algorithms. Unfortunately, there is a substantial risk of facing impractical or even infeasible computational loads when aiming to do full Bayesian inference. In the following sections, we present approaches to Bayesian computation which do not target probabilistic sampling of the full posterior directly, in order to circumvent the limitation with standard MCMC mentioned above.

4.2 Modal Approximations

Finding modes of the posterior, local and potentially global, is a way to begin characterizing a complex and multi-modal distribution. Modes can, and often do, serve as distribution summaries and/or point estimates in statistical applications. They can also serve as a basis for distributional approximation [23]. A central quantity is the *maximum a posteriori* (MAP) estimate; the point in the parameter space corresponding to the global maximum of the

posterior density. The MAP is sometimes referred to as the mode of a distribution, assuming unimodality. In here, we will use the term mode to refer to local maxima of the posterior density function. For the full posterior derived from our model of rocket dispersion in Eq. (3.24), the joint MAP estimator is given by

$$(\langle \hat{\mathbf{x}}^i \rangle, \hat{\boldsymbol{\theta}})_{MAP} = \arg \max_{(\langle \mathbf{x}^i \rangle, \boldsymbol{\theta})} \pi(\langle \mathbf{x}^i \rangle, \boldsymbol{\theta} | \langle \mathbf{y}^i \rangle) \quad (4.9)$$

Maximizing the posterior is equivalent to minimizing the negative log posterior, so Eq. (4.9) can be reformulated as

$$(\langle \hat{\mathbf{x}}^i \rangle, \hat{\boldsymbol{\theta}})_{MAP} = \arg \min_{(\langle \mathbf{x}^i \rangle, \boldsymbol{\theta})} -\log \pi(\langle \mathbf{x}^i \rangle, \boldsymbol{\theta} | \langle \mathbf{y}^i \rangle) \quad (4.10)$$

Moreover, we only care for the maximizing argument and can disregard proportional constants. Taking the logarithm, proportionality up to a normalizing constant now corresponds to discarding additive constants. Thus, in order to find a global mode we seek to solve

$$\begin{aligned} \arg \min_{(\langle \mathbf{x}^i \rangle, \boldsymbol{\theta})} -\log \pi(\langle \mathbf{x}^i \rangle, \boldsymbol{\theta} | \langle \mathbf{y}^i \rangle) = \\ \arg \min_{(\langle \mathbf{x}^i \rangle, \boldsymbol{\theta})} \sum_{i=1}^N \frac{1}{2\sigma_{\epsilon^2}} \|\mathbf{y}^i - \mathcal{M}(\mathbf{x}^i, \mathbf{d}^i)\|^2 \\ + \sum_{j=1}^k \left[\frac{1}{\theta_j} \left(\sum_{i=1}^N \frac{\|\mathbf{x}_j^i\|^2}{2} + \beta_j \right) + \left(\frac{m_j N}{2} + \alpha_j + 1 \right) \log(\theta_j) \right] \end{aligned} \quad (4.11)$$

The task of finding modes is an optimization problem, to which a rich variety of numerical methods can be applied. The choice of an appropriate method depends on the application and required accuracy. We give some examples of simple optimization algorithms that are easy to implement, serving as a baseline routine to begin characterizing the modes of the target distribution. Finding multiple modes can be done by initializing the optimization algorithm at different points in the parameter space. More advanced optimization schemes include deterministic and stochastic, Newton's method and Simulated Annealing being widely used examples of the two types, respectively.

Gradient Descent

Gradient based optimization, and gradient descent type algorithms in particular, has gained popularity through its extensive use in Machine Learning

applications. For an objective function $L(z)$, $z \in \mathbb{R}^k$ for some k , consider the optimization problem of finding z^* such that

$$z^* = \arg \min_z L(z). \quad (4.12)$$

Gradient descent (GD) is an iterative algorithm to find (local) minima, where the iterates are updated in the direction of the negative gradient of the objective function. Pseudo-code for a classic GD scheme is given in Algorithm 3.

Algorithm 3 Gradient descent

```

Initiate  $z_0$ 
for  $n = 0, \dots, N$  do
    Update  $z_{n+1} \leftarrow z_n - \eta_n \nabla L(z_n)$ 
end for

```

The step size η_n , known as learning rate in the ML community, governs the distance covered in the the loss landscape in each update. Tuning the learning rate appropriately is important to assure efficient convergence. In general, it does not have to be constant over iterations [38].

Conditional Maximization

As described in [23], conditional maximization is a simple optimization algorithm that can leverage a known hierarchical structure of the underlying model. In the presence of conditional conjugate prior distributions, the procedure is especially convenient. In each step we aim to decrease the negative log posterior, but we alter only a sub-set of the parameters, keeping the remaining fixed. We do this by considering the conditional distributions, taking steps as to always decrease the conditional negative log posterior of the considered individual sub-set.

4.3 Modular Bayesian Analysis

Limitations in pursuing probabilistic sampling of the full posterior distribution has sparked creativity within the field of Bayesian analysis; particularly in how one can make simplifications or modifications to the posterior distribution to enable inference. Modularization is an approach that has been developed partly to this end. It is a way of partitioning the problem into separate components, or modules, aiming to reduce total computational complexity but also as a way of preventing "suspect" parts of a full system from

influencing the end results too much [39]. After having formulated the full problem and derived a true posterior distribution via Bayes' theorem, one proceeds with modifying or approximating some modules of the full model. Replacing posterior distributions of parameters with Maximum Likelihood estimates as in [20] is an example of such a process.

Herein, we propose a modular approach as an alternative to full Bayesian inference. Ideally, we want a model that does not suffer from the same ignorance as the minimal one, yet mitigates the extreme complexity of the full Bayesian approach to the multidimensional model. The modularization allows finding approximate posterior distributions of the QoIs, although they might be inconsistent with Bayes' theorem. Thus, it still enables probability statements on the QoIs.

Recall that the marginal posterior distribution for θ is obtained by integrating out the nuisance variables. Using the general product rule and assuming independence of θ and $\langle \mathbf{y}^i \rangle$, we obtain

$$\begin{aligned} \pi(\theta|\langle \mathbf{y}^i \rangle) &= \int \pi(\langle \mathbf{x}^i \rangle, \theta|\langle \mathbf{y}^i \rangle) d\langle \mathbf{x}^i \rangle \\ &= \int \pi(\langle \mathbf{x}^i \rangle|\theta, \langle \mathbf{y}^i \rangle) \pi(\theta|\langle \mathbf{y}^i \rangle) d\langle \mathbf{x}^i \rangle \end{aligned} \quad (4.13)$$

$$\begin{aligned} &= \int \pi(\langle \mathbf{x}^i \rangle|\theta, \langle \mathbf{y}^i \rangle) \pi_{\Theta}(\theta) d\langle \mathbf{x}^i \rangle \\ &\approx \pi(\langle \hat{\mathbf{x}}^i \rangle|\theta, \langle \mathbf{y}^i \rangle) \pi_{\Theta}(\theta) \end{aligned} \quad (4.14)$$

The modularization takes place in the last step Eq. (4.14). The conditional posterior of $\langle \mathbf{x}^i \rangle$ is approximated by replacing it with an appropriately chosen point mass. We end this section with a brief discussion on doing full versus modular Bayesian analysis. In general, when possible and there is no reason to believe that sub-parts of the full model are flawed, the full Bayesian approach would probably be preferred from a purist viewpoint. Theoretically, full Bayesian analysis is superior; modifying posterior distributions is not entirely without mathematical unease and should be done carefully, avoiding ad-hoc fixes of inadequate mathematical modeling. However, a key advantage of modularization is the increase computationally tractability, enabling us to extract more information about posterior QoIs than allowed by point estimates.

Chapter 5

Surrogate Modeling

Probabilistic sampling like MCMC or iterative optimization schemes containing the likelihood function rely on a huge number of calls to potentially computationally costly forward model codes, growing with the number of data points. This can make such approaches infeasible in full scale engineering applications. In IUQ, a common way to remedy this issue is to introduce *surrogate* or *meta* models. Such emulators captures the input/output relationship of the forward model to an acceptable accuracy with the purpose of speeding up computations. In fact, in light of the previous section covering modularization, we can think of the surrogate model as a replacement of the module consisting of the deterministic forward model. The surrogate model should be cheap to evaluate, easy to interpret and preferably compatible with the computational framework in place. Common choices of modeling include Polynomial Chaos Expansion [40], Gaussian Process regression [20] and Artificial Neural Networks (ANN) [41, 42]. The focus in this work will be on ANNs, specifically on fully connected feed-forward neural networks (FFNNs). The first section Section 5.1 introduces the concept of NNs, whereas the second Section 5.2 accounts for constructing and evaluating models emulating the rocket simulation software provided by SSC.

5.1 Introduction to Neural Networks

In line with the massive development and progress in the field of machine learning, neural network based solutions to inverse problems in physics and engineering applications have received increasing attention, e.g. in [43, 44]. The generality and adaptability of neural networks as function approximators is an appealing feature. In addition, ANN's enjoy an inherent paral-

lel structure and gradient preserving property. Thus, they are attractive as computational units in parallel algorithms and automatic differentiation schemes, for example in GPU-employed gradient based optimization.

5.1.1 Fully Connected Feed-Forward Neural Networks

The mathematical foundation motivating neural network (NN) approximators of non-linear mappings relies on the Universal Approximation Theorem. With some constraints on regularity, the theorem states that any function can be approximated arbitrarily well by a neural network [45]. The classical form of the theorem considers a simple, multilayer feed-forward network, often referred to as multilayer perceptron (MLP). It is one of the simplest network architectures, often used as a prototypical example of an ANN.

An MLP consists of simple interconnected nodes, or neurons, serving as computational units. Information flows through the network via layers of neurons, where the outputs of the one layer is multiplied with corresponding connection weights and summed to form the input to the nodes in the next layer. Non-linearity is introduced through so called activation functions; a non-linear function mapping the input to output associated with each neuron. The term *fully connected* means that each neuron in a layer is connected to all neurons in the next layer. An illustration of an MLP with an input layer, two hidden layers and an output layer and flow of information is presented in Fig. 5.1.

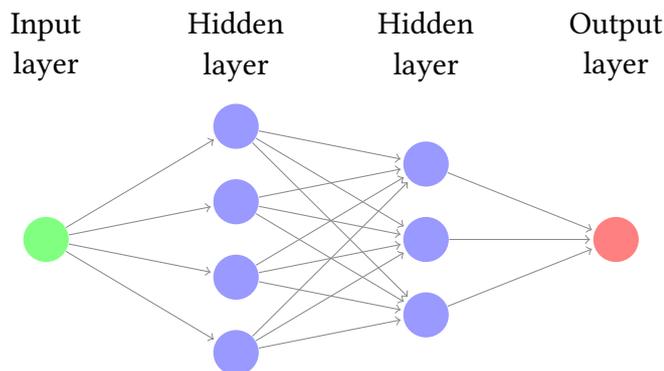


Figure 5.1: Illustration of a two-layer MLP architecture.

Deep learning refers to NNs with a potentially very large number of hidden layers. This kind of architectures has shown immense success in a vast

variety of learning problems, but often pose specific challenges e.g. vanishing/exploding gradients as well as degradation [46]. The use of *skip connections* to redeem these two issues gained popularity with the introduction of ResNet [47]. In a skip connection, also referred to as a residual block, the output of the preceding layer is fed through the next layer but also, directly into the the layer after the block a few layers ahead. A conceptual illustration of a residual block is presented in Fig. 5.2.

