Layout Optimization Methods for Offshore Wind Farms Using Gaussian Processes

Konstantinos Gkoutis

Master of Science Thesis

TU Delft

Layout Optimization Methods for Offshore Wind Farms Using Gaussian Processes

by

Konstantinos Gkoutis

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday January 30, 2018 at 14:00 PM.

Student number:	4514440	
Project duration:	October 28, 2016 – January 30, 2018	
Thesis committee:	Prof. dr. G. van Bussel,	TU Delft, Group Head, Research Group of Wind Energy
	Dr. ir. E. Quaeghebeur,	TU Delft, PostDoc Researcher, Research Group of Wind Energy
	Dr. R. Dwight,	TU Delft Assistant Professor, Research Group of Aerodynamics

Delft University of Technology

Department of Wind Energy

Aerospace Engineering

The following academic staff certifies that it has read and recommends to the Faculty of Aerospace Engineerging (AE) and Electrical Engineering, Mathematics and Computer Science (EEMCS) for acceptance a thesis entitled

Layout Optimization for Offshore Wind Farms Using Gaussian Processes

by

Konstantinos Gkoutis

in partial fulfillment of the requirements for the degree of Master of Science Sustainable Energy Technology

Dated, January 30, 2018

Head of Wind Energy Department

Prof. Dr. G. Van Bussel

Supervisor:

Dr. ir. E. Quaeghebeur

Reader:

Dr. R. Dwight

Abstract

Surrogate modeling is a family of engineering techniques that attracts great interest today and can be applied in many challenging fields. A big advantage of it is that surrogate models (models based on these techniques) offer reliable results by being computationally cheaper than other candidate models. The savings in computational time is usually paramount for problems that involve a lot of variables and parameters and many iterative processes.

In the wind energy industry in particular, the design of the best layout of the wind farm is a problem that has been presented in the literature as an optimization problem; that is, a problem to optimize the wind farm layout in respect to some objective the modeler deems appropriate. More often than not, maximizing the expected power of the layout is mainly considered as this objective. The layout's expected power is – among other things – heavily dependent on the layout and the wake interactions between the turbines. The iterative search among many layouts to find the best one can be done with the help of a well-known optimization tool, the binary genetic algorithm. However, this tool cannot work alone, it solely facilitates the search over an adequate number of candidate solutions. To make it work, the modeler should provide it with some model that assesses how good in terms of the objective that has been set.

In this thesis therefore, the theory, the development and the use of two models of interest are investigated: Gaussian Process Regression (a surrogate model) and the Monte Carlo Method (a method based on random sampling). Great care was given to compile the theoretical basis of these models in order to be a good reference point for the non-experienced reader. The nature of these two models differs quite a bit, but they both can be used by the modeler to yield interesting results. These results will be compared to each other and against a third model's results, a specific wake model. This third model is the Original Model which the Gaussian Process Regression model and the Monte Carlo Method model utilize and compare against. The reliability of the results and computational speed will be the measure of success and ranking for these three models.

Finally, the comparison of the three models continues in how potent they are to propose an optimized layout for a wind farm. Each of the three models is coupled with the binary genetic algorithm that is developed specifically to connect with them. Afterwards, the proposed best layouts are discussed. The results show that the Gaussian Process Regression model performs reliably and very fast in comparison to the Original model. On the other hand, the Monte Carlo model, although also fast when it is used to find an optimized layout, could not be verified that it performs reliably and therefore, its results cannot be trusted without going into further investigation. After the comparison, further discussion follows with some recommendations for future research.

Acknowledgements

As a token of respect I would like to express my gratitude to some persons that have helped me a lot with my thesis, my courses and in general made my time really enjoyable here in TU Delft. These are the following individuals:

- First and foremost, my parents, Thanasis and Eva, and my brother Bill for their continuous love and support,
- Anna-Maria for always being beside me these two long years,
- my classmates, mostly Panos, Nikos, Ilias and Thekla with whom we formed a beautiful relationship and had each other's back,
- my roommate Rohan for his help that let me continue with my thesis when my laptop broke down,
- my supervisor Postdoc Erik Quaeghebeur for about everything relating to this thesis but mostly because he opened a new door for me in scientific fields totally unexplored from my part and made me, unintentionally of course, confront my fears over my abilities relating to Math,
- professors Bierbooms, Dwight and Van Bussel from TU Delft for sharing their wisdom with me upon my work,
- professor Nikos Mamasis from NTUA for suggesting me to enroll to SET in the first place,
- all of my professors in SET and the the professors in TU Delft Center for Languages and Academic Skills and the Dutch people who made my stay in the Netherlands a lot more than welcome.

Konstantinos Gkoutis
Delft University of Technology
18 September 2017

Contents

Nomenclature

Abbreviations

AP	Average power production of all wind speeds and wind directions without considering the probability weights
BGA	Binary Genetic Algorithm
cdf or $\mathrm{cdf}_{\tilde{x}}$	The probability distribution function of a random variable $\tilde{x}$
cmf or $\mathrm{cmf}_{\tilde{x}}$	The probability mass function of a random variable $\tilde{x}$
FTVP	Fixed Time Varying Performance
FPVT	Fixed Performance Varying Time
MAP	Maximum A Posteriori Estimation
$\mathcal{MC}$	Monte Carlo
MLE	Maximum Likelihood Estimation
$\mathcal{GP}$	Gaussian Process
OM	Original Model
pdf or $\mathrm{pdf}_{\tilde{x}}$	The probability distribution function of a random variable $\tilde{x}$
pmf or $\mathrm{pmf}_{\tilde{x}}$	The probability mass function of a random variable $\tilde{x}$

Greek Symbols

$\tilde{\boldsymbol{\Upsilon}}$	The vector of the hyperparameters for which MAP estimation takes place
$\tilde{\theta}$	Wind direction. Notice it is a random variable with evaluations θ
$\mu_{\tilde{x}}$ or $\mathbb{E}(\tilde{x})$ or μ	The expected value of the random variable $\tilde{x}$
σ_k	The standard deviation of the stochastic process. It is one of the hyperparameters of the extended covariance function of the Gaussian Process
σ_n	Noise standard deviation. It is one of the hyperparameters of the extended covariance function of the Gaussian Process
$\sigma_{\tilde{x}}$ or σ	The standard deviation of the random variable $\tilde{x}$
$\sigma_{\tilde{x}}^2$ or $\mathrm{Var}(\tilde{x})$ or σ^2	The variance of the random variable $\tilde{x}$
$\boldsymbol{\Phi}$	The design matrix of the Gaussian Process Regression. Its elements are column vectors $\boldsymbol{\phi}_i,\ 1 \leq i \leq d$ of some dimensionality d, whose particular expression depends on the input vector(s) and is decided by the modeller
Ω	The sample space of all possible outcomes
ω	A possible outcome of a random experiment, an element of the sample space Ω

Roman Symbols

A	A set A or an event A
$\boldsymbol{A}$	A determinate matrix of arbitrary dimensionality

$|A|$ Cardinality of a set A. When $|A| \in \mathbb{N}$, the set A is finite

$|\boldsymbol{A}|$ or $\det(\boldsymbol{A})$ The determinant of a matrix $\boldsymbol{A}$

$\mathrm{Cov}(\tilde{x}, \tilde{y})$ or $\mathrm{Cov}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ Covariance between two random variables $\tilde{x}$ and $\tilde{y}$ or covariance between two random vectors $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$

$D = \{(x_{+_1}, y_{+_1}), \ldots, (x_{+_{|D|}}, y_{+_{|D|}})\}$ Data set of the training points. The values y_+ are all the evaluations $\boldsymbol{y}_+$ of the sampled partition $\tilde{\boldsymbol{y}}_+$ of the output vector $\tilde{\boldsymbol{y}}$ and the values x_+ are all the elements of the corresponding partition $\boldsymbol{x}_+$ of the input vector $\boldsymbol{x}$

D_t Diameter of the rotor of the wind turbine

$\mathbb{E}(P)$ Expected power production of a specific layout for all wind speeds and wind directions, assuming some joint probability mass function of the wind speeds and wind directions

F^{-1} The inverse function of an invertible function $F: A \mapsto B$, so that $F^{-1}(F(x)) = x$, for all values x of its image B

$F_{\tilde{x}}$ The cumulative probability density function (cdf) of a random variable $\tilde{x}$

$F_{\tilde{x},\tilde{y}}$ The joint cumulative probability density function of two random variables, $\tilde{x}$ and $\tilde{y}$

$f_{\mathcal{W}}(\mathcal{W} = W_\mathrm{sp} | \mathcal{H} = H_\mathrm{sp})$ The probability density function of a sub-collection of continuous random variables $\mathcal{W}$ conditioned by the evaluation of another sub-collection of continuous random parameters as $\mathcal{H} = H_\mathrm{sp}$

$f_{\tilde{w}}(\tilde{w} = w_\mathrm{sp} | \tilde{h} = h_\mathrm{sp})$ The probability density function of a continuous random variable $\tilde{w}$ conditioned by the evaluation of a continuous random parameter $\tilde{h}$ as $h = h_\mathrm{sp}$

$f_{\tilde{x}}$ The probability density function (pdf) of a random variable $\tilde{x}$. This can be expressed also as $\tilde{x} \sim f_{\tilde{x}}$ or $\tilde{x} \sim \mathrm{pdf}_{\tilde{x}}$

$f_{\tilde{x},\tilde{y}}$ The joint probability density function of two random variables, $\tilde{x}$ and $\tilde{y}$

$G_{\tilde{x}}(a) = \inf\{x \in \mathbb{R} : a \leq F_{\tilde{x}}(x)\}$ The generalized inverse distribution of a random variable $\tilde{x}$

$g : [v_\text{cut-in}, v_\text{cut-out}] \times [0°, 360°] \to \mathbb{R}$ The Original Model function which calculates the power production **of a specific layout** for a given wind speed and a given wind direction

$\tilde{\boldsymbol{g}}_+$ A random vector that has been already sampled. The symbol "+" symbolizes sampled quantities

$\tilde{\boldsymbol{g}}_*$ A random vector that has not been sampled. The symbol "*" symbolizes non-sampled quantities

$h_{\mathcal{GP}} : [v_\text{cut-in}, v_\text{cut-out}] \times [0°, 360°] \to \mathbb{R}$ The Gaussian Process Regression model function which calculates the power production **of a specific layout** for a given wind speed and a given wind direction

$h_{\mathcal{MC}} : \mathbb{N} \to \mathbb{R}$ The Monte Carlo model function which calculates the expected power production **of a specific layout** for some random samples of wind speed and wind direction

$\boldsymbol{I}$ The identity matrix whose its diagonal elements are equal to one and the rest are equal to zero

I_A The indicator function of an event A

$\boldsymbol{K}$ or $\boldsymbol{K}(\bullet, \bullet)$ Covariance matrix of two inputs $\bullet$ and $\bullet$. These inputs can be scalars or vectors but should have the same dimensionality

$\tilde{\boldsymbol{K}}_{++}$	The submatrix of the covariance matrix $\boldsymbol{K}$ containing the covariances of all the sampled random vectors
$\tilde{\boldsymbol{K}}_{+*}$ and $\tilde{\boldsymbol{K}}_{*+}$	The submatrices of the covariance matrix $\boldsymbol{K}$ containing the covariances of the sampled random vectors and the non-sampled ones. These partitions are transpose each other's
$\tilde{\boldsymbol{K}}_{**}$	The submatrix of the covariance matrix $\boldsymbol{K}$ containing the covariances of all the non-sampled random vectors
$k(\bullet, \bullet)$	Covariance function for the inputs $\bullet$ and $\bullet$. These inputs can be scalars or vectors but should have the same dimensionality
$k_{\text{extended}}(\bullet, \bullet)$	The extended form of the covariance function for the inputs $\bullet$ and $\bullet$. These inputs can be scalars or vectors but should have the same dimensionality. This form also incorporates the effect of noise
$\mathbb{L}(\,\bullet\,; w_{\text{sp}})$ or $\mathbb{L}(\,\bullet\,; W_{\text{sp}})$	The likelihood function of a parameter $\bullet$ that can be either a determinate or random quantity. The function is parametrized by an evaluation w_{sp} of a random variable $\tilde{w}$ or by an evaluation W_{sp} of a finite sub-collection of random variables $\mathcal{W}$
l	Length scale. It is one of the hyperparameters of the extended covariance function of the Gaussian Process
$\mathbb{N}$	The set of natural numbers
N_{t}	Number of wind turbines in a layout
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Normal Distribution of a random vector $\tilde{\boldsymbol{x}}$ with expected value vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{N}(x; \mu, \sigma)$	Univariate Normal Distribution of a random variable $\tilde{x}$ with expected value μ and variance σ
P	Power production of a specific layout for corresponding to a specific wind speed (at reference height) and a specific wind direction
$\mathbb{P}(A)$	The probability of an event A
$\mathbb{P}(A\|B)$	The conditional probability of an event A given the occurrence of event B
$\mathbb{P}(\tilde{w} = w_{\text{sp}})\|_{h=h_{\text{sp}}}$	The probability of a discrete random variable $\tilde{w}$ parametrized by a determinate parameter h with evaluation $h = h$
$\mathbb{P}(\tilde{w} = w_{\text{sp}} \| \tilde{h} = h_{\text{sp}})$	The probability of a discrete random variable $\tilde{w}$ conditioned by the evaluation of a discrete random parameter $\tilde{h}$ as $h = h_{\text{sp}}$
$\mathbb{R}$	The set of real numbers
$R = \{(x_{*_1}, y_{*_1}), \ldots, (x_{*_{\|R\|}}, y_{*_{\|R\|}})\}$	Data set of the testing points. The random variables $\tilde{y}_*$ are all the elements of the non-sampled partition $\tilde{\boldsymbol{y}}_*$ of the output vector $\tilde{\boldsymbol{y}}$ and the values x_* are all the elements of the corresponding partition $\boldsymbol{x}_*$ of the input vector $\boldsymbol{x}$
$\boldsymbol{T} = [\boldsymbol{x}\ \boldsymbol{y}]^{\top}$	Vector $\boldsymbol{T}$ of the geographical coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$ of a specific wind farm layout.
$U(\tilde{x}; a, b)$ or $U(a, b)$	The uniform distribution of a random variable $\tilde{x}$ with parameters a and b
$\tilde{v}$	Wind speed (at reference height, not at hub height) of a wind turbine. Notice it is a random variable with evaluations v
$v_{\text{cut-in}}$	The cut-in wind speed of a wind turbine. For the Vestas V80 model this is 3 m/s

$v_{\text{cut-out}}$ The cut-out wind speed of a wind turbine. For the Vestas V80 model this is 24 m/s

$\mathcal{X} = \{\tilde{x}_t : t \in T\}$ A stochastic process $\mathcal{X}$ as a collection of random variables $\tilde{x}_t$, indexed by elements t of an ordered set T. If $|T| \in \mathbb{N}$, then $\mathcal{X}$ is a subcollection of a stochastic process and can be written as a random vector $\tilde{\boldsymbol{x}}$. Notice that when the index set is finite, it can be written as a vector $\boldsymbol{t}$. For $\mathcal{GP}$ regression theory, the first vector is also termed the output vector while the second vector is the input vector

$\tilde{x}$ or $\tilde{x}(\omega)$ A random variable $\tilde{x}$ that is related to elementary events ω of the sample space Ω

$\boldsymbol{x}$ A determinate vector of arbitrary dimensionality

$\tilde{\boldsymbol{x}}$ A random vector with some pdf $f_{\tilde{\boldsymbol{x}}}$ which is also the joint pdf of its random variables

$\tilde{\boldsymbol{x}} = [\tilde{\boldsymbol{x}}_A \; \tilde{\boldsymbol{x}}_B]^\top$ A partition of a random vector $\tilde{\boldsymbol{x}}$ into two parts, $\tilde{\boldsymbol{x}}_A$ and $\tilde{\boldsymbol{x}}_B$

$\tilde{\overline{x}}_n$ The sample mean of n random variables. Notice that it is a random variable itself

$\tilde{y} = y(\tilde{x})$ A transformation of a random variable $\tilde{x}$ to the random variable $\tilde{y}$ through the function y

List of Figures

1 Introduction

1.1. Background Information

One of the main functions of every engineer, regardless his specialization, is to design reliable models with the help of a computer. Engineering models require careful considerations and meticulous planning in order to conform with the laws of physics. When for example one has to model the flow of a fluid around an airfoil this cannot be done without considering fundamentals of fluid dynamics theory. Engineering models eventually become computer models which need to be numerically reliable and stable. In essence, more accurate and versatile computer models need to be designed. For example, today's finite-element or finite volume solvers have expanded their computational capabilities in the sense that they can solve for more complex geometries, they can couple with other solvers to pass their results in an input-output relationship and offer a variety of features in reasonable amounts of time [16]. The development of better hardware and software during the last decades is responsible for such progress in computational power. However, it can be the case that even with some clever algorithmic model and some system with high capacity, the computational effort might be too much for the system's resources to manage. Since increasing computational effort means an increasing delay in delivering results, the modeler is usually faced with the dilemma of whether he should use a more simplified model for his problem or not. One approach for this is to start from the beginning and simplify the physics behind the model (we assume that the model concerns some physical process) by linearization of differential equations, omitting the effect of various variables, accepting parameters as constants etc. These approaches of course take their toll on the degree of accuracy of the model, so they include uncertainty about their predictions. If one is not sure about this accompanying uncertainty of these models, he has to test it by running the **original model** to "calibrate" the simplified models.

Another viable option though is to use surrogate models. **Surrogate models** are a well-known application in the engineering world and attract a lot of attention nowadays [24]. Their goal is to *emulate a model* so that they can generate outputs that are statistically sound and reliable in comparison to those of the **original model** [37]. Under proper emulation, they can yield acceptable results that replicate the results of the original model without the need to run this original model directly but rely on simpler and computationally less expensive routines that only generate outputs similar to the original model. The way in which they emulate the original model is the following:

- The original model can be thought of as a mapping of the form $g : A \rightarrow B$ where it accepts inputs $\boldsymbol{x} \in A$ and delivers outputs $\boldsymbol{y} \in B$ (inputs and outputs can be vectors or scalars of course)

- The surrogate model samples some n pairs of inputs and outputs (elements of $\boldsymbol{x}$ and $\boldsymbol{y}$) that uses to form a finite data set $D = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)\}$ (termed the **data set of sampled points**) and then emulates the outputs for all the remaining inputs that were not sampled. The surrogate model *does not compute those outputs* as the original model or the aforementioned simplified models do, but acts as a black-box that simply connects inputs to outputs – it establishes a rule of correspondence between inputs and outputs [16]

This procedure should be simple, fast and reliable so that the surrogate model can cut down computational time. If it is indeed fast and reliable, searches through the search space (parametric or variable sweeps) are easily facilitated [16]. The accuracy now of the surrogate model gets better the larger the aforementioned data set D gets. Accuracy (or fidelity) has to do with how small the error of the outputs of the surrogate model is when compared to the outputs of the original model. This accuracy of the surrogate model, if it is based on statistical methods, is defined within acceptable margins of confidence as shall be seen in Section 3.4. Popular surrogate models are Polynomial Regression, Gaussian Processes (also known as Kriging), Radial Basis Functions [24], Polynomial Chaos Expansions, etc.

Surrogate models can be distinguished in three categories [50]: *interpolation* (or data-fitting), *projection-based* and *hierarchical models*. Interpolation models as said treat the original model as a black-box and act as a "curve-fit" between inputs-outputs based on regression or interpolation theory. An example of this category would be the least squares linear regression model that interpolates between known points in the 2D plane, as seen by comparing two potential surrogate models in Figure 1.1. Projection based models deal with methods that project high-dimensional spaces onto low-dimensional subspaces. Often it is the case that solutions to a high-dimensional problem are contained in a low-dimensional subspace. This technique is also referred as feature extraction when only a few of the dimensions are of value. The third class of models which are named also multi-fidelity models [2] aim primarily to capture trends of the original model and not replicate accurately their results. An example is the well-known linearization of partial differential equations in order to approach the behavior of the model numerically, although numerical resolution may still be quite inaccurate [2]. In this thesis, only the first type of model will be considered. Lastly, just to complete the puzzle, *classification* is also performed by surrogate models, but it does not constitute a fourth category. Classification can be performed with the use of some step function that separates outputs to classes, so it can easily fall under the scope of interpolation models.

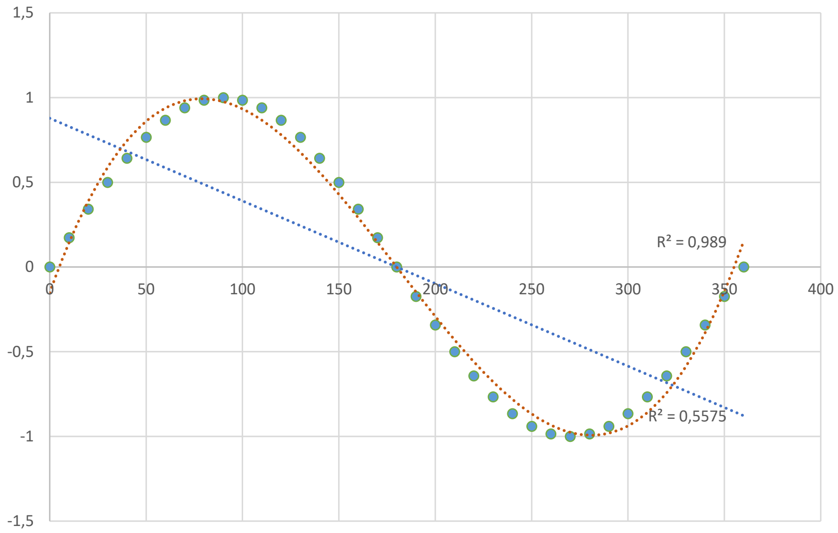


Figure 1.1: Comparing two surrogate models. The green points are the sampled inputs, the blue line is a non-reasonable surrogate model to consider and the orange line is a more reasonable surrogate model to consider. The surrogate modeling method used is called polynomial regression.

Another usage of the surrogates is, when doing regression, to mitigate noise effects by smoothing data through autoregression. The same happens when the meshing size is inadequate. Lastly, surrogate models are used for data mining in order to unveil functional relationships between variables [16].

When examining interpolation models, like the ones in Figure 1.1, both the original model and the surrogate model have a specific similarity: they establish a rule of correspondence, a *functional relationship*, between inputs and outputs, although the surrogate models have first to use the data set D and do not work as the original models do, as explained before. In Figure 1.2 one sees the original model (blue color) establishing the rule of correspondence $g : A \to B$ and on the same time the surrogate model (red color) with the aid of the data set D establishes another rule of correspondence $h_{\mathcal{GP}} : A \to B_{\mathcal{GP}}$, where $B_{\mathcal{GP}}$ might not be the same set as B, because as seen in Figure 1.1, the surrogate model may not emulate the results correctly. This means that the surrogate model may have outputs that have an error so they might not even be elements of the set B. The data set D which we talked about before takes samples of both inputs and outputs from the sets $S_{\text{in}_{\mathcal{GP}}} \subseteq A$ and $S_{\text{out}_{\mathcal{GP}}} \subseteq B$. It is with these subsets that the emulation done by the surrogate model becomes possible. In general, the larger these samples, the fewer errors will be from interpolation and the more $B_{\mathcal{GP}}$ will tend to approach B.

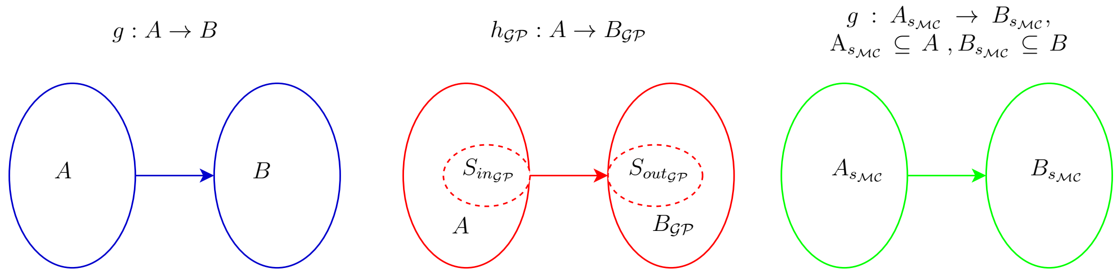

$g : A \to B$
$h_{\mathcal{GP}} : A \to B_{\mathcal{GP}}$
$g : A_{s_{\mathcal{MC}}} \to B_{s_{\mathcal{MC}}}$,
$A_{s_{\mathcal{MC}}} \subseteq A$, $B_{s_{\mathcal{MC}}} \subseteq B$
A
B
$S_{in_{\mathcal{GP}}}$
$S_{out_{\mathcal{GP}}}$
A
$B_{\mathcal{GP}}$
$A_{s_{\mathcal{MC}}}$
$B_{s_{\mathcal{MC}}}$

Figure 1.2: The conceptual map of three models, the original model (blue color), the surrogate model (red color) and the random sampling model (green color).$A_{s_{\mathcal{MC}}}$ is the set of *randomly* sampled values for the random sampling model while the sets $S_{\text{in}_{\mathcal{GP}}}$ and $S_{\text{out}_{\mathcal{GP}}}$ form the training data set D that was referred above for the surrogate model. The fact that in this Figure these three aforementioned sets are portrayed as “continuous” sets is only for illustration purposes – they need not to be that way.

Another totally different approach from the two above as to how one can derive results that relate to the ones of the original model is using **random sampling methods**. In particular, a method of interest is Monte Carlo ($\mathcal{MC}$). Its purpose is to compute through random sampling the *expected output* and the *expected variability* around it. This method differs from the previous ones due to the fact that it does not establish a rule of correspondence for the totality of the domain A, but for a subset of this domain $A_{s_{\mathcal{MC}}} \subseteq A$, as can be seen in Figure 1.2, (green color). This set is the set of the samples one draws *randomly*. $B_{\mathcal{MC}}$ is the subset of B that is computed by the Monte Carlo model. The computation happens with the use of the original model. Monte Carlo models can yield reliable results about what is expected of the output(s) if the number of the samples one draws is sufficiently high. However it is true that, since it is *underrepresenting* the possible outputs, since its set of outputs $B_{\mathcal{MC}}$ is a subset of B, there is always some error involved compared to the expected output of the set B. Without going into the specifics of the $\mathcal{MC}$ method, we should note that it is widely used for approximating the behavior of models, as do the surrogate models.

1.2. Problem Analysis

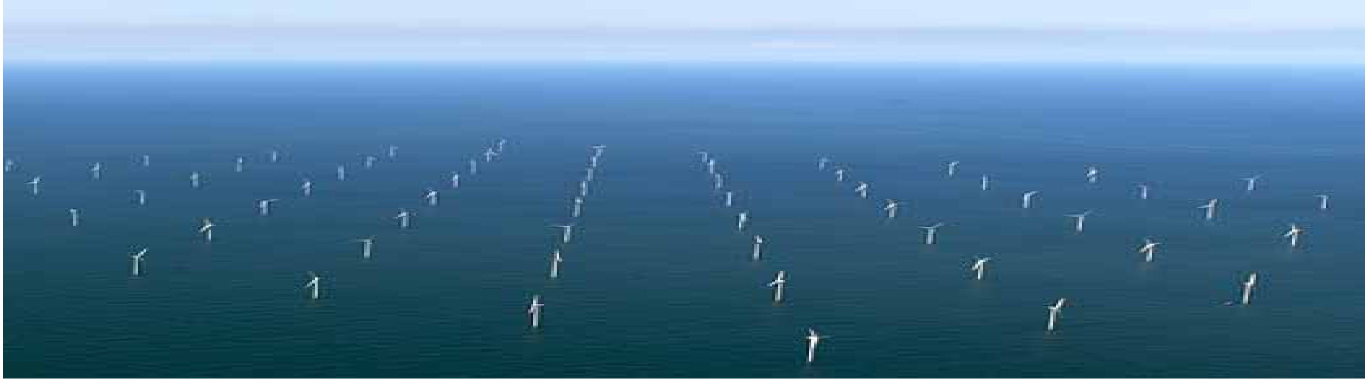


Figure 1.3: Horns Rev layout

In the wind energy industry, there are many areas of study that make use of complex models. Some models may require multi-physics modeling while others might take into consideration transient phenomena and extreme value analysis. Other models may require manipulating a lot of input data that many times have a stochastic nature such as e.g. the wind velocity or the wave height in offshore wind farms etc. Lastly, some models might actually incorporate all of the previous features.

The results of physics-related models may be passed as inputs to some other critical decision-guiding models used for inferring quantities of interest such as e.g. the expected power production of a wind farm layout or the net present value of the wind farm etc. These last models may be quite complex since the inter-dependency of the various variables they are affected from may cause them to go through multiple feedback loops in order to achieve an acceptable result and simultaneously satisfy criteria that have possibly been set.

A problem that is of concern lately [17, 45] is the **wind farm layout optimization** problem. This is, how to place the individual turbines inside a given wind farm so as to yield the highest possible wind farm power output. The solution to this problem (which wind farm layout is the best) should meet, *at least* the following criteria: a) the installation area for the turbines is defined and no turbines should be placed outside of it b) the turbines should be oriented as best as possible to face the direction of the wind that is the most probable and has the highest power content and c) the positioning of the turbines should be restricted so that it minimizes their wake interactions. Of course, there can be more criteria to this problem (e.g. relating to cost) but here we just laid down the most basic ones.

How did this wind farm layout optimization problem come to be though? A very common solution to the positioning of the wind turbines was historically given by installing all of them in a rectilinear formation with a fixed spacing along each direction [46]. An example of such formation is seen in Figure 1.3 for the Horns Rev offshore wind farm. In this case, the spacing in the two main directions is calculated with respect to some computed minimum distance between the wind turbines – more on this in Chapter 5. However, a natural question is to ponder whether this was enough. Since one can imagine that the wind turbines can be placed almost anywhere within the installation area and form extremely many (if not infinite) layouts, an *exhaustive search* to test and compare all those layouts would be futile: the **search space** of all layouts, (all candidate **solutions**) is too big. However, during the latest decades, new paradigms arose to find potent solutions to such search problems, collectively termed *heuristic search* methods. These methods give solutions similar to the ones of the exhaustive search approach and can be used in a great variety of problems. The solutions that these methods give are not the best (the most *optimal*), but are good approximations of the latter and they are produced much more faster than the latter. A particular heuristic method is Genetic Algorithms and specifically the Binary Genetic Algorithms (BGA's) that could in reasonable amounts of time approach a potent solution that for many problems might also be the most optimal one. So for the case of wind energy, the "restriction" to rectilinear layout formation is removed and through the use of some heuristic method, one can examine all kinds of possible layouts in an efficient and reliable way.

It would be better also if the prediction of such a power output for every layout examined could also be accompanied by an indication of how much reliable is this prediction. The financial viability of a wind farm usually requires an analysis of the risks involved when one invests in a specific wind power project. Banks have various probabilities of exceedance categories that classify wind farm projects such as e.g. P50, P75, P90 etc [3]. For example, a P90 wind farm project has only a 10% probability of not meeting the level of expected annual power production. The probability of exceedance of the predicted expectation of annual power production should be high enough in order for a project to be certified as “bankable” (capable of receiving funding from banks).

Moreover, it should be noted that the BGA algorithm compares layouts but does not compute the expected power output for a layout itself. Because this is of course needed to make the comparison possible, the BGA algorithm uses a physics model that calculates the expected power production and the wake-induced power losses for every layout. Naming this physics model as an original model as done above, if the best wind farm layout is to be found, a question arises whether the same goal can also be achieved with the help of a surrogate model. One then has to solve the problem using both an original model and its surrogate alternative and compare their results in terms of performance and delay of delivery of results. The surrogate model we will use is Gaussian Process Regression ($\mathcal{GP}$). Another approach could also be to test and compare the same original model with a random sampling method such as $\mathcal{MC}$ as introduced above. It was already discussed how $\mathcal{MC}$ method predicts the expected value of the output(s) we are interested in, based on a smaller randomly sampled set of inputs. Some kind of indication of how reliable is the prediction made by the $\mathcal{MC}$ model should be provided as well. Therefore, models based on $\mathcal{GP}$ and $\mathcal{MC}$ can be tested and compared against this original model over their solutions to the wind farm layout optimization problem.

1.3. Objectives and Methodology

This thesis examines the layout optimization problem using three models in order to assess their computational speed and reliability. First, an original model is introduced (of the form $g : A \to B$) that is founded on the theoretical considerations of aerodynamic theory, stochastic modeling of the meteorological data and medium-fidelity wake modeling, in order to compute (as output) the expected power production of the wind farm for a specific layout. This model will have as inputs the meteorological data (namely wind speed and wind direction) and their frequency of occurrence. Secondly, two alternative models will be built and introduced:

- a Gaussian Process surrogate model that samples from some of the inputs and their corresponding outputs of the original model and then interpolates for the rest. In this way it relates, as shown in Figure 1.2 *for all inputs* in the domain A, the value of the expected power production

- and a Monte Carlo model that samples *randomly* from a domain A of the inputs of the original model and then computes, *for all its domain* $A_{s_{\mathcal{MC}}}$, the value of the expected power production

Notice now that the original model and the surrogate model, based on their outputs for all inputs of the domain A, can each estimate the expected power production, *for all of the domain A*. Assuming that the subset $A_{s_{\mathcal{MC}}}$ is formed from enough samples so it is indicative enough the expected power produced from the Monte Carlo model and the other two can be considered comparable.

Thirdly, a BGA is constructed that accepts the expected power production of all these models for each specific layout and selects the best layout. This layout is selected on the basis of how each model computes or approximates its expected power production.

Lastly, for all of these models, there will be comparisons of their outputs and the error they produce as well as their efficiency in computational time. For the alternative models, the $\mathcal{GP}$ and the $\mathcal{MC}$ models, careful consideration should be given in how much and what type of sampling they need in order for their error to be as minimal as possible.

1.4. Thesis Outline

This chapter has introduced some background concerning the wind farm layout optimization problem and the various types of models that can be conceived to solve it. In particular, for the surrogate models, we referred briefly to the Gaussian Process Regression model, for the random sampling models we referred briefly to the Monte Carlo model and for the heuristic models we referred briefly to the Binary Genetic Algorithm model.

In Chapter 2 we will begin our analysis by introducing the mathematical toolbox that will be needed in the later chapters. This chapter refers to fundamental probability and stochastic process theory as well as explains parameter estimation and the Bayesian framework.

In Chapter 3 we will address the theory behind Gaussian Process Regression using the findings of Chapter 3 and many of the findings that can be found in the Appendix A

In Chapter 4 we will address the theory behind Monte Carlo methods utilizing some fundamental theorems of probability theory and by touching upon the different variations of these methods.

Chapter 5 is devoted to the wind farm layout optimization problem. It starts by explaining the main concepts of optimization theory, how it applies to wind farm layouts and how Binary Genetic Algorithms work.

In Chapter 6 the main analysis and the results of this thesis will be presented, explaining how all the models we built in computer language work, how they were verified and how should their results be perceived. The models include everything, from the original model, the Gaussian Process Regression model, the Monte Carlo model and the Binary Genetic Algorithm.

In Chapter we end our discussion on this thesis by giving some conclusions and recommendations on how the work here may be extended for future research.

2
Probability Fundamentals

This chapter is meant to present some theoretical notions from probability theory in order to prepare the reader for what follows in the next chapters. If the reader is already familiar with these notions, he may as well skim this chapter.

2.1. Events, Probability and Random Variable

Sets are well-defined collections of objects. We will assume here that the reader is acquainted with the basic set operations. In probability theory, when for example one asks what is the probability of a cast die to be lower than 5, one thinks of a set $\{1,2,3,4\}$. Such sets that contain **outcomes** of a random phenomenon are called **events**. Notice here two things:

- Events are sets, outcomes are not (they might be numbers, faces of a coin etc.).
- Outcomes are mutually exclusive - it cannot be the case that the die is both 1 and 3 for example. Such types of mutually exclusive events are termed **singletons**.

Elementary events $\{\omega\}$ are sets that contain only one outcome ω and as noted before, they are mutually exclusive, that is, their intersection is equal to $\emptyset$. The union of all elementary events is called the **sample space** Ω. Events in general therefore are subsets of this sample space Ω. One can **partition** the sample space **up to** all elementary events.

Now let us examine probabilities. We need probabilities to be defined for a collection of events and not necessarily for all sets. In order to provide the answer for the previous question ("lower than 5") one needs to construct a mathematical object that accepts as an argument an event A and produces a number x. So, the **probability measure (or simply probability)** $\mathbb{P}$ is defined such that $\mathbb{P}(A) = x$. It is necessary [36] that

- $\mathbb{P}(A) \geq 0$ for every event A;
- $\mathbb{P}(\Omega) = 1$;
- if $A \cap B = \emptyset$ then $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$

It can be proven that with this setup for probability measure, $\mathbb{P}(\emptyset) = 0$ and $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$ where A^c is **the complementary event** of A, that is $A \cup A^c = \Omega$

A real valued random variable $\tilde{x}$ or $\tilde{x}(\omega)$ now is a function that maps **outcomes** ω to a real number. By using it, one can **specify collections of events in general**. For example, imagine now that we cast two dice and ask which are outcome pairs which sum to 7. These are $(1,6), (2,5), (3,4), (4,3), (5,2)$ and $(6,1)$. Since this is the case we can also write the following in order to give this collection of events more concisely:

$$\{(i,j) : \tilde{x} = \tilde{x}(i,j) = i + j = 7\} \tag{2.1}$$

where i, j are the outcomes of each die (the numbers) and $\tilde{x}$ is a function called "random variable" such that $\tilde{x}(i,j) = i + j$. The *specification of which collection of events* are characterized by a sum of 7 is made possible by defining a random variable. In a similar fashion, with the help of the random variables, one can specify collections of events with the help of the following expressions:

$$\{\omega \in \Omega : \tilde{x} = x_0\}, \qquad \text{or} \qquad \{\omega \in \Omega : \tilde{x} \leq x_1\}, \qquad \text{or} \qquad \{\omega \in \Omega : x_2 \leq \tilde{x} \leq x_3\} \qquad \text{etc.} \tag{2.2}$$

where x_0, x_1, x_2 and x_3 are real numbers and ω is an outcome of the sample space Ω.

For a real-valued random variable $\tilde{x}$, one can define its **cumulative probability distribution** as

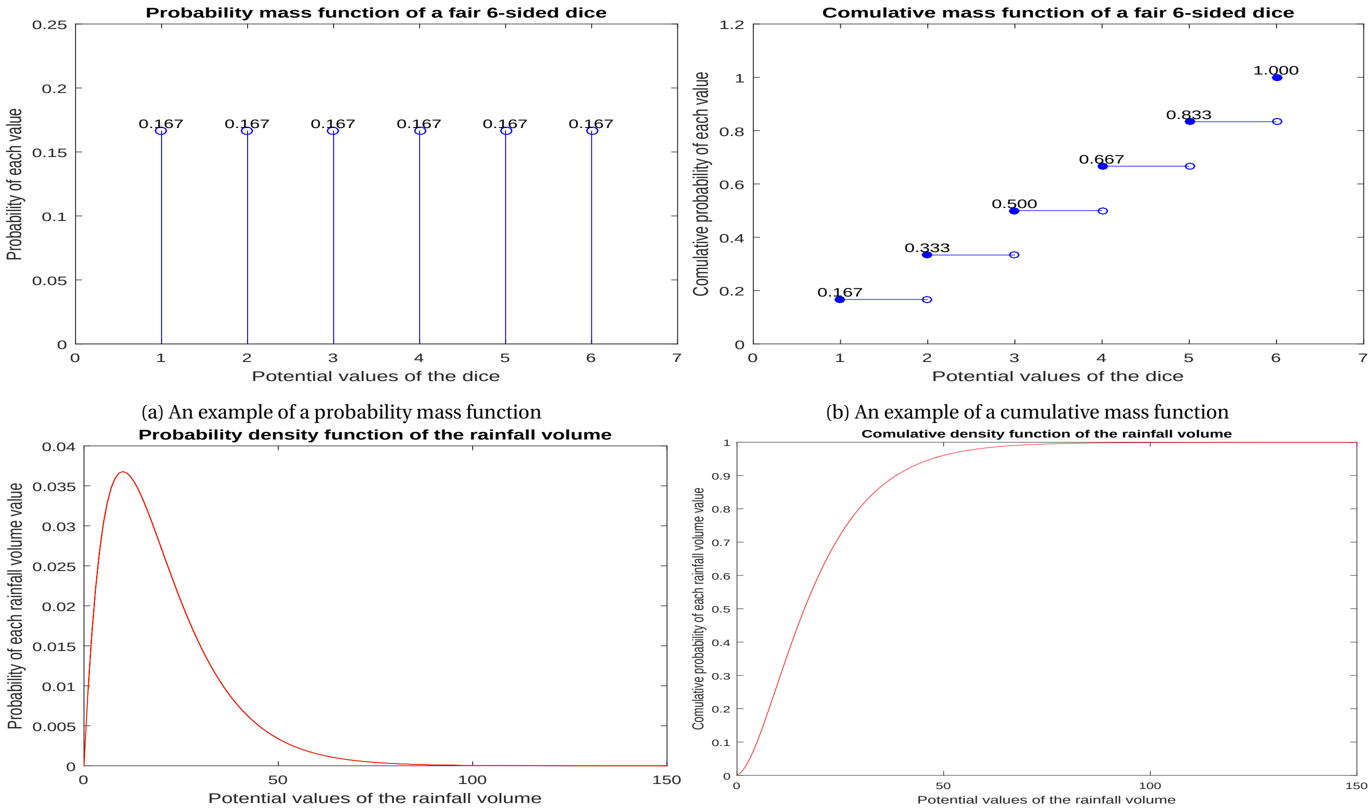
$$F_{\tilde{x}}(x) = \mathbb{P}(\tilde{x} \leq x) = \mathbb{P}(\{\omega \in \Omega : \tilde{x}(\omega) \leq x\}) \tag{2.3}$$

Notice here that the random variable $\tilde{x}$ could have been any kind of function – for example the sum of two dice or more complex phenomena. Since all such functions $\tilde{x}$ specify collections (sets) of events, for every value of x, one needs to determine *how to assign and compute probabilities of events.*

There are various approaches [36] as to how the probability measure is determined. One that will be shown in Subsection 4.1.3 is the **frequentist approach** to probability. Another school of thought, for example, considers that the probability is a mathematical object that represents a **subjective degree of belief**, i.e. how sure one is whether an event would happen or not. This brings questions such as e.g. a gambler's question about the final score of a tomorrow's football match into the probability framework without "testing" how valid such a belief is. This approach is of particular interest when one examines events that will happen or have happened that are non-repeatable or we do not have any further knowledge about their occurrence (or re-occurrence). Since the debate between all those approaches is ongoing, for practical applications at least, we shall consider directly a random variable that follows a cumulative probability distribution without specifying how this cumulative probability distribution was derived. Therefore, we will accept that for every x, the probability $\mathbb{P}(\tilde{x} \le x)$ is known.

2.2. Main Probability Distributions

Before discussing further the cumulative probability distribution we shall introduce some probability distributions that are generally known. Suppose for a moment that the random variable $\tilde{x}$ is real-valued and discrete – it accepts distinct values. If one would plot the probability found for every possible value that the random variable can take then one would derive a plot like that of Figure 2.1a. This figure concerns the outcome of a 6-fair sided dice, a discrete variable. Notice that since it is a fair die, we assume that all probabilities amount to 1/6 and their sum is 1. This type of plot is called a probability mass function plot. A **probability mass function** (pmf) is a function since for each value x and its evaluation is $\text{pmf}_{\tilde{x}}(x) = \mathbb{P}(\tilde{x} = x) = \mathbb{P}(\{x\})$. Accordingly one may define a **cumulative mass function** (cmf) and its plot as seen in Figure 2.1b, which is a cumulative summing of all individual probabilities of the various x values of the $\tilde{x}$ variable. This function is monotonically increasing function whose maximum value is 1, when all possible values $\mathbb{P}(\{x\})$ are summed so that its evaluation is $\text{cmf}_{\tilde{x}}(x) = \sum_{\tilde{x} \le x} \mathbb{P}(\{x\})$. Notice that because a die can have only integer values from 1 to 6, we assume that the rest of the values in between 1 to 6 shall certainly never happen, so their individual probabilities are 0. This explains why the sum of the probabilities does not change in those points. This reason explains why the term "cumulative mass function" is usually substituted for the term "cumulative distribution function".