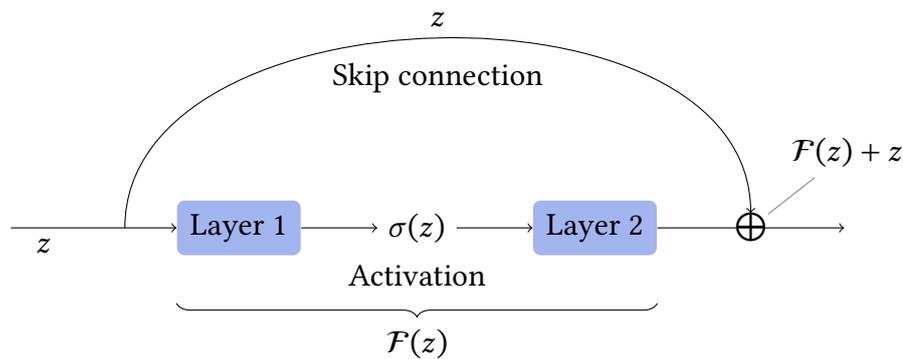


Figure 5.2: Illustration of skipping connections in neural networks. In addition to being fed forward through the succeeding layers in the residual block, the input z feeds directly into the layers after the block.

5.1.2 Training a Neural Network

Training a neural network aims to update the parameters defining the network, θ say, given training data. Supervised learning means that the data has a prescribed structure with input variables, or *features*, $x \in X$ and associated responses or ground truths $y \in Y$.

In order to fit a model, $f_\theta : X \rightarrow Y$, $f_\theta(x) = \hat{y}$, given the data set $\{(x_i, y_i)\}_{i=1}^N$, one defines a loss function $L : Y \times Y$ to quantify the prediction errors of the model. Define the empirical risk R as the sum of losses on the training data by

$$R(\theta) = \sum_{i=1}^N L(y_i, \hat{y}_i) = \sum_{i=1}^N L(y_i, f_\theta(x_i)) \quad (5.1)$$

Common choices of loss functions in regression tasks are L^1 - and L^2 norms, i.e. $L(y, \hat{y}) = \|y - \hat{y}\|^p$, for $p = 1, 2$ respectively. Training the neural net-

work equates to solving the minimization problem of finding θ^* such that

$$\theta^* = \arg \min_{\theta} R(\theta) \quad (5.2)$$

Finding an approximate solution to Eq. (5.2) is almost exclusively done through gradient based optimization schemes, where gradients are obtained via the backpropagating algorithm [48].

5.2 Surrogate Models of Rocket Simulator

In this section, we account for the process of training and evaluating the surrogate models that will be used in the numerical experiments conducted in this thesis. The neural networks emulating the rocket simulation software are built with PyTorch, a python based framework for machine learning and automatic differentiation [49]. It is beyond the scope of this thesis to train a highly optimized neural network; this is indeed a research question on its own. For our purpose, a simple and small scale model to retain interpretability and computational efficiency is sufficient. We settle for an accuracy that is contained within the magnitude of modeled noise in the observations.

The network architecture comprises 12 hidden layers with 200 neurons per layer, with skip connections at hidden layers 4, 6, 8 and 10. This yields a total of 490,002 trainable parameters. All hidden layers are equipped with ReLU activation functions defined by Equation (5.3).

$$\sigma(x) = \max(0, x), \quad x \in \mathbb{R} \quad (5.3)$$

5.2.1 Data Generation

The training and evaluation data is generated by running launch simulations with input variables sampled uniformly from intervals as specified in Table 5.1. The intervals are constructed to cover all relevant launch configurations, based on launches conducted at Esrange. The resulting simulated impact points serve as ground truths when computing the loss in the neural networks.

Aerodynamical responses of a 2-stage rocket like the VSB-30 are more intricate than for one stage rockets like Improved Orion. To accommodate for the more complex target function the network is learning, a larger training

data set of $2 \cdot 10^5$ examples is generated for the former. For the latter, 10^5 examples constitutes the training set.

Variable Description	Unit	VSB-30	Improved Orion
Wind speed, north	[m/s]	[-8,8]	[-8,8]
Wind speed, east	[m/s]	[-8,8]	[-8,8]
Launch elevation, γ	[deg]	[84,90]	[77,88]
Launch azimuth, δ	[deg]	[-50,50]	[-50,50]
Thrust misalignment, pitch	[deg]	[-0.1,0.1]	[-0.2,0.2]
Thrust misalignment, yaw	[deg]	[-0.1,0.1]	[-0.2,0.2]

Table 5.1: Input variable intervals to generate training data for surrogate models.

5.2.2 Training and Evaluation Setup

Models are trained with an L^1 loss function, employing an Adam optimizing algorithm as implemented in PyTorch [49] with an exponentially decaying learning rate starting at $3 \cdot 10^{-4}$ and decay rate 0.9. Gradients are computed on batches of 20 examples for 30 epochs.

To monitor the training dynamics, models are evaluated on an unseen set of 1000 data points every 100:th iteration. Loss logs from training the model associated with the Improved Orion are presented in Figure 5.3. We also introduce a more informative metric when evaluating the trained models on a test data set. For a given tolerance τ , we say that a prediction is correct if the Euclidian distance between the original and surrogate model output, denoted by y and \hat{y} , respectively, is beneath the tolerance: $\|\hat{y} - y\| < \tau$. Accuracy is defined as the fraction of correct predictions on the test set.

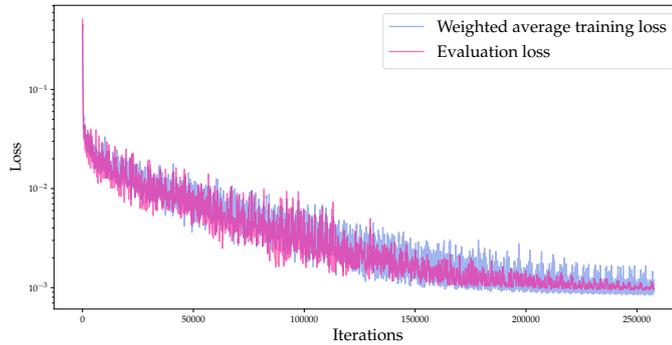


Figure 5.3: Training loss logs for the Improved Orion surrogate model.

5.2.3 Performance

Finally, we convince ourselves that the use of surrogate models is not compromising the end results by not being accurate enough. The trained models are evaluated on test sets of 1000 launches with resulting mean displacements of 1.1 km and 0.54 km for VSB-30 and Improved Orion respectively. Note that the average range of VSB-30 trajectories is considerably larger than for Improved Orion. Thus, we do expect absolute errors to be larger in general for the former; the relative error is comparable. The prediction accuracy for different tolerances is presented in Table 5.2.

τ [km]	VSB-30	Improved Orion
2	0.98	0.99
1	0.8	0.98
0.5	0.36	0.80

Table 5.2: Obtained accuracy with neural network surrogate models of rocket simulation software.

We look closer at how the models perform on examples corresponding to actual launches at Esrange. This is the region in the input variable space that the models will operate within the larger algorithmic framework. A visual comparison between the surrogate models and the original model is presented in Figure 5.4 and shows good resemblance. Quantitatively, all displacements are below 0.95 km and 0.49 km, with a mean of 0.49 km and 0.2 km for VSB-30 and Improved Orion respectively. In all, the accuracy of

the trained models are deemed satisfactory for the purpose of emulating the original simulation model.

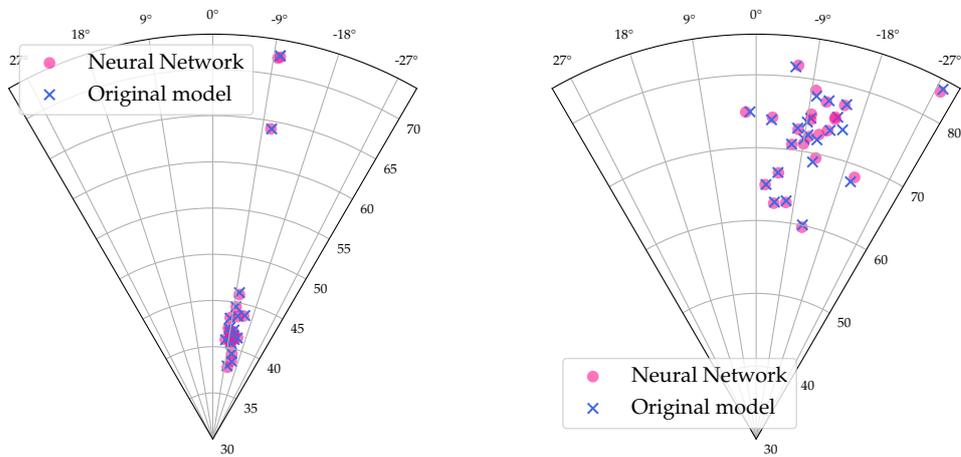


Figure 5.4: Surrogate model predictions with input variables from actual launches. **Left:** Improved Orion. **Right:** VSB-30.

As a final note, we highlight the huge gain in computation time. With the surrogate models, a million input examples can be evaluated in 0.01 seconds, whereas for the original simulation tool a single VSB-30 launch run takes about 1 second. Especially in parallelizable algorithms, we can truly enjoy this improvement.

Chapter 6

Methods and Model Selection

The models of sounding rocket dispersion proposed in Chapter 3 are formulated to be generally applicable to any rocket configuration and launch site. We proceed by narrowing the focus of the thesis by studying the specific case of sounding rockets launched from Esrange. In this chapter, we specify the selection of models and the methods we will employ to obtain the numerical results presented in Chapter 7. This includes specifying hyperparameters and considered variables in the probabilistic model derived in Section 3.3.2, as well as outlining implemented algorithms.

6.1 Choosing Hyperparameters

In this section, we discuss the process and impact of choosing prior hyperparameters. We will treat hyperparameters as deterministic; an alternative option would be to include them in the probabilistic model by prescribing priors over these parameters as well.

6.1.1 Informative Priors on Qols

In this work, we have chosen the route of informative priors. Thus, the distribution $\pi_{\Theta} \in \{\pi_{\Theta}(\cdot; \alpha, \beta) : \Theta \sim IG(\alpha, \beta), \alpha, \beta > 0\}$ should reflect current beliefs about the level of model input uncertainty. We achieve this by choosing α and β accordingly.

In the succeeding numerical experiments, the choice of the hyperparameter β will be based on the currently used individual σ -values in the dispersion estimates. Considering wind as a specific example, the value used at Esrange

is $\sigma_{wind}^2 = 1$ m/s. We proceed by constructing the prior π_{Θ} to have its mean at $\sigma_{wind}^2 = 1$. The expected value of an inverse-gamma is given by $\frac{\beta}{\alpha-1}$. Thus, the scale parameter β is chosen such that $\beta = \sigma_{wind}^2(\alpha - 1)$. We are left with the task of determining the shape parameter α .

The effect of the shape parameter α on the prior distribution is illustrated in Fig. 6.1. A larger α acts to narrow the density around the mean, effectively implying greater confidence in prior beliefs, i.e. current σ -values. Thus, increasing α is a more conservative modeling choice in the sense that it implies a stronger preference toward staying with current σ -values. Conversely, reducing α both shifts the peak of the prior towards smaller values and flattens the density. Essentially, we are reducing the confidence in the current beliefs and putting more probability mass on smaller (and larger) values.

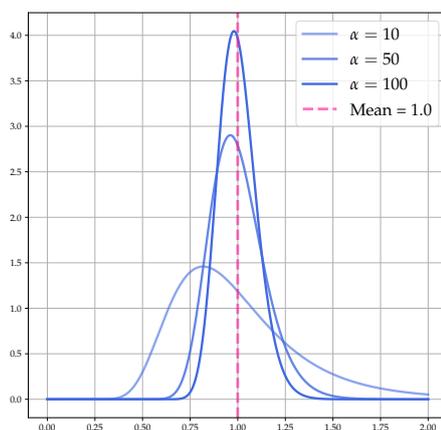


Figure 6.1: Effect of shape parameter α on prior probability density function $\pi_{\Theta}(\theta)$.

Ultimately, deciding the final value to use for inference is up to the practitioner. For some of the model inputs, we have a chance of making a more informed decision based on prior evidence. Specifically, this applies to wind and launch settings; quantities that allow for empirical data collection to support modeling choices. Nonetheless, it is not trivial how to relate the shape parameter to physical quantities, which makes the modeling choice more intricate. In this study, we map subjective beliefs about current σ -values to a choice of α by estimating a lower bound on probable values of the QoI θ . We will denote this limit by σ_{min}^2 , below which there is a negligible probability mass. Thus, subjective beliefs about actual model input uncertainty go

into choosing σ_{min} , after which we define the corresponding α implicitly as the solution to Eq. (6.1). We consider machine precision, denoted by ε , as tolerance for what is deemed numerically negligible.

$$\int_0^{\sigma_{min}^2} \pi_{\Theta_j}(\theta; \alpha) d\theta = \varepsilon_{machine}, \quad \alpha > 0, \quad \varepsilon_{machine} \sim 10^{-16} \quad (6.1)$$

In the next sections, we propose and motivate lower bounds σ_{min} for the variables wind and launch settings. For model inputs that we have no evidence of or reason to have strong beliefs about, we will always prefer a conservative choice and take $\alpha = 100$.

Prior knowledge: Wind

A major contribution to the uncertainty in wind is that final launch settings are not based on wind measurements at $T = 0$, i.e. the (instantaneous) time of launch. Instead, data from a few minutes before serve as basis when delivering the final launcher elevation and azimuth angles. The resulting discrepancy between accounted for and actual wind is likely to be larger than the wind measurement error, and assumed comparable to the strength of wind gusts during flight. Common practice includes logging a wind measurement also at $T = 0$; we begin to investigate the statistical properties of "at launch" wind discrepancy by looking at this data collection. We make a note that the atmospheric conditions measured at $T = 0$ are still not exactly those experienced by the sounding rocket, as wind is not logged continuously throughout all wind layers. For all launches considered in this study, the largest ever observed discrepancy in ballistic wind pre and at launch is 0.95 m/s, considering both head and cross wind. The absolute arithmetic mean is 0.24 m/s. This is indeed considerably smaller than the current 3σ value of 3 m/s. Recall that σ is commonly interpreted as the standard deviation of a normal distribution, and thus supposed to be a 99.7% limit of all values. Based on this observation, a proposed lower bound on the possible values for wind is $\sigma_{min} = 0.5$ m/s, retaining some conservative margin. The resulting prior for the corresponding choice of α is depicted in Fig. 6.2.

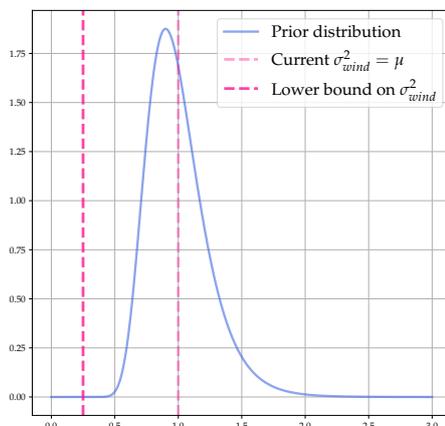


Figure 6.2: Choosing α from a defined lower bound.

Prior knowledge: Launch elevation and azimuth

The launcher is set automatically via commands from the control room, however, it is possible to measure the rail post launch to ensure consistency with actual launch elevation and azimuth angles. Launch operators report that the post-launch measurements of elevation and azimuth have never shown **any** discrepancy to the settings in the system. This supports the belief that current parameters are overshooting, but does not exactly provide a lower bound estimate. However, the setting resolution for the azimuth angle is bounded as it is always rounded to the nearest integer. Based on this, a 3σ below 1° is implausible; thus, we choose σ_{min} as $1/3^\circ$ for launch azimuth. Although elevation is set with one decimal, which would result in $\sigma_{min} = 1/30^\circ$ with the above reasoning, we choose a slightly more conservative lower bound on this variable based on recommendations from launch operators. Instead, the accuracy limit corresponds to one standard deviation; i.e. we set $\sigma_{min} = 1/10^\circ$ for elevation.

6.1.2 Additive Noise

The parameter ε governing the magnitude of additive noise introduced in our statistical model Eq. (3.23) has a clearer physical interpretation. For Improved Orion, we choose $\sigma_\varepsilon = 1$ km in the results presented here. This is enough to include the error introduced by the surrogate model. For VSB-30, we increase the noise to account for a known modeling error discussed in Chapter 7 and increased scale in range and use $\sigma_\varepsilon = 4$ km.

6.2 Variable Selection

In this section, we first conduct a small sensitivity study of the Improved Orion rocket in Section 6.2.1 to conclude which perturbations are contributing most to the dispersion estimates. Based on this, we then specify the models that will be studied in the numerical experiments.

6.2.1 Sensitivity Analysis of Forward Model

We run simulations when perturbing the dispersion factors with their individual 3σ -values and log the impact displacement in downrange and crossrange directions. Figure 6.3 shows a comparison of the absolute value of displacements. It is clear that perturbations to wind is the by far most dominant factor, followed by aerodynamic drag and launch elevation. Thrust misalignment also contributes substantially to the dispersion, and we choose to include it in the multidimensional model. Finally, we will also include launch azimuth, as there is reason to believe that the σ -value associated with this variable is overshooting the actual uncertainty.

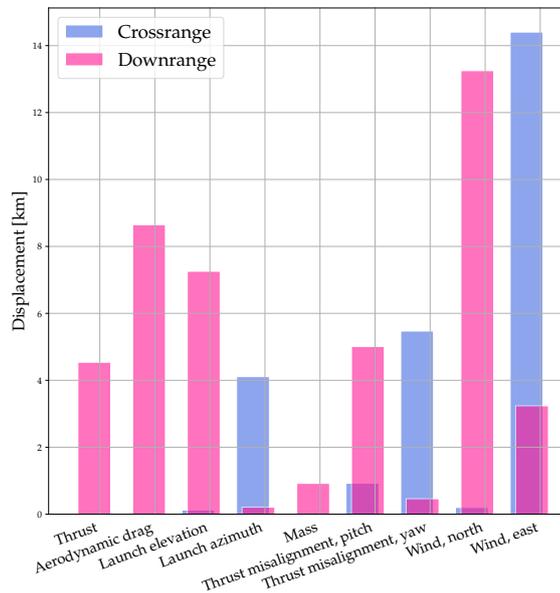


Figure 6.3: Quantifying contributions to dispersion from individual perturbations.

Rocket type	Payload mass	Launch settings	Nominal IP
Improved Orion	100 kg	$\gamma = 82.5^\circ, \delta = 0^\circ$	$r = 58.9 \text{ km}, \alpha = 0.07^\circ$

Table 6.1: Launch configuration for sensitivity study.

Effectively, this results in an input space isomorphic to \mathbb{R}^6 .

6.2.2 Model Specifications

We will study two different models as defined by the following considered configurations:

M.1 According to the minimal model in Section 3.3.2, we only attribute impact point dispersion to uncertainty in wind conditions. We take an inverse gamma prior density on the parameter Θ with hyperparameters α, β according to Section 6.1 and a centered Gaussian structural prior on wind disturbance \tilde{W} given Θ . Note that formally, the minimal model is a special case of the general Eq. (3.24), with number of blocks $k = 1$ with dimension $m_1 = m = 2$ and number of launches considered $N = N_{\text{launch}}$.

M.2 We aim to fit a multidimensional model derived in Section 3.3.2 to available launch data. Only the subset of perturbation factors dominating the dispersion defined in Section 6.2.1 will be considered, in order to reduce the dimensionality of the problem. In Table 6.2, we formalize the configuration of the multidimensional model in accordance with the general model formulation in Section 3.3.2. The variables x_j considered are described in terms of their current individual σ -values and their modeled prior distributions. Hyperparameters α_j, β_j for $\pi_{\Theta_j}(\theta_j)$ are chosen in accordance with the reasoning in Section 6.1.

Variable	Description	σ -value	$f_{X_j \Theta_j}$	α_j	σ_{\min}
$x_1 \in \mathbb{R}^2$	Thrust misalignment	$\sigma_1 = \frac{1}{30}^\circ$	$\mathcal{N}(0, \theta_1 I)$	100	N/A
$x_2 \in \mathbb{R}^2$	Wind disturbance	$\sigma_2 = 1 \text{ m/s}$	$\mathcal{N}(0, \theta_2 I)$	19	$\frac{1}{2} \text{ m/s}$
$x_3 \in \mathbb{R}$	Elevation misalignment	$\sigma_3 = \frac{1}{3}^\circ$	$\mathcal{N}(0, \theta_3)$	5	0.1°
$x_4 \in \mathbb{R}$	Azimuth misalignment	$\sigma_4 = \frac{4}{3}^\circ$	$\mathcal{N}(0, \theta_4)$	3.7	$\frac{1}{3}^\circ$

Table 6.2: Characterization of considered uncertain model inputs in multidimensional model **M.2**.

6.3 Methods for Posterior Exploration

The constructed algorithms that will be utilized for characterizing the joint posterior distribution associated with the selected models are presented in the following subsections. Throughout the numerical experiments conducted in this thesis, the deterministic forward model \mathcal{M} is replaced by a surrogate neural network \mathcal{F} .

6.3.1 MCMC Algorithm

We begin by proposing a M-H based sampling algorithm for the target posterior distributions derived in Section 3.3.2, aiming to implement a M-H within Gibbs algorithm. Recall that there is already a natural block structure in the probabilistic model formulation. We construct a sampler where the j :th block is divided into two Gibbs sampling blocks, one for the set of nuisance variables $\langle x_j^i \rangle$ and one for the dispersion parameter θ_j , for $j = 1, \dots, k$. All conditional posteriors of θ_j can be sampled directly from the derived distribution in Eq. (3.31) due to conditional conjugacy. For the perturbation variables $\langle x_j^i \rangle$, candidates are proposed and accepted/rejected in an M-H step by introducing a symmetric proposal with tuning parameters ρ_j . To denote the l :th iterate, introduce an additional second subscript, i.e. the l :th iterate in the sampling procedure of the j :th component is denoted by $\theta_{j,l}$. We will implement a Random Walk Metropolis Hastings (RWMH) algorithm by drawing candidates according to

$$\langle x_j^i \rangle^* = \langle x_j^i \rangle_k + \rho Z_j, \quad Z_j \sim \mathcal{N}(0, I), \quad (6.2)$$

This yields a symmetric proposal distribution. The sampling routine is summarized in pseudo-code in Algorithm 4.