(a) An example of a probability mass function

(b) An example of a cumulative mass function

(c) An example of a probability density function

(d) An example of a cumulative density function

Figure 2.1: Examples of distribution graphs for discrete (a),(b) and continuous variables (c)(d)

Now assume that $\tilde{x}$ is a continuous variable, for example the rainfall volume on a specific area. The same analogies can be drawn. In Figure 2.1c one can see the plot of a **probability density function** (pdf) and in 2.1d the plot of a **cumulative distribution function** (cdf). Now from Equation 2.3 one may find that $\mathbb{P}(a < x \le b) = F_{\tilde{x}}(b) - F_{\tilde{x}}(a)$ [15]. Then, the pdf is related to the cdf as:

$$f_{\tilde{x}}(x) = \frac{\mathrm{d}F_{\tilde{x}}(x)}{\mathrm{d}x} = \lim_{a\to x}\frac{F_{\tilde{x}}(x) - F_{\tilde{x}}(a)}{x-a} \qquad \text{so that} \qquad F_{\tilde{x}}(x) = \int_{-\infty}^{x} f_{\tilde{x}}(u)\mathrm{d}u \qquad \text{and} \qquad \int_{a}^{b} f_{\tilde{x}}(x)\mathrm{d}x = \mathbb{P}(a < \tilde{x} \le b) \tag{2.4}$$

Notice that the above Equation cannot be considered for the pmf since the cmf is not continuous while the cdf is. It can be also observed that the pmf has “concentrated” probability “mass” in certain x values while the pdf does not. It can be seen in the above equation that the pdf of a specific value $a \in \mathbb{R}$ is always equal to zero. Nevertheless, one can compare between two values a and b on the real axis the **relative expectancy** of their occurrence – **how much more likely** they are to happen. A question then may arise: what would the expected value μ for $\tilde{x}$ be? How much do all values of the $\tilde{x}$ variable differ with respect to this expected value μ? For computing such **statistical measures**, the pdf is used, the cdf is implicated only indirectly – it just introduces the pdf from Equation 2.4. Before we introduce the answer to these questions, we shall first introduce some other distribution functions for the multivariate case for the consideration of two random variables.

2.3. Joint Probability, Marginal Probability, Conditional Probability, Independence

Assume the existence of two continuous real-valued random variables $\tilde{x}$ and $\tilde{y}$. A **joint probability density function** $f_{\tilde{x},\tilde{y}}(x, y)$ is a non-negative function satisfying the following Equation [40]:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{\tilde{x},\tilde{y}}(x, y)\mathrm{d}x\mathrm{d}y = 1 \tag{2.5}$$

. Then we can define the following joint cumulative distribution function that is [43]:

$$F_{\tilde{x},\tilde{y}}(a, b) = \mathbb{P}(\tilde{x} \le a, \tilde{y} \le b) = \int_{-\infty}^{b}\int_{-\infty}^{a} f_{\tilde{x},\tilde{y}}(x, y)\mathrm{d}x\mathrm{d}y \tag{2.6}$$

, which is used to compute their mutual probability of occurrence. This happens when their events of occurrence **intersect**. The **marginal probability density function** $f_{\tilde{x}}$ or $f_{\tilde{y}}$ is then computed as [44]:

$$f_{\tilde{x}}(x) = \int_{-\infty}^{\infty} f_{\tilde{x},\tilde{y}}(x, y)\mathrm{d}y \qquad \text{and} \qquad f_{\tilde{y}}(y) = \int_{-\infty}^{\infty} f_{\tilde{x},\tilde{y}}(x, y)\mathrm{d}x \tag{2.7}$$

for $\tilde{x}$ and $\tilde{y}$ respectively. These concern the probability of $\tilde{x}$ or $\tilde{y}$ regardless of the probability of the other event ($\tilde{y}$ or $\tilde{x}$). This happens by “integrating out” the effect of this other event. Lastly, for all x or y such that $f_{\tilde{x}}(x) \ne 0$ or $f_{\tilde{y}}(y) \ne 0$, one can define the **conditional probability density function** or conditional probability as [44]:

$$f_{\tilde{y}}(y|\tilde{x} = x) = \frac{f_{\tilde{x},\tilde{y}}(x, y)}{f_{\tilde{x}}(x)} \qquad \text{and} \qquad f_{\tilde{x}}(x|\tilde{y} = y) = \frac{f_{\tilde{x},\tilde{y}}(x, y)}{f_{\tilde{y}}(y)} \tag{2.8}$$

which within an event context, for two events A and B we would write [41]:

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A\cap B)}{\mathbb{P}(A)} \qquad \text{and} \qquad \mathbb{P}(B|A) = \frac{\mathbb{P}(A\cap B)}{\mathbb{P}(B)} \tag{2.9}$$

Similar equations can be written if there are more than two random variables occurring. Also, similar equations can be written to introduce analogous mathematical objects for the discrete multivariate case (called “mass functions”).

We shall finish this section by discussing the **condition of independence** of two events A and B for a given probability $\mathbb{P}(A\cap B)$. This happens only if $\mathbb{P}(A\cap B) = \mathbb{P}(A)\mathbb{P}(B)$. Note that this also implies that $\mathbb{P}(A|B) = \mathbb{P}(A)$ and $\mathbb{P}(B|A) = \mathbb{P}(B)$, which means that knowing the realization of an event B (or A) does not change the probability of A (or B). For the bivariate case with two continuous real-valued variables $\tilde{x}$ and $\tilde{y}$ we have:

$$f_{\tilde{x},\tilde{y}}(x, y) = f_{\tilde{x}}(x)f_{\tilde{y}}(y) \tag{2.10}$$

Lastly, in the scientific literature when one samples from a specific probability distribution function $f_{\tilde{x}}$ of a random variable $\tilde{x}$ in order to evaluate it, usually one writes [15]:

$$\tilde{x} \sim f_{\tilde{x}} \qquad \text{or} \qquad \tilde{x} \sim \mathrm{pdf}_{\tilde{x}} \tag{2.11}$$

2.4. Expected Value, Variance and Covariance

As said before the pdf is used in order to compute the aforementioned statistical measures. The expectation or expected value $\mathbb{E}(\tilde{x})$ or $\mu_{\tilde{x}}$ or simply μ of a continuous real-valued random variable $\tilde{x}$ is calculated as [41]:

$$\mu_{\tilde{x}} \qquad \text{or} \qquad \mu: \qquad \mathbb{E}(\tilde{x}) = \int_{-\infty}^{\infty} x f_{\tilde{x}}(x)\mathrm{d}x \tag{2.12}$$

provided that this value exists [31].

For the case of two continuous real-valued random variables $\tilde{x}$ and $\tilde{y}$ if we consider a function $g(x,y)$ one may define also the following expected value of the **function of the random variables** $\tilde{x}$ and $\tilde{y}$ [31]:

$$\mathbb{E}\big(g(\tilde{x},\tilde{y})\big) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y) f_{\tilde{x},\tilde{y}}(x,y)\mathrm{d}x\mathrm{d}y \tag{2.13}$$

provided that this value exists [31].

Similar expressions can be written for the case of the discrete bivariate case. So now the **property of linearity** for the expected value can be proven [42]:

$$\mathbb{E}(a\tilde{x}+b\tilde{y}) = a\mathbb{E}(\tilde{x}) + b\mathbb{E}(\tilde{y}) \tag{2.14}$$

where a and b are constants. Also for two independent variables one has [43]:

$$\mathbb{E}(\tilde{x}\tilde{y}) = \mathbb{E}(\tilde{x})\mathbb{E}(\tilde{y}) \tag{2.15}$$

The variance $\mathrm{Var}(\tilde{x})$ or $\sigma_{\tilde{x}}^2$ or simply σ^2 of random variable $\tilde{x}$ always has a non-negative value and can be computed as follows [39]:

$$\sigma_{\tilde{x}}^2 \qquad \text{or} \qquad \sigma^2: \qquad \mathrm{Var}(\tilde{x}) = \mathbb{E}[(\tilde{x}-\mu)^2] = \mathbb{E}(\tilde{x}^2) - [\mathbb{E}(\tilde{x})]^2 = \mathbb{E}(\tilde{x}^2) - \mu^2 \tag{2.16}$$

It can also be proven that [42]:

$$\mathrm{Var}(a\tilde{x}+b\tilde{y}) = a^2\mathrm{Var}(\tilde{x}) + b^2\mathrm{Var}(\tilde{y}) + 2ab\mathrm{Cov}(\tilde{x},\tilde{y}) \tag{2.17}$$

where a and b are constants as before and $\mathrm{Cov}(\tilde{x},\tilde{y})$ is a statistical measure of joint-variability of two random variables which is defined as [15]:

$$\mathrm{Cov}(\tilde{x},\tilde{y}) = \mathbb{E}\Big(\big(\tilde{x}-\mu_{\tilde{x}})\big)\big((\tilde{y}-\mu_{\tilde{y}})\big)\Big) \tag{2.18}$$

For this to be valid the above random variables should have finite second moments, or in other words, **standard deviations** (the positive square-root of their variance) $\sigma_{\tilde{x}}$ and $\sigma_{\tilde{y}}$. It can be proven also that [15]:

$$\mathrm{Cov}(\tilde{x},\tilde{y}) = \mathbb{E}(\tilde{x}\cdot\tilde{y}) - \mu_{\tilde{x}}\mu_{\tilde{y}} \tag{2.19}$$

and

$$\mathrm{Var}(\tilde{x}) = \mathrm{Cov}(\tilde{x},\tilde{x}) = \mathbb{E}(\tilde{x}^2) - \mathbb{E}(\tilde{x})^2 = \mathbb{E}(\tilde{x}\cdot\tilde{x}) - \mathbb{E}(\tilde{x})\mathbb{E}(\tilde{x}) \tag{2.20}$$

Lastly we will add that the standard deviation of a random variable $\sigma_{\tilde{x}}$ or σ is also a statistical measure to quantify dispersion of the evaluations of a random variable around its expected value $\mu_{\tilde{x}}$ and is related to its variance as:

$$\sigma_{\tilde{x}} = \sqrt{\mathrm{Var}(\tilde{x})} \tag{2.21}$$

2.5. The Uniform Distribution

One probability distribution that is going to be of use in later chapters is the uniform distribution. We state that a variable $\tilde{x}$ follows a uniform distribution $U(a,b)$ of parameters a,b if the function describing this probability distribution is [51]:

$$U(\tilde{x};a,b) = \mathrm{pdf}_{\tilde{x}}(x) = f_{\tilde{x}}(x) = \begin{cases} \frac{1}{b-a}, & \text{for } x \in [a,b] \\ 0, & \text{otherwise} \end{cases} \tag{2.22}$$

The cumulative distribution function is the following:

$$\mathrm{cdf}_{\tilde{x}}(x) = F_{\tilde{x}}(x) = \begin{cases} 0, & \text{for } x < a \\ \frac{x-a}{b-a}, & \text{for } x \in [a,b) \\ 1, & \text{for } x \geq b \end{cases} \tag{2.23}$$

So for the uniform distribution $U(0,1)$ with parameters $a = 0$, $b = 1$ we have:

$$U(\tilde{x};0,1) = f_{\tilde{x}}(x) = \begin{cases} 1, & \text{for } x \in [a,b] \\ 0, & \text{otherwise} \end{cases} \tag{2.24}$$

and:

$$\mathrm{cdf}_{\tilde{x}}(x) = F_{\tilde{x}}(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{for } x \in [0,1) \\ 1, & \text{for } x \geq 1 \end{cases} \tag{2.25}$$

2.6. Cardinality of a Set

The cardinality $|A|$ of a set A is its "number of elements". For example the set $A = \{2, 7, 9\}$ has cardinality $|A| = 3$. One can consider the cardinality of every set, the most fundamental of which is $\mathbb{N}$, the set of natural numbers. There are three types of sets according to their cardinality measure:

- the **finite sets** A whose cardinality is less than the cardinality of $\mathbb{N}$ so that $|A| < |\mathbb{N}|$. If this is the case then $|A|$ is always some natural number, so that $|A| \in \mathbb{N}$.
- the **infinite sets** A in any other case. These sets can be either **countable or uncountable** though we do not need to define either class of them, but just to give an example, we would say that the set of natural numbers $\mathbb{N}$ is countable and the set of real numbers $\mathbb{R}$ is uncountable. If A is an infinite set, then $|A|$ **cannot be linked** to any number.

2.7. Real-Valued Stochastic Processes

Here we shall refer briefly to what a stochastic process is, in order to prepare the discussion of Chapter 3.

A real-valued stochastic process indexed by an ordered set T is a collection $\{\tilde{x}_t : t \in T\}$ of random variables $\tilde{x}_t(\omega)$ that take values in $\mathbb{R}$ for every element $\omega \in \Omega$. Notice that every random variable $\tilde{x}_t(\omega)$ takes an argument ω and maps it to the line of real numbers as introduced before, but now one examines an ordered set of these random variables, marked by the index set T. Therefore a stochastic process is also a **function**, written as $X = \tilde{x}(\omega, t)$ mapping every $t \in T$ and every $\omega \in \Omega$ to the real line. The only distinction between a real-valued stochastic process perceived as a function and any determinate function is that a stochastic process is a real-valued **random function**, that is, its image on the set $\mathbb{R}$ is not defined deterministically as seen in Figure 2.2.

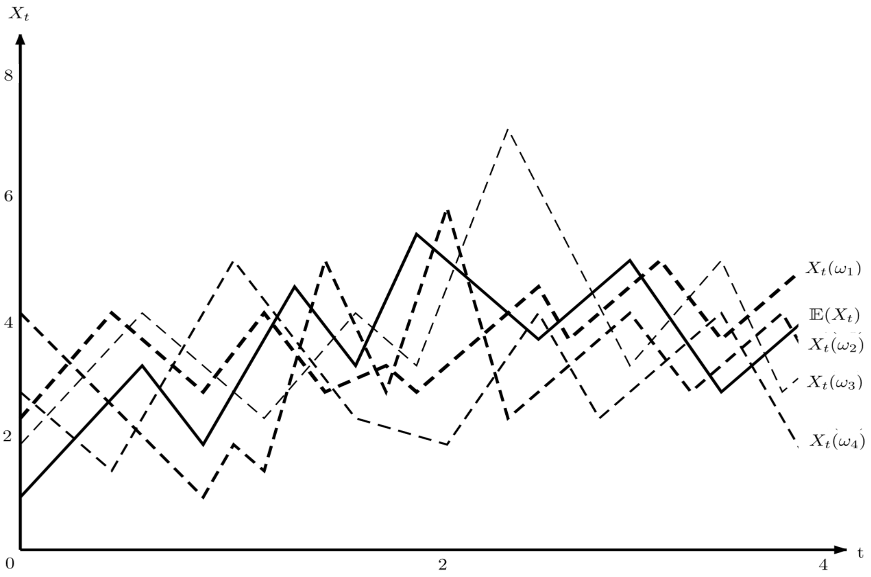


Figure 2.2: An example of a stochastic process. The dashed lines $X_t(\omega_1)$, $X_t(\omega_2)$, $X_t(\omega_3)$ etc. are various **representations** of the stochastic process (a *random function*). Each one of those representations is the collection of sampled evaluations of the random variables $\tilde{x}_t(\omega)$ for all index values $t \in T$. The continuous black line $\mathbb{E}(X_t(\omega))$ is the realization of the stochastic process that has the *expected values* of those random variables $\tilde{x}_t(\omega)$ for all index values $t \in T$.

The collection $\{\tilde{x}_t : t \in T\}$ of random variables $\tilde{x}_t(\omega)$ has an index set T as mentioned above that can be infinite or finite. In Figure 2.2 one sees a stochastic process that is indexed with the aid of a infinite set, namely $\mathbb{R}$. In a **continuous stochastic process**, as in Figure 2.2, T is infinite . As far as the next chapter is concerned, although we will assume the existence of an underlying **continuous** real-valued stochastic process, called "Gaussian Process", indexed by an infinite set T, we will always study it for **finite subsets** of the index set T. That is, if a Gaussian Process is a continuous stochastic process – an infinite collection of random variables indexed by an infinite index set T – then one can define a **finite** subset $T_s \subseteq T$, $|T_s| \in \mathbb{N}$ such that the focus now lies on a **finite sub-collection of the random variables** $\{\tilde{x}_t : t \in T_s\}$, $|T_s| \in \mathbb{N}$. Such **finite sub-collections** are of course *not all of the stochastic process.*

We notice that if we restrict ourselves and consider solely **finite sub-collection of the random variables** indexed on T_s, $T_s \subseteq T$, $|T_s| \in \mathbb{N}$, we will end up studying finite collections of random variables whose probability distribution is a **multivariate joint probability distribution**. On the other hand, if we consider the initial infinite index set T, the probability distribution defined on the totality of the infinite collection of random variables is too complex for practical purposes; one needs to define new mathematical objects to describe the probability distribution of a continuous stochastic process [57].

So the motive is: can we can "conclude our business" with Gaussian Process Regression in the next chapter by only using **multivariate joint probability distributions**? It turns out we can [38] and this, of course, makes things easier as we do not need to introduce any new mathematical objects.

2.8. Transformation of Random Variables

This section will be necessary for Chapter 5. When the time comes we will refer to it.

When one has a random variable $\tilde{x} = \tilde{x}(\omega)$ and a determinate function y that accepts it as an argument, the image of it is also a random variable because: $y(\tilde{x}) = y(\tilde{x}(\omega)) = \tilde{y}(\omega) = \tilde{y}$. However, this does not mean that $f_{\tilde{x}}$ and $f_{\tilde{y}}$ are the same. To see this compare for example $\tilde{x} \sim U(-1,1)$ and $\tilde{y}$ derived from the function $y = x^2$. It is easy to see that e.g. for the interval $(-1,0)$, $\mathbb{P}(\tilde{x} \in (-1,0)) = 0.5$ while $\mathbb{P}(\tilde{y} \in (-1,0)) = 0$, so clearly $f_{\tilde{x}} \neq f_{\tilde{y}}$.

2.9. Every Probability is Also an Expectation

This section will be necessary for Chapter 5. When the time comes we will refer to it.

For a random variable $\tilde{x}$, every probability $\mathbb{P}(\tilde{x} \in A)$ of an event $A \subseteq \Omega$, Ω being the sample space, can be written as:

$$\mathbb{P}(\tilde{x} \in A) = 1 \cdot \mathbb{P}(\tilde{x} \in A) + 0 \cdot \mathbb{P}(\tilde{x} \in A^c) = \mathbb{E}\big(I_A(\tilde{x})\big) \tag{2.26}$$

where I_A is the indicator function defined as [51]:

$$I_A(x) = \begin{cases} 1, \text{ if } x \in A \\ 0, \text{ if } x \in A^c \end{cases} \tag{2.27}$$

2.10. Parameter Estimation and Bayes' Rule

It seems impossible not to refer to some basic theory of parameter estimation since it will help the discussion of the next chapters. When one studies a random variable $\tilde{w}$, its pdf may depend on the existence of a parameter h. Since the pdf of $\tilde{w}$ is affected by the value of this parameter, when one has sampled data, which pdf is it supposed to follow? Parameter estimation is a set of methods for choosing the value of the parameter in question in order to find the pdf that conforms to the sampled data.

Let us say for convenience that we study a **discrete** random variable $\tilde{w}$ and a parameter $\tilde{h}$ that we assume is also a discrete random variable. Of all the evaluations $\tilde{h} = h_\text{s}$ of $\tilde{h}$, which one is "the best fit", given some evaluation w_s of the random variable of interest $\tilde{w}$? A "natural" estimation process is to consider a value for $\tilde{h}$, h_MLE, for which the term $\mathbb{P}(\tilde{w} = w_\text{s}|\tilde{h} = h_\text{s})$ is maximal. In order to find this maximizing value h_MLE one may assume the following term, called **likelihood** which is the following function $\mathbb{L}$:

$$\mathbb{L}(h_\text{s}; w_\text{s}) = \mathbb{P}(\tilde{w} = w_\text{s}|\tilde{h} = h_\text{s}), \tag{2.28}$$

where for h_{MLE} this function has its maximum value.

Notice that likelihood is a function of $\tilde{h}$ parametrized by the evaluations w_s of the random variable $\tilde{w}$ and our goal is to find the value h_MLE that maximizes it. Now the situation is a little bit different when one considers a **continuous** random variable $\tilde{w}$. Here the parameter estimation calls for the maximization of the following quantity [9]:

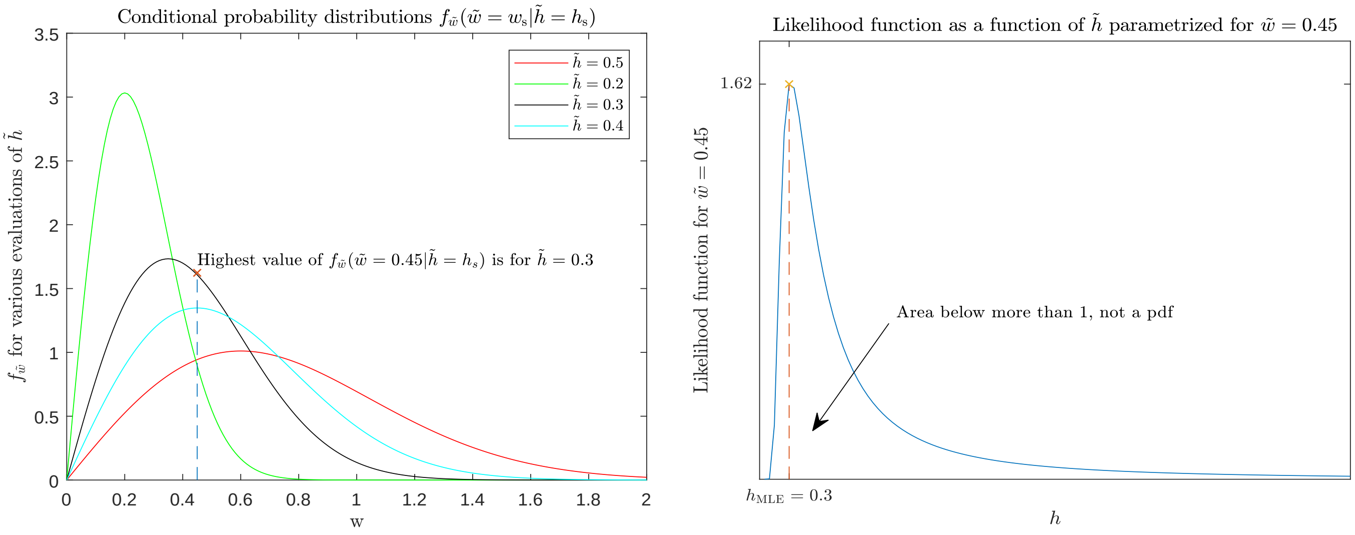
$$\mathbb{L}(h_\text{s}; w_\text{s}) = f_{\tilde{w}}(\tilde{w} = w_\text{s}|\tilde{h} = h_\text{s}). \tag{2.29}$$

The goal is again to somehow find the value h_MLE that maximizes the likelihood function as re-introduced in Equation 2.29. Figures 2.3a and 2.3b illustrate how this can be conceived. Remember that the likelihood function is a function of $\tilde{h}$ parametrized by the evaluations w_s of the random variable $\tilde{w}$. Therefore, to maximize the likelihood function is to find the evaluation $\tilde{h} = h_{MLE}$ whose corresponding pdf value $f_{\tilde{w}}(\tilde{w} = w_\text{s}|\tilde{h} = h_\text{MLE})$ is higher than the pdf value $f_{\tilde{w}}(\tilde{w} = w_\text{s}|\tilde{h} = h_\text{s})$ for any other evaluation h_s of the parameter $\tilde{h}$. In Figure 2.3a one sees how the family of all pdf's parametrized by $\tilde{h}$ are plotted. For a given evaluation $\tilde{w} = 0.45$, one finds that from this family of pdf's for different h_s the value of the $f_{\tilde{w}}(\tilde{w} = 0.45|\tilde{h} = h_\text{s})$. The maximum value is for $h_\text{MLE} = 0.3$ and this can be seen in Figure 2.3b where we also noted that a likelihood function should not be considered necessarily a pdf since it is not always true that $\int_{-\infty}^{\infty} \mathbb{L}(h; w_\text{s})\text{d}h$ is equal to 1.

If instead of one random variable $\tilde{w}$ and one parameter (as a random variable) $\tilde{h}$ we have a **finite sub-collection** $\mathcal{W}$ of continuous random variables $\{\tilde{w}_1, \tilde{w}_2, \tilde{w}_3, \ldots, \tilde{w}_n\}$ and a finite sub-collection $\mathcal{H}$ of parameters $\{\tilde{h}_1, \tilde{h}_2, \tilde{h}_3, \ldots, \tilde{h}_m\}$ again as continuous random variables, with $n, m \in \mathbb{N}$ we can write:

$$\mathbb{L}(H; W_\text{s}) = f_{\mathcal{W}}(\mathcal{W} = W_\text{s}|\mathcal{H} = H), \tag{2.30}$$

where W_s is a set of evaluations $\{\tilde{w}_1 = w_1, \tilde{w}_2 = w_2, \tilde{w}_3 = w_3, \ldots, \tilde{w}_n = w_n\}$.



(a) Different conditional pdf's that define the values that the likelihood function takes.

(b) Plotting the values of the likelihood function for different values of the parameter. The maximum value for the likelihood function that comes from the argument $h_{\mathrm{MLE}} = 0.3$

Figure 2.3: An example of a continuous random variable $\tilde{w}$ parametrized by a parameter $\tilde{h}$ of various evaluations, its conditional pdf's, the likelihood function and the value h_{MLE} of the parameter $\tilde{h}$ for MLE. In the second figure it is also noted that a likelihood function should not be considered necessarily a pdf since it is not always true that $\int_{-\infty}^{\infty} \mathbb{L}(h; w_s)\mathrm{d}h$ is equal to 1 – here it is approximately equal to 2.

Equations 2.28 to 2.30 can be used not only for **Maximum Likelihood Estimation** (MLE) but for **Maximum A Posteriori Estimation** (MAP) as well. Remember now that the parameter $\tilde{h}$ follows some pdf $f_{\tilde{h}}$ (and accordingly a finite sub-collection $\mathcal{H}$ follows some joint pdf of its elements $f_{\mathcal{H}}$). This means that not all evaluations h_s of the parameter $\tilde{h}$ have equal expectancy to be sampled from the pdf $f_{\tilde{h}}$. How can we use this information in MAP estimation? Let us here first introduce Bayes' rule in order to understand the following points better.

Suppose two events $A, B \subseteq \Omega$ where Ω is their sample space. Suppose also $\mathbb{P}(B) \neq 0$ and $\mathbb{P}(A) \neq 0$. Then:

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \bigcap B)}{\mathbb{P}(B)}, \tag{2.31}$$

or

$$\mathbb{P}(B|A) = \frac{\mathbb{P}(A \bigcap B)}{\mathbb{P}(A)}, \tag{2.32}$$

and

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)}. \tag{2.33}$$

Equation 2.33 is Bayes' rule. As it can be seen it is valid even for disjoint events since $\mathbb{P}(A|B) = \mathbb{P}(B|A) = 0$. Bayes' rule is the central idea in Bayesian statistics whose modus operandi for making statistical inference is to assign probabilities to every event and work with the laws of probability theory (using Bayes rule) in order to derive predictions. Equation 2.33 should be seen in the light of what is termed "diachronic interpretation". "Diachronic" is a greek word which means that something is happening over two (or more) instances of time. This use of Bayes' rule is to **update one's initial belief** with the likelihood of the underlying "conditions" in the presence of new acquired "sampled data". In order to understand what "conditions" and "sampled data" signify let us return to the likelihood term which we have already discussed in detail. We will assume the case of two discrete random variables as presented in Equation 2.28. We would write as above:

$$\mathbb{P}(\tilde{h} = h_s|\tilde{w} = w_s) = \frac{\mathbb{P}(\tilde{w} = w_s|\tilde{h} = h_s)\mathbb{P}(\tilde{h} = h_s)}{\mathbb{P}(\tilde{w} = w_s)} \tag{2.34}$$

By "sampled data" we mean the event $B = \{\tilde{w} = w_s\}$ that is, we have some sampled data w_s from $\tilde{w}$. By "conditions" we mean the event $A = \{\tilde{h} = h_s\}$. Remember that $\tilde{h}$ is a parameter that determines what the pdf of $\tilde{w}$ would look like, so it can be said, it "conditions" the sampling procedure that yields "sampled data".

Furthermore we have:

- the probability term $\mathbb{P}(\tilde{w} = w_s|\tilde{h} = h_s)$ is the ***likelihood*** term as defined above. It is a conditional probability of $\tilde{w}$ given some evaluation for the parameter $\tilde{h}$. As said above, in MAP estimation it is not the goal to maximize it as in MLE. We want to find the most probable parameter evaluation given data so clearly this is term is not the focus of our attention.

- the probability term $\mathbb{P}(\tilde{w} = w_s)$ is the ***data*** term. It considers only the sampled data and does not care what the parameter evaluation was. Also since one can write:

$$\mathbb{P}(\tilde{w} = w_s) = \sum_{\text{all } h_s} \mathbb{P}(\tilde{w} = w_s|\tilde{h} = h_s)\mathbb{P}(\tilde{h} = h_s) \tag{2.35}$$

it is clear that that this term is a *marginal* probability that "sums out" all potential evaluations of $\tilde{h}$.

- the probability term $\mathbb{P}(\tilde{h} = h_\text{s})$ is the ***conditions*** or ***prior*** term. This term is the pdf of $\tilde{h}$ and is significant for MAP estimation (and Bayesian statistics in general) since it involves all the information about how likely every evaluation h_s is considered to be. Since the modeller cannot specify evaluations $\tilde{h} = h_\text{s}$ at will he has to work in a probabilistic context. There are cases that the modeller does know that the parameter is a random variable and follows some pdf but does not know which one! The good thing about Bayes' rule is that the modeller *can still work with it*: the **prior** term will signify the assumptions of the modeller, his **belief** over the occurrence of the parameter's evaluations (remember that probability can be viewed as a degree of belief), while the right hand of Equation 2.33 will **update his belief** given the occurrence of the sampled data – the **data** term.

- the probability term $\mathbb{P}(\tilde{h} = h_\text{s}|\tilde{w} = w_\text{s})$ is the ***posterior*** term that is computed to **update one's belief** given the occurrence of **data**. It is a conditional probability term which assigns probabilities to the parameter evaluations given data. According to MAP estimation we have to find **the most probable parameter evaluation** so one wants to **maximize this term**.

MAP is in a way an extension of MLE in the sense that it tries to bring into consideration how the evaluations h_s of the parameter $\tilde{h}$ have more or less relative expectancy to occur according to the pdf $f_{\tilde{h}}$ – the ***prior*** term. That is the same as to say how much the ***prior** favors* specific evaluations h_s of the parameter $\tilde{h}$ more than others. It can be understood that if a prior is **strong**, that is if it favors too much a specific evaluations of the parameter $\tilde{h}$, it plays a great role in the numerator of Equation 2.33. If on the other hand we have a **weak** prior, then in the numerator, the "dominating" term in terms of computation is still the likelihood term. These remarks can be observed in Figures 2.4a and 2.4b. Notice in Figure 2.4b that a weak prior is not necessary a uniform one. The reason the ***data*** term does not show up in these Figures is that as we have already explained, the ***data*** term is only a normalization term since it does not depend on the effect of specific evaluations of $\tilde{h}$.

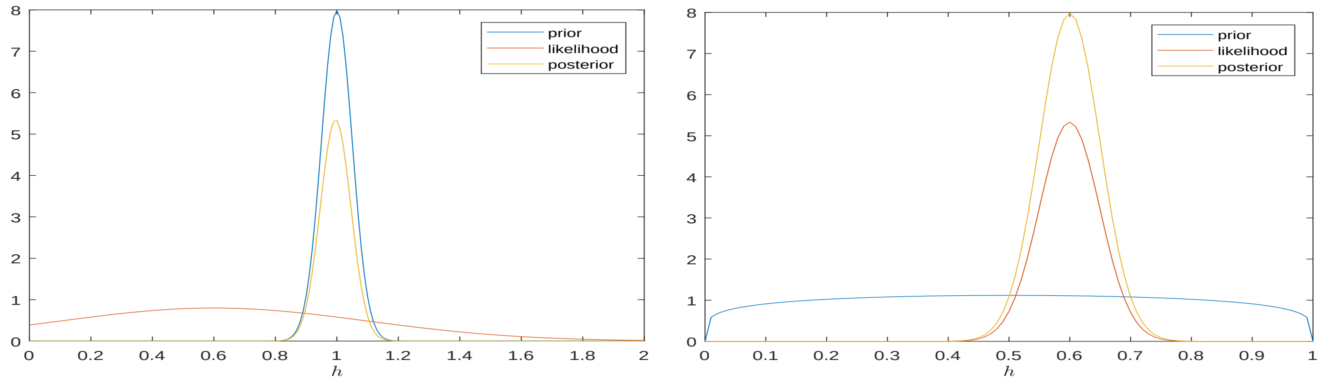
For weak priors as can be observed in Figure 2.4b their pdf is approximately equal to a constant. This means that when one examines again the Bayes' rule and the objective of MAP is to find the argument (some evaluation of $\tilde{h}$) that maximizes the posterior term, then one can see that:

$$\text{argmax}_{\tilde{h}}\Big(\mathbb{P}(\tilde{h} = h_\text{s}|\tilde{w} = w_\text{s})\Big) = \text{argmax}_{\tilde{h}}\Big(\frac{\mathbb{P}(\tilde{w} = w_\text{s}|\tilde{h} = h_\text{s})\mathbb{P}(\tilde{h} = h_\text{s})}{\mathbb{P}(\tilde{w} = w_\text{s})}\Big) = \text{argmax}_{\tilde{h}}\Big(\mathbb{P}(\tilde{w} = w_\text{s}|\tilde{h} = h_\text{s})\mathbb{P}(\tilde{h} = h_\text{s})\Big) \tag{2.36}$$

and since when we use a weak prior then $\mathbb{P}(\tilde{h} = h_s)$ is more or less constant so:

$$\text{argmax}_{\tilde{h}}\Big(\mathbb{P}(\tilde{h} = h_\text{s}|\tilde{w} = w_\text{s})\Big) \approx \text{argmax}_{\tilde{h}}\Big(\mathbb{P}(\tilde{w} = w_\text{s}|\tilde{h} = h_\text{s})\Big) \tag{2.37}$$

so that **when weak priors are used both MAP and MLE end up maximizing the likelihood function as discussed above**.



(a) An example of a strong prior and its effect on the posterior

(b) An example of a weak prior and its effect on the posterior

Figure 2.4: Examples of how the prior affects the posterior term. The strong prior directs the expected value h_MAP towards where itself is maximized and not where the likelihood does. The opposite is true for the weak prior where one can see that $h_\text{MAP} = h_\text{MLE}$. Notice that the weak prior is almost a uniform pdf ($f_{\tilde{h}}$ is approximately constant) but not necessarily.

But why should one use a weak prior in the first place? As marked above when we first introduced the ***prior*** term, this term expresses the **belief** of the modeller over the probability distribution of the parameter evaluations (when one uses Bayes' rule). Many times in Bayesian statistics in MAP estimation **weak** priors are advocated [4] so that the modeller can avoid expressing a strong belief over what the parameters would be expected to be. This is because the more the data are updated through sampling the more the likelihood function changes, and therefore h_MLE. For example, above in Figures 2.3a and 2.3b we had $\tilde{w} = 0.45$, if more sampling continues for we can estimate the expected value of $\tilde{w}$, $\mathbb{E}(\tilde{w})$, that will also give us the expected value of h_MLE that conforms with our data better. Therefore it is better for the analysis to leave the ***likelihood*** function be the "dominant" term for the maximization of the ***posterior*** and not the modeller's initial belief, the ***prior***. In this way h_{MAP} converges to h_{MLE} and not some other evaluation. In Figures 2.4a and 2.4b one sees the effect of a strong and a weak ***prior*** on the ***posterior*** term.

In Chapter 3 we will continue this MAP estimation analysis by assuming the use of a weak prior. Before we jump to this chapter however we have to note the following:

- For two continuous random variables $\tilde{w}$ and $\tilde{h}$, which is the case we care most for this thesis, Bayes' rule becomes

$$f_{\tilde{h}}(\tilde{h} = h_{\text{s}} | \tilde{w} = w_{\text{s}}) = \frac{f_{\tilde{w}}(\tilde{w} = w_{\text{s}} | \tilde{h} = h_{\text{s}}) f_{\tilde{h}}(\tilde{h} = h_{\text{s}})}{f_{\tilde{w}}(\tilde{w} = w_{\text{s}})} \tag{2.38}$$

  and in the same way one has to maximize the ***posterior*** term for MAP estimation.

- if one examines finite sub-collections $\mathcal{W}$ and $\mathcal{H}$, for Bayes' rule one would have:

$$f_{\mathcal{H}}(\mathcal{H} = H_{\text{s}} | \mathcal{W} = W_{\text{s}}) = \frac{f_{\mathcal{W}}(\mathcal{W} = W_{\text{s}} | \mathcal{H} = H_{\text{s}}) f_{\mathcal{H}}(\mathcal{H} = H_{\text{s}})}{f_{\mathcal{W}}(\mathcal{W} = W_{\text{s}})} \tag{2.39}$$

  so that one has to maximize the ***posterior*** term for MAP estimation. It is from Equation 2.39 that we will continue our analysis on Chapter 3.

3
Gaussian Processes

This chapter is about Gaussian Processes ($\mathcal{GP}$) and how one proceeds with $\mathcal{GP}$ regression. Here we will set up the fundamental knowledge basis in order to understand $\mathcal{GP}$.

3.1. Gaussian Process – an Introduction

Gaussian Processes ($\mathcal{GP}$) are stochastic processes that are regularly used to tackle machine learning problems. Nevertheless, they can be applied to other applications or fields that involve data manipulation [48]. Their fame in the field can be attributed to the fact that their approach is less algorithmic in nature (compared to e.g. Neural Networks methods) but is instead based on probability theory. $\mathcal{GP}$ methods fall within the Bayesian Statistics framework [4] and are non-parametric methods. The simplicity of the $\mathcal{GP}$ methods can facilitate good estimations in more than acceptable combinations of computer memory usage and computational time. This highlights them as potent substitutes of other more computationally demanding models – the original models as discussed in Chapter 1, whose $\mathcal{GP}$ are the surrogate ones. Taking into account all the above and the inherent link of $\mathcal{GP}$ to the univariate normal distribution, their application shows no limits, whether this is for learning algorithms, non-linear optimization problems or any other stochastic process analysis. Their implementation may include various types of problems such as regression, classification, clustering, optimization etc. [38]. Here our study will focus only on regression.

What is the link between the univariate normal distribution and the $\mathcal{GP}$ process as mentioned before? The univariate normal distribution is a probability distribution over an independent variable. $\mathcal{GP}$ is a **stochastic process** and is a generalization of a multivariate normal probability distribution, and so of the univariate as well. In the following sections, we first provide with some theoretical foundations upon which $\mathcal{GP}$ regression is based.

3.2. Theoretical Prerequisites

3.2.1. The Univariate and the Multivariate Gaussian Probability Distribution Function

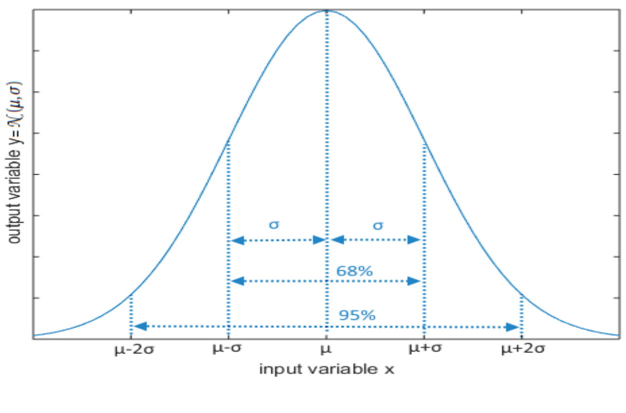


Figure 3.1: Univariate Normal Distribution $\mathcal{N}(x;\mu,\sigma)$ of a random variable $\tilde{x}$

In Figure 3.1 one sees a typical univariate normal (Gaussian) probability distribution function $\mathcal{N}(x;\mu,\sigma)$ of a random variable $\tilde{x}$. We denote the expected value $\mu = \mathbb{E}(\tilde{x})$ and the variance $\sigma = \mathrm{Var}(\tilde{x})$ of the pdf that also defines the confidence intervals. Its pdf has the following expression:

$$f_{\tilde{x}}(x) = \mathcal{N}(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.1}$$

On the other hand the multivariate Gaussian probability distribution function is a multivariate probability distribution of the dimension n which has the following expression:

$$f_{\tilde{\boldsymbol{x}}}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{3.2}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the expected value vector and the covariance matrix of this probability distribution. The definition of the covariance matrix will be introduced in the next subsection.

Here we used matrix notation for $\tilde{\boldsymbol{x}}$ because it is a collection of random variables $\{\tilde{x}_t : t \in T\}$, $|T| \in \mathbb{N}$. With this matrix notation – e.g. $\tilde{\boldsymbol{x}}$ – we mean to denote a **random vector** while with the usual matrix notation – e.g. $\boldsymbol{\mu}$ – we denote the determinate vectors. The above multivariate Gaussian distribution is the **joint** distribution for all individual elements of the random vector, that is, the finite collection of random variables $\{\tilde{x}_t : t \in T,\}$, $|T| \in \mathbb{N}$.

Imagine therefore such a finite collection of random variables $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_t$, $t \in T$, $|T| \in \mathbb{N}$ with joint Gaussian distribution as above, which are also the elements of the random vector $\tilde{\boldsymbol{x}}$. Consulting the Appendix Sections A.10 and A.11, one can verify that this condition ensures that: a) all individual random variables follow univariate Gaussian distributions and b) all conditional probabilities of the form $f_{\tilde{\boldsymbol{x}}_A}(\tilde{\boldsymbol{x}}_A|\tilde{\boldsymbol{x}}_B = \boldsymbol{x}_B)$ for any partition $\tilde{\boldsymbol{x}} = [\tilde{\boldsymbol{x}}_A \;\; \tilde{\boldsymbol{x}}_B]^\top$ is also Gaussian. So for a finite collection of random variables, by knowing only that their joint distribution is Gaussian, one knows implicitly everything about them. However for regression there are more things needed.

We have not yet introduced $\mathcal{GP}$ other than the fact that they are stochastic processes. But, based upon what has been discussed in the previous chapter about the link between $\mathcal{GP}$ regression and the study of **finite sub-collections of random variables**, for the next sections we can consider these finite sub-collections eligible to be written as random vectors. Also, we inform the reader that, as usually done in scientific literature about $\mathcal{GP}$ regression [12, 19, 32, 38, 55], we shall make arguments about what happens in $\mathcal{GP}$ regression using matrix notation, without denoting dimensions of vectors or matrices explicitly and the matrix notation – e.g. $\tilde{\boldsymbol{x}}$ – and the scalar notation – e.g. $\{\tilde{x}_t : t \in T\}$, $|T| \in \mathbb{N}$ – will be used interchangeably as appropriate.

3.2.2. Gaussian Process

But what is a $\mathcal{GP}$? Now comes the definition:

For any set X, a $\mathcal{GP}$ is set of random variables $\{\tilde{g}_x, : x \in X\}$ of which any finite sub-collection has a joint probability that is a multivariate Gaussian pdf.

It is already apparent that the implication of finite sub-collections as discussed above is not arbitrary; it is through them a $\mathcal{GP}$ is understood. It can be comprehended as a generalization of a multivariate Gaussian distribution for infinite-sized collections of real-valued variables (for example $X = \mathbb{N}$) or even uncountable collections of real-valued variables (for example $X = \mathbb{R}$). As already written, any stochastic process can be viewed also as a random function $\mathcal{G} = \tilde{g}(\omega, x)$ mapping every $x \in X$ and every $\omega \subseteq \Omega$ to the real line. This holds also for the Gaussian Process which has two important parameters that describe it, its **expected value** and its **covariance function**, which we will explain in more detail later. We will focus on finite sub-collections of $\mathcal{GP}$ so that we can write those as multivariate Gaussian probability distribution functions so that one may write:

$$\tilde{\boldsymbol{g}} \sim \mathcal{N}(\boldsymbol{g};\boldsymbol{\mu}_g,\boldsymbol{K}) \tag{3.3}$$

where $\boldsymbol{\mu}_g$ and $\boldsymbol{K}$ refer to the finite sub-collection only. This equation is exactly the same as Equation 3.2 because every finite sub-collection is a multivariate Gaussian random vector.

3.2.3. Covariance Matrix and Covariance Function

In Chapter 2, we encountered in Equation 2.17 the covariance of two random variables $\tilde{x}$ and $\tilde{y}$, $\mathrm{Cov}(\tilde{x},\tilde{y})$. If the variables are not two but many, they co-vary pairwise. If e.g. we assume $n \in \mathbb{N}$ random variables $\tilde{g}_1, \tilde{g}_2, \tilde{g}_3, \ldots, \tilde{g}_n$ we may want to construct an $n \times n$ representative matrix, the so-called covariance matrix $\boldsymbol{K}$ so that an element of the matrix $K_{i,j}$ [48]:

$$K_{i,j} = \mathrm{Cov}(\tilde{g}_i, \tilde{g}_j) \qquad \text{for all integers } 1 \le i,j \le n \tag{3.4}$$

It is also known as the variance-covariance matrix. The reason for that is that in Equation 3.4 for $i = j$ the values in the diagonal are the variance of the random variable. It is a symmetric matrix due to the fact that $\mathrm{Cov}(\tilde{g}_i, \tilde{g}_j) = \mathrm{Cov}(\tilde{g}_j, \tilde{g}_i)$.

Now, there is a **function concerning specifically Gaussian Processes**, that is used to **populate** the covariance matrix $\boldsymbol{K}$ (to define the value of its elements). This function is called **covariance function**. Given a Gaussian Process $\mathcal{G} = \{\tilde{g}_x, : x \in X\}$, indexed by a set X, for two elements of this index set $x_1, x_2 \in X$ the covariance function is a function that for those two arguments x_1 and x_2 it would have the expression $k(x_1, x_2) = \mathrm{Cov}(\tilde{g}_{x_1}, \tilde{g}_{x_2})$, where $\tilde{g}_{x_1}$ and $\tilde{g}_{x_2}$ are the corresponding random variables indexed by x_1 and x_2. Therefore for a random vector $\tilde{\boldsymbol{g}}$ of n elements, one would write:

$$K_{i,j} = \mathrm{Cov}(\tilde{g}_{x_i}, \tilde{g}_{x_j}) = k(x_i, x_j) \tag{3.5}$$

for all $x_i, x_j \in X$ and $1 \leq i \leq n,\ 1 \leq j \leq n$.

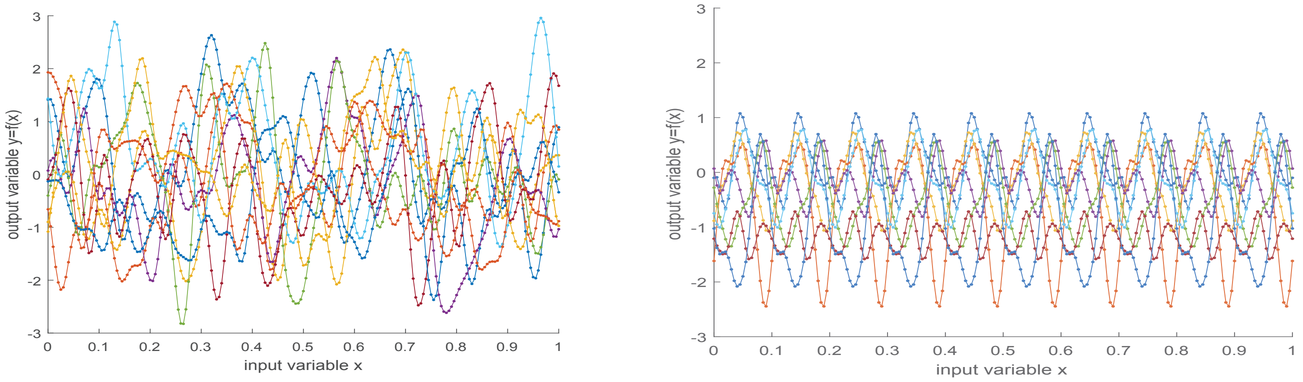
What exactly is the meaning of introducing such a function?. This function is introduced to represent the pairwise covariance of the random variables $\tilde{g}_1, \tilde{g}_2, \tilde{g}_3, \ldots, \tilde{g}_n$. **Altering the form of this function ends up affecting the structural properties of the stochastic process.** Let us take a look at the Figures 3.2a and 3.2b. In Figure 3.2a the covariance function is the so-called squared exponential function given as:

$$k(x, x') = \sigma_k^2 e^{\left(-\frac{1}{2}|x-x'|^2\right)} \tag{3.6}$$

for a predefined value of σ_k that signifies the **standard deviation of the stochastic process** – it is essentially the equivalent of the standard deviation for a univariate variable. For Figure 3.2b the covariance function is the periodic function:

$$k(x, x') = \sigma_k^2 e^{-2\sin^2\left(\left|\frac{x-x'}{2}\right|^2\right)} \tag{3.7}$$

It can be seen that specifying what the structural characteristics of the covariance function will be the realizations of the stochastic process will change dramatically. Because realizations of a stochastic process can be perceived as individual functions, it can be stated that the covariance function represents classes of functions that share structural characteristics, such as stationarity, isotropy, periodicity etc [38] (notice for example in Figure 3.2b all realizations of the stochastic process are periodic). We will not go into details here about all those structural properties; the meaning of those examples as seen in Figures 3.2a and 3.2b is to showcase that they play a big role on $\mathcal{GP}$ processes.



(a) Samples drawn drawn from the Gaussian process generated with the covariance function of the Equation 3.6

(b) Samples drawn drawn from the Gaussian process generated with the covariance function of the Equation 3.7

Figure 3.2: Two sets of sampled realizations of a Gaussian Process based on different covariance functions

It turns out that a covariance matrix **always needs to be positive semi-definite**. Conversely, it can also be proven that a symmetric, positive semidefinite matrix is always a covariance matrix of a random vector as well [52]. The covariance function should *always* ensure the construction of a symmetric, positive semi-definite matrix in order for it to be a proper covariance function.

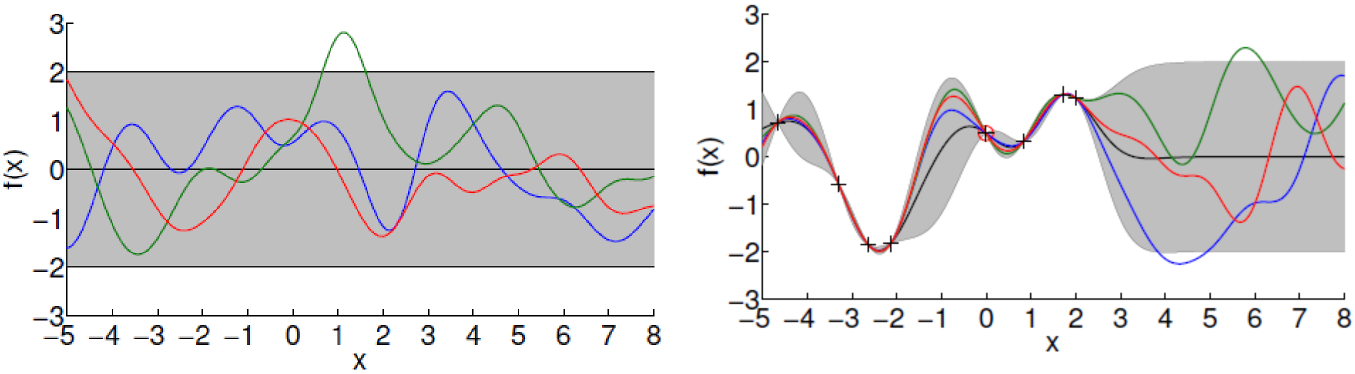
3.2.4. Sampling from a Finite Sub-Collection of a Gaussian Process

As already stated, despite the fact that we refer to continuous Gaussian Processes, we only need to work with finite sub-collections of their random variables. In this way also we can "safely" use matrix notation. Suppose therefore a finite sub-collection of a Gaussian process $\mathcal{G} = \{\tilde{g}_x : x \in X\}$, $|X| \in \mathbb{N}$. From this finite sub-collection we want to **draw samples** – that is, to specify evaluations for *some* random variables inside the collection of $\mathcal{G}$. We decide to draw samples for a subset X_{sampled} of X so that the initial stochastic process can be described as:

$$\mathcal{G} = \{\tilde{g}_x : x \in X_{\text{sampled}} \subset X\} \cup \{\tilde{g}_x : x \in X \setminus X_{\text{sampled}} \subset X\} \tag{3.8}$$

where the set difference $A \setminus B$ of two sets A and B is the set of objects that belong to A but not to B. Suppose for a moment that we have not yet sampled, but we know totally the collection of random vectors which we want to sample. By denoting these targeted random variables with the + symbol and the rest with the $*$ symbol we can write the stochastic process (in matrix notation) as $\tilde{\boldsymbol{g}} = [\tilde{\boldsymbol{g}}_+\ \tilde{\boldsymbol{g}}_*]^\top$. *After we have sampled*, the following event would have taken place; $\tilde{\boldsymbol{g}}_+ = \boldsymbol{g}_+$. A question now may arise: given the event $\tilde{\boldsymbol{g}}_+ = \boldsymbol{g}_+$ what is the expected value and the variability around it for the random vector $\tilde{\boldsymbol{g}}_*$? Essentially we are asking how does the pdf of $\tilde{\boldsymbol{g}}_*$ change, or better written, what is the **conditional probability distribution function** of $\tilde{\boldsymbol{g}}_*$ given that $\{\tilde{\boldsymbol{g}}_+ = \boldsymbol{g}_+\}$.

The most indicative example to understand this is to view the changes in Figures 3.3a and 3.3b. Figure 3.3a shows how the Gaussian process is perceived before sampling. As already explained, there is an expected value and a covariance function describing it. The expected value of the stochastic process is symbolized by the black line which for reasons of simplicity we considered to be zero for every index value x. The covariance function is implicitly represented by the grey area. The grey area represents the expected variability which here, for simplicity, is expected to be equal everywhere, for every index value x. This is how the situation stands before sampling the stochastic process. After sampling the stochastic process the situation changes and it is depicted in Figure 3.3b. We see the points $\{(x_+, g_+)\}$ we sampled and all possible realizations (colored lines) that as explained before, have to pass through those points. The black line now and the grey area also changed; this would mean that the probability distribution function of the stochastic process has also changed. The realizations of the stochastic process must now pass through the points we sampled from. Moving to the conditional probability distribution of the finite sub-collection of the Gaussian Process as stated before, means that the statistic measures of it – the expected value and the covariance matrix – have to change.



(a) Some realizations of a stochastic process before sampling

(b) Some realizations of a stochastic process that pass through the points $\{(x_+, g_+)\}$ we sampled.

Figure 3.3: A Gaussian Process before and after sampling. Retrieved from [10]. Here the output variable $f(x)$ should be substituted to g, to match out analysis above.

For the next sections we are going to consider the aforementioned framework by symbolizing sampled and not sampled quantities by the subscripts $+$ and $*$ and distinguish the following quantities:

- sampled and not sampled random vectors like $\tilde{\boldsymbol{g}}_+$ and $\tilde{\boldsymbol{g}}_*$ and
- determinate vectors $\boldsymbol{x}_+$, $\boldsymbol{g}_+$ and $\boldsymbol{x}_*$, the last symbolizing the index values corresponding to the elements of the random vector $\tilde{\boldsymbol{g}}_*$.
- Lastly, the vector $\tilde{\boldsymbol{g}}$ symbolizes the whole finite sub-collection $\mathcal{G}$ of the Gaussian process, $\boldsymbol{x}$ the whole collection of its index values, so essentially it is the set X written in matrix notation. It is also called the index vector

.

3.2.5. Noise and Variability

As considered above, after the sampling process is finished, the event $\{\tilde{\boldsymbol{g}}_+ = \boldsymbol{g}_+\}$ has taken place. This event led us to the statement that the variance around the expected value of the $\tilde{\boldsymbol{g}}_+$ is zero since we already designated its value through sampling. However, this is true as long as we believe that the measuring process involves **no effect from noise**.

Noise represents our uncertainty over the values we acquire through measurements. Noise, in general, is present in a data set because, whether one knows this data set from experiments or simulations, their internal computations **may be questionable** in terms of accuracy and precision. This leaves room for uncertainty in the data, expressed by **added variability** around the output values. This variability is not the same as what was discussed in previous subsections, noise comes from uncertainty over accuracy and precision coming from the model, not randomness coming from the expected value of random variables.

For example, imagine that we have a row of wind turbines and we measure the wind speed in front of them, with sample size e.g. 10 measurements. Each sample has its own variance $\mathrm{Var}(U_0)$ and since we expect that the upstream turbines affect the wind speed in the downstream turbines through wake propagation, we expect that their values co-vary so we can also estimate the $\mathrm{Cov}(U_{0_{\text{upstream}}}, U_{0_{\text{downstream}}})$ for all turbines. This constructs the covariance matrix $\boldsymbol{K}$. But we have to also take into account the variability from the anemometer that adds noise, **added variability**, to our wind speed data. Therefore, when one measures the **total variability** around the estimated value of a random variable, we should add the effect of both the covariance matrix $\boldsymbol{K}$ and the variability coming from the **noise**, since they are distinct!

In the concept of what was written before, there might be noise around the sampled vector $\boldsymbol{g}_+$ even after we have sampled. Remember now that for every element of $\tilde{\boldsymbol{g}}$ of the finite sub-collection $\mathcal{G}$ corresponds to an element of $\boldsymbol{x}$. We note here that the noise does not have to be the same for all elements of the vector $\boldsymbol{x}$, but may vary for different elements of it; For many problems though the usual practice is to consider the noise **homoscedastic**, that is irrespective of the values of the index values x. This means that the **standard deviation of the effect of noise** σ_n is common for all elements of the random vector $\tilde{\boldsymbol{g}}$, if the noise is modelled as **white Gaussian noise**. White Gaussian noise is a Gaussian process of the form:

$$\mathcal{E} = \{\tilde{\varepsilon}_x : x \in X\} \qquad \text{so that} \qquad \tilde{\varepsilon}_x \sim \mathcal{N}(\varepsilon; 0, \sigma_n^2) \tag{3.9}$$

If we consider $X_s \subset X$, $|X_s| \in \mathbb{N}$, we would take into account a finite sub-collection so that we would have a multivariate Gaussian pdf. This, as already written above can be expressed in matrix notation as [38]:

$$\tilde{\boldsymbol{\varepsilon}} \sim \mathcal{N}(\boldsymbol{\varepsilon}; \boldsymbol{0}, \sigma_n^2 \boldsymbol{I}) \tag{3.10}$$

Notice that $\sigma_n^2 \boldsymbol{I}$ is a covariance matrix as expected. The covariance between two different elements of the random vector $\tilde{\boldsymbol{\varepsilon}}$, e.g. random variables $\tilde{\varepsilon}_{x_1}$ and $\tilde{\varepsilon}_{x_2}$ is always zero due to the identity matrix $\boldsymbol{I}$, whose only non-zero elements are the diagonal ones catering for the variance of all $\tilde{\varepsilon}_x$. Because all the elements of the vector $\tilde{\boldsymbol{\varepsilon}}$ are normally distributed, zero covariance ensures independence and this means that noise $\tilde{\varepsilon}_{x_1}$ does not interfere with the noise $\tilde{\varepsilon}_{x_2}$, for any x_1 and x_2.

3.2.6. Regression with Noise Considered

Let us take a look now in Figures 3.3a and 3.3b: this is what regression would look like if we accepted no effect of noise involved. Knowing how the situation stands with the stochastic process before sampling as in Figure 3.3a, we observe a set of sampled data as in Figure 3.3b and the situation changes. In general, when we know that some points are valid and we suspect that there is an undergoing Gaussian Process justifying them as points, we believe in essence that they represent **samples from an underlying Gaussian Process**. That is, there exists some Gaussian Process with various possible realizations that a number of them, would pass through the points we observed.

Now let us assume that our knowledge of those points is not perfect because there is noise around the data involved and let us examine Figure 3.4. We see again some sampled data as yellow dots but now, there is also some variability around them which is explained by the involvement of noise. The red line is the expected value of the Gaussian Process but notice that it does not necessarily pass through the points as before. It does as best as it can to pass close to those points but the interference of noise affects how close to the points can the red line pass. The same would have happened with all realizations of the Gaussian Process after sampling (after considering those “sampled points”).

Since this is the case we would like to describe the situation above in mathematical formulation. First, we accept that the noise is independent as a phenomenon of the stochastic process whose existence we are assuming. Therefore there is no inner mechanism that links those two quantities and their joint effect can be considered to be the superposition of their individual effects. Following this last remark, it is time to follow up with the formulation of the fundamental theory of $\mathcal{GP}$ regression .

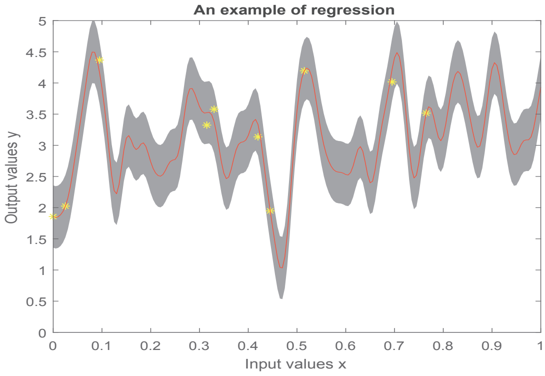


Figure 3.4: An example of regression with noise. Around the yellow points that represent the sampled data there is added variability from noise involved marked by noise. Such a situation is different with the one that was presented in Figure 3.3b since there the variability around the expected value of the sampled data points is zero, which makes their values **certain**.

3.3. Gaussian Process Regression - Fundamental Theory

As stated before we shall “conclude our business” with $\mathcal{GP}$ regression using finite sub-collections of random variables even if we assume the existence of continuous stochastic processes, that is, infinite collections of random variables. The consideration of noise brings forth the following Equation in terms of **continuous stochastic processes**:

$$\mathcal{Y} = \mathcal{G} + \mathcal{E} \tag{3.11}$$

It involves three things:

- A Gaussian Process $\mathcal{G}$ that we may sample from it (as done in subsection 3.2.4) which we assume is the "underlying Gaussian Process" that gives meaning to the samples that have been observed – more on that in Section 3.2.6.
- The added variability coming from noise $\mathcal{E}$, which we consider also a Gaussian Process (in accordance with what is written above and in subsection 3.2.5). Again we are considering the existence of noise and the existence of the underlying Gaussian Process to be independent so that their effects can be superimposed, as done in Equation 3.11.
- The resulting stochastic process $\mathcal{Y}$ which is the result of what the observer would obtain when these two independent Gaussian Processes ($\mathcal{G}$ and $\mathcal{E}$) add their effects.

So we accepted that $\mathcal{G}$ and $\mathcal{E}$ are Gaussian Processes and specifically, defined on the same set of index values and of equal cardinalities – in order for Equation 3.11 to hold any meaning. What is the resulting stochastic process $\mathcal{Y}$ then? Is it a Gaussian Process as well?

To answer this, it is easier to study the problem when it is written in matrix notation, so for finite sub-collections of all stochastic processes involved. For these finite sub-collections, one then would write:

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{g}} + \tilde{\boldsymbol{\varepsilon}} \tag{3.12}$$

Consulting the Appendix Section A.8, we reach the conclusion that the sum of two **independent** Gaussian random vectors is also a Gaussian random vector whose expected value is the sum of the expected values and its covariance matrix is the sum of its covariance matrices! So one can write:

$$\boxed{\begin{array}{c} \tilde{\boldsymbol{\varepsilon}} \sim \mathcal{N}(\boldsymbol{\varepsilon}; \boldsymbol{0}, \sigma_n^2 \boldsymbol{I}) \\ \tilde{\boldsymbol{g}} \sim \mathcal{N}(\boldsymbol{g}; \boldsymbol{\mu}_g, \boldsymbol{K}) \\ \tilde{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_g, \sigma_n^2 \boldsymbol{I} + \boldsymbol{K}) \end{array}} \qquad \begin{array}{r} (3.13\text{a}) \\ (3.13\text{b}) \\ (3.13\text{c}) \end{array}$$

Equations 3.13a to 3.13c are the three of the four most fundamental Equations for $\mathcal{GP}$ regression! Now let us proceed with the fourth and most fundamental Equation, whose derivation also comes after consulting the Appendix Section A.11.

Assume a partition of the random vector $\tilde{\boldsymbol{g}}$ in two random vectors $\tilde{\boldsymbol{g}} = [\tilde{\boldsymbol{g}}_+ \ \tilde{\boldsymbol{g}}_*]^\top$ and a partition of the random vector $\tilde{\boldsymbol{\varepsilon}}$ in two random vectors $\tilde{\boldsymbol{\varepsilon}} = [\tilde{\boldsymbol{\varepsilon}}_+ \ \tilde{\boldsymbol{\varepsilon}}_*]^\top$. As before the partition element $+$ will symbolize random variables from which we have sampled from, while partition element $_*$ will symbolize which we have not sampled. Considering these partitions above we can end up writing $\tilde{\boldsymbol{y}} = [\tilde{\boldsymbol{y}}_+ \ \tilde{\boldsymbol{y}}_*]^\top = [\tilde{\boldsymbol{g}}_+ + \tilde{\boldsymbol{\varepsilon}}_+ \ \ \tilde{\boldsymbol{g}}_* + \tilde{\boldsymbol{\varepsilon}}_*]^\top$. Therefore we start by partitioning the vectors $\tilde{\boldsymbol{g}}$ and $\tilde{\boldsymbol{\varepsilon}}$. We would have:

$$\begin{bmatrix} \tilde{\boldsymbol{g}}_+ \\ \tilde{\boldsymbol{g}}_* \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{\mu}_{g_+} \\ \boldsymbol{\mu}_{g_*} \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{K}_{++} & \boldsymbol{K}_{+*} \\ \boldsymbol{K}_{*+} & \boldsymbol{K}_{**} \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} \tilde{\boldsymbol{\varepsilon}}_+ \\ \tilde{\boldsymbol{\varepsilon}}_* \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{I}_{++} & \boldsymbol{I}_{+*} \\ \boldsymbol{I}_{*+} & \boldsymbol{I}_{**} \end{bmatrix} \sigma_n^2 = \begin{bmatrix} \boldsymbol{I}_{++} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{**} \end{bmatrix} \sigma_n^2 \tag{3.14}$$

Therefore the partitioning $\tilde{\boldsymbol{y}} = [\tilde{\boldsymbol{y}}_+ \ \tilde{\boldsymbol{y}}_*]^\top$ will be of the following form:

$$\begin{bmatrix} \tilde{\boldsymbol{y}}_+ \\ \tilde{\boldsymbol{y}}_* \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{\mu}_{g_+} \\ \boldsymbol{\mu}_{g_*} \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{K}_{++} + \boldsymbol{I}_{++}\sigma_n^2 & \boldsymbol{K}_{+*} \\ \boldsymbol{K}_{*+} & \boldsymbol{K}_{**} + \boldsymbol{I}_{**}\sigma_n^2 \end{bmatrix} \tag{3.15}$$

Let us proceed to the following assumptions/considerations:

- The matrices $\boldsymbol{K}_{++}$, $\boldsymbol{K}_{**}$, $\boldsymbol{K}_{+*}$ and $\boldsymbol{K}_{*+}$ symbolize the pairwise covariance between elements of the random vectors $\tilde{\boldsymbol{g}}_+$ and $\tilde{\boldsymbol{g}}_*$. Their elements are populated by the covariance function $k(x, x')$ for all $x, x' \in X$. The matrices $\boldsymbol{I}_{++}$, $\boldsymbol{I}_{**}$, $\boldsymbol{I}_{+*}\sigma_n^2$ and $\boldsymbol{I}_{*+}\sigma_n^2$ symbolize the pairwise covariance between elements of the random vectors $\tilde{\boldsymbol{\varepsilon}}_+$ and $\tilde{\boldsymbol{\varepsilon}}_*$. Their elements are populated by the factor σ_n^2 multiplied with the Kronecker delta function $\delta(x, x')$, that yields $\delta(x, x') = 0$ for all $x \neq x'$ and $\delta(x, x') = 1$ for all $x = x'$, where $x, x' \in X$ in both cases.

  Therefore we can introduce a new matrix $\boldsymbol{C}$ whose elements are $\left\{C_{i,j}\right\} = \left\{k(x_i, x_j) + \sigma_n^2 \delta(x_i, x_j)\right\}$ for all $x_i, x_j \in X$ and for all integers $1 \leq i \leq n, 1 \leq j \leq n$ so that $\boldsymbol{C} = \boldsymbol{K} + \sigma_n^2 \boldsymbol{I}$. This would lead to the partitioning of the initial vector $\tilde{\boldsymbol{y}}$ as:

$$\begin{bmatrix} \tilde{\boldsymbol{y}}_+ \\ \tilde{\boldsymbol{y}}_* \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{\mu}_{g_+} \\ \boldsymbol{\mu}_{g_*} \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{C}_{++} & \boldsymbol{C}_{+*} \\ \boldsymbol{C}_{*+} & \boldsymbol{C}_{**} \end{bmatrix} \tag{3.16}$$

Consulting the Appendix Section A.10, one finds that both partition elements $\tilde{\boldsymbol{y}}_+\tilde{\boldsymbol{y}}_*$ and $\tilde{\boldsymbol{y}}_*$ follow a Gaussian pdf. Then one can find the following conditional distribution, again consulting the Appendix Section A.10:

$$\boxed{f_{\tilde{\boldsymbol{y}}_*}(\boldsymbol{y}_* | \tilde{\boldsymbol{y}}_+ = \boldsymbol{y}_+) = \mathcal{N}(\boldsymbol{y}_*; \ \boldsymbol{\mu}_g + \boldsymbol{C}_{*+}\boldsymbol{C}_{++}^{-1}(\boldsymbol{y}_+ - \boldsymbol{\mu}_g), \ \boldsymbol{C}_{**} - \boldsymbol{C}_{*+}\boldsymbol{C}_{++}^{-1}\boldsymbol{C}_{+*})} \tag{3.17}$$

This one is the fourth and most useful Equation in the $\mathcal{GP}$ regression model, the results of which we are going to discuss in the following section. We inform the reader however that this set of four Equations hold only if the noise is **homoscedastic** and **independent** from the underlying stochastic process, that is, if their joint probability distribution satisfies the condition of independence shown in Equation 2.10.

3.4. Discussion of the Previous Results

We partitioned the $\tilde{\boldsymbol{y}}$ in two random vectors, the random vector from which we sampled $\tilde{\boldsymbol{y}}_+$ and the one we did not $\tilde{\boldsymbol{y}}_*$. If we pair them with their index vectors $\boldsymbol{x}_+$ and $\boldsymbol{x}_*$ one would get the **set of non-sampled points** R whose elements are the pairwise combinations of the vectors $\boldsymbol{x}_*$ and $\tilde{\boldsymbol{y}}_*$ so that $R = \{(x_{*_1}, \tilde{y}_{*_1}), (x_{*_2}, \tilde{y}_{*_2}), (x_{*_3}, \tilde{y}_{*_3}), \ldots, (x_{*_{|R|}}, \tilde{y}_{*_{|R|}})\}$ and the **data set of sampled points** $D = \{(x_{+_1}, y_{+_1}), (x_{+_2}, y_{+_2}), (x_{+_3}, y_{+_3}), \ldots, (x_{+_{|D|}}, y_{+_{|D|}})\}$ whose elements are the pairwise combinations of the vectors $\boldsymbol{x}_+$ and $\boldsymbol{y}_+$, where $\boldsymbol{y}_+$ is the evaluation of the "sampled" random vector $\tilde{\boldsymbol{y}}_+$.

Let us get familiarized with the submatrices of $\boldsymbol{C}$:

- $\boldsymbol{C}_{++}$ is the matrix that expresses the pairwise covariance of the elements of the random vector $\tilde{\boldsymbol{y}}_+$. This covariance is calculated from the expression $k(x_{+_i}, x_{+_j}) + \sigma_n^2 \delta(x_{+_i}, x_{+_j})$ for two elements of the index vector $\boldsymbol{x}_+$, x_{+_i} and x_{+_j}.

- $\boldsymbol{C}_{+*}$ is the matrix that expresses the pairwise covariance of the elements of the random vector $\tilde{\boldsymbol{y}}_+$ with elements of the random vector $\tilde{\boldsymbol{y}}_*$. This covariance is calculated from the expression $k(x_{+_i}, x_{*_i})$ for an element of the index vector $\boldsymbol{x}_+$, x_{+_i} and an element of the index vector $\boldsymbol{x}_*$, x_{*_i}.

- $\boldsymbol{C}_{*+}$ is the matrix that expresses the pairwise covariance of the elements of the random vector $\tilde{\boldsymbol{y}}_*$ with elements of the random vector $\tilde{\boldsymbol{y}}_+$. This covariance is calculated from the expression $k(x_{*_i}, x_{+_i})$ for an element of the index vector $\boldsymbol{x}_*$, x_{*_i} and an element of the index vector $\boldsymbol{x}_+$, x_{+_i}. Notice that because a covariance matrix must always be symmetric we have always $\boldsymbol{C}_{*+} = \boldsymbol{C}_{+*}^\top$

- $\boldsymbol{C}_{**}$ is the matrix that expresses the pairwise covariance of the elements of the random vector $\tilde{\boldsymbol{y}}_*$. This covariance is calculated from the expression $k(x_{*_i}, x_{*_j}) + \sigma_n^2 \delta(x_{*_i}, x_{*_j})$ for two elements of the index vector $\boldsymbol{x}_*$, x_{*_i} and x_{*_j}.

By Equation 3.17 we know that:

$$\mathbb{E}(\tilde{\boldsymbol{y}}_* | \boldsymbol{y}_+, \boldsymbol{x}_*, \boldsymbol{x}_+) = \boldsymbol{\mu}_g + \boldsymbol{C}_{*+} \boldsymbol{C}_{++}^{-1} (\boldsymbol{y}_+ - \boldsymbol{\mu}_g) \tag{3.18}$$

$$\operatorname{Cov}(\tilde{\boldsymbol{y}}_*, \tilde{\boldsymbol{y}}_* | \boldsymbol{y}_+, \boldsymbol{x}_*, \boldsymbol{x}_+) = \boldsymbol{C}_{**} - \boldsymbol{C}_{*+} \boldsymbol{C}_{++}^{-1} \boldsymbol{C}_{+*} \tag{3.19}$$

Here we offer some remarks about the results. First it can be seen from Equations 3.18 and 3.19 that the expected values for the elements of the "unsampled" random vector $\tilde{\boldsymbol{y}}_*$ are a linear combination of the elements of the evaluation $\boldsymbol{y}_+$ of the "sampled" random vector $\tilde{\boldsymbol{y}}_+$ which is expected in linear regression analysis. This linearity is also affected by the variability of the sampled vector $\tilde{\boldsymbol{y}}_+$ represented by the submatrices $\boldsymbol{C}_{++}, \boldsymbol{C}_{*+}$ and the expected value of the $\mathcal{GP}$, as depicted by the expected value vector $\boldsymbol{\mu}_g$.

About the covariance for the elements of the "unsampled" random vector $\tilde{\boldsymbol{y}}_*$, $\operatorname{Cov}(\tilde{\boldsymbol{y}}_*, \tilde{\boldsymbol{y}}_* | \boldsymbol{y}_+, \boldsymbol{x}_*, \boldsymbol{x}_+)$: the submatrix $\boldsymbol{C}_{**}$ represents our belief concerning the variability of the elements of the "unsampled" random vector $\tilde{\boldsymbol{y}}_*$ *before we sample*. We have already stated that the pdf of the finite sub-collection of the Gaussian Process changes *after sampling*. This change is depicted as far as the variability around the expected value of the "unsampled" random vector $\tilde{\boldsymbol{y}}_*$ is concerned by subtracting the *always non-negative* term $\boldsymbol{C}_{*+} \boldsymbol{C}_{++}^{-1} \boldsymbol{C}_{+*}$ from the initial submatrix $\boldsymbol{C}_{**}$. In the scientific literature [13, 19, 32, 38, 48, 55] one speaks about **prior covariance matrix** ($\boldsymbol{C}_{**}$) and **posterior covariance matrix** ($\boldsymbol{C}_{**} - \boldsymbol{C}_{*+} \boldsymbol{C}_{++}^{-1} \boldsymbol{C}_{+*}$ or simply $\operatorname{Cov}(\tilde{\boldsymbol{y}}_*, \tilde{\boldsymbol{y}}_* | \boldsymbol{y}_+, \boldsymbol{x}_*, \boldsymbol{x}_+)$ as above). This effect was seen in Figures 3.3a and 3.3b where the variability around the expected value of every output (the grey area around the black line) changed after sampling.

Therefore, imagine now that we are considering the covariance function $k(x, x') = \sigma_k^2 e^{-\frac{1}{2}|x - x'|^2}$, which is a typical option for many problems. Assume that we are examining the submatrix $\boldsymbol{C}_{*+}$ so that, x' is an element of the index vector $\boldsymbol{x}_*$ and x is an element of the index vector $\boldsymbol{x}_+$. It should be apparent that if the horizontal distance between the already sampled points and the point (x', y') is too great so that $|x - x'| \to \infty$ then the covariance function converges to zero (that is $k(x, x') \to 0$), so that $\boldsymbol{C}_{+*} = \boldsymbol{C}_{+*} \to \boldsymbol{0}$. This would also lead to $\mathbb{E}(\tilde{\boldsymbol{y}} | \boldsymbol{y}_+, \boldsymbol{x}_*, \boldsymbol{x}_+) \to \boldsymbol{\mu}_g$ and $\operatorname{Cov}(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{y}} | \boldsymbol{y}_+, \boldsymbol{x}_*, \boldsymbol{x}_+) \to \boldsymbol{C}_{**}$.

Therefore, when co-variation of elements of the elements of the "sampled" random vector $\tilde{\boldsymbol{y}}_+$ and the elements of the "unsampled" random vector $\tilde{\boldsymbol{y}}_*$ goes to zero, then the value of the "unsampled" random vector $\tilde{\boldsymbol{y}}_*$ is determined by the expected value vector $\boldsymbol{\mu}_g$ and the variability around this value **stays the same**.

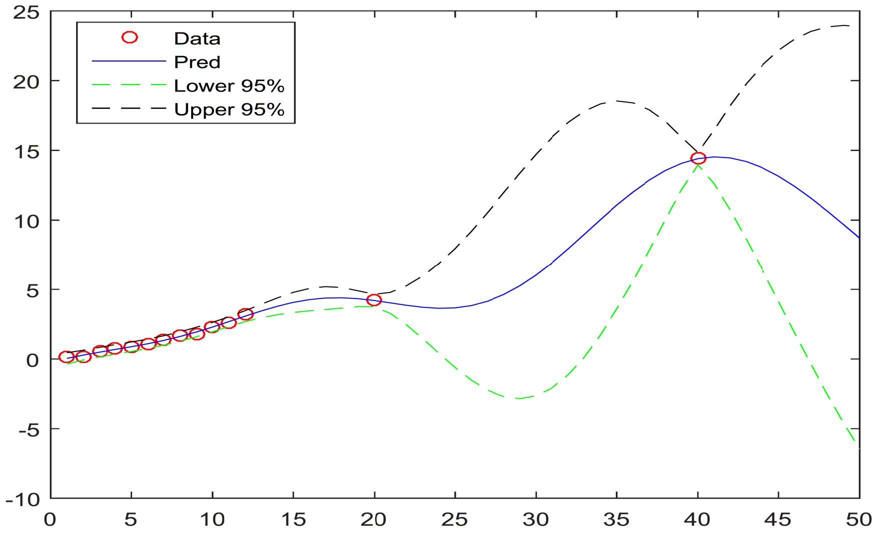


Figure 3.5: A simple example of Gaussian Process Regression

Figure 3.5 shows again the result of the $\mathcal{GP}$ regression. We considered again the resulting Gaussian Process $\mathcal{Y}$ being the sum of the underlying Gaussian Process $\mathcal{G}$ and the white noise (also a Gaussian Process) $\mathcal{E}$, as already explained before. In the graph one can see:

- the data set of the **sampled points** $D = \{(x_{+1}, y_{+1}), (x_{+2}, y_{+2}), (x_{+3}, y_{+3}), \ldots, (x_{+|D|}, y_{+|D|})\}$ (red dots),
- the set of the **non-sampled points** $R = \{(x_{*1}, \tilde{y}_{*1}), (x_{*2}, \tilde{y}_{*2}), (x_{*3}, \tilde{y}_{*3}), \ldots, (x_{*|R|}, \tilde{y}_{*|R|})\}$ which is not represented through dots but it is represented with
  1. the expected value of the resulting stochastic process $\mathcal{Y}$ (blue line), **after** we have sampled from it some of its random variables to form the data set of the **sampled points** D,
  2. the **confidence intervals** around this expected value, indicated with dotted lines.
- the added variability from noise can be seen in the confidence intervals around the red dots in the graph.

Notice that the expected value of the resulting stochastic process is **conditioned** through the points of the data set and has the tendency to drop to zero for other points away from it (notice the “drop” of the blue line after the rightmost red dot). This is because the covariance function used was the squared exponential function of Equation 3.6 and as stated above for points with a big horizontal distance from the points of the “sampled” vector, their expected value tends to the expected value vector. The confidence interval correspond to a certain confidence level and in our example is 95%.

It should be apparent already that the matrix $\boldsymbol{C}$ and its submatrices of Equations 3.18 and 3.19 play a big role in predicting the expected value and the covariances of the “unsampled” random vector $\tilde{\boldsymbol{y}}_*$. This matrix is itself defined by the covariance function and how we define it. How probable is our selection of the covariance function though? This question is going to be discussed in the next section.

3.5. Maximum A Posteriori Estimation of Hyperparameters

The covariance function was crucial in order to acquire the previous results. As was shown in the Figures 3.2a and 3.2b, it affects the structural properties of the stochastic process and as was shown in Equations 3.18 and 3.19 it underpins the results of the statistical measures of the “unsampled” random vector. Essentially, it plays the role of the parameter that was discussed in Section 2.10 concerning Parameter Estimation.

This parameter (covariance function) however is also underpinned by a set of some other parameters, termed *hyperparameters* – we will introduce them later on. All of these hyperparameters form the **random vector of hyperparameters** $\tilde{\boldsymbol{\gamma}}$. It is best that there is some probabilistic framework to work with all these hyperparameters and not leave the decision of them solely to the modeler. This is because Bayes’ rule and MAP can indicate which values for these hyperparameters conform best with the data. What is typically considered is that the covariance function can be selected by the modeler, since he might know beforehand the generic structural properties of the Gaussian Process, but instead the set of hyperparameters are continuous random variables and cannot beforehand be selected with full certainty.

Following then Equation 2.39 and writing everything in matrix notation we can introduce Bayes’ rule relating the “sampled” random vector $\tilde{\boldsymbol{y}}_+$ and the hyperparameters. Of course, in case the reader wonders, the estimation of the hyperparameters through MAP *will affect* the “unsampled” random vector $\tilde{\boldsymbol{y}}_*$, but we cannot implement Bayes’ rule on it, simply because by definition, we do not sample from it. The Equation we are looking for therefore is:

$$f_{\tilde{\boldsymbol{\gamma}}}(\boldsymbol{\gamma}|\tilde{\boldsymbol{y}}_+ = \boldsymbol{y}_+) = \frac{f_{\tilde{\boldsymbol{y}}_+}(\boldsymbol{y}_+|\tilde{\boldsymbol{\gamma}} = \boldsymbol{\gamma}) f_{\tilde{\boldsymbol{\gamma}}}(\boldsymbol{\gamma})}{f_{\tilde{\boldsymbol{y}}_+}(\boldsymbol{y}_+)}, \tag{3.20}$$

or as seen again:

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{data}}, \tag{3.21}$$

where $\tilde{\boldsymbol{y}}_+$ is the "sampled" random vector, $\boldsymbol{y}_+$ its evaluations and the random vector $\tilde{\boldsymbol{\gamma}}$ is the random vector of hyperparameters (more on that later) which the "sampled" random vectors $\boldsymbol{y}_+$ and $\boldsymbol{y}_*$ are affected by.

Again we have four terms as before: ***prior, data, likelihood*** and ***posterior***, from which we want to maximize the last one, as MAP estimation dictates. We have already explained in detail in Section 2.10 what is the reason why the ***data*** term is invariant for the maximization procedure, and what motivates the modeller to choose a weak prior. Therefore one can write:

$$\text{posterior} \propto \text{likelihood} \cdot \text{prior} \tag{3.22}$$

Then one would get that:

$$f_{\tilde{\boldsymbol{\gamma}}}(\boldsymbol{\gamma}|\tilde{\boldsymbol{y}}_+ = \boldsymbol{y}_+) \propto f_{\tilde{\boldsymbol{y}}_+}(\boldsymbol{y}_+|\tilde{\boldsymbol{\gamma}} = \boldsymbol{\gamma}) \tag{3.23}$$

So that only the ***likelihood*** term can be considered to play a significant role for the maximization of the ***posterior*** term.

3.6. Varying the Hyperparameters

First we have to introduce the hyperparameters that constitute the random vector $\tilde{\boldsymbol{\gamma}}$. We've already presented some of them before: the standard deviation of noise σ_n, the standard deviation of the stochastic process σ_k and the length scale l which we have not talked about before. The hyperparameters designate the behavior of the resulting stochastic process $\mathcal{Y}$ via its covariance matrix $\boldsymbol{C}$. Recall though that all elements of this matrix are populated as $\left[C_{i,j}\right]_{(1,1)}^{(n,n)} = \left[k(x_i, x_j) + \sigma_n^2 \delta(x_i, x_j)\right]_{(1,1)}^{(n,n)}$ for all $x_i, x_j \in X$, so that $\boldsymbol{C} = \boldsymbol{K} + \sigma_n^2 \boldsymbol{I}$. Let us now reintroduce the covariance function with the length scale which before we ignored – simply because we did not need at the time to initiate the discussion over it. We will have for example, the squared exponential covariance function as:

$$k(x, x') = \sigma_k^2 e^{\left(-\frac{1}{2l^2}|x-x'|^2\right)} = \sigma_k^2 e^{\left(-\frac{1}{2}\left|\frac{x}{l} - \frac{x'}{l}\right|^2\right)} \tag{3.24}$$

which differs from Equation 3.6 due to the length scale l. This triplet (σ_n, σ_k, l) is the vector of hyperparameters.