Algorithm 4 M-H within Gibbs sampler for multidimensional model

```

draw  $\langle x_j^i \rangle_0 \sim N(0, \sigma_j I)$  for  $j = 1, \dots, k$ 
draw  $\theta_{j,0} \sim \pi_{\Theta_j}(\theta)$  for  $j = 1, \dots, k$ 
for  $l = 0, \dots, L - 1$  do
  for  $j = 1, \dots, k$  do
    draw  $\langle x_j^i \rangle^* \sim p(\langle x_j^i \rangle^* | \langle x_j^i \rangle_l)$        $\triangleright$  Draw from proposal distribution
     $P_{accept} \leftarrow \alpha_j(\langle x_j^i \rangle^*, \langle x_j^i \rangle_l)$        $\triangleright$  Compute acceptance probability
    draw  $U \sim U(0, 1)$ 
    if  $U \leq P_{accept}$  then
       $\langle x_j^i \rangle_{l+1} \leftarrow \langle x_j^i \rangle^*$        $\triangleright$  Accept candidate with probability  $P_{accept}$ 
    else
       $\langle x_j^i \rangle_{l+1} \leftarrow \langle x_j^i \rangle_l$        $\triangleright$  Otherwise remain in current state
    end if
    draw  $\theta_{j,l+1} \sim \pi(\theta_j | \langle \mathbf{y}^i \rangle, \langle x_j^i \rangle_{l+1})$        $\triangleright$  Sample directly
  end for
end for

```

To derive explicit acceptance probabilities, we will introduce some additional notation. In the l :th iteration and j :th variable block, denote the derived conditional posterior of $\langle x_j^i \rangle$ by

$$f_j(\langle x_j^i \rangle) := \pi(\langle x_j^i \rangle | \langle \mathbf{y}^i \rangle, \langle x_1^i \rangle_{l+1}, \dots, \langle x_{j-1}^i \rangle_{l+1}, \langle x_{j+1}^i \rangle_l, \dots, \langle x_k^i \rangle_l, \theta_l) \quad (6.3)$$

The model outputs with the proposed candidate and previous candidate, respectively will be denoted by

$$\mathcal{M}_j^{i,*} := \mathcal{M}([\mathbf{x}_{1,l+1}^i, \dots, \mathbf{x}_{j-1,l+1}^i, \mathbf{x}_j^{i,*}, \mathbf{x}_{j+1,l}^i, \dots, \mathbf{x}_{k,l}^i]^T, \mathbf{d}^i) \quad (6.4)$$

$$\mathcal{M}_j^i := \mathcal{M}([\mathbf{x}_{1,l+1}^i, \dots, \mathbf{x}_{j-1,l+1}^i, \mathbf{x}_{j,l}^i, \mathbf{x}_{j+1,l}^i, \dots, \mathbf{x}_{k,l}^i]^T, \mathbf{d}^i) \quad (6.5)$$

With this notation in place and using the symmetry of the proposal, we ex-

and the acceptance probability to obtain

$$\begin{aligned}
 \alpha_j(\langle \mathbf{x}_j^i \rangle^*, \langle \mathbf{x}_j^i \rangle_l) &:= 1 \wedge \frac{f_j(\langle \mathbf{x}_j^i \rangle) p(\langle \mathbf{x}_j^i \rangle_l | \langle \mathbf{x}_j^i \rangle^*)}{f_j(\langle \mathbf{x}_j^i \rangle_l) p(\langle \mathbf{x}_j^i \rangle^* | \langle \mathbf{x}_j^i \rangle_l)} \\
 &= 1 \wedge \frac{f_j(\langle \mathbf{x}_j^i \rangle^*)}{f_j(\langle \mathbf{x}_j^i \rangle_l)} \\
 &= 1 \wedge \frac{\prod_{i=1}^N f_\varepsilon(\mathbf{y}^i - \mathcal{M}_j^{i,*}; \Sigma_\varepsilon) f_{X_j | \Theta_j}(\mathbf{x}_j^{i,*} | \theta_{j,l})}{\prod_{i=1}^N f_\varepsilon(\mathbf{y}^i - \mathcal{M}_{j,l}^i; \Sigma_\varepsilon) f_{X_j | \Theta_j}(\mathbf{x}_{j,l}^i | \theta_{j,l})} \\
 &= 1 \wedge \exp \left[-\frac{1}{2} \left(\frac{1}{\sigma_\varepsilon^2} \left(\sum_{i=1}^N \|\mathbf{y}^i - \mathcal{M}_j^{i,*}\|^2 - \|\mathbf{y}^i - \mathcal{M}_{j,l}^i\|^2 \right) \right. \right. \\
 &\quad \left. \left. + \frac{1}{\theta_{j,l}} \left(\sum_{i=1}^N \|\mathbf{x}_j^{i,*}\|^2 - \|\mathbf{x}_j^i\|^2 \right) \right) \right]
 \end{aligned} \tag{6.6}$$

An estimate of the posterior expectation of Θ by a Monte Carlo estimate from L draws is given by

$$\mathbb{E}[\Theta] = \int \theta \pi(\langle \mathbf{x}^i \rangle, \theta | \langle \mathbf{y}^i \rangle) d\langle \mathbf{x}^i \rangle d\theta \approx \hat{\theta}_N := \frac{1}{L} \sum_{l=1}^L \theta_l \tag{6.7}$$

Conditional Maximization for Modal Approximations

To find (local) maxima of the joint posterior density, we turn to Conditional Maximization as outlined in Section 4.2. Let $L(\langle \mathbf{x}^i \rangle)$ denote the unnormalized form of the conditional posterior distribution of the nuisance variables $\langle \mathbf{x}^i \rangle$, i.e. $L(\langle \mathbf{x}^i \rangle) = \pi(\langle \mathbf{x}^i \rangle | \theta, \langle \mathbf{y}^i \rangle)$ and $l(\langle \mathbf{x}^i \rangle) := -\log L(\langle \mathbf{x}^i \rangle)$. We use a gradient descent scheme to update the nuisance variables $\langle \mathbf{x}^i \rangle$. For θ_j , we simply update each parameter by its conditional mode given all other variables, as it is known in closed form. An example algorithm for our target posterior is outlined in Algorithm 5.

Algorithm 5 Conditional maximization

```

Initiate  $\langle \mathbf{x}^i \rangle_0, \theta_0$ 
for  $n = 0, \dots$  do
    Update  $\langle \mathbf{x}^i \rangle_{n+1} \leftarrow \langle \mathbf{x}^i \rangle_{n+1} - \eta_n \nabla l(\langle \mathbf{x}^i \rangle_n)$ 
    for  $j = 1, \dots, k$  do
        Update  $\theta_{j,n+1}$  to the conditional mode given  $\langle \mathbf{x}^i \rangle_{n+1}$ 
    end for
end for
    
```

6.3.2 Method of Modularization

Finally, we propose a modular Bayesian approach to IUQ for the specific problem of rocket dispersion as accounted for in Section 4.3. By the construction of this method, we aim to mitigate the extreme computational complexity of sampling from the full posterior, while retaining more information compared to doing mere point estimates. The workflow is illustrated in Fig. 6.4.

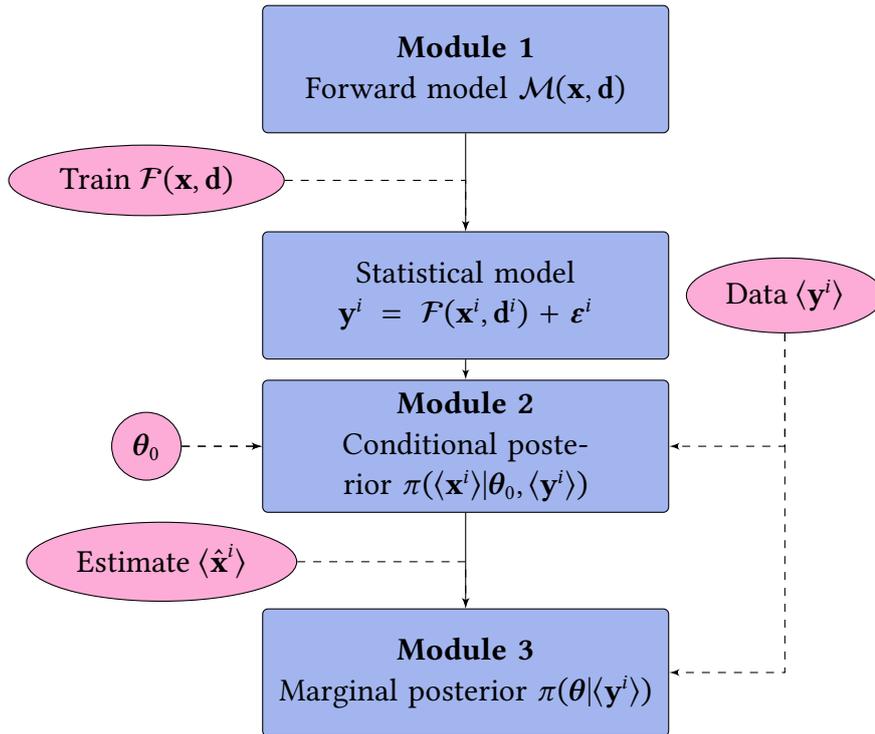


Figure 6.4: Workflow for the modular Bayesian approach. Module 1 replaces the forward model with a neural network. Module 2 approximates the conditional posterior of nuisance variables by a point mass. In Module 3 we sample from the resulting (approximate) marginal posterior of θ .

The first module consists of the deterministic forward model \mathcal{M} that we replace by a surrogate neural network \mathcal{F} . The conditional posterior of $\langle \mathbf{x}^i \rangle$ constitutes the second module; we proceed by approximating it with a point mass at $\langle \hat{\mathbf{x}}^i \rangle$, i.e. $\delta(\langle \mathbf{x}^i \rangle - \langle \hat{\mathbf{x}}^i \rangle)$. Here, we suggest taking a conditional mode as a point estimate. Specifically, for a fixed θ_0 ,

$$\langle \hat{\mathbf{x}}^i \rangle = \arg \max_{\langle \mathbf{x}^i \rangle} \pi(\langle \mathbf{x}^i \rangle | \theta_0, \langle \mathbf{y}^i \rangle) \quad (6.8)$$

The fixed θ_0 is taken as the mean of the prior distribution π_{Θ} . Consequently, by the construction of the prior we can let $\theta_{j,0} = \sigma_j^2$, where σ_j corresponds to the currently use dispersion parameters for the j :th block as specified in Table 6.2.

A salient feature of neural networks is gradient preservation. This makes gradient based optimization algorithms a natural choice and particularly easy to carry out. We use gradient descent (GD), running the algorithm multiple times while sampling initial points from the structural priors $f_{X_j|\Theta_j}$. Gradients are computed via automatic differentiation using PyTorch. We choose the final point estimate with the smallest negative log loss. This serves as the "output" from the second module. Once a point estimate $\langle \hat{\mathbf{x}}^i \rangle$ has been established, we proceed to the third and last module, which is the marginal posterior of θ . We target sampling from Eq. (4.14). Now, the approximate marginal posterior $\pi(\langle \hat{\mathbf{x}}^i \rangle | \theta, \langle \mathbf{y}^i \rangle) \pi_{\Theta}(\theta)$ is very easy to sample from. As a consequence of choosing a conditional conjugate and modeling θ_j as independent from each other, the components can be sampled from their individual posterior inverse gamma distributions by built-in sampling using e.g. SciPy.

Chapter 7

Numerical Results

In this chapter, we present the numerical results obtained with the models and methods specified in Chapter 6 when conditioning on data from actual rocket launches at Esrange. Throughout all experiments, we consider data sets of size $N_{\text{launch}} = 24$ and $N_{\text{launch}} = 22$ for VSB-30 and Improved Orion, respectively.

DISCLAIMER FOR VSB-30 ROCKET SIMULATION MODEL

The torque roll boosters mounted on VSB-30 rockets are unfortunately not implemented in the simulation model that we base our numerical experiments on. This effectively implies that the simulated rocket will be less aerodynamically stable and more sensitive to perturbations, and thus exhibit more dispersion, than a more correct model of the actual launch vehicle would. In the following, we will focus on the smaller rocket Improved Orion for inferences about absolute numerical results. However, theoretically and conceptually, the VSB-30 is still an interesting example to study, and relative results can carry important findings and support conclusions about the developed method as a whole.

7.1 Minimal Model

We solve the minimal model **M.1** introduced in Section 6.2 with the main objective of validating the developed method; the resulting estimates are not of primary interest as the model is expected to yield overly conservative results in terms of the posterior of θ . We sample by means of Algorithm 4. The tuning parameters ρ_j in the M-H random walk proposals are chosen to have average acceptance rates in the range 20 – 40%.

7.1.1 Convergence Diagnostics

We rely on Gelman Rubin diagnostics to monitor convergence and assure that simulated Markov chains behave as expected. In this section, we restrict ourselves to only consider the rocket type Improved Orion. Four parallel chains of length 10^4 are simulated to assess mixing and stationarity. Resulting trajectories of θ and the north component of wind perturbations for the i :th launch denoted by \tilde{n}_i for two of the chains are presented in Fig. 7.1. The wind perturbations $\langle \tilde{w}^i \rangle$ are initialized by draws from centered Gaussian distributions with twice the variance as the structural prior distributions over $\langle \tilde{w}^i \rangle$, i.e. we sample $\langle \tilde{w}^i \rangle_0$ from $\mathcal{N}(0, 2\sigma_{wind}^2)$. The wind variance parameter θ is initialized by draws from its corresponding prior; specifically we sample θ_0 from $IG(\alpha, \beta)$. Visually inspecting the trace plots in Fig. 7.1, we see no signs of anomaly. The dispersively initialized chains seem to have mixed well after a few hundreds of iterations and reached the same stationary distribution, implying convergence.

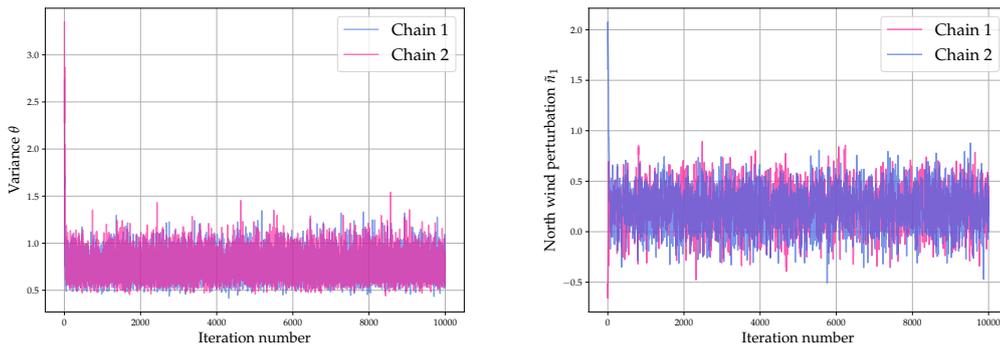


Figure 7.1: Trace plots of two parallel Markov chains. **Left:** Wind variance parameter, θ . **Right:** North component of wind perturbation, \tilde{n}_1 .

To compute the scale reduction factor \hat{R} , each chain is split in half as suggested in [23], resulting in a total of 8 á 5000 samples. The between and within sequence variances B and W are estimated every 100 iteration. We plot the evolution of \hat{R} and the sample mean of θ for increasing number of iterations in Fig. 7.2. We see that the scale reduction factor drops quickly beneath the proposed convergence limit 1.1, implying satisfactory mixing. The posterior mean converges as the number of samples increase and stay virtually constant after approximately 8000 draws.

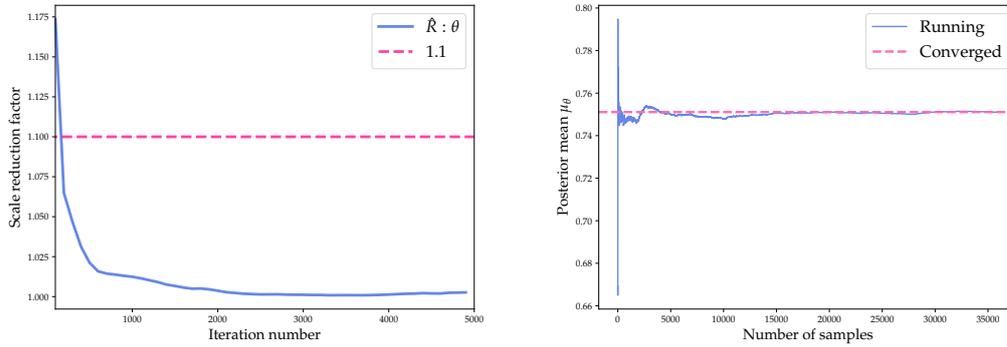


Figure 7.2: Convergence checks. **Left:** Scale reduction factor \hat{R} for θ sequences. **Right:** Convergence of posterior mean μ_θ .

The autocorrelation functions associated with θ and \tilde{n}_1 chains are presented in Fig. 7.3. For both samples, the autocorrelation has dropped to be practically zero by 30 lags. In this context, it is assessed as an acceptable level of mixing speed not impairing the sample quality.

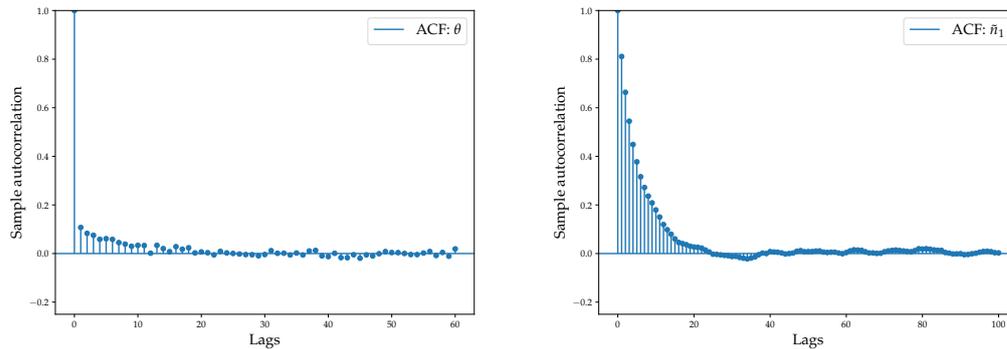


Figure 7.3: Sample Auto Correlation Function. **Left:** θ . **Right:** \tilde{n}_1 .

7.1.2 Resulting Posteriors

Finally, we present posterior results produced by the conservative model. The one stage rocket Improved Orion is at the focal point for most presented results, however, to ensure robustness of the sampler identical simulations are run for the VSB-30 rocket type. Posterior statistical summaries for are presented in Table 7.1.

As a general rule, we disregard the initial 10% of all simulated chains as

warm-up period. With 4 chains, the total number of draws is 36000. In Fig. 7.4, we present marginal posterior samples for our ultimate QoI, the wind variance parameter θ and hyperparameters as specified in Section 6.1. Learning from data has narrowed the prior distribution, implying a decrease in uncertainty in our QoI θ , but also shifted more probability mass towards smaller values.

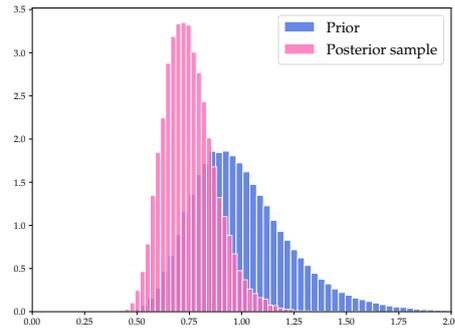


Figure 7.4: Posterior marginal samples of wind perturbation variance θ .

Recalling the discussion on hyperparameters in Section 6.1, for instructive reasons we illustrate the effect on the posterior by presenting results for two additional choices of the prior shape parameter α . Indeed, a larger α yields more conservative results in the sense that the posterior distribution does not alter from the prior in shape or location to the same extent as for smaller α .

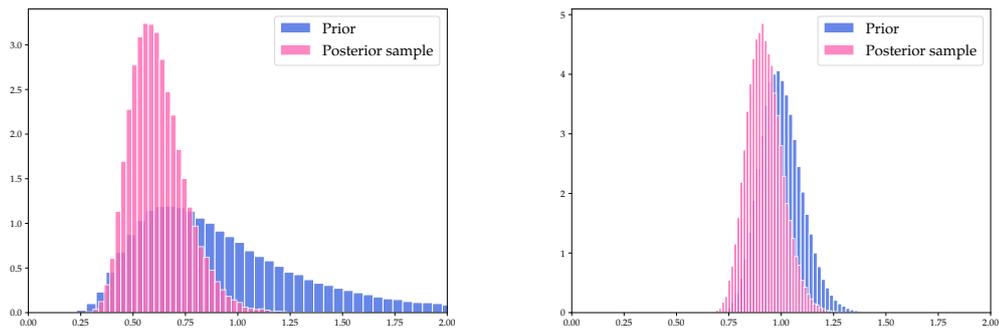


Figure 7.5: Posterior marginal samples for different values of prior hyperparameter α . **Left:** $\alpha = 5$. **Right:** $\alpha = 100$.

Statistical summaries, i.e. sample mean $\bar{\theta}$ and standard deviation SD, of the marginal posterior of θ are presented in Table 7.1 for a selection of α -values.

The final point estimate of the updated wind dispersion parameter, denoted by $\hat{\sigma}_{wind}$, is taken as the posterior mean. We also present the probability that the posterior θ is below the current σ_{wind}^2 . Recall that the prior mean is equal to $\sigma_{wind}^2 = 1$.

For all results studying the Improved Orion, we observe a **decrease** in the dispersion parameter, i.e. $\hat{\sigma}_{wind} < \sigma_{wind}$. The probability of being below the current σ_{wind} ranges from 0.83 to 0.99 for increasing α , the former corresponding to the most conservative choice of hyperparameters in this experiment. These results are particularly remarkable in light of the fact that we are ascribing all impact displacement to wind only. In reality, several additional sources of uncertainty contributes to dispersion, implying that the current estimate of σ_{wind} is indeed likely to be overshooting the actual uncertainty.

α	$\bar{\theta}$	SD	$\hat{\sigma}_{wind} = \sqrt{\bar{\theta}}$	$P(\sqrt{\bar{\theta}} \leq \sigma_{wind})$
Improved Orion				
5	0.619	0.132	0.787	0.990
10	0.680	0.130	0.824	0.981
50	0.866	0.105	0.930	0.892
100	0.922	0.085	0.960	0.828
VSB-30				
5	1.13	0.23	1.063	0.308
10	1.10	0.207	1.053	0.314
50	1.05	0.128	1.024	0.368
100	1.03	0.0948	1.015	0.386

Table 7.1: Statistical summaries of posterior distribution of wind variance θ with minimal model **M.1**.

Regarding the results for the VSB-30 rocket in Table 7.1 we see a small increase in the posterior mean. However, in this case we must consider the disclaimer. We are using a flawed computational model, as it is known not to capture the dynamics of the spin-up motors that are acting to reduce the dispersion. Thus, the absolute numbers for this rocket type are not directly comparable or necessarily representative due to the modeling discrepancy. Nevertheless, the output results seem reasonable given the circumstances,

and do not decrease our confidence in the validity of our approach to uncertainty modeling.

The principal finding from this section is twofold. First and perhaps foremost, the converged results validate the formulated framework and implemented methods, implying both mathematical sanity and algorithmic correctness. Second, the resulting posteriors support the hypothesis that currently estimated model uncertainties, i.e. σ -values, are overly conservative and thus not consistent with reality. Ultimately, this fact further justifies the objectives of this study as a whole. Moving to the next section, we expand the model to include more sources of uncertainty.

7.2 Multidimensional Model

In this section we go beyond the conservative assumption of attributing all uncertainty to wind perturbations. We fit the model **M.2** considering the four most prominent factors in impact dispersion selected in Section 6.2.1. We begin in Section 7.2.1 by attempting to do full Bayesian analysis by sampling from the joint posterior Eq. (3.24). As we will see, the multidimensional model poses challenges for standard MCMC, with bad mixing severely hampering convergence. In Sections 7.2.2 and 7.2.3, we turn to alternative methods to characterize the posterior distribution of θ .

7.2.1 Full Bayesian MCMC sampling

By implementing an MCMC algorithm as outlined in Algorithm 4, we aim to sample from the full posterior in Eq. (3.24). However, the limitations of such an approach becomes evident and illustrates the typical shortcomings of standard MCMC simulations. Specifically the nuisance variable chains $\langle \mathbf{x}^i \rangle$ suffer from severely bad mixing, illustrated in Fig. 7.6 by a trace plot showing Markov Chains that traverse the parameter space extremely slowly. Judging from the auto-correlation function (ACF), we also have strong correlations between samples, as the autocorrelation decreases very slowly with the number of lags.