As noted above we have to view these hyperparameters as random variables for applying MAP estimation so one wants to find which exact evaluations $\sigma_{n_{\text{MAP}}}, \sigma$

In this way one can state that $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{g}}$, thanks to the fact that $\mathbb{E}(\tilde{\boldsymbol{\varepsilon}}) = \boldsymbol{0}$! This of course is means that we can immediately "ignore" the random vector $\tilde{\boldsymbol{\varepsilon}}$ as we have already incorporated its effect inside the random vector $\tilde{\boldsymbol{g}}$. The ***likelihood*** term therefore is the following:

$$f_{\tilde{\boldsymbol{y}}_+}(\tilde{\boldsymbol{y}}_+|\tilde{\boldsymbol{\gamma}} = \boldsymbol{\gamma}) \sim \mathcal{N}(\tilde{\boldsymbol{y}}_+; \boldsymbol{0}, \boldsymbol{C}_{++}) \tag{3.25}$$

where we noted again that, as before, we designated $\boldsymbol{\mu}_g = \boldsymbol{0}$. Notice that the above equation makes sense because $\boldsymbol{C}_{++}$ *is* a function of $\boldsymbol{\gamma}$. The ***posterior*** term is maximized when this above Equation is maximized. With the aid of the logarithmic function that is an increasing monotonic function we may write:

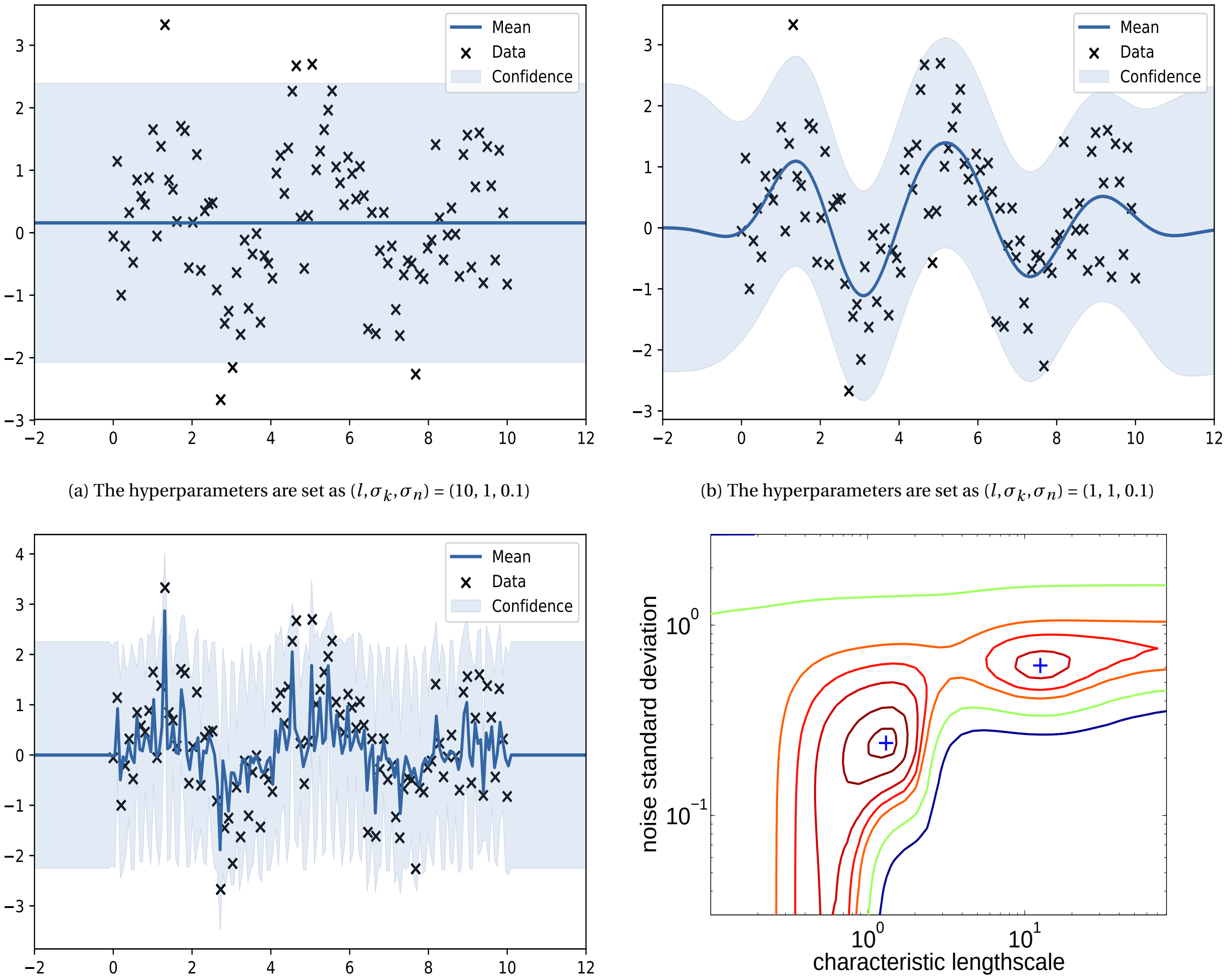
$$\log\!\left(f_{\tilde{\boldsymbol{y}}_+}(\boldsymbol{y}_+|\tilde{\boldsymbol{\gamma}} = \boldsymbol{\gamma})\right) \propto \log\!\left(\mathcal{N}(\boldsymbol{y}_+; \boldsymbol{0}, \boldsymbol{C}_{++})\right) = \log\!\left(\frac{1}{2\pi^{n/2}|\boldsymbol{C}_{++}|^{1/2}} e^{\left(-\frac{1}{2}\boldsymbol{y}_+{}^\top \boldsymbol{C}_{++}{}^{-1}\boldsymbol{y}_+\right)}\right) = -\frac{1}{2}\boldsymbol{y}_+{}^\top \boldsymbol{C}_{++}{}^{-1}\boldsymbol{y}_+ - \frac{1}{2}\log|\boldsymbol{C}_{++}| - \frac{n}{2}\log 2\pi, \tag{3.26}$$

so that $\boldsymbol{\gamma}_{\text{MAP}}$ is calculated by solving:

$$\frac{\mathrm{d}\overbrace{\left(-\frac{1}{2}\boldsymbol{y}_+{}^\top \boldsymbol{C}_{++}{}^{-1}\boldsymbol{y}_+ - \frac{1}{2}\log|\boldsymbol{C}_{++}|\right)}^{\text{function of } \boldsymbol{\gamma}}}{\mathrm{d}\boldsymbol{\gamma}} = 0 \tag{3.27}$$

Through this procedure one can estimate the triplet $(\sigma_{n_{\text{MAP}}}, \sigma_{k_{\text{MAP}}}, l_{\text{MAP}})$. In essence, one has to find the extremal (maximal) points of this function of $\tilde{\boldsymbol{\gamma}}$, where its derivative is equal to zero. We will not go into further detail how the derivation of the collection γ_{MAP} is done but we will note that there might be many *local* extremal point in the above noted function of $\tilde{\boldsymbol{\gamma}}$ so if an algorithm is employed to "search" for the value of $\tilde{\boldsymbol{\gamma}}$ that finds them, it has to take into account that the task is to find the *global extremal point and not just local ones.* This way the extremization of the posterior term would be surely the best achieved. There are various multivariate optimization algorithms that can be employed in order to tackle this that in general guarantee convergence to a global extremal point [13].

The length scale's effect can be better understood by examining the Figures 3.6a, 3.6b and 3.6c: it essentially determines the "area of influence" of the training points to other testing points. One can notice that as far as the expected value of the Gaussian Process is concerned, represented by the blue line: for a given vertical distance on the x axis, the y evaluation of the blue line each time varies all the more for decreasing lengths scales.



(a) The hyperparameters are set as $(l, \sigma_k, \sigma_n) = (10, 1, 0.1)$

(b) The hyperparameters are set as $(l, \sigma_k, \sigma_n) = (1, 1, 0.1)$

(c) The hyperparameters are set as $(l, \sigma_k, \sigma_n) = (0.03, 1, 0.1)$

(d) The search space for two hyperparameters, length scale and noise, for a given $\sigma_k = 1$. Reproduced from [38]

Figure 3.6: The effect of the length scale. In Figures 3.6a, 3.6b and 3.6c the effect of the length scale is showed by keeping the other hyperparameters constant. The crosses represent the training points and they are the same in all graphs. The shaded regions obtained represent 95% confidence intervals in each case. In Figure 3.6d one sees the use of a graphical representation to find where the function of $\boldsymbol{\gamma}$ of Equation 3.27 is extremal provided a given value of one of the hyperparameters. It can be seen that it may happen that more than one extremal point appear.

However the three Figures 3.6a, 3.6b and 3.6c were not made following MAP estimation and solving Equation 3.27; the triplet of the hyperparameters were designated by the modeler, simply to illustrate the effect of the length scale. If one does follow the procedure dictated by MAP estimation and does solve the Equation 3.27 then he should comprehend that the derivative of the function of $\tilde{\boldsymbol{\gamma}}$ that this Equation introduces may have more than one extremal local points. Therefore more than one solution for estimating the triplet $(\sigma_{n_{\mathrm{MAP}}}, \sigma_{k_{\mathrm{MAP}}}, l_{\mathrm{MAP}})$ could be possible. This means that if the modeler shows some preference to e.g. smaller length scale, that may mean by necessity he needs to have also the rest of the hyperparameters changed simultaneously in order to get to the possible solution of γ_{MAP}. In order to illustrate this better, Figure 3.6d for a given evaluation of σ_k equal to 1, how the two other hyperparameters behave. Of course, it is only the two out of three hyperparameters that change here, but the general notion is the same.

3.7. Gaussian Process Regression

Usually in the scientific literature this chapter is presented very early when introducing $\mathcal{GP}$ regression [32, 38, 55] but in our case we reversed the order as this section is **more complex** than the fundamental theory that was discussed before in Sections 3.3 and 3.4 and builds upon the discussion that was done there. Now that the main methodology on how to find the conditional pdf $f_{\tilde{\boldsymbol{y}}_*}(\tilde{\boldsymbol{y}}_*|\tilde{\boldsymbol{y}}_+ = \boldsymbol{y}_+)$ of Equation 3.17 is well deployed, everything else here can be understood quite easily.

Before going into more detail what all this means let us remind ourselves how the simple two-dimensional linear regression looks like:

$$g = g(x) = w_0 + w_1 x \qquad \text{for} \qquad x \in \mathbb{R}, \tag{3.28}$$

where the slope is w_1 and the intercept is w_0 and $x \in \mathbb{R}$. The output values are represented by y of course and can be visualized by a linear 2D plot. Similarly one may extend this model to n dimensions such that:

$$g = g(x) = w_0 + w_1 x + \cdots + w_{n-1}x^{n-1} + w_{n-2}x^{n-2} + w_n x^n \qquad \text{for} \qquad x \in \mathbb{R}, \tag{3.29}$$

with $w_n, w_{n-1}, w_{n-2}, \ldots, w_1, w_0$ being the weights and $x \in \mathbb{R}$. What would happen if we considered the fact that these weights varied randomly in order for the output to be a random variable as well? This turns the polynomial model to the study of a stochastic process since the weights are random!

To this end, let us consider therefore a finite sub-collection of a Gaussian process $\mathcal{W} = \{\tilde{w}_t \sim \mathcal{N}(\tilde{w}_t; 0, \sigma_w) : t \in T\}$, $|T| \in \mathbb{N}$ indexed with a set T of finite dimensionality. This finite sub-collection of the Gaussian process represents the weights; each one of them follows a univariate Gaussian distribution with expected value equal to zero and variance equal to σ_w. Lastly notice that we can write $\mathcal{W}$ also as a random vector $\tilde{\boldsymbol{w}} = [\tilde{w}_0\ \tilde{w}_1\ \tilde{w}_2\ \tilde{w}_3 \ldots\ \tilde{w}_n]^\top \sim \mathcal{N}(\tilde{\boldsymbol{w}}; \boldsymbol{0}, \boldsymbol{\Sigma}_w)$. Then, assume a finite sub-collection of a stochastic process:

$$\mathcal{G} = \{\tilde{g}_x = \tilde{w}_0 + x\tilde{w}_1 + x^2\tilde{w}_2 + x^3\tilde{w}_3 + \cdots + x^n\tilde{w}_n : x \in X\},\ |X| \in \mathbb{N} \tag{3.30}$$

indexed on the **some other finite set** X! This means that this set has potentially some other cardinality, e.g. $|X| \in \mathbb{N}$. In order for the reader to understand why this was necessary, he might return to Equation 3.28 and notice that the weights considered may be for example only two $(\tilde{w}_0, \tilde{w}_1)$ while the number of the random variables $\tilde{g}_x = \tilde{w}_0 + \tilde{w}_1 x$ might be arbitrary large to define their joint pdf and their finite sub-collection $\mathcal{G}$.

Because the expression $\tilde{w}_0 + x\tilde{w}_1 + x^2\tilde{w}_2 + x^3\tilde{w}_3 + \cdots + x^n\tilde{w}_n$ can be rewritten as $\boldsymbol{\phi}^\top\tilde{\boldsymbol{w}}$, with $\boldsymbol{\phi} = [1,\ x,\ x^2,\ x^3, \ldots,\ x^n]^\top$ (for $x \in X$) so each random variable $\tilde{g}_x$ of $\mathcal{G}$ may be written as:

$$\tilde{g}_x = \boldsymbol{\phi}^\top\tilde{\boldsymbol{w}} \qquad \text{with} \qquad \boldsymbol{\phi} = [1\ x\ x^2\ x^3 \ldots\ x^n]^\top \qquad \text{for } x \in X. \tag{3.31}$$

Now, consulting the Appendix Section A.12 one sees that the random vector $\tilde{g}_x$ is also a Gaussian random vector with pdf:

$$\tilde{g}_x \sim \mathcal{N}(g_x;\ 0, \boldsymbol{\phi}^\top\boldsymbol{\Sigma}_w\boldsymbol{\phi}) \qquad \text{with} \qquad \boldsymbol{\phi} = [1,\ x,\ x^2,\ x^3, \ldots,\ x^n]^\top \qquad \text{for } x \in X \tag{3.32}$$

Therefore all these random variables have a joint Gaussian pdf so again we are examining a $\mathcal{GP}$ process! Its random vector representation is once again $\tilde{\boldsymbol{g}} = \mathcal{N}(\tilde{\boldsymbol{g}}; \boldsymbol{0}, \boldsymbol{K})$ where here we set the covariance matrix $\boldsymbol{K} = \boldsymbol{\Phi}^\top\boldsymbol{\Sigma}_w\boldsymbol{\Phi}$. Of course the reader will ask what is this new matrix $\boldsymbol{\Phi}$. It is termed the **design matrix** and the necessity for its introduction here can be easily understood if the reader notices that there are as agreed m random variables $\tilde{g}_{x_1}, \tilde{g}_{x_2}, \tilde{g}_{x_3}, \ldots, \tilde{g}_{x_m} \in \mathcal{G}$ as well as m index values $x_1, x_2, x_3, \ldots, x_m \in X$! So the covariance matrix $\boldsymbol{K}$ is the following:

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{\phi}_1^\top \\ \boldsymbol{\phi}_2^\top \\ \boldsymbol{\phi}_3^\top \\ \vdots \\ \boldsymbol{\phi}_m^\top \end{bmatrix} \boldsymbol{\Sigma}_w \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \boldsymbol{\phi}_3 & \ldots & \boldsymbol{\phi}_m \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & x_2^3 & \ldots & x_2^n \\ 1 & x_3 & x_3^2 & x_3^3 & \ldots & x_3^n \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & x_m^3 & \ldots & x_m^n \end{bmatrix} \boldsymbol{\Sigma}_w \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_1 & x_2 & x_3 & \ldots & x_m \\ x_1^2 & x_2^2 & x_3^2 & \ldots & x_m^2 \\ x_1^3 & x_2^3 & x_3^3 & \ldots & x_m^3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^n & x_2^n & x_3^n & \ldots & x_m^n \end{bmatrix} = \boldsymbol{\Phi}^\top\boldsymbol{\Sigma}_w\boldsymbol{\Phi} \tag{3.33}$$

Notice also that e.g. $K_{12} = \mathrm{Cov}(\tilde{g}_{x_1}\tilde{g}_{x_2}) = \boldsymbol{x}_{\boldsymbol{1}}^\top\boldsymbol{\Sigma}_w\boldsymbol{x}_{\boldsymbol{2}}$ which is true since:

$$\mathrm{Cov}(\tilde{g}_{x_1}\tilde{g}_{x_2}) = \mathbb{E}\big(\boldsymbol{\phi}_1^\top\tilde{\boldsymbol{w}}(\boldsymbol{\phi}_2^\top\tilde{\boldsymbol{w}})^\top\big) = \boldsymbol{\phi}_1^\top\mathbb{E}(\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}^\top)\boldsymbol{\phi}_2 = \boldsymbol{\phi}_1^\top\big(\mathbb{E}(\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}^\top) - \boldsymbol{0}\big)\boldsymbol{\phi}_2 = \boldsymbol{\phi}_1^\top\big(\mathbb{E}(\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}^\top) - \mathbb{E}^2(\tilde{\boldsymbol{w}})\big)\boldsymbol{\phi}_2 = \boldsymbol{\phi}_1^\top\boldsymbol{\Sigma}_w\boldsymbol{\phi}_2 \tag{3.34}$$

And then as before we may write directly the four most fundamental equations:

$$\tilde{\boldsymbol{\varepsilon}} \sim \mathcal{N}(\boldsymbol{\varepsilon};\ \boldsymbol{0}, \sigma_n^2\boldsymbol{I}) \tag{3.35a}$$

$$\tilde{\boldsymbol{g}} = \boldsymbol{\Phi}^\top\tilde{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{g};\ \boldsymbol{0}, \boldsymbol{K}) \tag{3.35b}$$

$$\tilde{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{y};\ \boldsymbol{0}, \sigma_n^2\boldsymbol{I} + \boldsymbol{K}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}) \tag{3.35c}$$

$$\begin{gathered} f_{\tilde{\boldsymbol{y}}_*}(\tilde{\boldsymbol{y}}_*|\tilde{\boldsymbol{y}}_+ = \boldsymbol{y}_+) \sim \mathcal{N}(\boldsymbol{y}_*;\ \boldsymbol{C}_{*+}\boldsymbol{C}_{++}^{-1}\boldsymbol{y}_+, \boldsymbol{C}_{**} - \boldsymbol{C}_{*+}\boldsymbol{C}_{++}^{-1}\boldsymbol{C}_{+*}) = \\ \mathcal{N}\big(\boldsymbol{y}_*;\ \boldsymbol{\Phi}_*^\top\Sigma_{w_{*+}}\boldsymbol{\Phi}_+(\boldsymbol{K}_{*+} + \sigma_n^2\boldsymbol{I}_{*+})^{-1}\boldsymbol{y}_+\ ,\ \boldsymbol{\Phi}_*^\top\Sigma_{w_{**}}\boldsymbol{\Phi}_* - \boldsymbol{\Phi}_*^\top\Sigma_{w_{*+}}\boldsymbol{\Phi}_+(\boldsymbol{K}_{++} + \sigma_n^2\boldsymbol{I}_{++})^{-1}\boldsymbol{\Phi}_+^\top\Sigma_{w_{+*}}\boldsymbol{\Phi}_*\big) \end{gathered} \tag{3.35d}$$

It should be noticed that we partitioned also the design matrix $\boldsymbol{\Phi}$ to two partitions $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_+ \ \boldsymbol{\Phi}_*]^\top$ as was done with the rest of the random vectors as before.

The essence of what is written above is that if one considers a Gaussian Process whose every finite sub-collection is written in matrix notation as $\tilde{\boldsymbol{g}} = \boldsymbol{\Phi}^\top \tilde{\boldsymbol{w}} \sim \mathcal{N}(\tilde{\boldsymbol{g}};\ \boldsymbol{0}, \boldsymbol{K})$, then all random variables $\tilde{g}_x$ of the random vector $\tilde{\boldsymbol{g}}$ follow the polynomial model with random weights that each of those weights follows a univariate Gaussian distribution. The modeler should **first** define the joint pdf of the random vector $\tilde{\boldsymbol{g}}$ in order for its elements to have univariate Gaussian pdf's. *The reason we wrote everything here the other way around is to show how we would obtain the random vector $\tilde{\boldsymbol{g}}$ with such a joint pdf*! But specifying various random variables with univariate Gaussian pdf's does not mean that their joint pdf is also Gaussian!

The vector $\boldsymbol{\phi}$ now can include other elements than the ones written above. This way one changes the polynomial model to some other model. Possible examples are:

$$\boldsymbol{\phi} = [1\ x\ x^2\ x^3 \ldots\ x^m\ 1\ z\ z^2\ z^3 \ldots\ z^n]^\top \tag{3.36}$$

for $1 < n < m$ that leads to a polynomial model as:

$$g = g(x,z) = w_m x^m + w_{m-1}x^{m-1} + w_{m-2}x^{m-2} + \cdots + w_{n+1}x + w_n z^n + w_{n-1}z^{n-1} + w_{n-2}z^{n-2} + \cdots + w_1 z + w_0 \tag{3.37}$$

This last section, although not relevant to what will be showed later in this thesis, was added in order to present to the reader that the Gaussian Process model can incorporate the structure of a polynomial series model in order to be *enhanced* with more characteristics that are attributed to the polynomial series. In this sense, for example, since any continuous function can be approximated by its truncated Taylor expansion series, one can also "construct" a Gaussian Process that can approximate any realizations of any continuous stochastic process.

In general, there is no particular need to restrict one's self to polynomial sequences: interesting vectors might be the ones of the form $\boldsymbol{\phi}_b = h(\boldsymbol{\phi})$, with $\boldsymbol{\phi}$ being the vector of Equation 3.37, where h is a function of $\boldsymbol{\phi}$ whose image might look like:

$$h(\boldsymbol{\phi}) = [e\ e^x\ e^{x^2}\ e^{x^3} \ldots\ e^{x^m}\ e^z\ e^{z^2}\ e^{z^3} \ldots\ e^{z^n}]^\top \qquad \text{or} \tag{3.38}$$

$$h(\boldsymbol{\phi}) = [1\ \sin(x)\ \sin^2(x)\ \sin^3(x) \ldots \sin^n(x)]^\top \qquad \text{etc.} \tag{3.39}$$

which means that the modeler can populate the vector $\boldsymbol{\phi}_b = h(\boldsymbol{\phi})$ with whatever functions (or else termed ***basis functions***) he wishes under proper reasoning. It can be again shown that one can still "construct" a Gaussian Process.

4

Monte Carlo Methods

In this chapter, the basics of the Monte Carlo ($\mathcal{MC}$) method will be discussed along with some well-known sampling methods. The $\mathcal{MC}$ method is a method for solving various mathematical problems numerically by using what is called random sampling. Due to the inherent simplicity of $\mathcal{MC}$ and its potential to be combined with other computational methods it is a very useful tool for any scientist. The popularity of this method took off with the appearance and development of computers that made the iterative calculations of the $\mathcal{MC}$ method and the random sampling technique more accessible. In the upcoming sections, we shall refer to various well-known random sampling techniques as well.

4.1. Theoretical Prerequisites

4.1.1. The Laws of Large Numbers

Before we give the overview of the $\mathcal{MC}$ method we shall introduce first the weak law of large numbers and then the strong law of large numbers. Assume a sampling sequence of random variables $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \dots, \tilde{x}_n$ which are independent to each other and identically distributed (they have the same distribution, for which we know the expected value μ and the variance σ). Then the **weak law of large numbers** states that for every $\varepsilon > 0$ [41]:

$$\lim_{n\to\infty} \mathbb{P}\big(|\tilde{\bar{x}}_n - \mu| > \varepsilon\big) = 0, \qquad \text{where} \qquad \tilde{\bar{x}}_n = \frac{\tilde{x}_1 + \tilde{x}_2 + \tilde{x}_3 + \cdots + \tilde{x}_n}{n} \tag{4.1}$$

$\tilde{\bar{x}}_n$ is called the sample mean. The meaning of the weak law of large numbers is that for any positive threshold ε, however small, the probability that the sample mean will tend to approximate the expected value keeps on increasing if the sampling sequence forms an infinitely long sequence.

On the other hand the **strong law of large numbers** states that [41]:

$$\mathbb{P}\Big(\lim_{n\to\infty} \tilde{\bar{x}}_n = \mu\Big) = 1 \tag{4.2}$$

which means that the sample mean will converge almost surely to the common expected value of the random variables. This is a stronger statement than the previous one. We shall not discuss it explicitly, but there are cases where the weak law of large numbers holds while the strong law of large numbers does not. It can be proven that the strong law can be applied always for independent and identically distributed random variables that have the same expected value and variance, as is assumed above [41].

4.1.2. The Central Limit Theorem

Assume the previous sampling sequence of random variables $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \dots, \tilde{x}_n$. They are again independent of each other and identically distributed with the same estimated value μ and variance σ. The sample mean $\tilde{\bar{x}}_n$ is calculated as in Equation 4.1. Then the distribution of the random variable $\tilde{x}_c$ [39]:

$$\tilde{x}_c = \frac{\tilde{\bar{x}}_n - \mu}{\frac{\sigma}{\sqrt{n}}} \tag{4.3}$$

converges for $n \to \infty$ to a standard normal distribution $\mathcal{N}(\tilde{x}_c; 0, 1)$. This can be shown below in Figure 4.1 where a histogram of the values of the random variable $\tilde{x}_c$ is computed with respect to the random variables $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \dots, \tilde{x}_n$ which follow a normal distribution with expected value $\mu = 0.5$ and $\sigma = 0.0008$. To show this figure one needs to draw N **multiple random samples of the sample mean** $\tilde{\bar{x}}_n$, which itself is calculated for n $\tilde{x}_i$ elements every time. Here N was chosen as 10^3 and n as 10^7. Although we specified that the distribution we chose was a normal distribution, it can be shown that this theorem is valid regardless the probability distribution the random variables $\tilde{x}_i$, $i = 1,2,3,..,n$ follow. Consulting the Appendix Section A.12 one may also conclude that from the above Equation, $\tilde{\bar{x}}_n$ follows a Normal distribution as well, namely

$$\tilde{\bar{x}}_n \sim \mathcal{N}\left(\tilde{\bar{x}}_n; \mu, \frac{\sigma^2}{n}\right) \tag{4.4}$$

A similar normal distribution holds also for the sum of the random variables $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \ldots, \tilde{x}_n$.

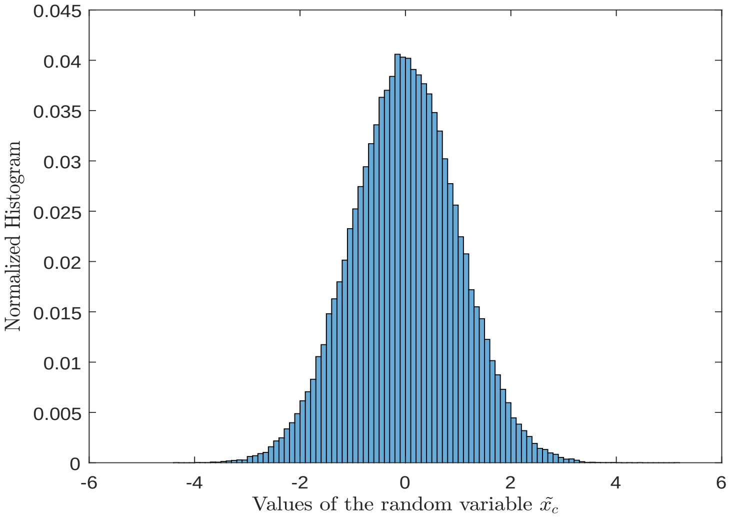


Figure 4.1: Conceptualization of the Central Limit Theorem.Normalized histogram of 100 bins of the $\tilde{x}_c$ variable, for $N = 10^3$ values of the $\tilde{\bar{x}}_n$ created from sample sizes of $n = 10^7$ $\tilde{x}_i$ elements, with $1 \le i \le n$

Notice that Equation 4.4 shows that:

$$\mathbb{E}(\tilde{\bar{x}}_n) = \mathbb{E}(\tilde{x}_1) = \mathbb{E}(\tilde{x}_1) \cdots = \mathbb{E}(\tilde{x}_n) = \mu \qquad \text{and} \qquad \mathrm{Var}(\tilde{\bar{x}}_n) = \frac{\sigma^2}{n} \tag{4.5}$$

4.1.3. The Frequentist Approach to Probability

Let us introduce the frequentist approach for the probability of an event B. The probability of its realization is [36]:

$$\mathbb{P}(B) = \lim_{n_T \to \infty} \frac{n_B}{n_T} \tag{4.6}$$

where n_T is the total number of trials we run the experiment in question and n_B is the number of trials where event B was realized. This formula relates well to our empirical understanding about what probability is. We will use it in the following section.

4.2. A Well-Known Example

As stated above the $\mathcal{MC}$ method is useful for a variety of applications in order to find numerical solutions. Here we shall refer to a widely known example in the literature [41, 51] – how to **numerically approach** the digits of π, through random sampling the ratio of the area A covered by a disk of radius $r = 1$ inside a 2×2 square of area $\Omega = 4$. By Euclidean geometry, we know the definition of pi as the ratio between the disk circumference and the diameter, $d = 2r$. We also know (again from Euclidean geometry) that $A = \pi r^2 = \pi d^2/4$, so we know the ratio A/S as $A/S = \pi/4$. We will show how to use random sampling in order to calculate the ration A/S which also will lead to approximate numerically the digits of π. We will use an **iterative method that converges** to π– we will obtain $\pi/4$.

But how can we use random sampling in such a problem, when no random phenomenon takes place? First, assume two independent uniform random variables $\tilde{x}$ and $\tilde{y}$ as $(\tilde{x}, \tilde{y}) \in \Omega = \{(x, y) : -1 \le x \le 1, -1 \le y \le 1\}$. Now assume the event $A \subseteq \Omega$, where $A = \{(x, y) : x^2 + y^2 \le 1\}$ It follows that $A^c = \{(x, y) : x^2 + y^2 > 1\}$.

Suppose we sample random points that may or may not be in the disk – they are always though inside the square. Some of these points will give us a portion of the area of the disk while all of them will give us the area of the square. Now by drawing many random sampling points, we can end up approaching as shown above in the frequentist approach, the probability of the event that a sampling point is inside the disk. So, on the limit of this process, one would get:

$$\mathbb{P}\big((\tilde{x}, \tilde{y}) \in A\big) = \lim_{n_T \to \infty} \frac{n_{\{(\tilde{x},\tilde{y})\} \in A}}{n_T} = \frac{\int_A \mathrm{d}x\mathrm{d}y}{\int_\Omega \mathrm{d}x\mathrm{d}y} = \frac{\pi}{4} \tag{4.7}$$

where we used the two random variables $\tilde{x}, \tilde{y}$ as above. We just saw that through this way we approached a deterministic problem by introducing random sampling. **But how do we draw random samples** in order to compute this probability – and consequently $\pi/4$?

It is for this purpose that we chose to introduce two independent random variables [34]: One can see that since for each random variable $\tilde{x}$ and $\tilde{y}$ their uniform distribution is $U(a,b)$ with $a=-1, b=1$ one would get that $f_{\tilde{x}}(x) = f_{\tilde{y}}(y) = \frac{1}{1-(-1)} = \frac{1}{2}$ one would have:

$$f_{\tilde{x},\tilde{y}}(x,y) = f_{\tilde{x}}(x) f_{\tilde{y}}(y) = \frac{1}{4} \tag{4.8}$$

Following the considerations of Section 2.9 one could write:

$$\mathbb{E}(I_A(\tilde{x},\tilde{y})) = \mathbb{P}\big((\tilde{x},\tilde{y}) \in A\big) = 1 \cdot \mathbb{P}\big((\tilde{x},\tilde{y}) \in A\big) + 0 \cdot \mathbb{P}\big((\tilde{x},\tilde{y}) \in A^c\big) = 1 \cdot \int_A f_{\tilde{x},\tilde{y}}(x,y)\mathrm{d}x\mathrm{d}y + 0 \cdot \int_{A^c} f_{\tilde{x},\tilde{y}}(x,y)\mathrm{d}x\mathrm{d}y \tag{4.9}$$

This has the form of Equation 2.13 with $g(x,y) = I_A(x,y)$. Because $|I_A(\tilde{x},\tilde{y})| < \infty$, the random variable $\tilde{v} = I_A(\tilde{x},\tilde{y})$, from now on called $\tilde{v}$ has expected value:

$$\mathbb{E}(\tilde{v}) = \int_\Omega I_A(x,y) f_{\tilde{x},\tilde{y}}(x,y)\mathrm{d}x\mathrm{d}y = \int_A 1 \cdot \frac{1}{4}\mathrm{d}x\mathrm{d}y + 0 = \frac{1}{4}\pi r^2 = \frac{\pi}{4}, \tag{4.10}$$

where here we used the fact that the disk area A can be computed by the integral of $\int_A \mathrm{d}x\mathrm{d}y$. So by introducing this random variable $\tilde{v}$, we were able through its expected value $\mathbb{E}(\tilde{v})$ to approximate numerically the value of the ratio between the disk and the square, eventually approximating the value of π. The more random samples we draw from this variable $\tilde{v}$ and its pdf, the smaller the variance of the resulting one would expect. Similarly, we can define the standard deviation around the sample mean, termed **the standard error of the sample mean**, denoted by $\sigma_{\tilde{\bar{v}}_n}$, using Equations 2.21 and 4.5 as:

$$\sigma_{\tilde{\bar{v}}_n} = \sqrt{\mathrm{Var}(\tilde{\bar{v}}_n)} = \frac{\sigma}{\sqrt{n}} \tag{4.11}$$

So one can see that the **standard error of the sample mean is proportional to** $\frac{1}{\sqrt{n}}$. This means for example that if one needs to increase the precision of the computations and reduce the error 10 times, the sampling size n should be increased 100 times.

4.3. Monte Carlo Method – the Big Picture

The previous example is indicative of what $\mathcal{MC}$ is about. First one has to set up the problem properly in order to implicate random sampling of a random variable $\tilde{x}$ that follows a probability distribution $f_{\tilde{x}}(x)$. By necessity as proved above, for its sample mean will converge in distribution to a normal distribution $\mathcal{N}(\mu, \frac{\sigma^2}{n})$, if the random variables are identically and independently drawn. Now assume that this random variable $\tilde{x} \sim f_{\tilde{x}}$ is also the argument to a function g, where this way we form a random variable $\tilde{g} = g(\tilde{x})$. Assume again a sequence of random samples which are independent to each other and identically distributed $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \ldots, \tilde{x}_i, \ldots, \tilde{x}_n$ where $1 \le i \le n$ and $\tilde{x}_i \sim f_{\tilde{x}}$. Following the same analysis as above we now have:

$$\mathbb{E}(\tilde{g}) = \mathbb{E}\big(g(\tilde{x})\big) \approx \frac{1}{n}\sum_{i=1}^{n} g(x_i), \qquad \tilde{x}_i \sim f_{\tilde{x}}, \tag{4.12}$$

$$\mathrm{Var}(\tilde{g}) = \mathrm{Var}\big(g(\tilde{x})\big) \approx \frac{1}{n}\sum_{i=1}^{n} g^2(x_i) - \Big(\frac{1}{n}\sum_{i=1}^{n} g(x_i)\Big)^2, \qquad \tilde{x}_i \sim f_{\tilde{x}}, \tag{4.13}$$

$$\mathbb{E}(\tilde{\bar{g}}) = \mathbb{E}(\tilde{g}), \tag{4.14}$$

$$\sigma_{\tilde{\bar{g}}} = \sqrt{\mathrm{Var}(\tilde{\bar{g}})} = \frac{\sqrt{\mathrm{Var}(\tilde{g})}}{\sqrt{n}}. \tag{4.15}$$

As can be seen the computed error is similar as above. Equations 4.12 to 4.15 show how to estimate the expected value and the variance of the random variable $\tilde{g}$. A big difference from before though is the fact that we did not use in these Equations the pdf of the random variable in question, $f_{\tilde{g}}$, but instead we sample from another pdf $f_{\tilde{x}}$. From Section 2.8 we saw that in general, if $\tilde{g} = g(\tilde{x})$, it may be that $f_{\tilde{g}} \neq f_{\tilde{x}}$. Are therefore Equations 4.12 to 4.15 adequate to use? Without going to detail it can be proven that always [29]:

$$\mathbb{E}(\tilde{g}) = \int_{-\infty}^{\infty} g f_{\tilde{g}}(g)\mathrm{d}g = \int_{-\infty}^{\infty} g(x) f_{\tilde{x}}(x)\mathrm{d}x = \mathbb{E}[g(\tilde{x})], \tag{4.16}$$

where here all random variables were considered continuous. **Analogous claims can be considered for multivariate distributions**. Therefore, if one knows the probability distribution of the argument of the function, the $\mathcal{MC}$ method can continue with the set of Equations 4.12 to 4.15 as described above.

And now, having shown the main set of Equations, we shall proceed to show how we can sample from the probability distribution function $f_x(x)$ adequately in order to ensure that random sampling conforms with the pdf we have in mind.

4.4. Sampling Methods for Monte Carlo

To sample the from the pdf $f_{\tilde{x}}(x)$ adequately there are two ways to go for: either one needs some kind of sampling method that generates samples of the variable of interest $\tilde{x}$ using its pdf $f_{\tilde{x}}(x)$ or cdf $F_{\tilde{x}}(x)$ directly; or one needs a sampling method that generates samples from an alternative pdf $f_{\tilde{t}}(t)$ or cdf $F_{\tilde{t}}(t)$ of some random variable $\tilde{t}$ which however yields the same result as if we sampled from the original pdf $f_{\tilde{x}}(x)$. Here we are only going to present the most well known sampling methods.

4.4.1. Inverting the CDF - Direct Sampling

Direct sampling is a common method that relies on inverting the cdf. Assume therefore a random variable $\tilde{x}$, a pdf $f_{\tilde{x}}(x)$ and a cdf $F_{\tilde{x}}(x)$. In general it is defined that if a cdf $F_{\tilde{x}} : \mathbb{R} \to (0,1)$ is **invertible** then there is some function $F_{\tilde{x}}^{-1} : (0,1) \to \mathbb{R}$ such that $F_{\tilde{x}}^{-1}\big(F_{\tilde{x}}(x)\big) = x$ for all $x \in \mathbb{R}$ and also $F_{\tilde{x}}\big(F_{\tilde{x}}^{-1}(a)\big)$ for all $a \in (0,1)$. To illustrate this better this, Figure 4.2 shows an invertible cdf and how for every value $y = F_{\tilde{x}}(x)$ the inverse distribution function $F_{\tilde{x}}^{-1}$ can accept it as an argument and give back the initial value x. It should be apparent that if a cdf is strictly monotonically increasing and continuous then there exists an invertible function for it.

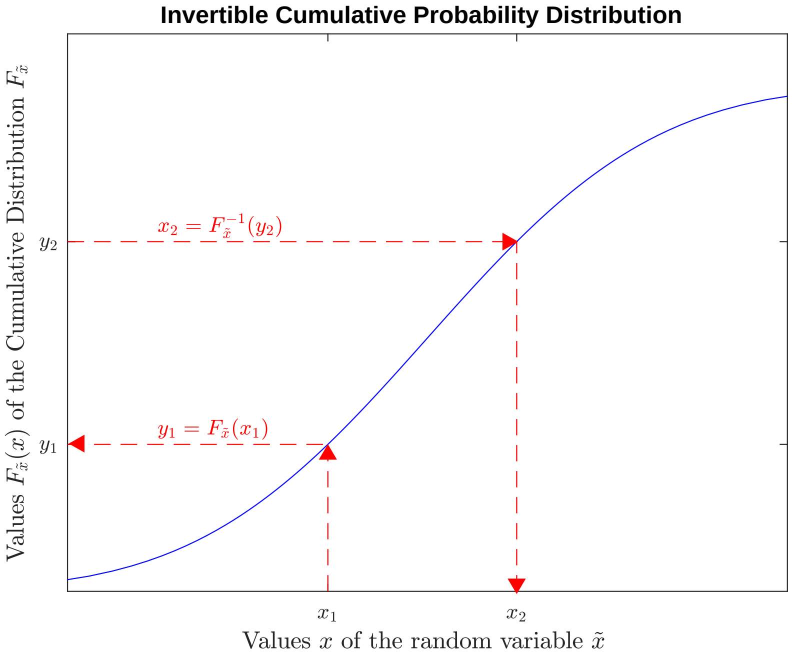


Figure 4.2: An example of an invertible cumulative distribution $F_{\tilde{x}}$ of a random variable $\tilde{x}$

Not all cdf's are invertible however, but we will examine this further below. For the time being we want to concentrate on the following remarks:

- For all cdf's, either invertible or not, the following relationship is true: for any random variable $\tilde{x}$ and for any $x_1 \in \mathbb{R}$ and $x_2 \in \mathbb{R}$ such that $x_1 \leq x_2$ it is valid that $F_{\tilde{x}}(x_1) \leq F_{\tilde{x}}(x_1)$.

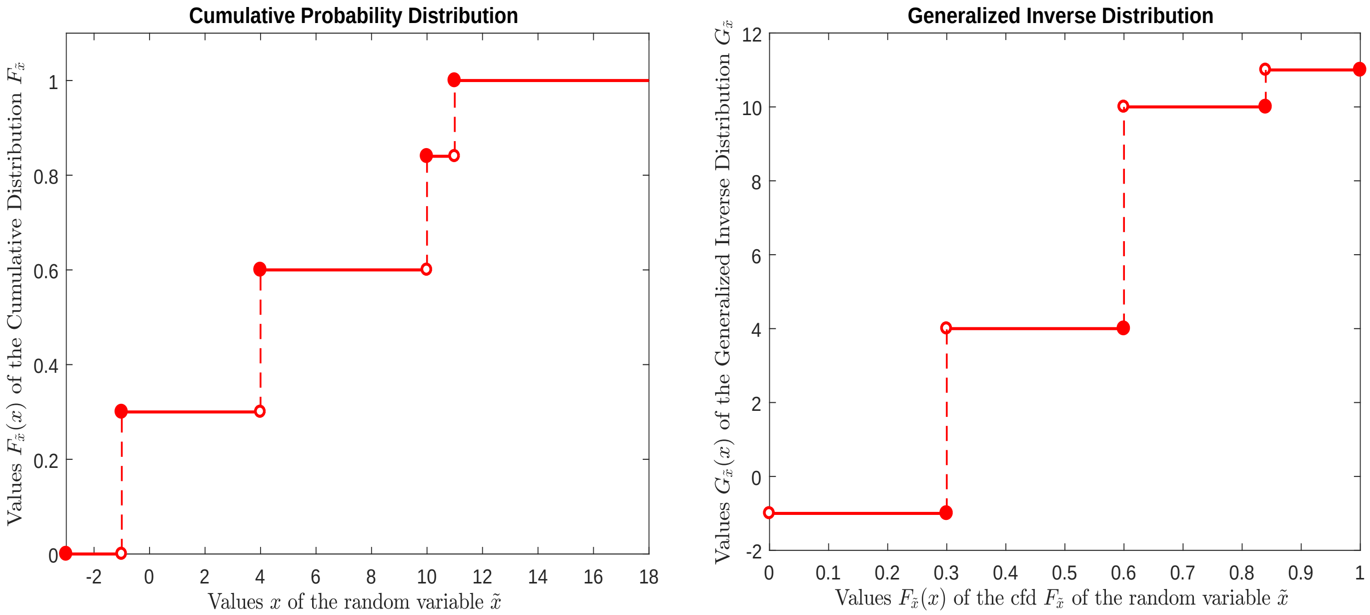
- For any random variable $\tilde{x}$ that **does have an invertible cdf** $F_{\tilde{x}}$ and for any $x_1 \in \mathbb{R}$ and $x_2 \in \mathbb{R}$ such that $x_1 < x_2$ it is valid that $F_{\tilde{x}}(x_1) < F_{\tilde{x}}(x_1)$. This has to do with the fact that the invertible cdf is strictly increasing as seen in Figure 4.2. Now, assume therefore the inverse distribution $F_{\tilde{x}}^{-1}$ of the aforementioned arbitrary invertible cdf $F_{\tilde{x}}$. Then, for any $a \in (0,1)$ and $b \in (0,1)$ such that $a \leq b$ it is always valid that $F_{\tilde{x}}(a) \leq F_{\tilde{x}}(b)$. This is because had it been otherwise, i.e. $F_{\tilde{x}}(a) > F_{\tilde{x}}(b)$ then $F_{\tilde{x}}^{-1}\big(F_{\tilde{x}}(a)\big) > F_{\tilde{x}}^{-1}\big(F_{\tilde{x}}(b)\big) \Rightarrow a > b$ as explained before, which is however a contradiction.