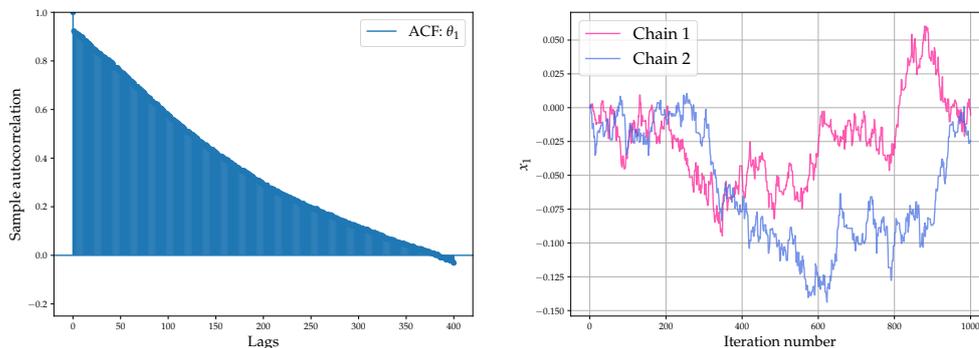


Figure 7.6: Illustrating convergence issues with simple MCMC sampling. **Left:** Autocorrelation function of θ_1 showing highly correlated samples. **Right:** Bad mixing of nuisance variable x_1 .

Even with parallel tempering, details about which can be found in e.g. [37], the implemented sampler fails to exhibit satisfying convergence within a feasible number of iterations. Whether this should be mainly attributed to poor mixing due to correlated samples from the hierarchical model, strong non-linear interactions within the model inputs, multimodality of the full posterior, high dimensionality, or fundamentally too complex a problem for the proposed method remains inconclusive at this point. Most likely, it is a combination of all of the above. The somewhat disappointing results with the naive sampling algorithm are not unexpected however. Developing efficient computational approaches to high dimensional, strongly correlated advanced Bayesian models in real world applications is an active field of research. Thus, it is considered out of the scope of this project to develop a tailored, specialized method to resolve the issues encountered above. However, options to remedy shortcomings of standard MCMC algorithms are definitely part of future improvements of the work herein, and further discussed in Chapter 8. In the subsequent sections, we turn to alternative ways to characterize the posterior distributions of the QoIs that do not involve iterative sampling of the full joint posterior distribution.

7.2.2 Posterior Modes

Abandoning probabilistic sampling from the full posterior, we summarize the target distribution by finding modes. The optimization algorithm in Algorithm 5 is run with starting points from different regions in the input space, sampling initial values from the prior distributions of $\langle \mathbf{x}^i \rangle$ and θ . We use a

constant step size $\eta = 5 \cdot 10^{-5}$ and monitor the decrease of the negative log posterior density function to assure convergence. In Fig. 7.7, we see how the loss decreases and eventually converges within approximately $3 \cdot 10^4$ iterations. A general comment on the results concerns the accuracy of the optimization algorithm. Gradient Descent is a first order approach, in the sense that it only uses information based on the first derivative. Future improvements includes employing e.g. higher order approaches to have faster convergence, such as a Newton-Raphson procedure enjoying quadratic convergence.

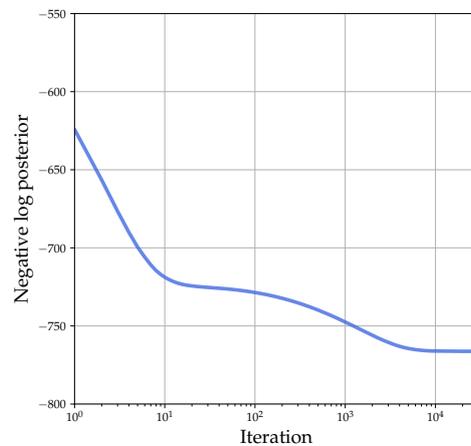


Figure 7.7: Convergence of Conditional Maximization iterations for MAP estimates.

The found mode with the smallest negative log posterior value serves as an approximate MAP estimate; in particular we are interested in the variance parameters θ_{MAP} . Regardless of the moderately scattered starting points, the algorithm always converges to the same mode; whether or not it is a global minimum is not of utmost importance, rather that it is deemed a sufficiently robust estimate and contained within a relevant region of the input parameter space. The resulting estimates are summarized in Table 7.2, showing good but not complete correspondence between the estimates based on data from the two different rocket types. A note of caution is due for the VSB-30 because of the model disclaimer, to which this discrepancy could partly be attributed. However, even with perfect forward models we could not hope for complete agreement due to the relative scarcity of data. Although the Bayesian approach counteracts over-fitting to available launch data, a larger amount of processed experiments will, in principle, always yield results truer to the underlying system in general. Assuming correctness of the overall

modeling and acceptable quality of data, the estimates are expected to converge to a common limit as the number of data points, i.e. launches, grows.

Rocket type	θ_{MAP}
Improved Orion	[0.0009, 0.567, 0.0272, 0.314]
VSB-30	[0.0010, 0.409, 0.0247, 0.287]

Table 7.2: Resulting MAP estimates.

The dispersion parameters associated with θ_{MAP} , denoted by $\hat{\sigma}$ and defined such that

$$\hat{\sigma} := [\hat{\sigma}_1, \hat{\sigma}_2, \hat{\sigma}_3, \hat{\sigma}_4]^T, \quad \hat{\sigma}_j = \sqrt{\theta_{j,MAP}}, \quad (7.1)$$

are compared to their prior values in Fig. 7.8. We emphasize that all posterior estimates of σ_j have decreased relative to the priors. The most significant decrease is for azimuth misalignment. The more conservative prior modeling of thrust misalignment uncertainty leads to a less pronounced decrease compared to the others. The largest difference found when comparing the two rocket types is in posterior wind uncertainty; the reason for which is difficult to infer at this stage, as the VSB-30 model flaw cannot be ruled out as a partial explanation.

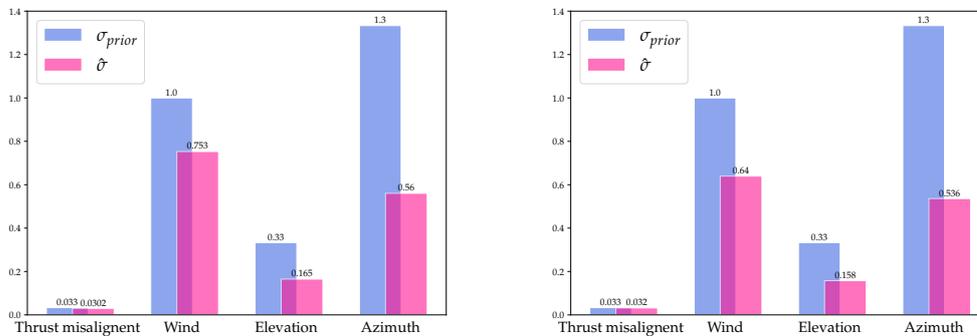


Figure 7.8: The posterior σ -values based on approximate MAP estimates, compared to the prior values. **Left:** Improved Orion. **Right:** VSB-30.

In summary, the results in this section indicate validity of the formulated full probabilistic model, having yielded seemingly reasonable outputs. They also confirm prior beliefs of overly-conservative dispersion parameters. The MAP estimates can serve as first point summaries of the posterior distributions and guidelines if SSC were to update the dispersion parameters for

future purposes. However, exclusively considering point estimates means stepping away from the pure Bayesian way of inference; one of the major draw-backs is that we do not obtain a distribution over the posterior parameters and thus cannot make any probability statements about them. In the next section, we aim to remedy the dissatisfaction of having only point summaries of the posterior distribution associated with the multidimensional model.

7.2.3 Modular Analysis

In here, we present results from taking a modular Bayesian approach as accounted for in Section 6.3.2. With a constant learning rate, convergence dynamics of the Gradient Descent algorithm to find an estimator $\langle \hat{\mathbf{x}} \rangle$ is assured in Fig. 7.9. We also plot the model outputs with $\hat{\mathbf{x}}^i$, $i = 1, \dots, N_{\text{launch}}$.

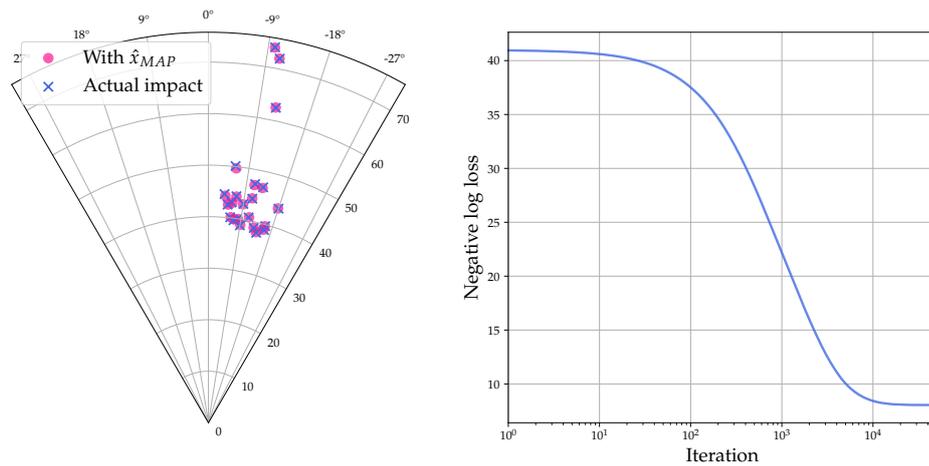
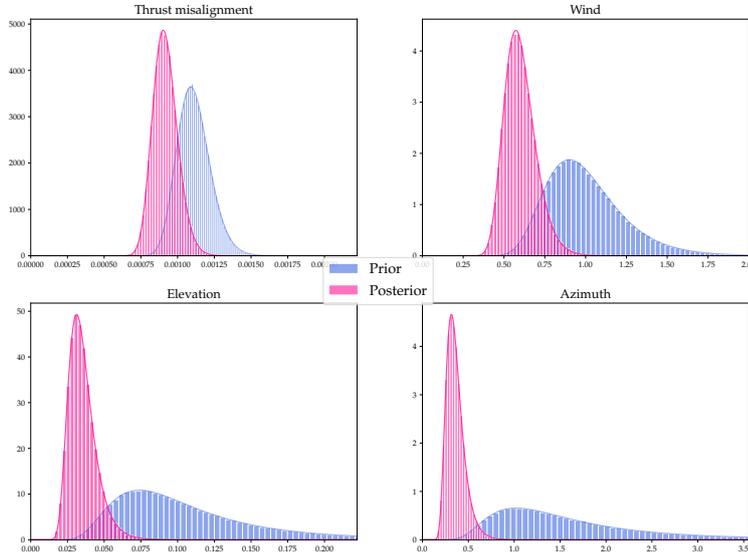


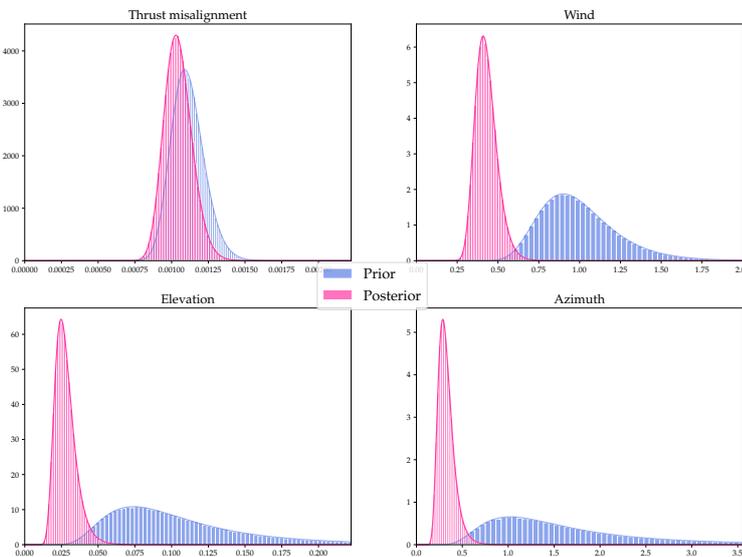
Figure 7.9: Solution to module 1: Conditional MAP estimate of $\langle \mathbf{x}^i \rangle$. **Left:** Forward model outputs $\mathcal{M}(\hat{\mathbf{x}}^i, \mathbf{d}^i)$ and data \mathbf{y}^i . **Right:** Convergence of GD iterations.