- **For any invertible cdf** it is valid, for any $x_1 \in \mathbb{R}$ and $a \in (0,1)$ that $F_{\tilde{x}}^{-1}(a) \leq x_1$ if and only if $a \leq F_{\tilde{x}}(x_1)$. This is because if it is true that $F_{\tilde{x}}^{-1}(a) \leq x_1$ then $F_{\tilde{x}}\big(F_{\tilde{x}}^{-1}(a)\big) \leq F_{\tilde{x}}(x_1) \Rightarrow a \leq F_{\tilde{x}}(x_1)$ and conversely if it is true that $a \leq F_{\tilde{x}}(x_1)$ then it is also true that $F_{\tilde{x}}^{-1}(a) \leq F_{\tilde{x}}^{-1}\big(F_{\tilde{x}}(x_1)\big) \Rightarrow F_{\tilde{x}}^{-1}(a) \leq x_1$.

From these remarks we can proceed to the following observation: Assume a uniform random variable $\tilde{u} \sim U(\tilde{u}; 0, 1)$, then, for a cumulative distribution F which is invertible, the random variable $\tilde{x} = F^{-1}(\tilde{u})$ has cumulative probability distribution $F_{\tilde{x}} = F$. But why is it true that $F_{\tilde{x}} = F$ for the random variable $\tilde{x} = F^{-1}(\tilde{u})$? This is because for any $x_1 \in \mathbb{R}$ [39, 40]:

$$F_{\tilde{x}}(x_1) = \mathbb{P}\big(\tilde{x} \leq x_1\big) = \mathbb{P}\big(F^{-1}(\tilde{u}) \leq x_1\big) = \mathbb{P}\big(\tilde{u} \leq F(x_1)\big) = F_{\tilde{u}}\big(F(x_1)\big) = F(x_1), \tag{4.17}$$

Therefore, if one has an invertible cdf $F_{\tilde{x}}$ he can use its inverse $F_{\tilde{x}}^{-1}$ to pass to it as an argument the drawn samples of a uniform random variable $\tilde{u}$ and the results will be such, as if they were sampled from the cdf $F_{\tilde{x}}$ directly. Here we make the implicit assumption that it is easy to draw samples of a uniform random variable $\tilde{u} \sim U(0,1)$; this is because all modern computer languages facilitate this because they use pseudorandom generating algorithms that generate sequence of numbers whose properties approximate the properties of sequences of random numbers [18]. It is this way one can obtain samples from the cdf he wants. **But what happens if the cdf is not invertible?**



(a) An example of a cumulative probability distribution

(b) An example of a cumulative density function

Figure 4.3: Example of a cumulative distribution graph and its generalized inverse distribution

This case is depicted with an example in Figure 4.3a. The cdf does not have an inverse distribution as was introduced before F^{-1} since, for example, the value $F_{\tilde{x}} = 0.6$ does not have only one evaluation of x it can be mapped to. In this case therefore, the generalized inverse distribution $G_{\tilde{x}} : (0,1) \to \mathbb{R}$ is introduced so that for any $a \in (0,1)$ we have:

$$G_{\tilde{x}}(a) = \inf\{x \in \mathbb{R} : a \leq F_{\tilde{x}}(x)\} \tag{4.18}$$

where $\inf(A)$ signifies the greatest lower bound of a set A. Some remarks follow [33]:

- For any $a \in (0,1)$ it is always valid that $G_{\tilde{x}}(a) \in \mathbb{R}$ and this means that the infimum is always some real number. This is because for every cdf one has $F_{\tilde{x}} \to 1$ as $x \to \infty$ and $F_{\tilde{x}} \to 0$ as $x \to -\infty$ so the set $\{x \in \mathbb{R} : a \leq F_{\tilde{x}}(x)\}$ is never a non-empty set, but always contains some evaluations of interest $x \in \mathbb{R}$ of $\tilde{x}$ of which one is the infimum.
- For any $a \in (0,1)$ and $x_1 \in \mathbb{R}$ it is always valid that $a \leq F_{\tilde{x}}(x_1)$ if and only if $G_{\tilde{x}}(a) \leq x_1$. This is true because:
  - if it is valid that $a \leq F_{\tilde{x}}(x_1)$ then because $G_{\tilde{x}}(a) = \inf\{x \in \mathbb{R} : a \leq F_{\tilde{x}}(x)\}$ one has always $G_{\tilde{x}}(a) \leq F_{\tilde{x}}(x_1)$ because $F_{\tilde{x}}(x_1)$ *is* inside this set, as $F_{\tilde{x}}(x_1) \leq a$.
  - if it is valid that $G_{\tilde{x}}(a) \leq x_1$ then there exists some sequence of evaluations of $\tilde{x}$, $x_1, x_2, x_3, \ldots, x_n \geq G_{\tilde{x}}(a)$ such that $a \leq F(x_n)$ for every $n \in \mathbb{N}$ (they are all members of the set $\{x \in \mathbb{R} : a \leq F_{\tilde{x}}(x)\}$) and $x_n \to G_{\tilde{x}}(a)$ as $n \to \infty$. Notice this means that they converge from the right. We will use this along with the fact that every cdf is always right-continuous. Then, due to the fact that $a \leq F_{\tilde{x}}(x_n)$ one can consider that $\lim_{n\to\infty} a \leq \lim_{n\to\infty} F_{\tilde{x}}(x_n) \Rightarrow a \leq F_{\tilde{x}}\big(\lim_{n\to\infty} x_n\big) = F_{\tilde{x}}\big(G_{\tilde{x}}(a)\big) \leq F_{\tilde{x}}(x_1) \Rightarrow a \leq F_{\tilde{x}}(x_1)$.
  - From the last remark we can see that the generalized inverse distribution $G_{\tilde{x}}$ has the same property as the inverse function $F_{\tilde{x}}^{-1}$ has. If a cdf $F_{\tilde{x}}$ is invertible then $F_{\tilde{x}}^{-1} = G_{\tilde{x}}$. In Figure 4.3b one sees the generalized inverse distribution $G_{\tilde{x}}$ for the case of Figure 4.3a.

Then, as we did above, we will assume a uniform random variable $\tilde{u} \sim U(0,1)$. Then, for any cumulative distribution F and its generalized inverse distribution $G = \{x \in \mathbb{R} : a \leq F(x)\}$. Then the random variable $\tilde{x} = G(\tilde{u})$ has cumulative probability distribution $F_{\tilde{x}} = F$. This is because for any $x_1 \in \mathbb{R}$ [33]:

$$F_{\tilde{x}}(x_1) = \mathbb{P}\big(\tilde{x} \leq x_1\big) = \mathbb{P}\big(G(\tilde{u}) \leq x_1\big) = \mathbb{P}\Big(\tilde{u} \leq F(x_1)\Big) = F_{\tilde{u}}\big(F(x_1)\big) = F(x_1) \tag{4.19}$$

So in the end we ended up using the expression $\tilde{x} = G(\tilde{u})$ that yields results as if we have sampled the cdf of $\tilde{x}$.

4.4.2. Rejection Sampling

Rejection sampling is another technique of interest that obtains samples, not from the probability distribution $f_{\tilde{x}}$ one would normally expect, but indirectly, through the help of another probability distribution as we will see further below. A reason why one might consider using rejection sampling is whenever the distribution $f_{\tilde{x}}$ is known analytically or graphically but drawing samples from it is computationally expensive or complicated. Instead what one does is to sample from a *proposal distribution* g that is much more convenient to work with. Why exactly that is true is not the heart of the matter, as long as we can use this proposal distribution for rejection sampling.

For rejection sampling we employ the use of two random variables $a, c \in (0, \infty)$ so that $af_{\tilde{x}} \leq cg$ for every $x \in \mathbb{R}$. This last remark can be seen in Figure 4.4; we see that for every input point x, the graph of cg is always above or at least at the same "height" comparably to the one of $af_{\tilde{x}}$.

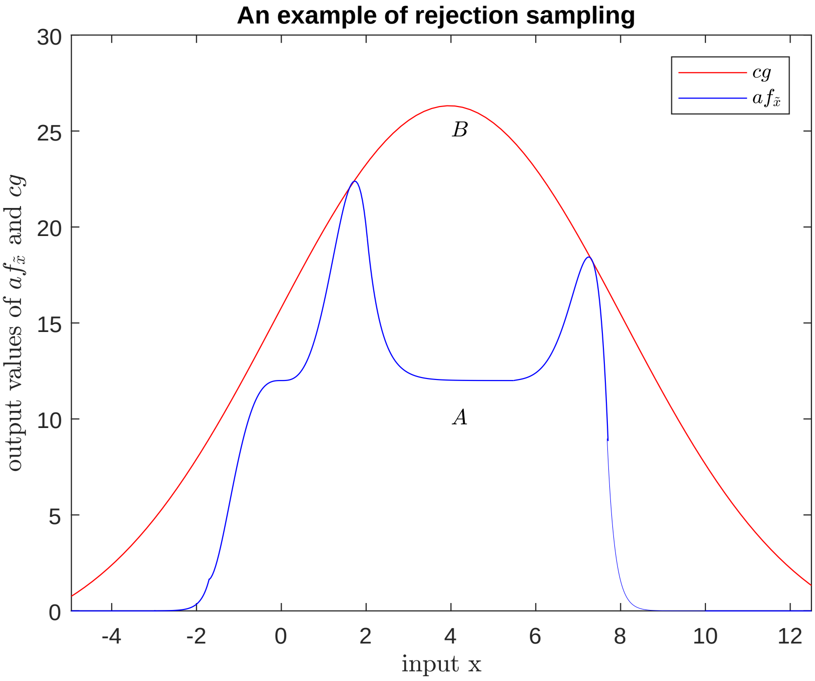


Figure 4.4: An example of rejection sampling concerning two pdf's $f_{\tilde{x}}$ and g.

Again in Figure 4.4 we see the delineation of two areas in the graph, the area below the curve of $af_{\tilde{x}}$, A, and the area below the curve of cg, B. This means that $A \subseteq B$. We would like our sampling to consider randomly drawn points in those areas, A and B. Therefore we would refer to a random vector $\tilde{\boldsymbol{u}} = [\tilde{x}\ \tilde{y}]^\top$ sampled from the area B, and sampled *uniformly*, as we shall define soon below. Conceptually the method of rejection sampling has to do with the following procedure: we select randomly from the area B a point, a realization of the random vector $\tilde{\boldsymbol{u}} = [\tilde{x}\ \tilde{y}]^\top$. If it does not belong also to the area A we "reject" it (we do not take it into account and repeat with sampling again), otherwise we "accept" it. With this procedure we sample every evaluation x with a pdf equal to $f_{\tilde{x}}$, which was our initial intent.

Before proceeding to prove why rejection sampling works, we have to state the following:

Lemma 4.4.1. *Assume a random vector* $\tilde{\boldsymbol{u}} = [\tilde{x}\ \tilde{y}]^\top$ *with* $\tilde{x}, \tilde{y} \in \mathbb{R}$, $\tilde{x} \sim f_{\tilde{x}}$ *and:*

$$f_{\tilde{y}}(y|\tilde{x} = x) = \begin{cases} \frac{1}{cf_{\tilde{x}}(x)}, & \textit{for } y \in [0, cf_{\tilde{x}}(x)], \\ 0, & \textit{otherwise}, \end{cases} \tag{4.20}$$

with $c \in (0, \infty)$. *Then:*

$$f_{\tilde{\boldsymbol{u}}}(\boldsymbol{u}) = \begin{cases} \frac{1}{c}, & \textit{for } \boldsymbol{u} \in A, \\ 0, & \textit{otherwise}, \end{cases} \tag{4.21}$$

where $A = \{x \in \mathbb{R}$ *and* $y \in [0, cf_{\tilde{x}}(x)]\}$

Proof. Multiplying both ends of Equation 4.20 with $f_{\tilde{x}}(x)$ for every $x \in \mathbb{R}$ because of the relationship $f_{\tilde{\boldsymbol{u}}}(\boldsymbol{u}) = f_{\tilde{y}}(y|\tilde{x} = x)f_{\tilde{x}}(x)$ proves the result. This lemma and proof were retrieved from [33]. When a random vector $\tilde{\boldsymbol{u}}$ has such a pdf as in Equation 4.21 for a set B, we will define it to be *uniformly distributed in A*. This will be useful for understanding the following lemmas. □

Lemma 4.4.2. *Assume a random vector* $\tilde{\boldsymbol{u}} = [\tilde{x}\ \tilde{y}]^\top$ *with* $\tilde{x}, \tilde{y} \in \mathbb{R}$ *and* $\tilde{x} \sim f_{\tilde{x}}$ *and a function* $f : \mathbb{R} \to \mathbb{R}$ *such that:*

$$f_{\tilde{\boldsymbol{u}}}(\boldsymbol{u}) = \begin{cases} constant & for\ \boldsymbol{u} \in A, \\ 0 & otherwise \end{cases} \quad and \quad \int_{-\infty}^{\infty} f(x)\mathrm{d}x = 1 \tag{4.22}$$

where $A = \{\boldsymbol{u} \in \mathbb{R}^2 : x \in \mathbb{R},\ and\ y \in [0, cf(x)]\}$ *and* $c \in (0,\infty)$*. Then* $f_{\tilde{x}} = f$*.*

Proof. Notice again that the random vector is supposed to be *uniformly distributed in A*. Assume the aforementioned constant term is $b \in \mathbb{R}$. Then we observe that:

$$\int_A f_{\tilde{\boldsymbol{u}}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} = 1 \Rightarrow \int_{\mathbb{R}} \int_0^{cf(x)} f_{\tilde{\boldsymbol{u}}}(\boldsymbol{u})\mathrm{d}y\mathrm{d}x = b \int_{\mathbb{R}} \int_0^{cf(x)} \mathrm{d}y\mathrm{d}x = bc = 1 \Rightarrow b = \frac{1}{c} \tag{4.23}$$

so then for all $x \in \mathbb{R}$:

$$f_{\tilde{x}}(x) = \int_{\mathbb{R}} f_{\tilde{\boldsymbol{u}}}(\boldsymbol{u})\mathrm{d}y = 0 + \frac{1}{c}\int_0^{cf(x)} \mathrm{d}y + 0 = f(x) \tag{4.24}$$

This lemma and proof were retrieved from [33] □

Lemma 4.4.3. *If* $A \subseteq B$ *and* $\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2, \tilde{\boldsymbol{u}}_3, \ldots$ *is a sequence of random vectors independent and identically distributed, uniformly in the set B, and* $\tilde{\boldsymbol{w}} = \tilde{\boldsymbol{u}}_k$ *where* $k = \min\{k : \tilde{\boldsymbol{u}}_k \in A\}$ *then* $\tilde{\boldsymbol{w}}$ *is uniformly distributed in the set A.*

Proof. The proof of this can be found in [11]. It is important for the next lemma below. □

Proposition 4.4.4. *Assume two numbers* $c, a \in (0,\infty)$ *and two pdfs* $f_{\tilde{x}}, g : \mathbb{R} \to \mathbb{R}$ *such that* $cg(x) \geq af_{\tilde{x}}(x)$ *for all* $x \in \mathbb{R}$*. Assume a sequence of independent and identically distributed random vectors* $\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2, \tilde{\boldsymbol{u}}_3, \ldots$ *such that* $\tilde{\boldsymbol{u}}_i = [\tilde{x}_i\ \tilde{y}_i]^\top$ *and* $\tilde{x}_i \sim g$ *and* $f_{\tilde{y}_i}(y_i | \tilde{x}_i = x) = U\big(0, cg(x)\big)$ *for* $i \in \mathbb{N}$*. If* $\tilde{\boldsymbol{u}}_k = [\tilde{x}_k\ \tilde{y}_k]$ *where* $k = min\{k : \tilde{y}_k \leq af(\tilde{x}_k)\}$*, then* $\tilde{x}_k \sim f_{\tilde{x}}$*.*

Proof. This proposition and proof were retrieved from [33]. It is here all above lemmas come to use. First from Lemma 4.4.1 one finds that because $\tilde{\boldsymbol{u}}_i = [\tilde{x}_i\ \tilde{y}_i]^\top$ and $\tilde{x}_i \sim g$ and $f_{\tilde{y}_i}(y_i | \tilde{x}_i = x) = U\big(0, cg(x)\big)$ for $i \in \mathbb{N}$ then it is also:

$$f_{\tilde{\boldsymbol{u}}_i}(\boldsymbol{u}_i) = \begin{cases} \frac{1}{cg(x_i)} g(x_i) = \frac{1}{c} & \text{for } \boldsymbol{u}_i \in B, \\ 0 & \text{otherwise} \end{cases} \tag{4.25}$$

where $B = \{\boldsymbol{u}_i \in \mathbb{R}^2 : x_i \in \mathbb{R}, \text{ and } y_i \in [0, cg(x_i)]\}$ for $i \in \mathbb{N}$. Consequently from Lemma 4.4.2 one sees that every $\tilde{x}_i$ in this case is distributed according to the pdf g. However we would like the random variables $\tilde{x}_i$ to follow the pdf $f_{\tilde{x}}$, so we implement the random vector $\tilde{\boldsymbol{u}}_k$ to help. This random vector's $\tilde{y}_k$ components should be strictly less than $af_{\tilde{x}}(\tilde{x}_k)$, therefore the evaluations of this random vector $\tilde{\boldsymbol{u}}_k$ have a special subset they are elements of. This subset of course is the area below the curve $af_{\tilde{x}}$, $A \subseteq B$, or analytically expressed, $A = \{\boldsymbol{u}_k \in \mathbb{R}^2 : x_k \in \mathbb{R}, \text{ and } y_k \in [0, af_{\tilde{x}}(x_k)]\}$. This subset has elements only the evaluations of the random vector $\tilde{\boldsymbol{u}}_k = [\tilde{x}_k\ \tilde{y}_k]^\top$ and from Lemma 4.4.3 one sees that this random vector is *uniformly* distributed in the set A. That is:

$$f_{\tilde{\boldsymbol{u}}_k}(\boldsymbol{u}_k) = \begin{cases} \frac{1}{af_{\tilde{x}}(x_k)} f_{\tilde{x}}(x_k) = \frac{1}{a} & \text{for } \boldsymbol{u}_k \in A, \\ 0 & \text{otherwise} \end{cases} \tag{4.26}$$

where again $k = \min\{k : \tilde{y}_k \leq af_{\tilde{x}}(\tilde{x}_k)\}$. This proves, again from Lemma 4.4.2 that $\tilde{x}_k \sim f_{\tilde{x}}$ and this is the reason why indeed the rejection sampling technique works. □

4.4.3. Importance Sampling

As explained before, given a random variable $\tilde{x} \in (a, b)$ of probability density $f_{\tilde{x}}(x)$ and a function g on this random variable, the expected value of this function would be $\mathbb{E}(g(\tilde{x})) = \int_a^b g(x) f_{\tilde{x}}(x)\mathrm{d}x$. Drawing samples from a particular $f_{\tilde{x}}$ might be difficult however.

An example of this is shown in Figure 4.5. It can be seen that the bulk mass of the pdf $f_{\tilde{x}}(x)$ is not at the same input values x where the function $g(x)$ contributes the most (the left side of the graph). The probability to draw samples at those values from the pdf $f_{\tilde{x}}(x)$ is too small to be of practical use; the waiting time of the user to draw samples from the left side of the graph would be tremendous as well as the computational effort for the computing system. The implementation of the pdf $q_{\tilde{x}}(x)$ as shown in Figure 4.5 will make computations faster.

For the importance sampling method, we shall introduce such a new, more convenient probability distribution $q_{\tilde{x}}(x)$ such that when $q_{\tilde{x}}(x) = 0$ then also $f_{\tilde{x}}(x) = 0$. Then we can write the expected value $\mathbb{E}(g(x))$ as [42]:

$$\mathbb{E}\big(g(x)\big) = \int_a^b g(x)f(x)\mathrm{d}x = \int_a^b \frac{g(x)f(x)}{q(x)} q(x)\mathrm{d}x = \int_a^b g(x)w(x)q(x)\mathrm{d}x \approx \frac{1}{n}\sum_{i=1}^{n} w(x_i)g(x_i), \qquad x_i \sim q_{\tilde{x}} \tag{4.27}$$

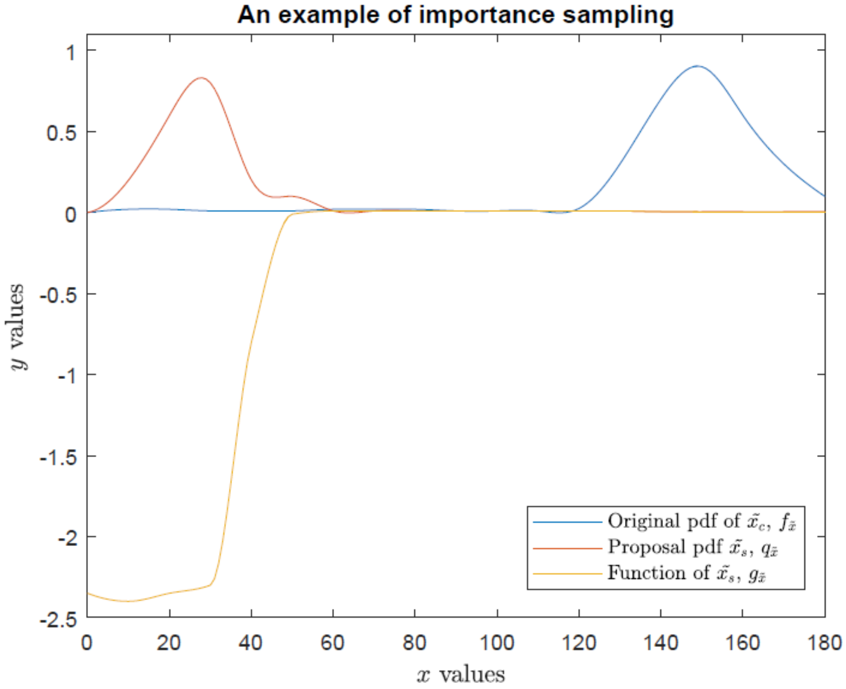


Figure 4.5: An example of the usage of importance sampling. The bulk mass of the pdf $f_{\tilde{x}}(x)$ is not at the same input values x where the function $g(x)$ contributes the most. The probability to draw samples at those values from the pdf $f_{\tilde{x}}(x)$ is too low to be of practical use. The implementation of the pdf $q_{\tilde{x}}(x)$ will make computations easier.

where $w(x) = \frac{f(x)}{q(x)}$ expresses the 'importance weight' of the one pdf to the other [51]. By this method we end up calculating the same expected value but from drawing samples from another pdf. If we would like to evaluate the standard error of the sample mean for this pdf then we would get the same results as shown before in Equations 4.12 to 4.15 except the fact that $x_i \sim q_{\tilde{x}}$

Comparing now Equations 4.12 and 4.27 one can see that in the first situation one directly samples a collection of $\{g(x_1), g(x_2), g(x_3), \ldots, g(x_n)\}$ values, while using the last Equation one ends up sampling a collection of different values, namely $\{w(x_1)g(x_1), w(x_2)g(x_2), w(x_3)g(x_3), \ldots, w(x_n)g(x_n)\}$. One then can apprehend that making an good selection of the $q_{\tilde{x}}$ might also lead to a collection of values which would have a smaller standard variation around their sample mean. Therefore, one may use such a method purely **in order to reduce the standard error of the sample mean**. The selection of $q_{\tilde{x}}$ is arbitrary as explained before. As a rule of thumb, the standard error of the sample mean is the smallest when $q_{\tilde{x}}(x) \propto f_{\tilde{x}}(x)$ [51].

Both the last two sections of this chapter will not be considered to our analysis later in the Thesis since there is not particular need for them, as they are more intricate methods than direct sampling. However, they are presented here in order to show that there are several considerations and different methods available to be used in the arsenal of Monte Carlo technique.

5

Wind Farm Layout Optimization

A common problem in the wind farm design process is to define a proper layout that allows maximizing the expected power production. This chapter displays the considerations about the wind farm layout and how to optimize it.

5.1. Optimization Theory

Optimization theory concerns a family of algorithmic concepts that seek optimal solutions by either maximizing or minimizing a function, termed the **objective function**, while in parallel respects the **constraints and limitations** (bounds on function values) of the problem [22]. Optimization theory is about solution techniques that are either particular to a specific set of problems or can be employed in a variety of unrelated problems with varying degrees of success. Their implementation is adjusted to the nature of the problem, mainly one has to identify the objective function and the constraints. In mathematical notation one may write [22]:

$$\begin{array}{cc} \text{Maximize or minimize} & h_0(\boldsymbol{x}) \\ \text{subject to} & h_{1_i}(\boldsymbol{x}) \leq 0\ ,\ i\text{=}1\text{,}2\text{,}\dots,m \text{ for } m \in \mathbb{N} \\ \text{and} & h_{2_j}(\boldsymbol{x}) = 0\ ,\ j\text{=}1\text{,}2\text{,}\dots,l \text{ for } l \in \mathbb{N} \\ \text{and} & \boldsymbol{x}_\text{L} \leq \boldsymbol{x} \leq \boldsymbol{x}_\text{U} \end{array}$$

where $\boldsymbol{x} = [x_1, x_2, x_3, \dots, x_n]^\top$ is a column vector of n real-valued variables. Now, $h_0(\boldsymbol{x})$ is the objective (or cost) function, $h_{1_i}(\boldsymbol{x})$ is every inequality constraint function (except bounds) that the problem may include, $h_{2_j}(\boldsymbol{x})$ is every equality constraint functions that the problem may entail. Lastly the vector $\boldsymbol{x}$ can be bound between a lower value $\boldsymbol{x}_\text{L}$ and an upper value $\boldsymbol{x}_\text{U}$. Optimization analysis is used broadly in engineering, as well as in computer science, economics and finance, etc.

Popular optimization methods are evolutionary algorithms, genetic algorithms, hill climbing, particle-swarm optimization, tabu search, simulated annealing and many others [30]. The inspiration for their functioning comes from the understanding and mimicking of various natural processes and they are usually applicable to a vast variety of problems because they require only but a few assumptions.

- Genetic Algorithms [6] as their name suggests rely on the use of bio-inspired operations of **genes** translated in computer language such as **mutation, crossover and selection**. The same way as in nature, genes are contained in **chromosomes**. **Genes** now are in terms of computer language simply variables, but as far as the setting of the problem goes, they incorporate information about chromosomes. Because genes are variables, chromosomes are usually depicted as vectors, as implied from the vector $\boldsymbol{x}$ we used above. Since the vector space of all vectors $\boldsymbol{x}$ is the **search space** where the modeller seeks the solution to extremize $h_0(\boldsymbol{x})$, some **chromosomes are solutions** to this problem while others are not. Exactly as in biology, the best chromosomes are the "solutions" that nature selects because they hone some ability or property of the individual organism. Chromosomes in genetic algorithms constitute the modeler's field of search. This field of search is again as above subject to limitations and constraints – as in nature. Every genetic algorithm works with **populations** of chromosomes, groups of them, where the bio-inspired operations referred above (which we will soon examine) take place. In nature, how "optimal" is a population of organisms (and thus its chromosomes) depends on the environment and the initial conditions from which evolution of the species started. Nature optimizes the **fitness** of the species to the given environment by altering its chromosomes. Therefore for genetic algorithms, the objective function is termed also the **fitness function**. In genetic algorithm methods, the fitness function of the chromosomes is iteratively calculated in order to reach the optimum point. As far as what the bio-operations that were referred above, we shall continue this discussion in Section 5.3

- Hill Climbing [6] is an optimization method used mainly in problems oriented in finding local optima and therefore for localized search. It is a simple iterative approach that begins from an **initial random point** in the **search space** and by altering the variables of the column vector $\boldsymbol{x}$ it computes the value of the **objective function** and compares whether its value is maximized, minimized or has the same value as before. This way it either adopts the change in the column vector $\boldsymbol{x}$ or not. If not the search continues until the number of iterations exceeds a fixed limit or the optimum point is found. Because this approach is very prone to miss globally optimal solutions get stuck in local ones, it is used limitedly, only when it is ensured that local maxima are essentially global maxima (such as convex problems).

- Particle-swarm optimization [8] is a method that employs a **population** of candidate solutions, termed **particles**, and "moves" iteratively the **particles** across the **search space** defined to seek better "positions" (solutions in the search space) with regard to a given measure or quality (essentially an objective function). Each particle adjusts its movement to its own moving experience and the "experience" of the other particles. The "velocity" with which each particle moves is affected by the distance of it from the best solution position, the distance from its personal best position and its current position and velocity. Since all particles begin from random positions, after an adequate amount of iterations, if multiple particles converge in one position in the search space it is thought to be the best global position. This algorithm mimics the natural social behavior of living animals and insects and their dynamic system of movements through intercommunication. Variation may also hold indicative names (e.g. artificial bee colony optimization method).

- Tabu search [20] is a concept in order to boost the performance of local search whose general pitfall is to get stuck at **suboptimal solutions** in the search space (like hill climbing). Tabu search relaxes their basic rule, which is to allow only improving solutions, by **accepting conditionally worsening solutions**. The term tabu hails from the fact that **prohibitions are implemented** in order not to move again to solutions which the algorithm has visited in a defined short time-frame in the past. This way the algorithm enhances the **exploratory characteristics** of local search and can potentially reach global optimum solutions.

- Simulated Annealing [6] follows a more probabilistic concept in order to introduce **worsening moves**. The name is an allegorical use by the annealing process in metallurgy a technique that heats and then gradually cools the metal in order to cause atoms to migrate in the crystal lattice and thus reduce the presence of defects inside it. The high heat annealing speeds up the thermodynamic process, and this is necessary, because at room temperature the process of erasing dislocations in the metal is very slow. Therefore **the temperature gradient should increase** and not only that but also **afterwards decrease in a controllable way**, varying the temperature step-by-step. The atoms, seeking thermal equilibrium, end up in various energy states for various temperatures. The probability of occurrence of the i-th state with energy E_i is given by the Boltzmann distribution:

  $$\mathbb{P}\{\text{state}_i\} \propto e^{-\frac{E_i}{\mathrm{k_B}T}} \tag{5.1}$$

  where $\mathrm{k_B}$ is the Boltzmann constant. From the equation above one can see that as the temperature drops the probability of the i-th energy state (above a minimum-energy state) decreases significantly. Simulated annealing consequently is used as an approach that mimics the statistical mechanics of equilibrium for a multiatomic system such as a metal by exploring potential candidate energy states in the search space that conditionally, if their probability requirement is met, they **can lead to non-minimizing values of the objective function**. The steps of the iteration correspond to the step of the decreasing temperature of the metallurgical process after the metal is first heated up. The objective function needs to be formulated so that optimum values are expressed as minimizing ones.

The aforementioned optimization algorithms offer the advantage that they are derivative-free and depend on very few parameters [6]. Relevant literature [1, 25, 28] offers many more variants and/or other algorithms but in general not all algorithms are in all problems useful, in the sense that the wind farm layout problem aims to solve for the layout, or better said, a multidimensional variable that cannot be handled by all types of optimizers efficiently. For example, hill climbing as a method is way too simple when there can be a lot of turbines to model for. This last point will be understood further below where we will introduce the problem of Layout Optimization in more detail.

5.2. Wind Layout Optimization Considerations

How does one need to optimize the wind farm layout in general? First, one needs to define what model he would use for expected power production. This has to do with what features/considerations do the original and the surrogate models have. In general however, **every model of expected power production** is affected by the layout either positively, negatively or nothing at all. It follows from basic aerodynamic theory that regarding the operation of a wind turbine [7] two areas can be distinguished downstream of it: the near-wake region and the far-wake region. In the near-wake region, the flow field is heavily influenced by the wind turbine geometry (rotor-nacelle assembly and tower). Immediately downstream of a turbine, the axial pressure gradient is large enough to significantly affect the velocity deficit in the wake. Conversely, in the far-wake region the effect of the wind turbine geometry starts to matter less and the wake induced, which progressively decreases as the downstream axial distance increases, is influenced by the presence of wind shear due to the mixing of the surrounding non-turbulent flow field, other topographic effects, as well as interference with other wakes generated. The distinction of the near-wake region and the far-wake region is not exact and thus it may differ in different models of the expected power production. However, there are two things that apply for all expected power models:

- For the model of expected power production, the rule about the placement of subsequent turbines downstream of a wind turbine, is never to place them inside the near-wake region of it. The reason is that inside this region the wake effects and consequently the power losses are extremely large. There should be some minimum distance between wind turbines so that they are always inside the far-wake region, the only region which is interesting for the modeler to study the wake effects. This minimum distance is usually between three to seven diameters of the wind turbine model to be installed.

- For all models, wakes cause power losses and more importantly, the wake effect is affected by the distance of subsequent turbines downstream. The greater the distance downstream of a subsequent turbine, the smaller the effect.

As far as the *search space for the optimum layout* goes, it is a geometric shape, most likely a polygon that is the **parcel of land or water, the wind farm site**. An example of it can be seen in Figure 5.1. The solutions of the search space are the specific coordinates where the wind turbines are to be installed. Therefore the optimization procedure in general concerns an objective function $h_0(\boldsymbol{T})$, whose input can be expressed as $\boldsymbol{T} = [\boldsymbol{x}, \boldsymbol{y}]^\top$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the vectors of coordinates of the turbines on an x, y geographic coordinate system. Their dimensionality is equal to the number of turbines installed.

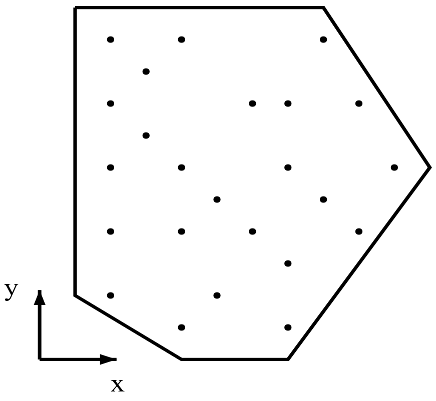


Figure 5.1: An example of a potential wind farm layout. The dots are the turbines while the black line symbolizes the parcel on which they are to be installed

Now a wind farm layout optimizer in general has a known search space, meaning that we focus on a particular wind park every time. It is there we are searching for an optimal layout, looking at various potential layouts. For this region one can have specific meteorological measurements that give us an overview of the expected **wind direction** and expected **wind speed** as well as their variability of their values. These data are needed because it is only this way we can compute the expected power of a specific layout. We start therefore this discussion by stating that **these meteorological data are valid for all potential layouts** the optimizer examines. We shall continue it further in the following sections.

Another consideration is how the optimizer examine different positions (solutions) in the search space. This has to do with the consideration of it being **discretized or not** [46], that is, if the geographic coordinate system (x, y) is a system of discrete or continuous values for (x, y).

If the coordinate system is not discretized, the wind turbines can take any position inside the wind park. To examine such a search space, one usually selects an initial layout and then by some operation, changes the position of the wind turbine by “moving” it in accordance to some distance and direction taking into consideration the limits of the wind park and the minimum distance between the wind turbines. After many iterations hopefully many turbines have been “moved” and the layout is optimized in terms of the objective function goal. Such approaches, however, need too much computational power in order to **explore** all of the search space for finalizing the wind turbine positioning, especially when the number of turbines is significant.

If the coordinate system is discretized, on the other hand, the search space is then a finite vector $\boldsymbol{T} = [\boldsymbol{x}, \boldsymbol{y}]^\top$. When this is the case, the optimization procedure concerns only those points alone. These points need to conform with the limits of the wind park and the minimum distance between the wind turbines. If that is the case then the optimization procedure is to **test and compare** layouts that concern these points. If new wind turbines are to be added they are to be positioned only at these points.

This discretized coordinate system approach is more appealing in terms of decreasing the computational effort, since the expected power predicted model can become a burden on its own when many iterations of expected power calculations are considered. Besides, up till now, approaches in the literature that adopt a continuous coordinate system are outnumbered by the alternative ones of a discretized coordinate system [46]. It is for this reason we will adopt for the following chapters such a discretized coordinate system.

The question that follows is what kind of optimization algorithm will the modeler have to work with. Of all the options analyzed above the more “obvious” one is the genetic algorithms: they have been implemented a lot of times in the past for wind farm layout optimization problems [14, 21, 23, 46], they are easy to model and they offer many variations for the modeler to choose. A well-known one is the **Binary Genetic Algorithm** (BGA) which is the first historically speaking approach for implementing an optimization method to wind farm layouts done by Mosseti et al. [17]. Their approach considers a discretized coordinate system and accepts a search space where with the help of the BGA new layouts are examined

In the next section we will follow this discussion concerning the Genetic Algorithms where we will introduce in more detail their features and their variations.

5.3. Genetic Algorithm

A good introduction of genetic algorithms was given above. We have already introduced:

- **genes**, the variables that give the solutions in the search space,
- **chromosomes**, the vector (or matrix) that contains all the variables (genes) whose values need to be optimized,
- **populations**, groups of chromosomes, usually the algorithm introduces more than one group to achieve variety of chromosomes,
- **fitness function**, the objective function of the algorithm that judges how optimized the chromosomes are,
- the **bio-operations** namely **mutation, crossover and selection** which we will present in detail here.

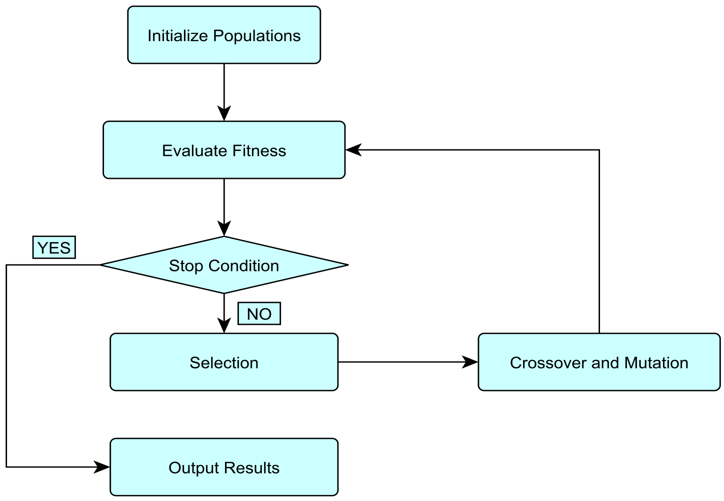


Figure 5.2: Genetic Algorithm general scheme

Here we will add a few more terms. Every iterative run of the genetic algorithm is termed a **generation** since every run the populations of chromosomes changes. In nature the same thing happens when the species evolves in order to optimize its fitness to the environment - it simply happens each generation. In each generation the bio-operations take place: **mutation, crossover and selection** – see Figure 5.2. These concern, of course, the genes of the chromosomes in each generation, they have to change in order to allow exploration of the search space to find an optimized solution (chromosome). The way this is done is not unique and thus the many variations in the genetic algorithmic design.

Before we go into more detail, it would be good to introduce what the BGA is. In BGA chromosomes are represented as strings of ones and zeros [49]. These values –zeros and ones– are the genes. Other encodings are also possible (e.g. more than two options for every gene), but they usually tend to converge to solutions slower than the BGA [26]. Figure 5.3 gives an example of what the chromosomes look like. Because BGAs are the most iconic example of what a genetic algorithm is, we shall refer to the three aforementioned bio-operations below by giving examples that are valid to BGAs without loss of generality.

0	1	0	1	0	1	1	1

Figure 5.3: Binary representation of a chromosome in BGA. The genes are represented by zeros and ones. The length of the binary string is dependent on how many genes (variables) are to be considered in the problem

It should be mentioned that there are other genetic algorithms that do not work with string representations at all but they work with genes that are *real numbers* and other objects which are called **phenotypes** [26]. Phenotypes do not “contain” ordered strings of all genes as chromosomes do, they rather *map all the real-valued genes* to a real number and then, the fitness functions are processing populations of phenotypes, not chromosomes. These type of genetic algorithms form the category of **real-coded genetic algorithms**. Their utility is usually highlighted when working with continuous search spaces, so finely discretized, where genetic algorithms with chromosomes of string representation would fail to examine all of the search space in some reasonable amounts of time [26]. To understand this, imagine when one has m real valued genes that take values in $(0,1) \subseteq \mathbb{R}$. We would like to allow genes to take values with some precision, so small, that it would mean that the genes could have e.g. 10^7 different evaluations. To work with chromosomes, either he would have to have chromosomes as strings of length m where every digit would have to be examined for 10^7 different values which is extremely big as a number. Then a small population of these chromosomes would take extremely large amounts of time to reach the true optimal solution. Analogously, if the length of the chromosome string is extremely long and the initial population of chromosomes is relatively too small, the search space would also remain unexplored at large and the true optimal solution might not be reached [26]. Fortunately, all of these considerations are not of concern of us as we shall see later below.

Now as far as the bio-operations are concerned, for all genetic algorithms, we always have:

- **Selection**: Its function is for each successive generation to select some portion of the existing population (the “parents”) that are destined to breed the population of the following generation (the “children”). Parents are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. There are several options for this to be done, but to name two of them, related to BGA’s, we can have:

  – *roulette selection* where each section of the “roulette” (the algorithm) represents an area equal to the probability of the candidate parent to be selected. This probability of being selected is proportional to the fitness value of every candidate parent (every individual chromosome in the population).

  – *tournament selection* a common choice which selects first randomly candidate parents (individual chromosomes) out of *populations of the same generation* and then selects the best out of these candidate parents. By “best” we do not mean only one, it usually is many we want, one has to order them. To do this, every candidate parent is compared against each other with respect to its fitness value. In order not to always choose “superindividuals”, chromosomes with extreme fitness values, the evaluation and comparison operation which selects either one candidate parent or another has a *reversal probability* which means that the candidate parent with the less optimal fitness value may be –potentially– also selected.

  Many options exist for the **selection** bio-operation. In general they all rely on some rating of the fitness for all chromosomes of the existing populations which can be either strict such as sorting them from most to least optimal and select the best, or allow some initial randomized selection in the beginning before any rating.

- **Crossover**: it follows the Selection bio-operation after the parents have been selected. Its function is to specify how the genetic algorithm combines two parents to form a child for the next generation. In Figure 5.4 one sees the result of crossover in the context of Binary Genetic Algorithm (BGA). It should be apparent from Figure 5.4, that crossover is employed to exchange information contained in the genes of the parents. The so-called “extent” of the crossover operation varies; for example in Figure 5.4 the last four digits were selected to be exchanged from the parents. This could have been the first two digits, the third and the sixth etc. Such choices are usually random. Also it should be mentioned that not all parents undergo crossover of genes by necessity. This is of course another one of the modeler’s choices.

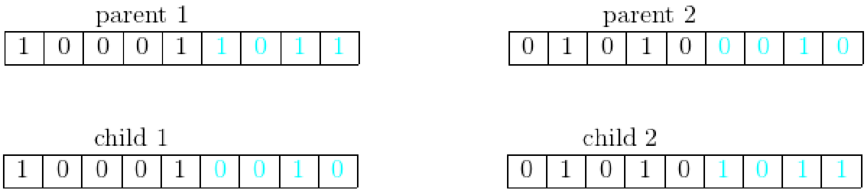


Figure 5.4: Binary Genetic Algorithm crossover scheme

- **Mutation**. After the crossover operation, mutation follows, whose purpose is to randomly change up to a small degree the information contained in the genes inside the children of the new populations. The same way as in nature mutation has only a small probability to happen, the random small change for the genes of children have only a small probability to be realized in the end. Again for BGA, in Figure 5.5 one sees how a random chromosome for a child in the new population is changed. It is up to the modeler to set how low the probability of allowing this mutation is or even the “extent” of this mutation.

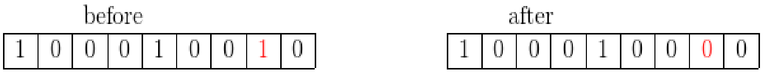


Figure 5.5: Binary Genetic Algorithm mutation scheme

Without delving deeper into this discussion we have to mention that all of these three bio-operations differ extensively when one works with real-coded genetic algorithms. The essence though of their function is the same.

The reader now might ask the following question: why would not the modeler choose *exhaustive search* for binary strings of small length as in Figure 5.3, that is to *test and compare* every possible configuration for a short binary string and find the one that is optimal? The answer is that yes, he probably *should* do this if the binary strings are short enough and not solve the problem with the use of BGA. But if the strings have sufficiently long length, or if the possible values of the genes (termed *alleles*) are too large in number, then genetic algorithms are expected to bring the final optimal result faster. That said, it is true that *for short binary strings exhaustive search is also a viable option.*

6 Analysis and Results

This chapter considers the testing and comparison of three models – the Original Model (OM), the Gaussian Process ($\mathcal{GP}$) model and the Monte Carlo ($\mathcal{MC}$) model – which we will introduce in detail directly below. This chapter uses these models as objective functions of a Binary Genetic Algorithm (BGA) in order to determine an optimized wind farm layout. First, the premises upon which the models were designed are explained, then the results are presented and commented on.

6.1. The Original Model

As explained in Chapter 1, the objective of this thesis is to compare the three aforementioned models when coupled with the genetic algorithm. In Chapter 5 it was explained that all models are supposed to be the objective (or *fitness*) function of the Binary Genetic Algorithm (BGA). In this sense, before starting our analysis directly with explaining what is going on with the Gaussian Process $\mathcal{GP}$ model and the Monte Carlo $\mathcal{MC}$ model, let us first introduce here the original model.

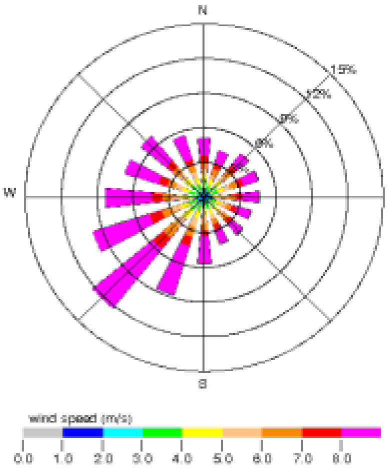


Figure 6.1: An example of a windrose

The original model (OM) is an expected power production model implemented by Bo Hu [27] at the TU Delft in order to assess the layout performance when considering different types of wakes, namely when the meandering effect of the wake is included or not. The necessary inputs for the model in order to work are some parameters of interest, like the model of the wind turbines, their diameter and hub height, and the following variables:

- the geographical coordinates of the layout of the wind farm and
- the windrose of the farm, which as far this thesis is concerned, is the graph of the joint probability mass function for various evaluations of the wind speed (at reference level) and the wind direction, as can be seen in Figure 6.1.

There is no true need to analyze thoroughly the inner workings of the original model because

- the $\mathcal{GP}$ emulates it. Namely, it does not go through the analytic derivation of results based on the rules of physics and wind energy engineering as the original model does, but relates inputs and outputs in its own fashion.

- in the same way, the $\mathcal{MC}$ model only uses it to find results after having acquired with an adequate method random input samples, with which it estimates the expected sample mean of the outputs. Developing a $\mathcal{MC}$ model is mainly to develop a good sampling scheme and get the corresponding outputs from the original model.

All in all, we will introduce the original model here only briefly in order to give the reader an idea on what we base our analysis. It would be informative however to state that the novelty of this particular wake model is based on simplifications to the momentum exchanged between the wind flow and the turbine rotor as well as defining the velocity deficit due to wake with a Gaussian asymmetrical profile, as seen in Figure 6.2 on the right. If the reader wants to search more information about this model, he should refer to Bo Hu's work [27] and the article of Bastankhah and Porté-Agel [5] that it was based upon.

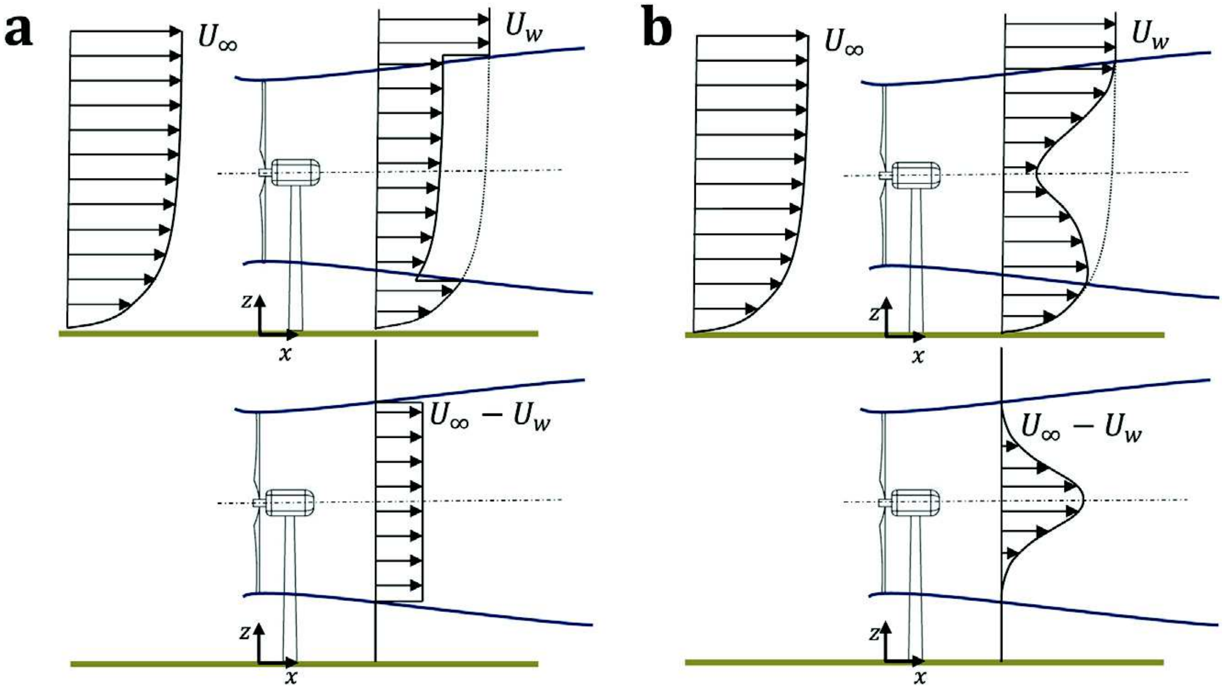


Figure 6.2: Different approaches in the development of wake. Schematic of the vertical profiles of the wind speed (top) and velocity deficit (bottom) downwind of a wind turbine obtained by assuming: (a) a top-hat and (b) an axisymmetrical Gaussian distribution for the velocity deficit in the wake. The first one (a) is a common simplifying modeling choice and appears usually in wake models such as e.g. the Jensen model, while the second one (b) offers a more realistic representation of the velocity deficit downstream of the turbine. Reproduced (with permission by the authors) from the article of Bastankhah and Porté-Agel [5]

So the original model as was coded by Bo Hu can be thought as a function:

$$\mathbb{E}(\tilde{P}) = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{P}(\tilde{v} = v_j|\tilde{\theta} = \theta_i)\mathbb{P}(\tilde{\theta} = \theta_i)g(v_j, \theta_i; \boldsymbol{T}) \qquad \text{with } n \in \mathbb{N} \text{ and } m \in \mathbb{N} \tag{6.1}$$

whose full explanation we will give in the following section after we clarify some things concerning the function g of the original model.

6.2. The General Scheme

To explain the above expression we have first to remind again some terms from previous chapters:

- To start with, as was introduced in Chapter 5, the vector $\boldsymbol{T} = [\boldsymbol{x}\ \boldsymbol{y}]$ of every layout is a vector containing all the geographical coordinates of a layout in a two dimensional $x - y$ plane
- In Figure 1.2 we presented the conceptual map of three models of which now we have some good idea of what they mean and we can introduce them more meticulously. These models were the original model (OM), the Gaussian Process Regression model ($\mathcal{GP}$) and the Monte Carlo model ($\mathcal{MC}$). However, in Figure 1.2 the functions were introduced rather abstractly. In the case of the wind farm layout optimization problem therefore, in order to be more precise, we will have to specify all the terms as follows:
  - The set A of Figure 1.2 is the set $[v_{\text{cut-in}}, v_{\text{cut-out}}] \times [0°360°]$, where the terms $v_{\text{cut-in}}$ and $v_{\text{cut-out}}$ refer to the cut-in and cut-out speed of the wind turbine model as provided by the manufacturer that the layout is populated with. The set A therefore refers to the idea that the power production of the wind turbines is a mapping from the wind speeds (at reference height) and wind directions the turbines face

- the sets B is the set of real numbers $\mathbb{R}$, since the result we would like to compute, the expected power of the layout, is a real number. For similar reasons, the sets $B_{\mathcal{GP}}$ and $B_{s_{\mathcal{MC}}}$ are also the set of real numbers $\mathbb{R}$.
- The sets $S_{\text{in}_{\mathcal{GP}}}$ and $S_{\text{out}_{\mathcal{GP}}}$ are, for a $\mathcal{GP}$ model, the inputs and outputs of the data set of sampled points D (as introduced in Chapter 3, Section 3.4), so only the latter will be specified in more detail in the following sections.
- the set $A_{s_{\mathcal{MC}}}$ is, as in Figure 1.2, a subset of A which depends on what the random samples will be, so the $\mathcal{MC}$ model needs to be reformulated for the *number* of random samples, as we will see in a few lines below.

What is important here is the fact that now we can define two mappings for the OM and the $\mathcal{GP}$ model, namely:

- for the OM model (and to be used for the $\mathcal{MC}$ model) the mapping $g : [v_{\text{cut-in}}, v_{\text{cut-out}}] \times [0°, 360°] \to \mathbb{R}$ and
- for the $\mathcal{GP}$ model the mapping $h_{\mathcal{GP}} : [v_{\text{cut-in}}, v_{\text{cut-out}}] \times [0°, 360°] \to \mathbb{R}$.

These two mappings work differently as explained in previous chapters but they both calculate the power production for a specific wind speed and wind direction of a certain layout. This last remark means that we have to parametrize these mappings by the vector $\boldsymbol{T} = [\boldsymbol{x}\ \boldsymbol{y}]$ as introduced above.

Then the setup of the three models can be looked at in the following way:

- Original model:

$$\mathbb{E}(\tilde{P}) = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{P}(\tilde{v} = v_j | \tilde{\theta} = \theta_i)\mathbb{P}(\tilde{\theta} = \theta_i) g(v_j, \theta_i; \boldsymbol{T}) \qquad \text{with } n \in \mathbb{N} \text{ and } m \in \mathbb{N} \tag{6.2}$$

  where $\tilde{P}$ is the produced power, $\tilde{v}$ and $\tilde{\theta}$ are the wind speed (at reference height) and the wind direction, v_j, θ_i are the possible discrete evaluations of the two quantities whose joint probability $\mathbb{P}(v_j, \theta_i) = \mathbb{P}(\tilde{v} = v_j | \tilde{\theta} = \theta_i)\mathbb{P}(\tilde{\theta} = \theta_i)$ is given by the meteorological data of a specific layout. The vector $\boldsymbol{T}$ that parametrizes the function g represents this layout. Here we consider the wind speed and wind velocity discrete random variables and not continuous. Also, the conversion of wind speeds at reference height to wind speeds at hub height is done internally inside the OM model. This has to be taken into account, when one computes the $v_{\text{cut-in}}$ and the $v_{\text{cut-out}}$ speeds. Lastly, the natural numbers n and m as declared above can be any numbers and it is only a matter of modelling preference how to discretize the continuous random variables using a regular discretization grid. In this thesis they were chosen to be $n = 24$ and $m = 180$, because the meteorological data we have at our disposal allowed us to increase the accuracy offered by a more fine discretization. We will refer to this point in the coming sections.

- $\mathcal{GP}$ model:

$$\mathbb{E}(\tilde{P}) \approx \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{P}(\tilde{v} = v_j | \tilde{\theta} = \theta_i)\mathbb{P}(\tilde{\theta} = \theta_i) h_{\mathcal{GP}}(v_j, \theta_i; D, \boldsymbol{T}) \qquad \text{with } n \in \mathbb{N} \text{ and } m \in \mathbb{N} \tag{6.3}$$

  where we emulate g with the $\mathcal{GP}$ model $h_{\mathcal{GP}}$. This is still an approximation because there might be some error involved, which has to do with the data set of the sampled points as introduced in Chapter 3, Section 3.4, where here it is $D = \{(x_{+1}, y_{+1}), \ldots, (x_{+|D|}, y_{+|D|})\}$ (more on this below). Notice that the parametrization to the data set of sampled points D is crucial, because the $\mathcal{GP}$ model cannot be run "on its own" without first accepting some sampled inputs from the original model. Therefore there is a question how many sampled points the $\mathcal{GP}$ model needs in order to perform adequately. Lastly, in order to compare the OM with the $\mathcal{GP}$ model on the same basis, the natural numbers n and m were again chosen to be as above $n = 24$ and $m = 180$

- $\mathcal{MC}$ model:

$$\mathbb{E}(\tilde{P}) = \mathbb{E}(\tilde{\overline{P}}_k) \approx \frac{1}{k}\sum_{s=1}^{k} P_s = h_{\mathcal{MC}}(k; \boldsymbol{T}) = \frac{1}{k}\sum_{s=1}^{k} g(v_s, \theta_s; \boldsymbol{T}) \text{ and } (\tilde{v}_s, \tilde{\theta}_s) \sim \text{pmf}_{v,\theta},\ \tilde{P}_s \sim \text{pmf}_P, \text{ and } n, m \in \mathbb{N} \tag{6.4}$$

  where $k \in \mathbb{N}$ is the number of samples we use for approximating the expected value of the produced power $\mathbb{E}(\tilde{P})$. Here, as in Chapter 4, we use *the sample mean of the random outputs of power $\tilde{P}_s$ which are determined from the random samples of wind speed $\tilde{v}_s$ and wind direction $\tilde{\theta}_s$*. The reader can compare the above equation with Equation 4.12. We also introduce a new mapping that relates specifically to the $\mathcal{MC}$ model which is $h_{\mathcal{MC}} : \mathbb{N} \to \mathbb{R}$. The need to parametrize this mapping with the vector $\boldsymbol{T}$ to declare a specific layout appears also here. There is no discretization scheme but a sampling scheme, this is why the natural number k plays an important role in $\mathcal{MC}$ model. The joint probability mass function $\text{pmf}_{v,\theta}$ comes from the meteorological data. The probability mass function pmf_P however we do not know. Lastly the observation that $\mathbb{E}(\tilde{P}) = \mathbb{E}(\tilde{\overline{P}}_k)$ was referred on Equation 4.5, Chapter 4.

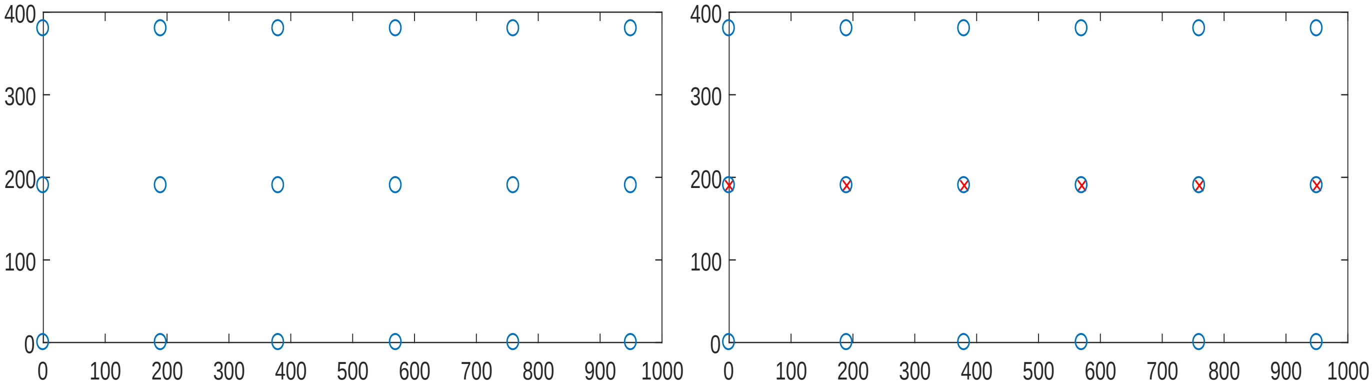
Notice that since all these Equations are comparable with each other in the sense that the $\mathcal{GP}$ and the $\mathcal{MC}$ provide an approximation to what the expected produced power is, so an approximation of the result of the OM model. Since their results are comparable ***for the same layout***, we can use them as fitness function to the genetic algorithm for every specific layout that we examine and compare them as objective functions to a wind farm layout optimization setup. We are ready now to continue with the analysis and first tests of the BGA for the layouts.

6.3. The Binary Genetic Algorithm - Development and Verification

In Chapter 5 we laid down the foundation of the features the Binary Genetic Algorithm should have. It was decided that we will use a discretized search space with fixed possible positions for an also pre-fixed number of wind turbines to be installed. These positions should be inside a given wind farm site of interest for which we have meteorological data (namely the joint wind speed and the wind direction probability mass function). Also, the minimum distance between the turbines, *in every wind direction*, should always be in the far-wake region and *never* in the near-wake region, as mentioned in Chapter 5, Section 5.2.

With these prerequisites in mind, we developed a code in MATLAB using the inbuilt MATLAB Optimization Toolbox and specifically its BGA. In order to express the search space, we have to enumerate the possible positions of the wind farm site. When a wind turbine is placed in a possible position, the string cell (gene) gets a 1, otherwise a 0. There are other parameters that we have to specify in the BGA, however before we build the final version, it was decided that there should be some simple tests so as to **verify how the BGA actually performs**. It would be unwise to simply run the BGA without testing it first on simpler wind farm layouts than the one we are planning on using it for.

To this end, we designed a test version of BGA with fitness function specifically the original model. The search space was kept small and simple, namely, it was decided to consist of a rectilinear wind park of 3 rows by 6 columns of 18 possible positions. Its minimal row-wise and column-wise pairwise distance of any two turbines was decided to be equal to 190 m. The grid of 18 possible positions can be seen in Figure 6.3a.



(a) A grid of 18 possible positions

(b) The best layout for wind directions 0° and 180° found by the BGA

Figure 6.3: An overview of the grid for which the BGA is verified and an example of a best layout that the BGA indicated

To test the BGA for this specific layout, we needed to determine which meteorological conditions are of interest, in the sense that they would be indicative what would we expect for the best layout to be. Before we continue explaining the modeling details of the BGA, we would like to discuss its results so that the reader can appreciate that it works as desired. All the modeling details will be given at the end of this section.

So to proceed directly to the findings of the tests we run, we present in Figure 6.3b the best layout inside this rectilinear grid when:

- the direction of the wind is **always** 0° (coming from the North – see Figure 6.1) , so that $\mathbb{P}(\tilde{\theta} = 0°) = 1$ and
- the wind speed of this direction is **always** equal to 10 m/s (at reference height), so that $\mathbb{P}(\tilde{v} = 10\ \mathrm{m/s} \mid \tilde{\theta} = 0°) = 1$

As one can see in Figure 6.3b the result of the developed BGA is the desired result: all turbines, when facing the wind coming from the North, cause no other turbine to suffer from wake losses, since there are no downstream turbines for this layout and this combination of wind direction and wind speed. The following remarks however are true:

- the best layout is not only one for this particular grid of potential positions; if any of the turbines changes its position northwards or southwards by placing itself in any other row, then the layout still has no wake losses. Therefore the layout of Figure 6.3b – and all others that will be presented as results to the tests – are **indicative** and *might* not be the only best layouts for given wind direction and wind speeds
- it is not only wind direction $\theta = 0°$ that has no downstream turbines for this particular proposed best layout – e.g. for $\theta = 180°$ the situation is the same
- the wind speed as $\tilde{v} = 10\ \mathrm{m/s}$ coming from this direction is not the only one that can "guarantee" that the best layout is the one of Figure 6.3b. One can imagine that if e.g. one assumes $\mathbb{P}(\tilde{v} = 12\ \mathrm{m/s} \mid \tilde{\theta} = 0°) = 1$ the same way we assumed deliberately $\mathbb{P}(\tilde{v} = 10\ \mathrm{m/s} \mid \tilde{\theta} = 0°) = 1$ above the result will be the same. **However** there is an **"upper limit"** for increasing the wind speed that this continues to happen and it is directed by the power curve of the turbine model, namely for Vestas V80 (rated power is 2000 kW), as seen in Figure 6.4. That is, if e.g. the outermost turbine experiences free-stream

wind speed that is 25 m/s and all the downstream turbines experience wakes (i.e. velocity deficits) that still lead to wind speeds above 20 m/s then *whichever the layout*, it does not eventually matter, because all turbines will produce always 2000 kW. Of course setting $\mathbb{P}(\tilde{v} = 25 \text{ m/s} \mid \tilde{\theta} = 0°) = 1$ is an extreme approach due to the fact that these types of high wind speeds are not frequent at all in usual wind farms.

Acknowledging therefore that for the tests we would like to conduct the wind speed v does not play a crucial role below a certain upper limit it was deemed enough to have it always set to 10 m/s $\left(\mathbb{P}(\tilde{v} = 10 \text{ m/s} \mid \tilde{\theta} = \theta_i) = 1, 1 \leq i \leq n = 180\right)$ where θ_i is *any* evaluation for the wind direction. Also, it was deemed necessary to test the BGA also for $\theta = 180°$ and the result (the best layout) was also the layout of Figure 6.3b, which is satisfying.

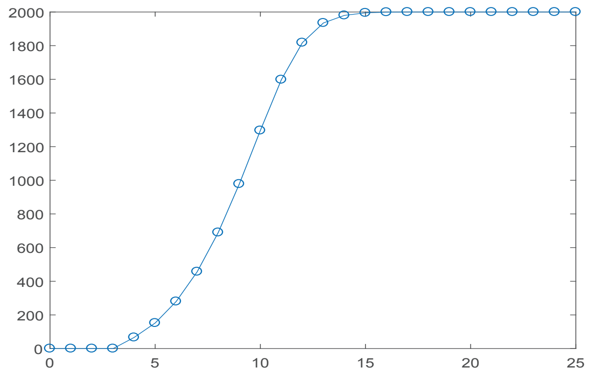
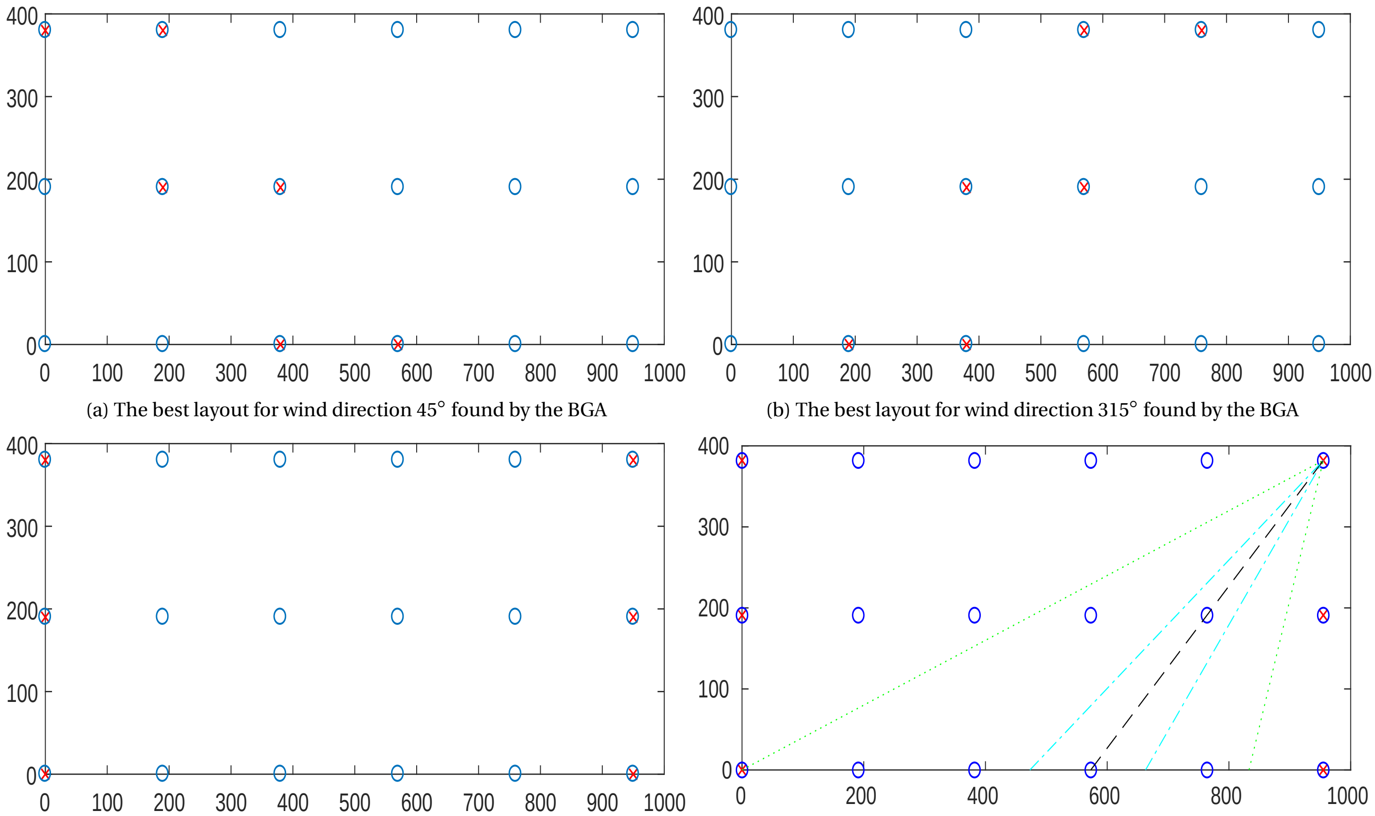


Figure 6.4: The power curve of Vestas V80, the turbine model for which we verify the BGA. The wind speeds here are denoted at hub height. Notice that for very large wind speeds the power produced is equal to the rated power, 2000 kW.

Continuing with the tests, we experimented with more wind directions. In Figures 6.5a and 6.5b and in Figure 6.5c, one sees the best layout indicated by the BGA we tested for wind directions: 45° for Figure 6.5a, 315° for Figure 6.5b and 90° or/and 270° for Figure 6.5c, all with respect to clockwise angle measurements and with 0° again being the North.



(a) The best layout for wind direction 45° found by the BGA

(b) The best layout for wind direction 315° found by the BGA

(c) The best layout for wind direction 90° or 270° found by the BGA. In our case this is a good layout also for wind direction 45° or 315°, because the rate of radius expansion for the wake is adequate

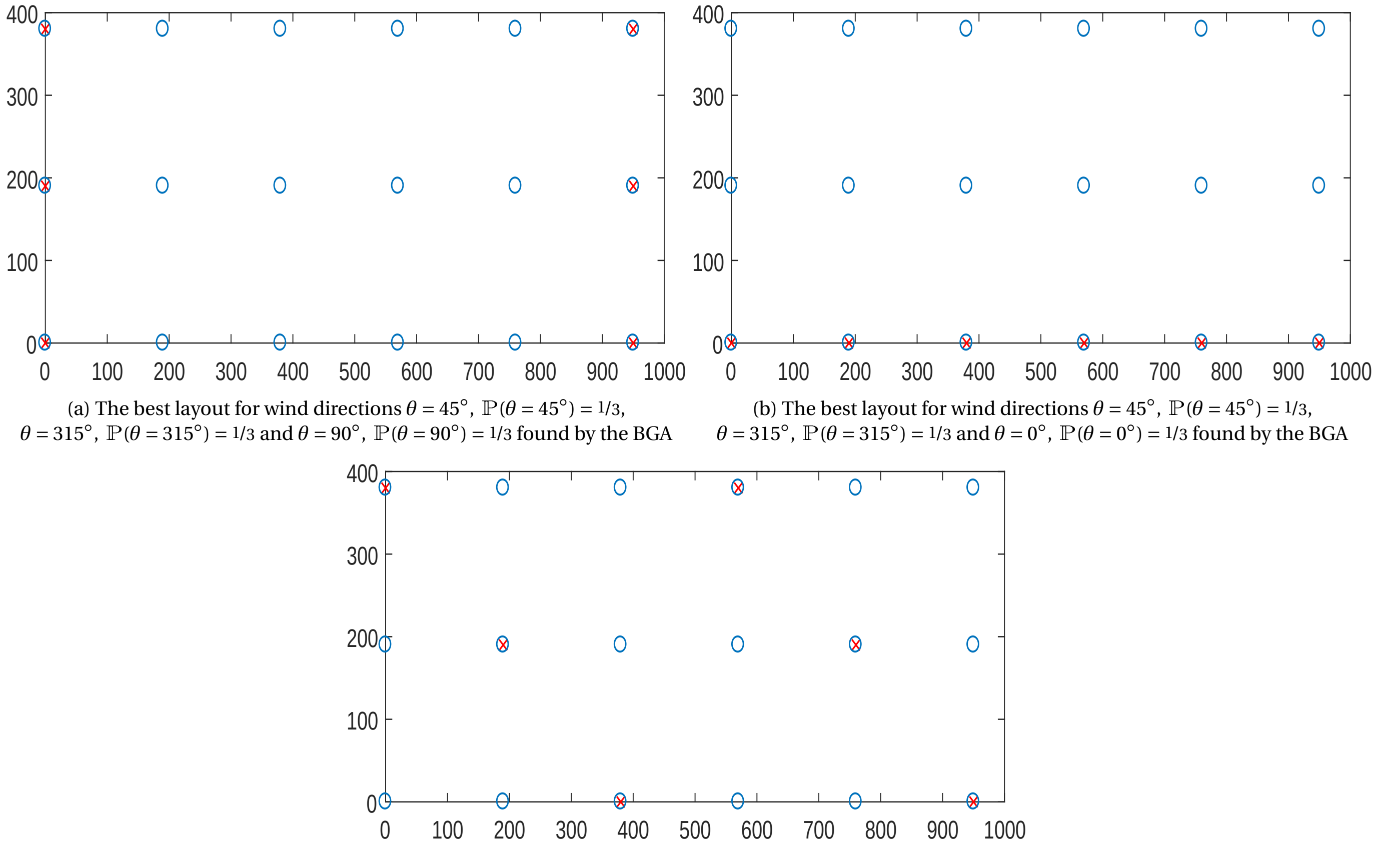
(d) Visual explanation why the previous layout with different rates of radius expansion for the wake (green dotted or blue dash-dotted lines) might or might not be one of the best layouts for 45°.

Figure 6.5: Discussion and analysis about the best layouts indicated by the BGA for various wind directions

These results are also what one would want because for these wind directions there is again no downstream turbine placed behind any single turbine. Starting with Figure 6.5c, one sees that the turbines have to be placed in two vertical columns that are placed as far apart from each other as possible in order to reduce the resulting wake losses. This is the same situation, either one examines wind direction of 90° or 270°, due to the symmetry of the layout. This layout is indeed the single best layout due to the available positions in the grid. On the other hand, the layouts in Figure 6.5a and 6.5b are not the single best layouts, because one could move all turbines either Westwards or Eastwards and have again equally potent good layouts.

But for wind directions 45° or 315° even the layout in Figure 6.5c could be another potent candidate! To prove this, one has to turn to Figure 6.5d. There, we suppose a random turbine in this layout, e.g. the one in (950,0), as being the outermost turbine facing wind or 45° direction. The axis of the wake propagating downstream in this direction is shown as the black dashed line. Now the wake radius is expanding as the downstream distance from the upstream turbine is increased. This is shown in Figure 6.5d with two types of delineating areas where the wake effects take place, one area determined from a set of green dotted lines and another determined from a set of cyan dash-dotted lines. These two areas involve the same axis of the wake propagating downstream (the black dashed line), but the rate of radius expansion of the wake is different, so one area does include and affect another turbine (the one at (0,0)), while the other does not. This means that in the first case at least, this layout cannot be considered one of the best since there are wake losses. Therefore, it all comes down to this rate of radial expansion for the wake, which is determined by the modeller. **The only reason this discussion is mentioned here** is because indeed this rate of radial expansion for the wake in our wake model – the expected power production model implemented by Bo Hu [27], the original model (OM) – is small enough and the layout in Figure 6.5c is also one of the best layouts either for 45° or 315°. **This discussion will also shed light on the results of all the following figures of this section and prove why they are acceptable results of the BGA.**

Next, we examine **combination of wind directions with proper probabilities of occurrence for each of them.** Again we set the wind velocity to $v = 10$ m/s, for all wind directions we will present. In Figure 6.6a we see the best layout indicated by the BGA when the wind direction is either 45° or 315° or 90° with $\mathbb{P}(\tilde{\theta} = 45°) = \mathbb{P}(\tilde{\theta} = 315°) = \mathbb{P}(\tilde{\theta} = 90°) = 1/3$. From what was pointed out in the previous paragraphs it should be apparent to the reader why this should be the case. In Figure 6.6b we have the best layout found by the BGA when the wind direction is either 45° or 315° or 0° with $\mathbb{P}(\tilde{\theta} = 45°) = \mathbb{P}(\tilde{\theta} = 315°) = \mathbb{P}(\tilde{\theta} = 0°) = 1/3$. Again, as told above, this may not be the single best layout for this case, because all turbines can move on the upper or the middle row simultaneously and still get the same expected power production with no wake losses involved. Lastly, in Figure 6.6c we have the best layout found by the BGA when the wind direction is either 0° or 90° with $\mathbb{P}(\tilde{\theta} = 0°) = \mathbb{P}(\tilde{\theta} = 90°) = 1/2$. Notice that in this case, the horizontally mirrored layout is also an option.



(a) The best layout for wind directions $\theta = 45°$, $\mathbb{P}(\theta = 45°) = 1/3$, $\theta = 315°$, $\mathbb{P}(\theta = 315°) = 1/3$ and $\theta = 90°$, $\mathbb{P}(\theta = 90°) = 1/3$ found by the BGA

(b) The best layout for wind directions $\theta = 45°$, $\mathbb{P}(\theta = 45°) = 1/3$, $\theta = 315°$, $\mathbb{P}(\theta = 315°) = 1/3$ and $\theta = 0°$, $\mathbb{P}(\theta = 0°) = 1/3$ found by the BGA

(c) The best layout for wind directions $\theta = 0°$, $\mathbb{P}(\theta = 0°) = 1/2$ and $\theta = 90°$, $\mathbb{P}(\theta = 90°) = 1/2$ found by the BGA

Figure 6.6: Discussion and analysis over the best layouts indicated by the BGA for various combinations of wind directions and various probability distributions

So as it seems the BGA passed all the tests that we put it through and we considered them as the most indicative ones. This was done with some parameter modeling of the BGA that we are going to discuss in the next paragraph. For the time being, with these results, – we argue – that the modeling of the BGA allows us to trust that the results of the MATLAB algorithm we developed are reliable.

As far as modelling the BGA goes, it was decided that Table 6.1 of parameters is adequate for proceeding with the use of the BGA in bigger layouts. These parameters which gave the previous results were acquired through trial and error. However, since examining bigger layouts of turbines means expanding the search to bigger grids (more than 18 possible positions) such parametrization of the BGA might not be perfect, especially since there is a MATLAB inbuilt tolerance on how precisely (on how many decimal digits) the objective function gets calculated and differs from its evaluations in previous iterations. In simpler words, the iterative process might yield sub-optimal solutions because not all of the search space, if it is big enough, can be examined. Such problems though appear in all optimization algorithms and as discussed in Chapter 5, not every one of them reaches the global extremal points.

Table 6.1: Binary Genetic Algorithm parametrization

Parameters	Input	Information
Generations	50	Stall generation limit is 20
Population Size	100	The number of chromosomes in every population
Selection Type	Tournament (2 random parents)	Refer to Chapter 5
Mutation Rate	0.5	Probability of mutation for every chromosome
Crossover Rate	0.8	Probability of crossover for every chromosome
Objective Function Tolerance	10^{-3}	Lower bound of difference between evaluations of the objective function in each successive iteration

6.4. Meteorological Data

To judge which wind farm layout is the best among the many candidate layouts, one must know the meteorological conditions that the site faces. Such knowledge has to do with static estimators of the wind and comes from historical meteorological data gathered in time series of significant amount of years. Such data often come into the form of measurements from various measuring units and sensors that provide in fixed intervals of time the magnitude of the quantity observed. A big amount of these retrieved data is in essence samples which justifies statistical inference and deductions about it, such as what is the average wind speed in the site, etc.

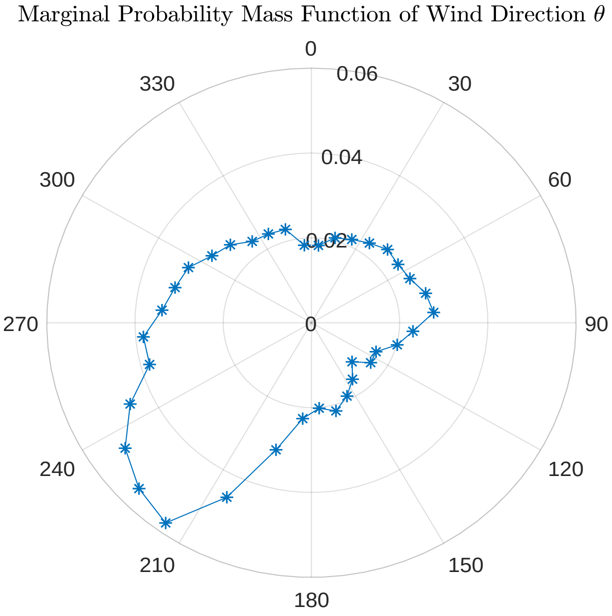


Figure 6.7: Marginal probability mass function of wind direction θ. All probabilities of the form $\mathbb{P}(\tilde{\theta} = \theta)$ are shown here for the wind farm of our study

In our case, our most important wind data are of course wind speed and wind direction as already been discussed. In order to have realistic wind data to work with for the site of our selection, wind speed and wind direction time series was retrieved from the OWEZ farm met mast in the North Sea [35]. A time series of 273209 10-minute measurements retrieved over a period of 6 years was processed and the two following figures were produced. In Figure 6.7 we see the marginal probability mass function of the wind direction. It is from here that one can assess what is the probability $\mathbb{P}(\tilde{\theta} = \theta)$ for every $\theta \in (0°, 360°)$. In Figure 6.8 one sees the group of histograms of 24 bins of the wind speed υ referring to 36 bins of the wind direction θ ranging from 0° to 72°. For wind directions that do not range between the interval $(0°, 72°)$ there are other sets of histograms for the wind speed that can be seen in the Appendix Section B.1 – here we showcase only one.

It is from here that one can assess what is the probability $\mathbb{P}(\tilde{v} = v | \tilde{\theta} = \theta)$ for every $v \in [v_{\text{cut-in}} = 3\text{ m/s}, v_{\text{cut-out}} = 24\text{ m/s}]$ given some $\theta \in (0°, 360°)$. The range of wind speeds of interest has to do with the cut-in and cut-out speeds of the wind turbine model; for the Vestas V80 model is $v_{\text{cut-in}} = 3\text{ m/s}$ and $v_{\text{cut-out}} = 25\text{ m/s}$ [27]. Notice that we placed a lower limit for the cut-out speed (24 m/s) because the reference height that the meteorological data were collected was 10 meters below the hub height of our wind turbines' assumed hub height (80 meters) and it is known the wind profile increases vertically with increasing height due to wind shear [7].

We can address again the discretization scheme we discussed in Equations 6.2 and 6.3. We chose $n = 24$ and $m = 180$. This means that we would like wind speeds represented in bins of 1 m/s width and wind directions represented in bins of 2°, which agrees with what was written above and can be distinguished in Figures 6.8 and in the Appendix Section B.1. Lastly, notice that this gives $nm = 24 \times 180 = 4320$ different joint probabilities of wind speed and wind direction. We will meet this number again in the following sections.

We are ready to introduce the upcoming sections. It is with this type of statistic data that

- we will assess the joint probability $\mathbb{P}(\tilde{v}_j, \tilde{\theta}_i) = \mathbb{P}(\tilde{\theta} = \theta_i)\mathbb{P}(\tilde{v} = v_j | \tilde{\theta} = \theta_i)$ with $1 \leq i \leq n \in \mathbb{N}$ and $j \leq m \in \mathbb{N}$ that shows up both in Equations 6.2 and 6.3 as well as

- draw samples through the use of direct sampling for evaluating Equation 6.4

.

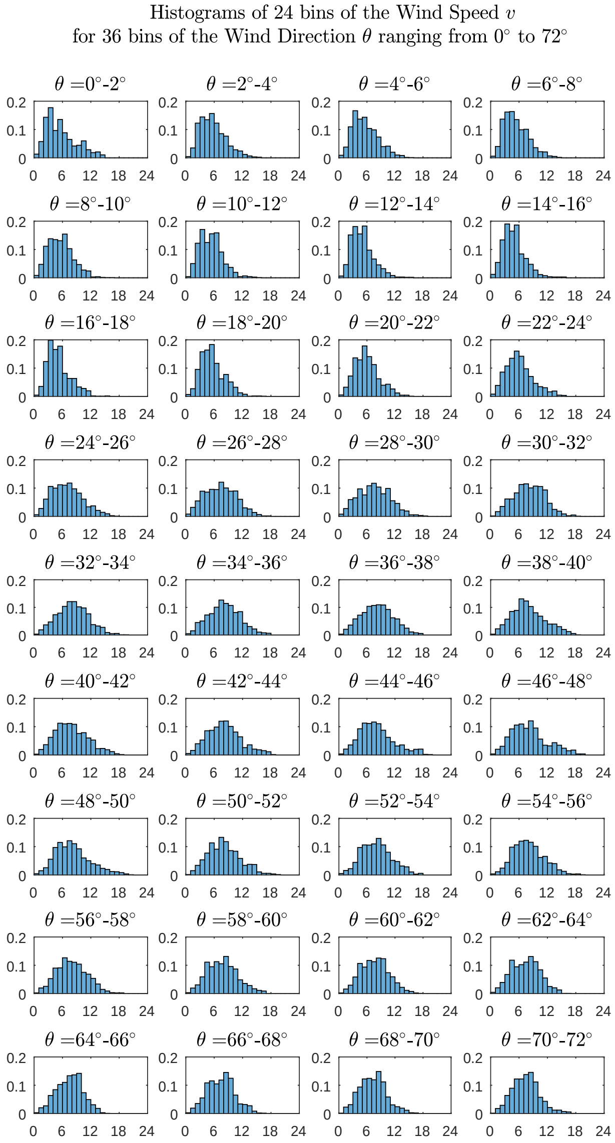


Figure 6.8: Histograms of wind speed v arranged in 36 bins for given range of wind direction θ ranging from 0° to 72°. All probabilities of the form $\mathbb{P}(\tilde{v} = v | \tilde{\theta} = \theta)$ are shown here as well as in Appendix Section B.1 for the wind farm site of our study

6.5. The Gaussian Process Model

In the previous section we showed how the joint probability of the random variables $\tilde{v}$ and $\tilde{\theta}$ can be determined. Once this is done for every combination of wind speed and wind direction, one can find the expected power production of a specific wind farm layout with the use of the Original Model (OM). As explained in Equation 6.3, we would like to emulate the results of the OM with the Gaussian Process ($\mathcal{GP}$) model. Then when the $\mathcal{GP}$ model is ready we can use it as an objective function for the BGA in order to find the best layout in terms of expected power production.