Posterior samples of 10^5 draws for the two studied rocket types are presented in Fig. 7.10. Learning from data has decreased the variance and shifted the location towards smaller values in the prior distributions, for all considered model inputs $j = 1, 2, 3, 4$. The impact of the more cautious prior modeling of thrust misalignment uncertainty is clearly illustrated by a smaller shift in associated marginal posterior, compared to parameters modeled with a less distinguished peak around the prior σ -values. The latter exhibits a more pronounced effect of conditioning on observed launches, as the location of the

posterior distributions have translated further away from the priors. These qualitative trends apply to both rocket types; again we see some expected quantitative differences.



Improved Orion



VSB-30

Figure 7.10: Marginal posterior samples of θ with modular approach. All distributions have shifted toward smaller values of θ_j . **Top:** Improved Orion. **Bottom:** VSB-30.

With the modular approach, we have gained more information of the posterior QoIs. In addition to point estimates, probability statements about the updated σ -values are readily available. A brief statistical summary of the resulting distributions is provided in Table 7.3, where the marginal posterior mean and probability that θ_j is below the current σ_j^2 are itemized for all j . The latter quantifies the degree of confidence with which we could decrease the currently used dispersion parameters after conditioning on launch data. The only posterior parameter for which a decrease is not implied with almost 100 % certainty is thrust misalignment based on VSB-30 launches. However, bare in mind that this parameter is highly correlated with the effect of having spin-up motors, and we again refer to the aforementioned simulation model disclaimer to explain this particular deviant result.

Rocket type	$\bar{\theta}$	$\mathbb{P}(\theta_j \leq \sigma_j^2)$
Improved Orion	[0.0009, 0.60, 0.035, 0.36]	[0.98, 0.99, 1.0, 1.0]
VSB-30	[0.0011, 0.43, 0.028, 0.33]	[0.67, 1.0, 1.0, 1.0]

Table 7.3: Resulting posterior summaries with modular approach.

As previously, we take the square root of the posterior means as point estimators, $\hat{\sigma}$, of the dispersion parameters. Resulting $\hat{\sigma}$ are presented in Fig. 7.11 and contrasted to their prior values. We observe a decrease in the posterior $\hat{\sigma}_j$ for all model inputs, for both rocket types. Compared to the previous results based on MAP estimates, we observe minor discrepancies. The overall trend is that MAP estimates result in somewhat smaller posterior estimates. A probable explanation as to why is that we have imposed priors on θ with modes slightly shifted toward smaller values compared to their means. Given that we would only aim to do modal estimates, we might reconsider the way of choosing hyperparameters for the priors on θ . For example, one could choose the prior mode as the current σ -values instead of the expected value as done here. In all, the small differences between modal estimates and the modular approach are encouraging; the consistency supports the validity of the modularization.

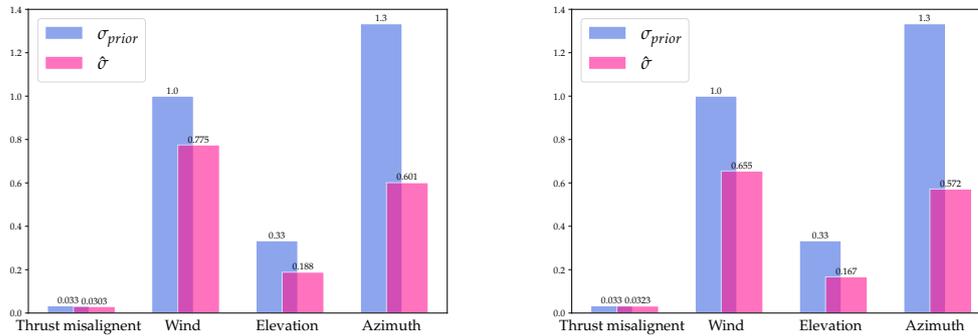


Figure 7.11: Resulting $\hat{\sigma}$ with modular approach. **Left:** Improved Orion. **Right:** VSB-30.

7.3 Posterior Rocket Dispersion Estimates

Once the uncertainty in model inputs has been quantified, i.e. the individual σ -values, we can propagate them forward through the simulation model to obtain an estimate of the output uncertainty, here the theoretical dispersion area. In this section, we present and compare the resulting dispersion areas using the estimated posterior σ -values from three different procedures. First, via MCMC sampling from the minimal model **M.1** via Algorithm 4. For the multidimensional model **M.2**, we compare the methods of MAP estimation in Section 6.3.1 and a modularization as in Section 6.3.2.

The theoretical dispersion is computed by an RSS procedure as outlined in Section 2.3.2. First, we determine benchmark dispersion areas, in terms of the total σ radii, with the currently used dispersion parameters. The launch configurations and results are presented in Table 7.4. Note that these are not consistent with the established theoretical dispersion areas for the two considered rockets at SSC. Small differences are foreseen, as the dispersion depends on the launch settings for the simulations. In addition, a larger difference is expected for the VSB-30, as the simulation model used here is incomplete. In this context, however, the absolute values are not of immediate interest, rather the relative differences when altering the underlying dispersion parameters.

Rocket type	Payload mass	Launch settings	Total σ
Improved Orion	100 kg	$\gamma = 82.5^\circ, \delta = 0^\circ$	5.86 km
VSB-30	396.3 kg	$\gamma = 87.7^\circ, \delta = 0^\circ$	22.1 km

Table 7.4: Launch configuration for benchmark dispersion estimates.

Based on the hyperparameters in Table 6.2, the resulting posterior dispersion parameters are estimated and used to compute the total σ -radius defining the dispersion area for Improved Orion and VSB-30, illustrated in Fig. 7.12. The resulting radii are itemized in Table 7.5

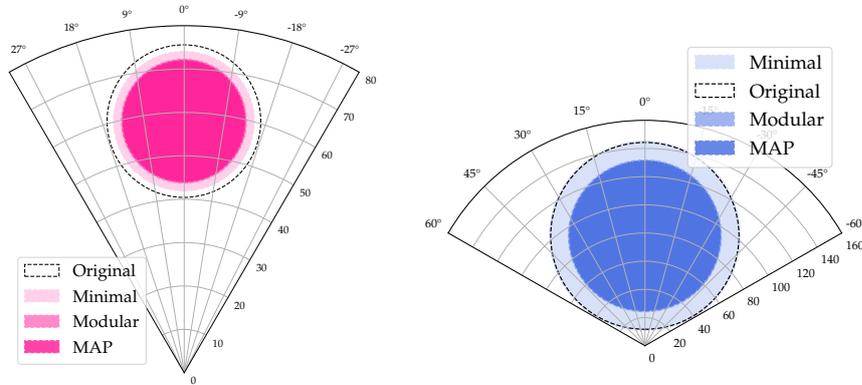


Figure 7.12: Resulting RSS dispersion estimates. **Left:** Improved Orion. **Right:** VSB-30.

Rocket type	Posterior total σ in [km]		
	Minimal	MAP	Modular
Improved Orion	5.36	4.65	4.76
VSB-30	22.39	17.66	17.94

Table 7.5: Resulting total σ after updating individual σ -values.

We relate the computed dispersion areas to the benchmark in Table 7.6. A decrease in σ is observed in all cases based on Improved Orion data, even for the case of the ignorant modeling choice in **M.1** of assuming that only wind causes dispersion, denoted by **Minimal** in the table. The only estimate with

a new dispersion area larger than the benchmark is for the minimal model based on VSB-30 data, although the increase is only with 2%. These results should also be viewed in the light of having a faulty rocket model and that we have increased the noise in the statistical model. Using the MAP estimates yields the largest decrease in dispersion with 21.6% and 19.8% for Improved Orion and VSB-30 data, respectively. This is a direct consequence of the fact that the MAPs resulted in the smallest estimated dispersion parameters as discussed in Section 7.2.3.

Rocket type	Percentage of prior total σ		
	Minimal	MAP	Modular
Improved Orion	91.5 %	79.4 %	81.2 %
VSB-30	102 %	80.2 %	81.5%

Table 7.6: Resulting relative changes in dispersion after updating individual σ -values.

As a concluding remark, we can also relate the results in Table 7.5 to the empirical dispersion associated with the considered data points. These are ~ 15 km and ~ 3.2 km for VSB-30 and Improved Orion, respectively. Note that the latter is smaller than what was presented in Table 2.4. This is because the earliest launches of the Improved Orion were not deemed relevant data points for this study, and thus not used in fitting the model. All posterior estimated areas are larger than the empirical one; we interpret this as a sign of not having over-fitted the parameters to data.

7.3.1 Sensitivity to Hyperparameters

In all the above presented results, we used a fixed set of hyperparameters chosen based on lower bounds on probable values for the dispersion parameters σ_j . In this final section we perform a short sensitivity study to the shape hyperparameter α . Specifically, we conduct experiments scaling the α_j s with up to $\pm 50\%$ of its original value and compute the resulting relative change in σ_j using the modular approach as in Section 7.2.3. From Fig. 7.13, a first observation is that the dependence on α is non-linear. Furthermore, we infer that the launch setting uncertainties exhibit a larger sensitivity to its associated prior shape parameter, azimuth angle showing the most pronounced relative change in resulting $\hat{\sigma}_j$. The explanation for this is likely to be that both elevation and azimuth priors are modeled more aggressively, in

the sense that their absolute α is smaller compared to the other considered perturbation variables.

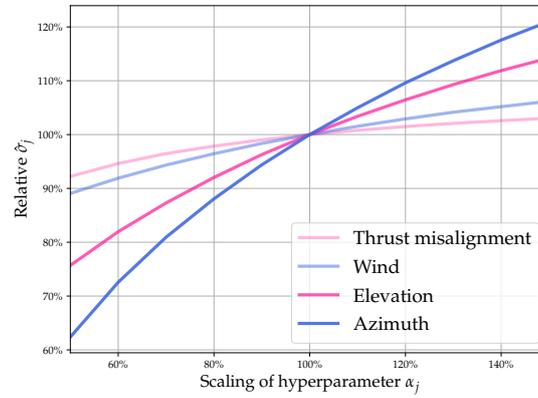


Figure 7.13: Dispersion parameter estimates sensitivity to the choice of prior shape parameter α .

Chapter 8

Conclusions and Outlook

8.1 Conclusions

A statistical method to quantify uncertainty in sounding rocket simulation models based on acquired launch data has been developed. One of the most significant findings to emerge from this study is the reformulation of the problem into the framework of Bayesian analysis. To simultaneously deal with epistemic **and** aleatory uncertainty in the computational model, we have proposed a hierarchical Bayesian approach. It is a versatile and robust way of addressing target model uncertainties on multiple levels, naturally letting practitioners induce regularizing structure to the inverse problem by incorporating prior knowledge about the considered system. We demonstrated the method and presented numerical results based on the operations at SSC; the method however can be readily adapted and specifically tailored to other rocket configurations and launch sites. In addition, we have shown how the use of surrogate models, specifically deep neural networks, in inverse uncertainty quantification circumvents issues with computationally prohibitive simulation codes and facilitates gradient based optimization algorithms. We successfully employed Markov Chain Monte Carlo techniques to sample from the derived posterior distributions associated with a low-dimensional model that attributed all rocket impact point dispersion to a single source of uncertainty. The resulting estimates supported the belief that the currently used dispersion parameters are overly conservative, and served as a first method validation. In the expansion to a more realistic multidimensional model we found the simple MCMC sampler to be inadequate and turned to alternative approaches. In addition to maximum a posteriori estimations, this study also shed light on modular Bayesian approaches and

how such can be utilized in cases where full Bayesian analysis is infeasible. The numerical study implied an up to a 20% possible decrease in impact point dispersion, after updating the dispersion parameters according to the results using the developed method. After conditioning on data, the resulting posterior distributions over dispersion parameters imply that decreasing current values is supported by observations, for all considered input variables.

8.2 Future Work

The process of constructing a mathematical model should always be iterative. Identifying and resolving model inadequacies is therefore a natural topic for future improvements. A particularly relevant matter is that of observed non-mixing in MCMC-sampling of the multidimensional probabilistic model in Section 7.2.1. The limitations of standard MCMC-sampling in high dimensions with correlated variables were discussed in Section 4.1.4, but we should not rule out that the poor performance could also partly be due to some model inadequacy. Although the proposed modular approach seemingly resolves this issue, this hypothesis deserves further investigation, parallel to researching more advanced techniques for posterior distribution exploration. To facilitate full Bayesian analysis, which should be considered the ultimate goal, it is of great interest to implement e.g. a Hamiltonian Monte Carlo scheme to perform probabilistic sampling. This could hopefully also allow for even more complex problem formulations, e.g. increasing dimension to include all dispersion factors. On a related note, the incomplete model for the VSB-30 rocket sparks the discussion of having a more elaborate representation of model discrepancy, as accounted for in Section 3.3.2. A more thorough investigation on this topic is of interest for future purposes.

Further improvements also include examining alternative choices of prior distributions, e.g. the effect of choosing non-informative priors, other parametric families of distributions or even a probabilistic treatment of hyperparameters. Another future topic of research concerns improving performance of surrogates, as well as alternative approaches to meta modeling. The relative scarcity of data does not violate the validity of the presented method, however it can limit the robustness of numerical results. It is therefore desirable to incorporate more data in the pipeline as it becomes available, as well as corroborate results for different rocket types to rule out strong dependencies on the data the model is fit to. Finally, we hope that this study can incite continued research within the formulated Bayesian framework, to

enable reliable and efficient sounding rocket dispersion analysis by abandoning ad-hoc uncertainty estimation in favor of a data driven approach. In the bigger picture, this work adds to the list of fruitful use of multilevel Bayesian inference for uncertainty quantification in engineering applications.

Bibliography

- [1] Timothy John Sullivan. *Introduction to uncertainty quantification*. Vol. 63. Springer, 2015.
- [2] Dave Higdon et al. “A Bayesian approach for parameter estimation and prediction using a computationally intensive model”. In: *Journal of Physics G: Nuclear and Particle Physics* 42.3 (2015), p. 034009.
- [3] Rijan Shrestha and Tomasz Kozłowski. “Inverse uncertainty quantification of input model parameters for thermal-hydraulics simulations using expectation–maximization under Bayesian framework”. In: *Journal of Applied Statistics* 43.6 (2016), pp. 1011–1026.
- [4] Alberto Malinverno and Victoria A Briggs. “Expanded uncertainty quantification in inverse problems: Hierarchical Bayes and empirical Bayes”. In: *Geophysics* 69.4 (2004), pp. 1005–1016.
- [5] Andrew Gelman and Donald B Rubin. “Inference from iterative simulation using multiple sequences”. In: *Statistical science* (1992), pp. 457–472.
- [6] James L Beck, Lambros S Katafygiotis, et al. “Updating models and their uncertainties. I: Bayesian statistical framework”. In: *Journal of Engineering Mechanics-Proceedings of the ASCE* 124.4 (1998), pp. 455–462.
- [7] Tomasz Noga and Rachita Puri. “Microgravity, atmosphere sounding, astronomy, technology validation-an overview of suborbital rockets’ missions and payloads”. In: *International Journal of Space Science and Engineering* 6.2 (2020), pp. 179–208.
- [8] Günther Seibert and Bruce T Battrick. *The history of sounding rockets and their contribution to European space research*. ESA Publications division Noordwijk, 2006.

- [9] Alexandre Garcia et al. “VSB-30 sounding rocket: history of flight performance”. In: *Journal of Aerospace Technology and Management* 3 (2011), pp. 325–330.
- [10] Paul D Wilde. “Range safety requirements and methods for sounding rocket launches”. In: *Journal of space safety engineering* 5.1 (2018), pp. 14–21.
- [11] Robert L James and Ronald J Harris. *Calculation of Wind Compensation for Launching of Unguided Rockets*. National Aeronautics and Space Administration, 1961.
- [12] Commercial Space Transportation. *Supplemental Application Guidance for Unguided Suborbital Launch Vehicles (USLVs)*. Tech. rep. Tech. rep. Federal Aviation Administration (FAA), 2007. url: <https://www...>, 2007.
- [13] *Esrange Safety Manual, SCIENCE-60-4208*. SSC, url: <https://sscspace.com/wp-content/uploads/Esrange-Safety-Manual.pdf>. 2020.
- [14] Tomasz Noga, Maciej Michałow, and Grzegorz Ptasiński. “Comparison of dispersion calculation methods for sounding rockets”. In: *Journal of Space Safety Engineering* 8.4 (2021), pp. 288–296.
- [15] Alan Julian Izenman. “Modern multivariate statistical techniques”. In: *Regression, classification and manifold learning* 10 (2008), pp. 978–.
- [16] Ronald A Fisher. “On the mathematical foundations of theoretical statistics”. In: *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character* 222.594-604 (1922), pp. 309–368.
- [17] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [18] MM Siddiqui. “Statistical inference for Rayleigh distributions”. In: *Journal of Research of the National Bureau of Standards, Sec. D* 68.9 (1964), p. 1007.
- [19] Pauli Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature methods* 17.3 (2020), pp. 261–272.
- [20] Xu Wu et al. “Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian process, Part 1: Theory”. In: *Nuclear Engineering and Design* 335 (2018), pp. 339–355.
- [21] Andrei Nikolaevich Tikhonov et al. *Numerical methods for the solution of ill-posed problems*. Vol. 328. Springer Science & Business Media, 1995.

- [22] Armen Der Kiureghian and Ove Ditlevsen. “Aleatory or epistemic? Does it matter?” In: *Structural safety* 31.2 (2009), pp. 105–112.
- [23] Andrew Gelman et al. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.
- [24] W Keith Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: (1970).
- [25] Adrian FM Smith and Gareth O Roberts. “Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 55.1 (1993), pp. 3–23.
- [26] Dani Gamerman and Hedibert F Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC press, 2006.
- [27] Masoumeh Dashti and Andrew M Stuart. “The Bayesian approach to inverse problems”. In: *Handbook of uncertainty quantification*. Springer, 2017, pp. 311–428.
- [28] Sai Hung Cheung and James L Beck. “Bayesian model updating using hybrid Monte Carlo simulation with application to structural dynamic models with many uncertain parameters”. In: *Journal of engineering mechanics* 135.4 (2009), pp. 243–255.
- [29] Joseph B Nagel. “Bayesian techniques for inverse uncertainty quantification”. In: *IBK Bericht* 504 (2019).
- [30] Jingbo Wang and Nicholas Zabaras. “A Bayesian inference approach to the inverse heat conduction problem”. In: *International journal of heat and mass transfer* 47.17-18 (2004), pp. 3927–3941.
- [31] James L Beck. “Bayesian system identification based on probability logic”. In: *Structural Control and Health Monitoring* 17.7 (2010), pp. 825–847.
- [32] Andrew Gelman. “Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper)”. In: *Bayesian analysis* 1.3 (2006), pp. 515–534.
- [33] Christian P Robert and George Casella. *Monte Carlo statistical methods*. Vol. 2. Springer, 2004.
- [34] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

- [35] Richard A Levine and George Casella. “Optimizing random scan Gibbs samplers”. In: *Journal of Multivariate Analysis* 97.10 (2006), pp. 2071–2100.
- [36] Radford M Neal et al. “MCMC using Hamiltonian dynamics”. In: *Handbook of markov chain monte carlo* 2.11 (2011), p. 2.
- [37] Malcolm Sambridge. “A parallel tempering algorithm for probabilistic sampling and multimodal optimization”. In: *Geophysical Journal International* 196.1 (2014), pp. 357–374.
- [38] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).
- [39] MJ Bayarri, JO Berger, and F Liu. “Modularization in Bayesian analysis, with emphasis on analysis of computer models”. In: *Bayesian Analysis* 4.1 (2009), pp. 119–150.
- [40] Youssef M Marzouk and Habib N Najm. “Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems”. In: *Journal of Computational Physics* 228.6 (2009), pp. 1862–1902.
- [41] Tristan Hauser, Andrew Keats, and Lev Tarasov. “Artificial neural network assisted Bayesian calibration of climate models”. In: *Climate dynamics* 39.1 (2012), pp. 137–154.
- [42] C Balaji and Tamanna Padhi. “A new ANN driven MCMC method for multi-parameter estimation in two-dimensional conduction with heat generation”. In: *International journal of heat and mass transfer* 53.23-24 (2010), pp. 5440–5455.
- [43] Govinda Anantha Padmanabha and Nicholas Zabaras. “Solving inverse problems using conditional invertible neural networks”. In: *Journal of Computational Physics* 433 (2021), p. 110194.
- [44] Kailai Xu and Eric Darve. “The neural network approach to inverse problems in differential equations”. In: *arXiv preprint arXiv:1901.07758* (2019).
- [45] Kurt Hornik. “Approximation capabilities of multilayer feedforward networks”. In: *Neural networks* 4.2 (1991), pp. 251–257.
- [46] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*. Vol. 1. MIT press Cambridge, MA, USA, 2017.

- [47] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [48] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.
- [49] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019).

Appendix A

Supporting Material

A.1 Example Wind Profile

The ballistic wind computed in Eq. (2.2) is based on the wind profile in Appendix A.1. The north wind w_n and w_e components for an altitude h are calculated as the average speed between the positions of two neighboring altitude layers. A positive component implies that the wind is heading to the north or to the east, respectively.

h [m]	w_n [m/s]	w_e [m/s]	h [m]	w_n [m/s]	w_e [m/s]	h [m]	w_n [m/s]	w_e [m/s]
17.50	0.00	-0.20	2502.00	1.80	4.70	12001.31	-0.84	6.94
35.00	2.70	-0.80	3000.00	1.70	6.20	13001.56	-0.47	6.14
55.00	1.40	-1.40	3503.95	0.57	6.17	14001.69	-0.64	6.51
75.00	1.10	-0.80	4000.02	-1.96	4.77	15001.55	0.75	5.52
92.50	2.10	-2.60	5081.10	-4.87	5.29	16000.69	0.04	3.29
105.00	1.70	-1.60	5500.49	-7.28	4.01	17002.47	-0.35	4.04
141.00	1.00	-1.50	6000.23	-5.94	1.38	18001.75	4.38	4.48
301.00	1.00	2.20	6500.56	-2.17	2.52	19000.23	1.51	5.28
452.00	4.20	9.60	7003.43	-4.39	8.03	20000.10	1.29	8.05
600.00	3.80	8.80	7500.13	-5.87	6.34	21001.39	5.33	5.36
750.00	2.60	7.80	8001.31	-5.34	9.03	22000.30	-0.39	7.95
900.00	2.90	7.90	8500.72	-6.54	9.66	23002.13	-1.47	3.78
1201.00	1.20	5.90	9001.48	-6.56	11.27	24000.10	7.98	4.04
1500.00	-1.70	5.40	10000.18	-4.46	8.31	25002.60	3.80	13.13
2001.00	-1.40	3.80	11000.33	-4.65	8.36	28138.00	3.00	11.20