In order to proceed, we need to determine the data set of the **sampled points** D and the set of the **non-sampled points** R as were introduced in Chapter 3, Section 3.4. We remind the reader that as in Chapter 3:

- For the simple case of a Gaussian process $\mathcal{Y} = \{\tilde{y}_x : x \in X\}$, $|X| \in \mathbb{N}$ we can define the input vector $\boldsymbol{x}$ and the output vector $\tilde{\boldsymbol{y}}$
- We can partition the two vectors as $\boldsymbol{x} = [\boldsymbol{x}_+ \boldsymbol{x}_*]^\top$ and $\tilde{\boldsymbol{y}} = [\tilde{\boldsymbol{y}}_+ \tilde{\boldsymbol{y}}_*]^\top$ where
  - the "+" symbol of the vectors $\boldsymbol{x}_+$ and $\tilde{\boldsymbol{y}}_+$ represents the elements of the input vector $\boldsymbol{x}$ that we sampled from and the corresponding elements of the output vector $\tilde{\boldsymbol{y}}$, whose partition $\tilde{\boldsymbol{y}}_+$ when sampled, has evaluations $\boldsymbol{y}_+$,
  - The "*" symbol of the vectors $\boldsymbol{x}_*$ and $\tilde{\boldsymbol{y}}_*$ represents the elements of the input vector $\boldsymbol{x}$ that we did not sample from and the corresponding elements of the output vector $\tilde{\boldsymbol{y}}$
- the **set of non-sampled points** R is the set whose elements are the pairwise combinations of the vectors $\boldsymbol{x}_*$ and $\tilde{\boldsymbol{y}}_*$ so that $R = \{(x_{*_1}, \tilde{y}_{*_1}), (x_{*_2}, \tilde{y}_{*_2}), (x_{*_3}, \tilde{y}_{*_3}), \dots, (x_{*_{|R|}}, \tilde{y}_{*_{|R|}})\}$ and the set $D = \{(x_{+_1}, y_{+_1}), (x_{+_2}, y_{+_2}), (x_{+_3}, y_{+_3}), \dots, (x_{+_{|D|}}, y_{+_{|D|}})\}$, the **data set of sampled points** whose elements are the pairwise combinations of the vectors $\boldsymbol{x}_+$ and $\boldsymbol{y}_+$.

This case however is different because there is a two-dimensional stochastic process since the index values the power production for a given layout is evaluated for both the wind speed and the wind direction. Therefore, in order to be compatible with what is written above, the **data set of sampled points** in our case is $D = \Big\{\big((v,\theta)_{+_1}, P_{+_1}\big), \big((v,\theta)_{+_2}, P_{+_2}\big), \dots, \big((v,\theta)_{+_{|D|}}, P_{+_{|D|}}\big)\Big\}$ and the **set of non-sampled points** is $R = \Big\{\big((v,\theta)_{*_1}, \tilde{P}_{*_1}\big), \big((v,\theta)_{*_2}, \tilde{P}_{*_2}\big), \dots, \big((v,\theta)_{*_{|R|}}, \tilde{P}_{*_{|R|}}\big)\Big\}$.

As far as the data set D goes, the values of wind speed and wind direction we sample should be selected carefully in order for the data set to have the least number of elements possible. This is because, the output values P_+ that correspond to the sampled values $(v,\theta)_+$ are determined with the help of the OM and if the data set D is overpopulated with many elements, this would mean that the $\mathcal{GP}$ model relies too much on the help of the OM model. The goal is to make an effective $\mathcal{GP}$ model that can be competitive to the OM model by utilizing it only to draw the smallest number of samples with which it can determine the power output $\tilde{P}_*$ for any input values $(v,\theta)_*$ of the set R.

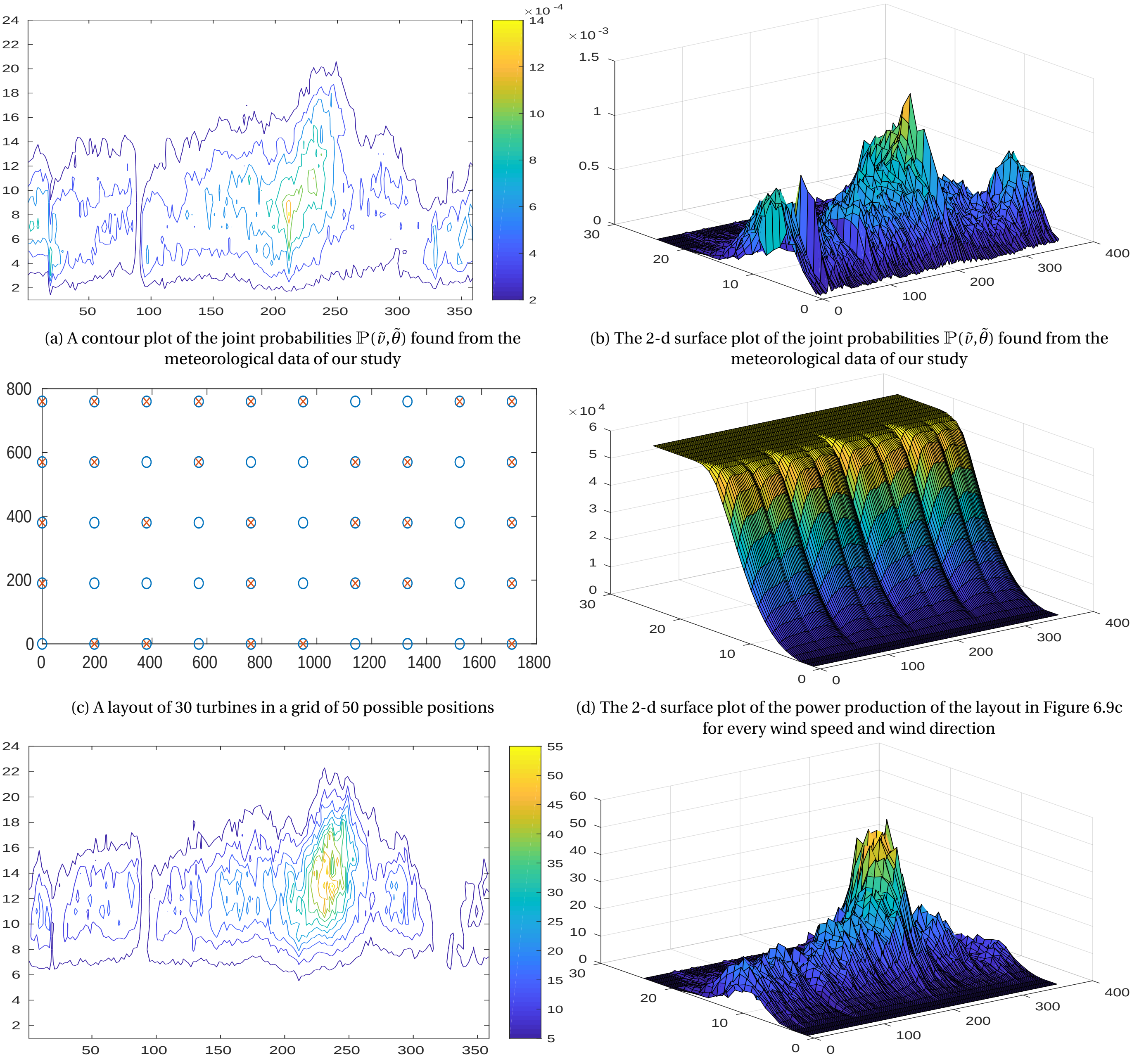
So a quick prioritization of the good modelling of the $\mathcal{GP}$ regression in this case requires the following:

- **a good sampling scheme**, so what would be the elements of the data set D,
- **a good selection of the covariance function** in order to estimate adequately well the expected values for every power output $\tilde{P}_*$ for any input values $(v,\theta)_*$ of the set R,
- **a good selection of the hyperparameters** in order for the hyperparameter optimization algorithm (more on that later) to reach the global extremal point of Equation. 3.27

It should be apparent to the reader that the three aforementioned prerequisites are interdependent on one another. For example, a bad selection for the covariance function will imply a bigger dependency on more samples so that the expected value of every power output $\tilde{P}_*$ of the set R for its input values $(v,\theta)_*$ will be closer to the true power output, the one of the OM model. On the other hand, a bigger data set D with more sample points, might potentially imply a worsening hyperparameter optimization performance; this is because as one sees in Equation 3.27, in which one sees the term $\boldsymbol{y}_+$, the output vector of the sampled points, the more the sampled points are involved, the more elements this vector $\boldsymbol{y}_+$ contains, so this might mean that the hyperparameter set satisfying all those points might not be easy to find easily, as the search space for solving Equation 3.27 might be non-smooth (refer to Figure 3.6d). This would mean that it would be difficult to find the adequate set of hyperparameters that "conforms" as best as it can with the given sampled points. Lastly there is the dependency on the set of hyperparameters and the covariance function which has been discussed already in Chapter 3.

Therefore, in order to model the GP model correctly, it is best if one has a good educated guess on what might be the results of the OM model. Better yet, one should first acquire the results of various layouts of interest and visualize them in order to learn from the observed outcomes how to model properly the $\mathcal{GP}$ surrogate model. To this end, various layouts were tested with the help of the OM model and results such as the ones shown in Figures 6.9c – 6.9e were retrieved. The layout in Figure 6.9c has **30 turbines in a grid of 50 possible positions**, with minimum distance between the turbines 190 m.

First let us recall the Equation 6.2 where we see the term $g(v_j, \theta_i; \boldsymbol{T})$ and the term $\mathbb{P}(\tilde{v}_j, \tilde{\theta}_i)$ with $1 \leq i \leq n \in \mathbb{N}$ and $j \leq m \in \mathbb{N}$. In Figures 6.9a and 6.9b one sees the joint probability mass function of the wind farm site retrieved by the meteorological data of the previous section. In Figure 6.9b the two horizontal axes depict the evaluations of $\tilde{v}_j$ and $\tilde{\theta}_i$, while the vertical axis depicts the probability $\mathbb{P}(\tilde{v}_j, \tilde{\theta}_i)$ with $1 \leq i \leq n \in \mathbb{N}$ and $j \leq m \in \mathbb{N}$. Notice how the indexes i and j, which are common for Equations 6.2 and 6.3, are not arbitrary but denote the discretization of the wind speed and wind direction and their joint probability which comes from the statistical data gathered in the previous section. Eventually, the number of joint probabilities calculated for Figure 6.9b was 4320 ($nm = 24 \times 180$), corresponding to an equal amount of combinations of wind speed and wind direction.



(a) A contour plot of the joint probabilities $\mathbb{P}(\tilde{v},\tilde{\theta})$ found from the meteorological data of our study

(b) The 2-d surface plot of the joint probabilities $\mathbb{P}(\tilde{v},\tilde{\theta})$ found from the meteorological data of our study

(c) A layout of 30 turbines in a grid of 50 possible positions

(d) The 2-d surface plot of the power production of the layout in Figure 6.9c for every wind speed and wind direction

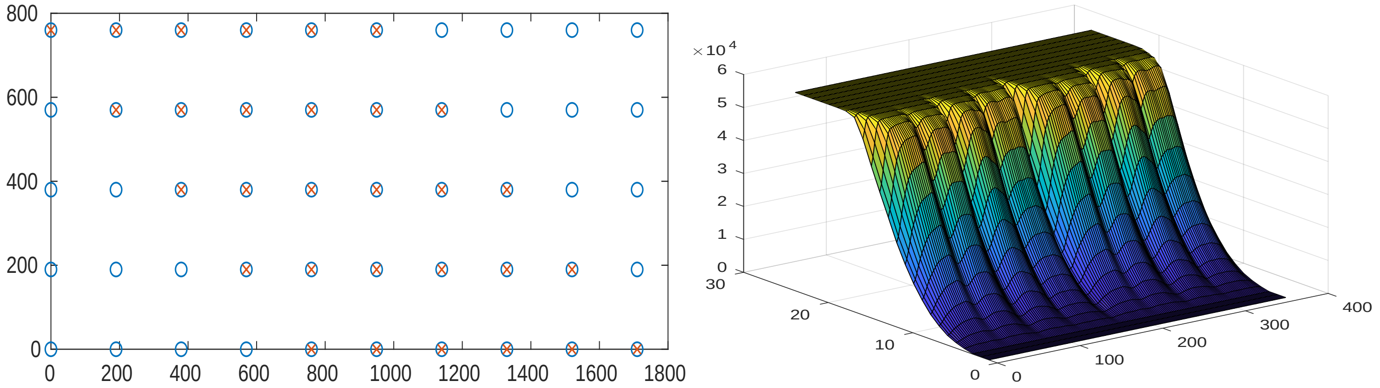
(e) A contour plot of the *probability-weighted* power production of the layout in Figure 6.9c for every wind speed and wind direction

(f) The 2-d surface plot of the *probability-weighted* power production of the layout in Figure 6.9c for every wind speed and wind direction. It is the element-by-element product of 6.9b and 6.9d

Figure 6.9: Various graphs of interest for the layout in Figure 6.9c **calculated with the OM model**

In Figure 6.9d one sees the results of the power production *for a given layout* calculated for all evaluations of wind speed and wind direction. This calculation has the equivalent expression of $g(v_j,\theta_i;\boldsymbol{T})$ of Equation 6.2. Their element-by-element product gives Figure 6.9f – a contour plot again is given in Figure 6.9e.

Figure 6.9d as told before, refers for a given layout, but one can draw conclusion from its form, regardless the specific layout or the number of turbines in it. What should be apparent immediately is the fact that across one wind direction, for all wind speeds, one sees a similar shape like that of the power curve (Figure 6.4), aside the fact that there might be wake losses for the particular wind direction. The latter would mean that there will be a vertical "dips" for the power production in this wind direction for all wind speeds. As was verified by several runs of the OM model, if the layout is different from the one of the Figure 6.9c, these vertical "dips" might be observed for different wind directions, but the "general" form will be similar to the one of Figure 6.9d. For example, in Figure 6.10b one sees another layout for which a similar power production graph is depicted.

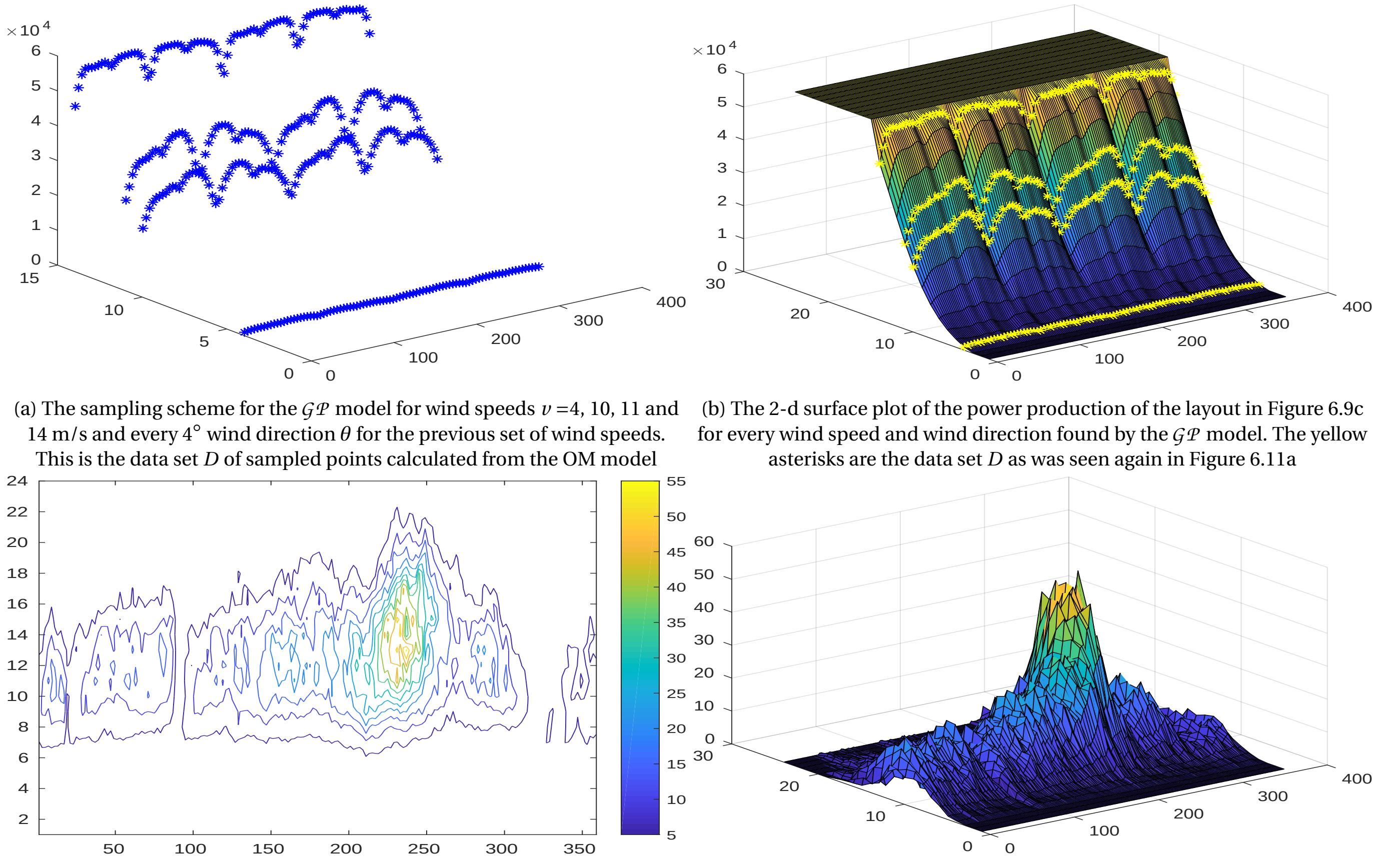


(a) A *different* layout of 30 turbines in a grid of 50 possible positions than in Figure 6.9c

(b) The 2-d surface plot of the power production of the layout in Figure 6.10a for every wind speed and wind direction. Although it is different than the previous graph in Figure 6.9d, the shape of the two curves is similar

Figure 6.10: Similarity between the power production graphs of two randomly selected layouts, calculated by the OM model for every wind speed and wind direction

This leads us to the discussion regarding how to model the $\mathcal{GP}$. For reasons to be discussed later, it was decided to model directly the curve of the power production for all wind speeds and all wind directions in Figure 6.9d and not the curve of the *probability-weighted* power production for all wind speeds and wind directions Figure 6.9f. To this end, the covariance function that was chosen to be tested was the squared exponential covariance function which is the most typical choice for most $\mathcal{GP}$ modelling problems. However, this problem concerns a two dimensional stochastic process with two indexes. One of the most challenging aspects of this problem was to find the adequate toolbox, able to perform regression in for a two dimensional stochastic process. In the end, the GPstuff MATLAB toolbox was utilized [53]. The toolbox can calculate for every input $(v,\theta)_*$ the expected value and the variance of the output $\tilde{P}_*$, but also can optimize the set of hyperparameters, as it has an inbuilt optimizer of its own.



(a) The sampling scheme for the $\mathcal{GP}$ model for wind speeds v =4, 10, 11 and 14 m/s and every 4° wind direction θ for the previous set of wind speeds. This is the data set D of sampled points calculated from the OM model

(b) The 2-d surface plot of the power production of the layout in Figure 6.9c for every wind speed and wind direction found by the $\mathcal{GP}$ model. The yellow asterisks are the data set D as was seen again in Figure 6.11a

(c) A contour plot of the *probability-weighted* power production of the layout in Figure 6.9c for every wind speed and wind direction calculated from $\mathcal{GP}$ model

(d) The 2-d surface plot of the *probability-weighted* power production of the layout in Figure 6.9c for every wind speed and wind direction calculated from $\mathcal{GP}$ model. It is the element-by-element product of 6.9b and 6.9d

Figure 6.11: Various graphs of interest for the layout in Figure 6.9c **calculated with the** GP **model**

So out of the three prerequisites given above, the set of the hyperparameters and the covariance function are decided. The last thing that was left to decide is the sampling scheme. The goal here was to utilize as little as possible the OM model in order to create the necessary data set D. The final sampling scheme was based on several tries, whose criterion of success was the minimization of error for the expected value $\mathbb{E}(\tilde{P}_*)$ for all values of input $(v,\theta)_*$ and all tried layouts, compared to the power production as calculated with the OM model. Therefore, the sampling scheme includes sampling for wind speeds v =4, 10, 11 and 14 m/s and every 4° wind direction θ for the previous set of wind speeds. This means that in the end one samples 360 points of the form $D = \Big\{\big((v,\theta)_{+1}, P_{+1}\big), \big((v,\theta)_{+2}, P_{+2}\big), \big((v,\theta)_{+3}, P_{+3}\big), \ldots, \big((v,\theta)_{+360}, P_{+360}\big)\Big\}$ from the OM model. Notice lastly that the ratio of the sampled points from the total points of the OM model is $^{360}/_{4320} \approx 8.35\%$. Which definitely helped the speed of the $\mathcal{GP}$ model as we will show below.

Figure 6.11a shows the sampling scheme for the layout in Figure 6.9c, while Figure 6.11b shows the emulated results of Figure 6.9d. The results are indeed very close with mean error when comparing the expected value $\mathbb{E}(\tilde{P}_*)$ for for every input $(v,\theta)_*$ to the power production calculated with the OM model is less than 2.5%. This is natural because the emulation is not perfect and the two surfaces are not entirely identical. **However**, and this is very interesting, if one calculates the product of the graph in Figure 6.11b with the joint pmf in Figure 6.9b in order to find the *probability-weighted* power production for every wind speed and wind direction, as was done for the OM model, then one gets the result in the form of Figures 6.11d and 6.11c. The reader has only to compare Figures 6.11d and 6.11c with the equivalent ones from the OM model – Figures 6.9f and 6.9e – to convince himself that the *probability-weighted* power production of the $\mathcal{GP}$ model is accurate enough. The total error in expected power production of a given layout for all wind speeds and wind directions never surpasses 500 kW, so for 30 turbines of rated power 2000 kW, this is 0.83% of error in prediction.

This section was devoted in discussing the analysis on how we developed tested and validated the $\mathcal{GP}$ model having in mind the results of the OM model. Already we have mentioned the produced error in the surrogate model, but a more general and full comparison follows in the last section of this chapter where we will also talk about the BGA again. In the next section, our discussion will revolve around the development of the MC model and how to acquire good samples in order to find the result of Equation 6.4.

6.6. The Monte Carlo Model

In order to find the result of Equation 6.4 one has to determine how many samples $k \in \mathbb{N}$ one needs and how one will draw them randomly so that they follow the random outputs $\tilde{P}_s$ follow their probability mass function pmf_P as they should. Instead of drawing directly random samples of the power production, we can draw random samples from the corresponding inputs, $\tilde{v}_s$ and $\tilde{\theta}_s$, so that they follow their joint pmf $\text{pmf}_{v,\theta}$ and calculate which power production evaluations $\tilde{P}_s$ these random samples give (with $1 \leq s \leq k$). The discussion *why* those random power production outputs $\tilde{P}_s$ corresponding to $\tilde{v}_s$ and $\tilde{\theta}_s$ will *indeed* follow their pmf pmf_P was discussed in Section 4.3.

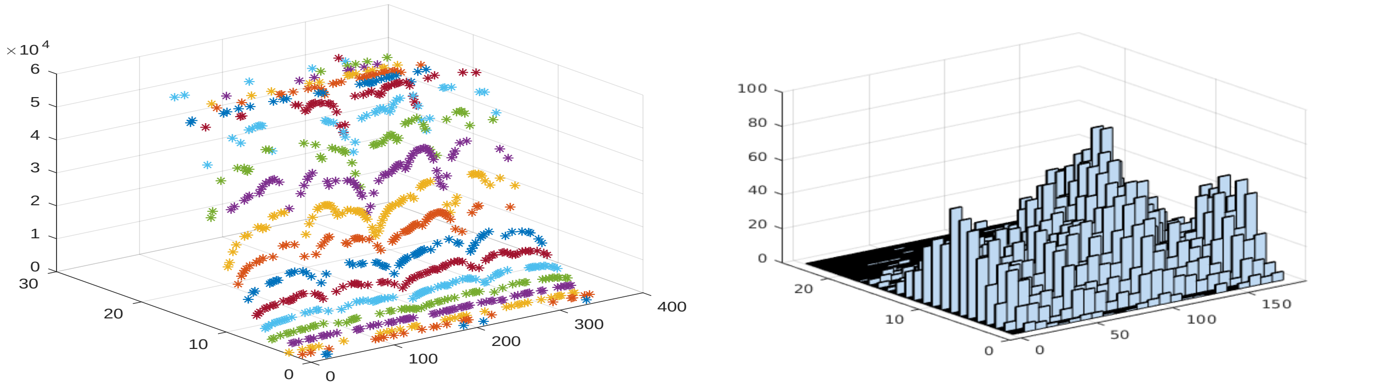
A common solution to draw random samples of wind speed $\tilde{v}_s$ and wind direction $\tilde{\theta}_s$ is to apply direct sampling based not on the joint cumulative mass function $\text{cmf}_{\tilde{v}_s,\tilde{\theta}_s}(v,\theta) = \text{cmf}_{\tilde{v},\tilde{\theta}}(v,\theta)$ directly, but on the marginal cumulative mass function $\text{cmf}_{\tilde{\theta}_s}(\theta) = \text{cmf}_{\tilde{\theta}}(\theta)$ and the conditional cumulative mass functions $\text{cmf}_{\tilde{v}_s}(\tilde{v}_s = v|\tilde{\theta}_s = \theta) = \text{cmf}_{\tilde{v}}(\tilde{v} = v|\tilde{\theta} = \theta)$. The MATLAB inbuilt function to generate random samples from a univariate random variable makes it easy to apply direct sampling after the cmf's are computed by modelling the corresponding generalized inverse mass functions (refer to 4.18). Afterwards the $\mathcal{MC}$ model continues to calculate with the use of the OM model the corresponding random outputs $\tilde{P}_s$ and calculates their average to estimate the expected value of the power production of the wind layout.

A question then arises, how many combinations of random samples of wind speed and wind direction should the modeller draw? Theoretically there is no upper limit of course and as it was denoted in Chapter 4 the standard deviation around the sample mean (so the error essentially) becomes smaller for larger sample sizes drawn. **However**, here it would be good if we restrict ourselves to sample sizes not greater than 4320. This is because as explained above, the OM model with which we compare the $\mathcal{MC}$ model with, calculates 4320 times the power production before assigning to it the joint probability of the wind speed and wind direction it was calculated upon. Therefore, if there is any hope to make the $\mathcal{MC}$ model faster than the OM model, one should avoid drawing more than 4320 samples because it would have to calculate the power production more times than the OM does and increase the computational effort.

In practice however this restriction is not so strict. After various trials it was verified that the difference in the result of the $\mathcal{MC}$ model of $k = 1000$ random samples with the OM model is less than 2% (either above or below the expected power calculated with the OM model). Drawing fewer combination of v_s and θ_s samples than $k = 1000$ however would mean to compromise the approximation of the expected power production in terms of accuracy. It can be seen from Equation 4.5 in Chapter 4 that the variance of the output mean $\text{Var}(\tilde{\bar{P}}_k)$ equals to $^{\sigma^2}/_k$ where k is the number of combination of random samples of wind speed and wind direction drawn and σ^2 is the variance of the probability mass function the random outputs P_s should have been drawn from (but instead we drew random samples from the wind speed and the wind direction mass functions and calculated the power outputs indirectly). Regardless of the value of σ^2 since it is a constant it can be seen that:

$$\text{Var}(\tilde{\bar{P}}_k) = \frac{\sigma^2}{k} \Rightarrow \text{Var}(\tilde{\bar{P}}_{k/10}) = \frac{\sigma^2}{k/10} = 10\frac{\sigma^2}{k} = 10\text{Var}(\tilde{\bar{P}}_k), \tag{6.5}$$

which means that if we draw the $^1/_{10}$ of the samples we did the sample variance gets 10 times bigger so we lose accuracy. In this sense, one can compute that in order for the $\mathcal{MC}$ model to sample as many samples (360) as the $\mathcal{GP}$ does and compete with it on this basis, the variance would grow approximately 2.78 times in comparison to the 1000 samples basis. Although this may not seem a lot with what we have written so far ($2.78 \times 2\% = 5.55\%$), it actually is in terms of the BGA performance of the $\mathcal{MC}$ model as we will see in the next section.



(a) 1000 random outputs $\tilde{P}_s$ corresponding to 1000 combinations of randomly sampled data $\tilde{v} = v_s, \tilde{\theta}_s = \theta_s$

(b) A histogram based on 1000 draws of random samples $\tilde{v}_s$ and $\tilde{\theta}_s$. Notice the bins here are 1 m/s for the wind speed and 2° for the wind direction.

Figure 6.12: Various graphs of interest for the layout in Figure 6.9c **calculated with the MC model**

To show some comparison, the $\mathcal{MC}$ model for the layout of Figure 6.9c yielded these random outputs $\tilde{P}_s$ as seen in Figure 6.12a which of course are ***indicative***, since they rely on random samples $\tilde{v}_s$ and $\tilde{\theta}_s$. However, it can be seen that they follow the general form of the power production for every wind speed and wind direction as in Figure 6.9d. In the same fashion, and to solidify our argument, the histogram of the random samples $\tilde{v}_s$ and $\tilde{\theta}_s$ drawn from direct sampling resembles the joint mass function of Figure 6.9b.

6.7. Comparison of Results

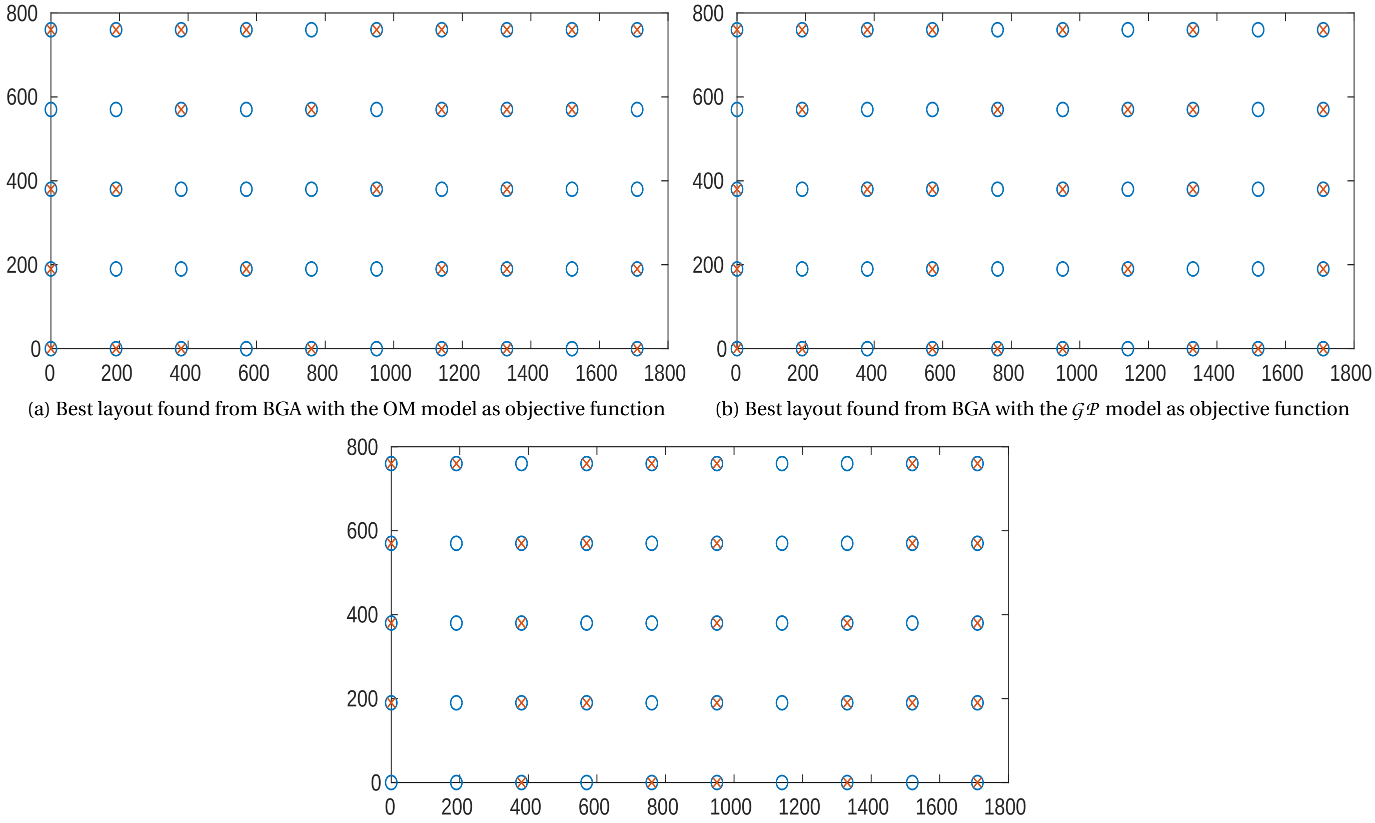
The only results we have mentioned in the previous sections have to do with the accuracy of the $\mathcal{GP}$ and the $\mathcal{MC}$ model. First, we estimated the $\mathcal{GP}$ model (in comparison to the OM model) to show maximum error in predicting the expected power production as 0.83%. Secondly, we estimated the error of the expected power production of the $\mathcal{MC}$ model for 1000 random samples to be less than 2%. These results seem good so we can now proceed with the coupling of the three models (OM, $\mathcal{GP}$ and $\mathcal{MC}$) with the BGA algorithm.

As has been stated in the previous chapters and sections, these models can be placed as the objective function of the BGA, because they all estimate the expected power of a certain layout. Essentially, the BGA algorithm has as its objective function one the Equations 6.2, 6.3 and 6.4. The setup of the comparison is as follows:

- The parameters of the BGA are the ones shown in Table 6.1
- The meteorological data are the ones presented in Section 6.4
- All layouts take place in a grid of 50 possible positions with minimum distance between the turbines 190 m, like the layouts in Figure 6.9c and 6.10a
- The layouts with more or less than 30 turbines are prohibited. In order to achieve this we force the BGA to check all chromosomes preemptively and if the sum of the genes marked with one is more other than 30, then the objective function rejects this layout by equating its expected power production to zero
- There are two approaches, both to be examined, the one after the other.:
  - one of **fixed time and varying performance**, where there is a time limit of 72 minutes for the three models but their performance may vary
  - one of **fixed performance and varying time** where there is no time limit but all models are timed until they reach the lowest performance, which was marked in the previous approach.
- In order to compare the three models (OM, $\mathcal{GP}$ and $\mathcal{MC}$) with the BGA algorithm fairly, one has to make sure that the *initial population* of chromosomes (i.e. layouts) is common for all three models. One way to do this is to initiate the random seed of the random number generating function in MATLAB always from the same number, so that when the pseudorandom chromosomes of the first population are created, essentially they will be the same for all three models.

The result of the BGA for all models is an optimized layout with an increased expected power production compared to the expected power production of all past layouts in all generations and populations. Of course, no one can assume that this is the "best" layout, because there might exist a better layout if the BGA does not converge to a global optimum. In Chapter 5 we explained that an optimization algorithm does not always reach a global extremal point like e.g. the exhaustive search method does. However, we can be sure that it produces as an outcome a *better* layout than every other examined in terms of expected power production.

We are ready to present the results. Detailed discussion follows after *all* plots and results are presented. The three layouts in Figures 6.13a, 6.13b and 6.13c are the results of the BGA with objective functions the OM, the $\mathcal{GP}$ and the $\mathcal{MC}$ model respectively.



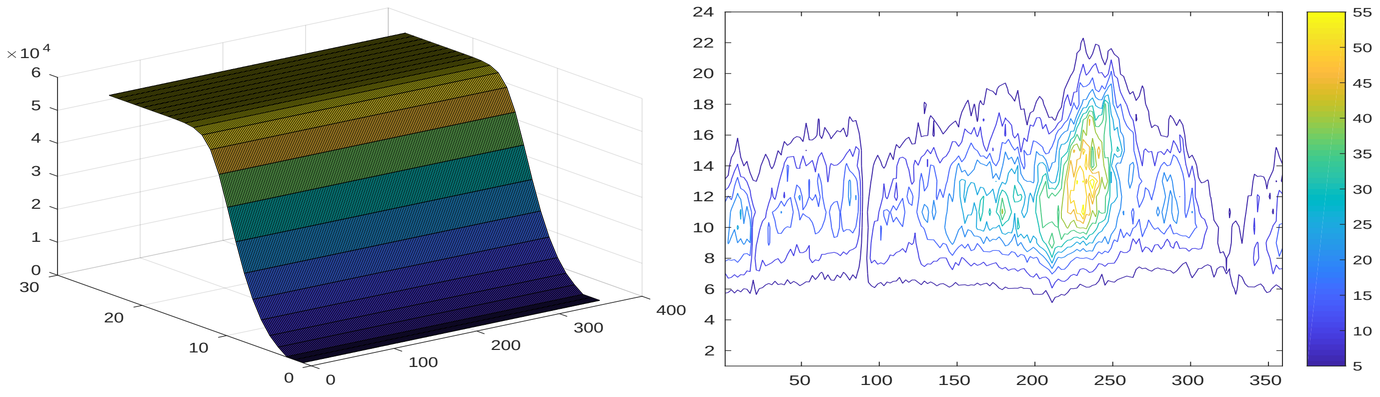
(a) Best layout found from BGA with the OM model as objective function

(b) Best layout found from BGA with the $\mathcal{GP}$ model as objective function

(c) Best layout found from BGA with the $\mathcal{MC}$ model as objective function

Figure 6.13: Best layouts found from the BGA when coupled with each of the three models

In Figure 6.14a one sees the power production for all wind speeds and wind directions if no wake is produced, while in In Figure 6.14b one sees the *probability-weighted* power production for all wind speeds and wind directions if no wake is produced. Notice that this case is the ***ideal case***, with which we compare the **performance** of every layout tested.



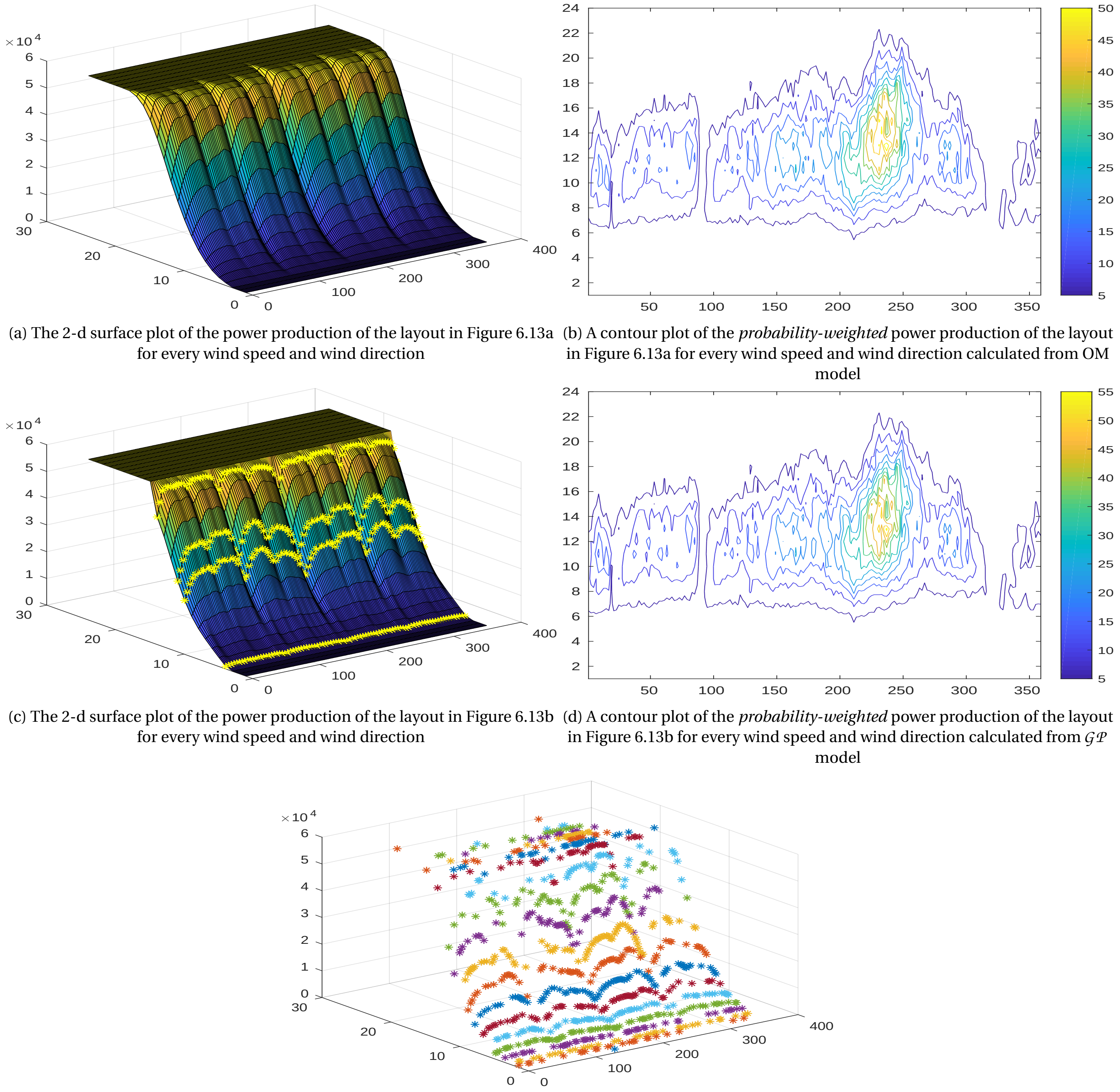
(a) A 2-d surface plot of the power production for all wind speeds and wind directions if no wake is produced

(b) A contour plot of the *probability-weighted* power production for all wind speeds and wind directions if no wake is produced

Figure 6.14: Power production graphs for the ideal case of no-wake generation

In Figures 6.15a, 6.15c and 6.15e one sees the power production for all wind speeds and wind directions of the optimized layouts, while in Figures 6.15b and 6.15d the *probability-weighted* power production for all wind speeds and wind directions is shown, only for the layouts found by OM and $\mathcal{GP}$.

The fact that all the other plots of Figures 6.15a-6.15e are similar with all relevant power production figures that have been shown in the previous section has to do with the fact that the ratio of installed turbines to possible positions in this layout is 30/50. This ratio is too close to one, which means that, in general, there will be many wake interactions in such a wind farm and most likely changing the position of individual turbines to some other position may not easily avoid any power losses induced. **Such a ratio however, was chosen because then the task of the BGA becomes more challenging to maximize the objective function result**. On the other hand, a low ratio such as in Figures 6.3a-6.6c (except the power curve in Figure 6.4) such as 6/18 means that there is a lot of potential for every turbine installed to avoid wake interactions or at least increase the spacing between other turbines (and subsequently reduce power losses). Such a ratio is good in order to observe particular "patterns" that may arise for indicative wind directions and wind speeds, exactly as we did in Section 6.3. We will come back to this point as it is important and explains also the numerical results found in our comparison analysis.



(a) The 2-d surface plot of the power production of the layout in Figure 6.13a for every wind speed and wind direction

(b) A contour plot of the *probability-weighted* power production of the layout in Figure 6.13a for every wind speed and wind direction calculated from OM model

(c) The 2-d surface plot of the power production of the layout in Figure 6.13b for every wind speed and wind direction

(d) A contour plot of the *probability-weighted* power production of the layout in Figure 6.13b for every wind speed and wind direction calculated from $\mathcal{GP}$ model

(e) 1000 random outputs $\tilde{P}_s$ corresponding to 1000 combinations of randomly sampled data $\tilde{v} = v_s, \tilde{\theta}_s = \theta_s$

Figure 6.15: Various graphs of interest for the best layouts found from the BGA coupled with the three models

As far as the in Table 6.2 one sees various numerical results of the optimized layouts of which the discussion follows below. We are ready to proceed to the main discussion and compare the models. By the best layouts in Figures 6.13a, 6.13b and

Table 6.2: Results of the Binary Genetic Algorithm for every model

Best layout	FTVP[1]			Calculated by (in MW)			Error[6] (%)		Performance[7] (%)			FPVT[8]	
	Ge.[2]	Ti.[3]		OM	$\mathcal{GP}$	$\mathcal{MC}$	$\mathcal{GP}$	$\mathcal{MC}$	OM	$\mathcal{GP}$	$\mathcal{MC}$	Ge.	Ti.
BGA+OM	10	72	$\mathbb{E}(\tilde{P})$[4]	24.41	24.53	24.59	0.463	0.70	83.48	83.86	84.06	10	72
			AP[5]	36.41	36.5	-	0.24	-	93.23	93.45	-		
BGA+$\mathcal{GP}$	43	72	$\mathbb{E}(\tilde{P})$	24.74	24.75	24.61	0.04	0.53	84.59	84.63	84.14	1	2
			AP	36.56	36.6	-	-	0.11	93.61	93.71	-		
BGA+$\mathcal{MC}$	33	72	$\mathbb{E}(\tilde{P})$	24.43	24.33	24.19	0.40	0.97	83.52	83.19	82.71	1	7
			AP	36.35	36.33	-	0.05	-	93.07	93.03	-		

(1): Fixed Time-Varying Performance: The time limit was set to 72 minutes.
(2): Generations reached until termination.
(3): Time duration (in minutes) until termination
(4): Expected power production as derived in Equations 6.2, 6.3 and 6.4.
(5): Average power production of all wind speeds and wind directions *without considering the probability weights*. This was done in order to verify that the error produced is not dependent on the joint probability mass function of wind speed and wind direction. Naturally, for the $\mathcal{MC}$ model the average power production is calculated not for all wind speeds and wind directions, but only for the *sampled* ones, that is, all of its randomly sampled combination (v_s, θ_s).
(6): The error is the error of *prediction* of $\mathbb{E}(\tilde{P})$ and AP for the final layout of that row compared to the results of the OM model and **not the error** of the performance of the model.
(7): By performance, we denote how optimal this layout is in comparison to the case of Figure 6.14 where no wake is produced. There are two types of performance that can be defined: performance in $\mathbb{E}(\tilde{P})$ and performance in AP. As far as the step of FPVT is concerned we specify performance **only** as performance in $\mathbb{E}(\tilde{P})$! We exhibit performance in AP just to offer a broader view of the results. The $\mathbb{E}(\tilde{P})$ and AP of the ideal case is 29249 kW and 39055 kW.
(8): Fixed Performance - Varying Time: The criterion of terminating the optimization procedure was passing the lowest $\mathbb{E}(\tilde{P})$ found in the FTVP approach, which was the one of the BGA+OM combination (24416 kW).

6.13c, show that there is more similarity between the layouts indicated with the BGA by the OM and the $\mathcal{GP}$ models, while the $\mathcal{MC}$ model seems to be more different than the other two. The most probable wind directions from the data in Figure 6.7 refer to wind coming from West-Southwest-South. It is not easy in layouts such as those in Figures 6.13a, 6.13b and 6.13c to tell easily if they are adequate for such winds due to the fact that there are no visible “patterns” of layout such in the case of the layouts used for verification of the BGA algorithm in Section 6.3. However, this is the reason we verified the function of the BGA algorithm in the first place – to build trust that its results are reliable.

We will start commenting the results of the fixed time and varying performance approach (FTVP). In Table 6.2 one sees that the layout (in terms of $\mathbb{E}(\tilde{P})$ and AP as defined in Table 6.2) of BGA+$\mathcal{GP}$ (Figure 6.13b) is better than the other two optimized layouts found from BGA+OM (Figure 6.13a) and BGA+$\mathcal{MC}$ (Figure 6.13c). This is because all algorithms completed their search on the same amount of time (72 minutes – none of them stalled before reaching the set maximum time limit) and the BGA+$\mathcal{GP}$ combination managed to search through more generations than any other combination. This means that this combination is much faster in examining the search space so it tends to test more layouts than the combinations BGA+OM and BGA+$\mathcal{MC}$. The second best layout is the layout of the BGA+$\mathcal{MC}$ combination apparently for the same reason.

As seen in the middle of Table 6.2, for all layouts in Figures 6.13a, 6.13b and 6.13c, there was an extra calculation step in order to assess their corresponding $\mathbb{E}(\tilde{P})$ and AP with all of the three models in order to see what the error involved in the $\mathcal{GP}$ and $\mathcal{MC}$ models is. It is essential to see how well the $\mathcal{GP}$ and the $\mathcal{MC}$ model can find what the average power production AP of a layout is *before* weighing the results with the joint probability distribution of the wind speeds and wind directions. The errors found both for $\mathbb{E}(\tilde{P})$ and AP do not contradict what was denoted in the beginning of this section for 0.83% maximum error in $\mathbb{E}(\tilde{P})$ for the $\mathcal{GP}$ model and less than 2% error in $\mathbb{E}(\tilde{P})$ for the $\mathcal{MC}$ model. This error concerns the accuracy of the two models in measuring $\mathbb{E}(\tilde{P})$ and AP, which seems to be small.

However, the $\mathcal{MC}$ model differs from the $\mathcal{GP}$ model in the sense that the first uses *random sampling* while the latter uses a *standardized sampled scheme* in order to populate the data set of sampled points D with elements. We will return to the $\mathcal{GP}$ model shortly, but let us first analyze the $\mathcal{MC}$ model. Random sampling for the $\mathcal{MC}$ model means the sample mean does not always obtain the same value every time one evaluates the randomly drawn samples (compare with Figure 4.1). In this sense, if one chooses a layout and calculates its expected power with the $\mathcal{MC}$ model two times or more, it is more than likely that he will obtain different results. This is because with the combination BGA+$\mathcal{MC}$, for a comparison of two layouts L1 and L2, if one draws for every layout $k \in \mathbb{N}$ random samples of combinations of wind speed and wind direction that yield k random outputs $\{P_{s_{\text{L1}}}\}_{s=1}^{k}$ and $\{P_{s_{\text{L2}}}\}_{s=1}^{k}$, where $P_s \sim \text{pmf}_{\tilde{P}}$ for all $1 \leq s \leq k$ the relationship between them can be:

$$\sum_{s=1}^{k} \frac{P_{s_{\text{L1}}}}{k} > \sum_{s=1}^{k} \frac{P_{s_{\text{L2}}}}{k} \qquad \text{or} \qquad \sum_{s=1}^{k} \frac{P_{s_{\text{L1}}}}{k} < \sum_{s=1}^{k} \frac{P_{s_{\text{L2}}}}{k} \qquad \text{or} \qquad \sum_{s=1}^{k} \frac{P_{s_{\text{L1}}}}{k} = \sum_{s=1}^{k} \frac{P_{s_{\text{L2}}}}{k} \tag{6.6}$$

This means that for the combination BGA+$\mathcal{MC}$ and the two layouts as above, the approximation of the expected power production (compare to Equation 6.4) may yield $\mathbb{E}(\tilde{P})_{L1} > \mathbb{E}(\tilde{P})_{L2}$ or $\mathbb{E}(\tilde{P})_{L1} < \mathbb{E}(\tilde{P})_{L2}$ or even $\mathbb{E}(\tilde{P})_{L1} = \mathbb{E}(\tilde{P})_{L2}$, depending on whether the the mean of the random outputs $\sum_{s=1}^{k} \frac{\tilde{P}_s}{k}$ of every layout underestimates, overestimates or predicts correctly the expected power production of every layout. This means that the results of the combination BGA+$\mathcal{MC}$ appearing in Table 6.2 are indicative and may change if the modeler begins the search again. That is, the combination BGA+$\mathcal{MC}$ may indicate a different optimized layout as the result! Since the most optimal layout is indicative for the BGA+$\mathcal{MC}$ combination, one may understand that decreasing the number k of combinations of random samples from the wind speed and wind direction, will increase the standard deviation around the expected power production of a layout and therefore the probability of underestimating or overestimating it. It is for this why we referred to the previous section about why one should not decrease without careful consideration the number of random samples drawn for the $\mathcal{MC}$ model.

On the other hand, one has the $\mathcal{GP}$ model. Although there might as very well be noise included in this model, the calculations of this model involve expressions of the form of the Equations 3.18 and 3.19 in Chapter 3. These expressions, however, involve expected values and variances of random vectors. As is well known however, these statistical measures are not random! Of course, they are dependent on the data set of the sampled points D, but as explained in the previous section, the sampling scheme of our choice (Figure 6.11a) is fixed. Therefore every time the combination BGA+$\mathcal{GP}$ compares two layouts as above, L1 and L2, it will yield always the same comparison between them, and so this combination, if not terminated prematurely, yields the same optimized layout as in 6.13b.

So, irrespective of the fact that these two models have comparable errors, **the combination BGA+MC is unreliable to indicate the best layout**, since its result is random.

As far as performance ($\mathbb{E}(\tilde{P})$) is concerned, the combination BGA+$\mathcal{GP}$ is the best while the combination BGA+$\mathcal{MC}$ is the worst. Comparing with the performance calculated by the OM model, it is good to see that there is no bias of over- or underestimating the performance of various layouts by the $\mathcal{GP}$ model since, as seen in Table 6.2, for the three found layouts. This has to do with how well the model can perform regression given the power output values of the data set D. We already know from e.g. Figure 3.5 that the confidence intervals indicate that the true output values may be equal, less or more than the expected values of the output vector predicted by the $\mathcal{GP}$ model. All models perform similarly when compared with the ideal case of Figure 6.14. This has to do with the discussion we had above, about the fact that the big ratio 30/50 of the layout plays an important role. There seems to be limited space for the turbines to avoid wake effects and realign themselves to avoid the main directions of wind. This is also the reason why after so many generations (compare for example 10 generations of BGA+OM with 43 generations of BGA+$\mathcal{GP}$) there seems to be little progress in optimizing the layout further.

A second fixed performance and varying time (FPVT) approach was tested with a lower bound of expected power $\mathbb{E}(\tilde{P})$ to be equal to the expected power of the layout found from the BGA+OM combination, the lowest found in general. There are fewer generations here even for the BGA+$\mathcal{GP}$ and BGA+$\mathcal{MC}$ combinations, since the BGA is programmed to terminate the search as soon as there is a layout found that has expected power greater than the lowest bound, so it does not continue to test any other layouts. It can be seen from Table 6.2 that the speed of $\mathcal{GP}$ and $\mathcal{MC}$ models to encounter such a better layout is indeed very good, since the first one needs only 2 and the second one only 7 minutes.

All in all, the comparison between the three models showed that the $\mathcal{GP}$ model seemed to be a potent competitor for the OM model in order to get as fast as possible a reliable layout that is optimized for the grid that it is destined to be installed in. When combined with BGA, the $\mathcal{GP}$ model is essentially 330% more productive than the OM model. The $\mathcal{MC}$ model, although very fast in making computations, cannot be coupled with the BGA to show reliable results as does the $\mathcal{GP}$ model, so it must be avoided. On top of that, the $\mathcal{GP}$ model is faster than the the $\mathcal{MC}$ model.

6.8. Modelling with the GPstuff Toolbox

This section is a general discussion around the GPstuff toolbox analyzing the modeling experience of the author while attempting to emulate the results of the OM model with the $\mathcal{GP}$ model using the GPstuff toolbox.

First, let us refer to the hyperparameter optimization. We used the hyperparameter optimization routine provided by the GPstuff Toolbox which performed adequately well, although we observed some failures when preparing the $\mathcal{GP}$ model to be used by the BGA algorithm. The initiation of the hyperparameter set was:

- the length scale vector $\boldsymbol{l}$ (notice here it is a vector, because there are two directions in play) was set to 1 in both directions (first axis: wind speeds, second axis: wind directions)

- the standard deviation of the stochastic process and the noise standard deviation were, on purpose, set very small equal to $\sigma_k = \sigma_n = 0.04$. The reason for this was to verify that the hyperparameter optimizer could propose a) the value of $\sigma_{k_{\text{MAP}}}$ high enough to be around the power outputs (this is because we considered the expected value vector of the $\mathcal{GP}$ model to be equal to zero and b) the value of $\sigma_{n_{\text{MAP}}}$ low enough so that added noise assumed is reduced as much as possible.

The set of hyperparameters for Maximum A Posteriori (MAP) estimation of a typical layout (the one in Figure 6.10a) was $\boldsymbol{\gamma}_{\text{M}\tilde{\text{A}}\text{P}} = [\sigma_{k_{\text{MAP}}}\ \sigma_{n_{\text{MAP}}}\ \boldsymbol{l}_{\text{MAP}}]^\top = [19672\ 0.921\ [4.85\ 6.78]^\top]^\top$. Of course, for every layout these numbers change but because all power production 2-d surface plots are similar, it was observed that they actually do not vary that much. These values were deemed to be reasonable in the sense that:

- The noise standard deviation $\sigma_{n_{\text{MAP}}}$ was found small as desired

- the length scale vector seems reasonable in the sense that the wind speed element has value 4.85 which can be justified if one notices that the sampling scheme is for wind speeds 4,10,11 and 14 m/s. Because it is these wind values that have effect in the rest of the rest of the wind speeds (e.g. 12 m/s), the rest of the wind speeds has to be inside the “area of effect". One may notice that (excluding 11 m/s) their difference of the rest is 14 − 10 = 4 m/s and 10 − 4 = 6 m/s which are both close to 4.85 m/s. The wind direction element has a value of 6.78 which is not too big since the range of wind directions is $(0°, 360°)$ but experience has proven that in practice changes in 5° of wind direction are negligible to the power production. Of course the length scales do not examine changes in the input values but "areas of effect", however, from the vertical "dips" in the 2-d surface plots of power production, such a small area of effect appear reasonable.

- The standard deviation of the stochastic process $\sigma_{k_{\text{MAP}}}$ was found to be reasonable in the sense that the maximum point-wise standard deviation $\sigma_{\bar{P}}$ in the layout after Gaussian Process Regression has taken place was found to be equal to 2174, which means that indeed the standard deviation reduces as it should, as can be seen in Equation 3.19 and the general discussion in Section 3.4 about prior and posterior covariance. The discussion there is for covariances but due to the indirect link between covariance and variance, this discussion eventually extends to the standard deviation. The only problematic part is that 10% of the power outputs of the $\mathcal{GP}$ model surpass the $2\sigma_{k_{\text{MAP}}} = 39344$ which (refer to Figure 3.1) should have been only 5%. ***This may indicate that the hyperparameter optimization procedure may not be perfect due to the fact that the expected value vector is considered to be zero.*** Unfortunately, whether this is indeed an indication of inability of the hyperparameter optimizer to converge to a proper maximum a posteriori standard deviation of the stochastic process $\sigma_{k_{\text{MAP}}}$ leads the investigation too far, so we will not proceed with it.

A last remark for the GPstuff toolbox was that although there is a good variety of covariance functions, for the 2-d case, they all perform regression so that they always yield smooth curves. This made our investigation turn to the power production surface plots and not to the *probability-weighted* power production graphs to work with, and examine specifically the squared exponential covariance function. It would be interesting if it there was a way to use reliably covariance functions for non-one-dimensional problems.

In general, due to the fact that the results of the $\mathcal{GP}$ model were good in terms of predicting the power outputs for all wind speed and wind direction with small error involved, the results the GPstuff Toolbox offered were trusted. All in all however, there is probably no other choice in the variety of toolboxes for implementing 2-d Gaussian Process Regression for the MATLAB environment, since we have searched for an alternative option to GPstuff Toolbox found none.

7 Conclusions and Recommendations

7.1. Conclusions

In this thesis, the use of Gaussian Process Regression technique was examined in order to see whether it can be a useful tool for substituting a high-fidelity model in the field of wind farm layout optimization. Three models were examined with two of them being developed by the author and the third one, implemented by Bo Hu [27] and retrieved from the TU repository, was modified by the author in order to conform to the needs of this thesis project. The results validated the initial argument that the use of Gaussian Process Regression is a reliable technique that can be useful for wind layout optimization problems. Of course, the proper mathematical theory supporting this argument was also provided in previous chapters.

The use of Gaussian Process Regression was found to be both reliable and computationally cheaper in comparison to the high-fidelity model dedicated to calculating the expected power production for a wind farm layout. Between the two models, it was found that the surrogate model was able to test 330% more wind farm layouts before the optimizer gave the final optimized layout. This meant that it ended up producing a better layout than the high-fidelity model. The $\mathcal{MC}$ model was found to be also very fast in processing a big number of candidate layouts but its contribution had to be discarded, due to the fact that as far as evaluating the expected power output of any two wind farm layouts and comparing them, its comparison of their expected power output is random and therefore arbitrary, so the results are not reliable to report and need further investigation.

The Binary Genetic Algorithm that was developed in order to be used for the layout optimization was verified and it did provide better layouts for a variety of cases. However, because the search space of a Binary Genetic Algorithm is discrete and finite, the turbines were restricted in being reallocated to specific possible positions which meant also that the layouts found by the optimizer might not be the best in reality. This does not mean that the results of the Binary Genetic Algorithm are not useful. It just means that in order for someone to use the algorithm, he should first select within a future wind farm installation area specific possible positions to install turbines. This restriction to choosing specific possible positions for the turbines was done deliberately in order to avoid specific computational problems encountered by the author. If one chooses to allow all possible geographic locations inside a future wind farm installation area, one needs to explore a continuous and not a discrete search space so the search would in principle be more time-consuming. On its own, this is not a prohibiting fact, but what is prohibiting is the fact that no wind turbine has to be inside the near-wake region, for all directions, due to the fact that the high-fidelity model does not compute any power output in such a case. Since in that case the search space would be continuous, searching for coordinates for *all* wind turbines to be installed such that no wind turbine is in the near-wake region of *any* other, would mean discarding a substantial amount of layouts, rendering the whole process impractical if not prohibitive in terms of computing power required.

Also, the wind turbines should never be reallocated outside the installation area. So every time a layout is tested in a continuous search space, the layout has to be discarded if any turbine causes some other turbine to be positioned in the near-wake region, for all directions, or if a turbine is outside of the perimeter of the installation area. When raising the numbers of turbines to 30 as it was done to this project, the interactions between the turbines themselves and the perimeter of the installation area are so many that the use of optimization algorithms considering a continuous search space is computationally impractical. On this basis, the Binary Genetic Algorithm offered a better solution that could be modeled so that it could always avoid near-wake placements for the turbines and all possible positions were inside the installation area.

Although bringing reliable results, the layouts found by the Binary Genetic Algorithm are too convoluted to be able to receive a visual interpretation in comparison to the meteorological data. This had to do with the high ratio of installed wind turbines to total possible positions. A high ratio was preferred on the basis that this would make the search of the Binary Genetic Algorithm more challenging. In general, it was found that increasing the aforementioned ratio, the progress of the Binary Genetic Algorithm in finding better layouts in successive generations is very slow.

7.2. Recommendations

The wind farm layout optimization problem is a problem in which many satisfying solutions can be given. Above we defended the validity of our effort since we verified the optimizing algorithm we used but also did not forget to mention its shortcoming. There are other optimizing algorithms that may be examined from other developers that may end up expanding the exploration in the search space of a wind farm site. During this project, there was also some experimentation and development of two other types of optimizing algorithms, the Continuous Genetic Algorithm and the Simulating Annealing algorithm. We would recommend the second one as it is shown to be more promising as it can avoid more easily positions of near-wake interactions or the ones outside the installation area. On the other hand, its parametrization is not a straightforward thing to justify.

The Gaussian Process Regression model that we developed performed very well and seems to be a good substitute for high-fidelity power production models that are more computationally expensive. However it should be tested if more simplified, lower-fidelity models such as the Jensen model may outperform the model which we developed here in terms of computational speed and reliability in results.

In the field of surrogate models, there are also other models of interests such as Polynomial Chaos Expansion, Neural Networks, Second Order Regression. Whatever the choice, a more challenging approach would be to model again with a Gaussian Process Regression model, but directly for the *probability-weighted* power production of all wind speeds and all wind directions. The shortcoming to this has to do with finding a proper numerical toolbox that can handle Gaussian Process Regression of non-smooth curves in a robust way.

Bibliography

[1] Andreas Antoniou. *Practical optimization : algorithms and engineering applications.* Springer, 2007.

[2] M. J. Asher, B. F. W. Croke, A. J. Jakeman, and L. J. M. Peeters. A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, 51(8):5957–5973, 2015.

[3] Asian Development Bank. Guidelines for wind resource assessment: Best practices for countries initiating wind development. Technical report, 2014.

[4] D. Barber. *Bayesian Reasoning and Machine Learning.* Cambridge University Press, 2012.

[5] Majid Bastankhah and Fernando Porté-Agel. A new analytical model for wind-turbine wakes. *Renewable Energy*, 70:116 – 123, 2014. Special issue on aerodynamics of offshore wind energy systems and wakes.

[6] Ashok D. Belegundu and Tirupathi R. Chandrupatla. *Optimization Concepts and Applications in Engineering, Second Edition.* Cambridge University Press, 2011.

[7] Tony Burton. *Wind energy handbook.* Wiley, 2011.

[8] Maurice Clerc. *Particle Swarm Optimization.* Wiley, 2006.

[9] David S. Matteson David Ruppert. *Statistics and Data Analysis for Financial Engineering: with R examples.* Springer-Verlag New York, 2015.

[10] Marc Deisenroth. Lecture notes on Gaussian processes for Big Data problems, April 2015.

[11] Luc Devroye. *Non-Uniform Random Variate Generation.* Springer, 1986.

[12] David Kristjanson Duvenaud, Miguel Hernández-Lobato, Yue Wu, Ryan Turner, and Roger Frigola. *Automatic Model Construction with Gaussian Processes.* PhD thesis, University of Cambridge, 2014.

[13] M. Ebden. Gaussian Processes: A Quick Introduction. *ArXiv e-prints*, 2015.

[14] Alireza Emami and Pirooz Noghreh. New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. *Renewable Energy*, 35(7):1559 – 1564, 2010.

[15] William Feller. *An introduction to probability theory and its application.* Wiley, 1971.

[16] Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering Design via Surrogate Modelling: A Practical Guide.* Wiley, 2008.

[17] C. Poloni G. Mosetti and B. Diviacco. Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. *Journal of Wind Engineering and Industrial Aerodynamics*, 51(1):105 – 116, 1994.

[18] James E. Gentle. *Random Number Generation and Monte Carlo Methods.* Springer, 2003.

[19] Agathe Girard. *Approximate Methods for Propagation of Uncertainty with Gaussian Process.* PhD thesis, University of Glasgow, 2004.

[20] Fred W. Glover and Manuel Laguna. *Tabu Search.* Springer, 1998.

[21] Javier Serrano Gonzalez, Manuel Burgos Payán, and Jesús Riquelme Santos. Optimization of wind farm turbine layout including decision making under risk. *IEEE Systems Journal*, 6:94–102, 2012.

[22] Byron S. Gottfried and Joel Weisman. *Introduction to optimization theory.* Prentice-Hall, 1973.

[23] S. A. Grady, M. Y. Hussaini, and M. M. Abdullah. Placement of wind turbines using genetic algorithms. *Renewable Energy*, 30(2):259 – 270, 2005.

[24] Zhong-Hua Han and Ke-Shi Zhang. *Real-World Applications of Genetic Algorithms*, chapter Surrogate-Based Optimization. Intech, 2012.

[25] Alexander Hartmann. *Optimization algorithms in physics.* Wiley-VCH, 2002.

[26] F. Herrera, M. Lozano, and J.L. Verdegay. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review*, 12(4):265–319, Aug 1998.

[27] Bo Hu. Design of a simple wake model for the wind farm layout optimization considering the wake meandering effect. Master's thesis, Delft University of Technology, 2016.

[28] C. T. Kelley. *Iterative methods for optimization*. SIAM, 1999.

[29] Imai Kosuke. Lecture notes on quantitative analysis, March 2006.

[30] Kenneth Lange. *Optimization*. Springer, 2013.

[31] Richard J. Larsen and Morris L. Marx. *An Introduction to Mathematical Statistics and Its Applications*. Pearson, 2011.

[32] David J. C. MacKay. Information theory, inference, and learning algorithms. *IEEE Transactions on Information Theory*, 50:2544–2545, 2003.

[33] Jeff Miller. Lecture notes on machine learning, June 2011.

[34] Zabaras Nikolaos. Lecture notes in Bayesian scientific computing, March 2013.

[35] OWEZ. OWEZ reports and data online repository. `http://www.noordzeewind.nl/en/knowledge/reportsdata/`. Accessed: 2017-10-31.

[36] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Europe, 2002.

[37] Nestor V. Queipo, Raphael T. Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P. Kevin Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1 – 28, 2005.

[38] Carl E. Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning. In *Adaptive computation and machine learning*, 2009.

[39] Sheldon M. Ross. *Probability Models for Computer Science*. Harcourt Academic Press, 2001.

[40] Sheldon M. Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. Elsevier Academic Press, 2004.

[41] Sheldon M Ross. *Simulation*. Elsevier Academic Press, 2006.

[42] Sheldon M. Ross. *Introduction to probability models*. Elsevier Academic Press, 2007.

[43] Sheldon M. Ross. *A First Course in Probability*. Prentice Hall, 2009.

[44] Sheldon M. Ross and Erol A Pekoz. *A Second Course in Probability*. www.ProbabilityBookstore.com, 2007.

[45] Michele Samorani. The wind farm layout optimization problem. 2011.

[46] Michele Samorani. *The Wind Farm Layout Optimization Problem*, pages 21–38. Springer Berlin Heidelberg, 2013.

[47] Thomas B. Schon and Fredrik Lindsten. Manipulating the multivariate Gaussian density. 2011.

[48] Matthias W. Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14 2:69–106, 2004.

[49] S. N. Sivanandam. *Introduction to genetic algorithms*. Springer, 2007.

[50] Ralph C. Smith. Uncertainty quantification - theory, implementation, and applications. In *Computational science and engineering*, 2014.

[51] Ilya M. Sobol. *A primer for the Monte Carlo method*. CRC Press, 1994.

[52] Dan Stefanica. *A Linear Algebra Primer for Financial Engineering*. FE Press, LLC, 2014.

[53] Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. Gpstuff: Bayesian modeling with Gaussian processes. *Journal of Machine Learning Research*, 14:1175–1179, 2013.

[54] Susana Vinga. Convolution integrals of normal distribution functions. 2004.

[55] C. K. I. Williams. Prediction with Gaussian processes from linear regression to linear prediction and beyond. 1997.

[56] Fuzhen Zhang. *The Schur Complement and Its Applications (Numerical Methods and Algorithms)*. Springer, 2005.

[57] Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications (Universitext)*. Springer, 2014.

A

Appendix – Mathematical Proofs

In this Appendix we demonstrate various proofs necessary for understanding deeper the main text, especially Chapters 3 and 4. The majority of these proofs can be found in the scientific literature, but so far as we know, there is no source that displays all of them readily available for the reader. The proofs are presented into different sections, for which it would be best that the reader follows them in order, one by one, before proceeding to more advanced context. The inter-dependency between the different sections is also declared at the beginning of each section.

A.1. The Schur Complement and the Woodbury Formula

Here we follow the derivation of Fuzhen Zhang in [56]. Assume the existence of an invertible block matrix as:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \tag{A.1}$$

we shall multiply it as follows:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{D}^{-1}\boldsymbol{C} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{D} \end{bmatrix} \tag{A.2}$$

assuming that $|\boldsymbol{D}| \neq 0$. Then for left multiplication:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{B}\boldsymbol{D}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D} \end{bmatrix} \tag{A.3}$$

So lastly we have:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} &= \left\{ \begin{bmatrix} \boldsymbol{I} & \boldsymbol{B}\boldsymbol{D}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{D}^{-1}\boldsymbol{C} & \boldsymbol{I} \end{bmatrix}^{-1} \right\}^{-1} \\ \Rightarrow \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} &= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{D}^{-1}\boldsymbol{C} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \left(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{B}\boldsymbol{D}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \end{aligned} \tag{A.4}$$

assuming that $|\boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}| \neq 0$. In a similar fashion we have:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{A}^{-1}\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \left(\boldsymbol{D}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}\right)^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{I} \end{bmatrix} \tag{A.5}$$

assuming that $|\boldsymbol{A}| \neq 0$ and $|\boldsymbol{D}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}| \neq 0$. The terms $\boldsymbol{\Delta}_A = \boldsymbol{D}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}$ and $\boldsymbol{\Delta}_D = \boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}$ are called the **Schur complement** of $\boldsymbol{A}$ or $\boldsymbol{D}$ respectively. By continuing the calculations we have:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1}+\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{\Delta}_A^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{\Delta}_A^{-1} \\ -\boldsymbol{\Delta}_A^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{\Delta}_A^{-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Delta}_D^{-1} & \boldsymbol{\Delta}_D^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{\Delta}_D^{-1}\boldsymbol{D}^{-1} & \boldsymbol{D}^{-1}+\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{\Delta}_D^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \end{bmatrix} \tag{A.6}$$

For a symmetric, invertible matrix $\boldsymbol{\Sigma}$ whose inverse is $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$. We have:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_{BB} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\Lambda}_{AA} & \boldsymbol{\Lambda}_{AB} \\ \boldsymbol{\Lambda}_{BA} & \boldsymbol{\Lambda}_{BB} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \boldsymbol{\Lambda}_{AA}^{-1}+\boldsymbol{\Lambda}_{AA}^{-1}\boldsymbol{\Lambda}_{AB}\boldsymbol{\Delta}_{\boldsymbol{\Lambda}_{AA}}^{-1}\boldsymbol{\Lambda}_{BA}\boldsymbol{\Lambda}_{AA}^{-1} & -\boldsymbol{\Lambda}_{AA}^{-1}\boldsymbol{\Lambda}_{AB}\boldsymbol{\Delta}_{\boldsymbol{\Lambda}_{AA}}^{-1} \\ -\boldsymbol{\Delta}_{\boldsymbol{\Lambda}_{AA}}^{-1}\boldsymbol{\Lambda}_{BA}\boldsymbol{\Lambda}_{AA}^{-1} & \boldsymbol{\Delta}_{\boldsymbol{\Lambda}_{AA}}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\Delta}_{\boldsymbol{\Lambda}_{BB}}^{-1} & \boldsymbol{\Delta}_{\boldsymbol{\Lambda}_{BB}}^{-1}\boldsymbol{\Lambda}_{AB}\boldsymbol{\Lambda}_{BB}^{-1} \\ -\boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}\boldsymbol{\Delta}_{\boldsymbol{\Lambda}_{BB}}^{-1}\boldsymbol{\Lambda}_{BB}^{-1} & \boldsymbol{\Lambda}_{BB}^{-1}+\boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}\boldsymbol{\Delta}_{\boldsymbol{\Lambda}_{BB}}^{-1}\boldsymbol{\Lambda}_{AB}\boldsymbol{\Lambda}_{BB}^{-1} \end{bmatrix} \end{aligned} \tag{A.7}$$

Therefore one finds the relationship between the submatrices of $\boldsymbol{\Sigma}$ to the Schur complements of $\boldsymbol{\Lambda}$. The above analysis of course can go the other way around for finding the relationship between the submatrices of to the Schur complements of $\boldsymbol{\Sigma}$.

Now, by comparing the elements in the first column and row of the last matrices of Equation A.6, and substituting with: $\boldsymbol{A} \rightarrow \boldsymbol{A}$, $\boldsymbol{B} \rightarrow -\boldsymbol{U}$, $\boldsymbol{C} \rightarrow \boldsymbol{V}$, $\boldsymbol{D} \rightarrow \boldsymbol{C}^{-1}$ we derive the **Woodbury formula**:

$$(\boldsymbol{A}+\boldsymbol{UCV})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1}+\boldsymbol{VA}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VA}^{-1} \tag{A.8}$$

Then by applying the $\boldsymbol{U} = \boldsymbol{V} = \boldsymbol{I}$ we derive:

$$(\boldsymbol{A}+\boldsymbol{C})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}(\boldsymbol{C}^{-1}+\boldsymbol{A}^{-1})^{-1}\boldsymbol{A}^{-1} \tag{A.9}$$

Finally, reversing the roles for $\boldsymbol{A}$ and $\boldsymbol{C}$ in the above Equation one gets:

$$(\boldsymbol{C}+\boldsymbol{A})^{-1} = \boldsymbol{C}^{-1} - \boldsymbol{C}^{-1}(\boldsymbol{A}^{-1}+\boldsymbol{C}^{-1})^{-1}\boldsymbol{C}^{-1} \tag{A.10}$$

The two Equations above yield the same result $(\boldsymbol{A}+\boldsymbol{C})^{-1}$.

A.2. Inverse Sum of Two Inverse Matrices

We notice that for two invertible matrices $\boldsymbol{A}$, $\boldsymbol{B}$ the following is valid:

$$\boldsymbol{A}^{-1}+\boldsymbol{B}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{BB}^{-1}+\boldsymbol{A}^{-1}\boldsymbol{AB}^{-1} = \boldsymbol{A}^{-1}(\boldsymbol{A}+\boldsymbol{B})\boldsymbol{B}^{-1} \Rightarrow \big(\boldsymbol{A}^{-1}+\boldsymbol{B}^{-1}\big)^{-1} = \boldsymbol{B}\big(\boldsymbol{A}+\boldsymbol{B}\big)^{-1}\boldsymbol{A}. \tag{A.11}$$

Reversing the order gives the same result so:

$$\big(\boldsymbol{A}^{-1}+\boldsymbol{B}^{-1}\big)^{-1} = \boldsymbol{A}\big(\boldsymbol{A}+\boldsymbol{B}\big)^{-1}\boldsymbol{B} = \boldsymbol{B}\big(\boldsymbol{A}+\boldsymbol{B}\big)^{-1}\boldsymbol{A} \tag{A.12}$$

A.3. Determinant Property

Assume two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The determinant of the matrix $\boldsymbol{AB}$ is:

$$|\boldsymbol{AB}| = |\boldsymbol{A}||\boldsymbol{B}| \tag{A.13}$$

and also, for every $n \in \mathbb{N}$:

$$|\boldsymbol{A}^n| = |\boldsymbol{A}|^n \tag{A.14}$$

From the Equations A.4 and A.5 we have that:

$$|\boldsymbol{M}^{-1}| = |\big(\text{left matrix} \cdot \text{middle matrix} \cdot \text{right matrix}\big)^{-1}| \Rightarrow \tag{A.15}$$

$$|\boldsymbol{M}|^{-1} = |\big(\text{left matrix} \cdot \text{middle matrix} \cdot \text{right matrix}\big)|^{-1} \Rightarrow \tag{A.16}$$

$$|\boldsymbol{M}| = |\text{left matrix}| \cdot |\text{middle matrix}| \cdot |\text{right matrix}|^{-1} \tag{A.17}$$

It can be seen that the determinants of the left and the right matrices are equal to 1 since they are upper and lower triangular matrices, so their determinants equal to the product of the diagonal elements which is in both cases 1. For the middle matrix we have in general for two submatrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ in the diagonal:

$$\det\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2 \end{bmatrix} = \det\left(\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_2 \end{bmatrix}\begin{bmatrix} \boldsymbol{I}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2 \end{bmatrix}\right) = \det\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_2 \end{bmatrix}\det\begin{bmatrix} \boldsymbol{I}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2 \end{bmatrix} = |\boldsymbol{A}_1||\boldsymbol{A}_2| \tag{A.18}$$

Therefore we have respectively that:

$$|\boldsymbol{M}| = |\boldsymbol{A}||\boldsymbol{\Delta}_{\boldsymbol{A}}| = |\boldsymbol{D}||\boldsymbol{\Delta}_{\boldsymbol{D}}| \tag{A.19}$$

A.4. Distinction Between Gaussian Distribution and Gaussian Function

It was not mentioned explicitly before, but in order not to confuse the reader on the following passages of the Appendix, the distinction between Gaussian Distribution and Gaussian Function has to be described now. A univariate Gaussian function has the form as expressed in 3.1. A univariate Gaussian distribution (or normal distribution) is **expressed** through such a function but the distinction still holds. For example, one may proceed to do unhindered algebraic manipulations between two Gaussian functions as:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} \tag{A.20}$$

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \tag{A.21}$$

even if $\mu_1 \neq \mu_2$ and $\sigma_1 \neq \sigma_2$ as is common in calculus. However, it cannot be that the two Equations represent a Gaussian distribution for a random variable $\tilde{x}$ since a variable cannot follow more than one probability distribution!

Therefore, in the following section where we should proceed with multiplying two Gaussian functions one should bear this distinction in mind. The reason we are doing this shall be apparent further below. As implied, the distinction between a Gaussian function and a Gaussian distribution is valid for the multivariate case.

A.5. Product of Two Gaussian Functions

The multivariate Gaussian function has the following equation:

$$f(x) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{A.22}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean value and the covariance matrix of the probability distribution. The expected value vector $\boldsymbol{\mu}$ has dimensions $d \times 1$ while the covariance matrix has dimensions $d \times d$ and it is symmetric. Suppose now that $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ and also $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu}$. Then because $\boldsymbol{\Lambda}$ is a symmetric matrix so $\boldsymbol{\Lambda}^\top = \boldsymbol{\Lambda}$ we obtain $\boldsymbol{\eta} = \boldsymbol{\Lambda}\boldsymbol{\mu} \Rightarrow \boldsymbol{\eta}^\top = \boldsymbol{\mu}^\top\boldsymbol{\Lambda}$ and also $\boldsymbol{\eta}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta} = \boldsymbol{\mu}^\top\boldsymbol{\Lambda}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}\boldsymbol{\mu} = \boldsymbol{\mu}^\top\boldsymbol{\Lambda}\boldsymbol{\mu}$. Then the multivariate Gaussian function can be also expressed as follows:

$$f(\boldsymbol{x}) = (2\pi)^{-n/2}|\boldsymbol{\Sigma}|^{-1/2}e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \Rightarrow \log(f(\boldsymbol{x})) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{\Sigma}|) - \frac{1}{2}\left(\boldsymbol{x}^\top\boldsymbol{\Lambda}\boldsymbol{x} - \boldsymbol{\mu}^\top\boldsymbol{\Lambda}\boldsymbol{\mu} + \boldsymbol{x}^\top\boldsymbol{\Lambda}\boldsymbol{\mu} - \boldsymbol{\mu}^\top\boldsymbol{\Lambda}\boldsymbol{x}\right) \tag{A.23}$$

Now $\boldsymbol{x}^\top\boldsymbol{\Lambda}\boldsymbol{\mu} = \boldsymbol{\mu}^\top\boldsymbol{\Lambda}\boldsymbol{x} = \left(\boldsymbol{x}^\top\boldsymbol{\Lambda}\boldsymbol{\mu}\right)^\top$ because both $\boldsymbol{x}^\top\boldsymbol{\Lambda}\boldsymbol{\mu} = \boldsymbol{x}^\top\boldsymbol{\eta}$ and $\boldsymbol{\mu}^\top\boldsymbol{\Lambda}\boldsymbol{x} = \boldsymbol{\eta}^\top\boldsymbol{x}$ are scalars we have:

$$\log(f(\boldsymbol{x})) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{\Sigma}|) - \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{\Lambda}\boldsymbol{x} + \boldsymbol{\eta}^\top\boldsymbol{x} - \frac{1}{2}\boldsymbol{\eta}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta} \tag{A.24}$$

Now we set $\boldsymbol{\zeta} = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{\Sigma}|) - \frac{1}{2}\boldsymbol{\eta}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$ and because from the properties of the determinant we know that $|\boldsymbol{A}^{-1}| = |\boldsymbol{A}|^{-1}$ we obtain $\boldsymbol{\zeta} = -\frac{d}{2}\log(2\pi) + \frac{1}{2}\log(|\boldsymbol{\Lambda}|) - \frac{1}{2}\boldsymbol{\eta}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$. Eventually we have the Equation:

$$f(\boldsymbol{x}) = \exp\left\{\boldsymbol{\zeta} + \boldsymbol{\eta}^\top\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{\Lambda}\boldsymbol{x}\right\} \tag{A.25}$$

This is the so-called **canonical form** of a Gaussian function. Now let us examine the product of two Gaussian functions $f(x)$ with $\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1$ and $g(x)$ with $\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2$ that have the same dimensionality d. We have:

- $\boldsymbol{\Lambda}_1 = \boldsymbol{\Sigma}_1^{-1} \Rightarrow \boldsymbol{\Lambda}_1^{-1} = \boldsymbol{\Sigma}_1,$ $\quad\quad$ $\boldsymbol{\Lambda}_2 = \boldsymbol{\Sigma}_2^{-1} \Rightarrow \boldsymbol{\Lambda}_2^{-1} = \boldsymbol{\Sigma}_2$
- $\boldsymbol{\eta}_1 = \boldsymbol{\Lambda}_1\boldsymbol{\mu}_1 \Rightarrow \boldsymbol{\eta}_{\mathbf{1}}^\top = \boldsymbol{\mu}_1^\top\boldsymbol{\Lambda}_1 \Rightarrow \boldsymbol{\eta}_1^\top\boldsymbol{\Lambda}_1^{-1}\boldsymbol{\eta}_1 = \boldsymbol{\mu}_1^\top\boldsymbol{\Lambda}_1\boldsymbol{\mu}_1$, $\quad\quad$ $\boldsymbol{\eta}_2 = \boldsymbol{\Lambda}_1\boldsymbol{\mu}_2 \Rightarrow \boldsymbol{\eta}_{\mathbf{2}}^\top = \boldsymbol{\mu}_2^\top\boldsymbol{\Lambda}_2 \Rightarrow \boldsymbol{\eta}_2^\top\boldsymbol{\Lambda}_2^{-1}\boldsymbol{\eta}_2 = \boldsymbol{\mu}_2^\top\boldsymbol{\Lambda}_2\boldsymbol{\mu}_2$
- $\boldsymbol{\zeta}_1 = -\frac{d}{2}\log(2\pi) + \frac{1}{2}\log(|\boldsymbol{\Lambda}_1|) - \frac{1}{2}\boldsymbol{\eta}_1^\top\boldsymbol{\Lambda}_1^{-1}\boldsymbol{\eta}_1$ $\quad\quad$ $\boldsymbol{\zeta}_2 = -\frac{d}{2}\log(2\pi) + \frac{1}{2}\log(|\boldsymbol{\Lambda}_2|) - \frac{1}{2}\boldsymbol{\eta}_2^\top\boldsymbol{\Lambda}_2^{-1}\boldsymbol{\eta}_2$

So the product of two Gaussian functions is the following:

$$f(\boldsymbol{x})g(\boldsymbol{x}) = \exp\left\{\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 + \boldsymbol{\eta}_1^\top\boldsymbol{x} + \boldsymbol{\eta}_2^\top\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{\Lambda}_1\boldsymbol{x} - -\frac{1}{2}\boldsymbol{x}^\top\boldsymbol{\Lambda}_2\boldsymbol{x}\right\} = \exp\left\{\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 + \left(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\right)^\top\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top\left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\right)\boldsymbol{x}\right\} \tag{A.26}$$

Now suppose the following scalar: $\boldsymbol{\zeta}_n = -\frac{d}{2}\log(2\pi) + \frac{1}{2}\log(|\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2|) - \frac{1}{2}\left(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\right)^\top\left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\right)^{-1}\left(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\right)$. We add it and subtract it from Equation A.26 and we get:

$$\begin{aligned} f(\boldsymbol{x})g(\boldsymbol{x}) &= \exp\left\{\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 - \boldsymbol{\zeta}_n + \boldsymbol{\zeta}_n + \left(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\right)^\top\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top\left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\right)\boldsymbol{x}\right\} \\ &= \exp\left\{\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 - \boldsymbol{\zeta}_n\right\}\exp\left\{\boldsymbol{\zeta}_n + \left(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\right)^\top\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top\left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\right)\boldsymbol{x}\right\} \end{aligned} \tag{A.27}$$

We first observe the second exponential term from Equation A.27. We have:

$$\begin{aligned} &\exp\left\{\boldsymbol{\zeta}_n + \left(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\right)^\top\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top\left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\right)\boldsymbol{x}\right\} \\ &\quad = \exp\left\{-\frac{d}{2}\log(2\pi) + \frac{1}{2}\log(|\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2|) - \frac{1}{2}\left(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\right)^\top\left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\right)^{-1}\left(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\right) + \left(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\right)^\top\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top\left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\right)\boldsymbol{x}\right\} \end{aligned} \tag{A.28}$$

By setting $\boldsymbol{\Sigma}_n = \left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\right)^{-1}$, $\boldsymbol{\Lambda}_n = \boldsymbol{\Sigma}_n^{-1}$, $\boldsymbol{v} = \boldsymbol{\eta}_1 + \boldsymbol{\eta}_2$ we have:

$$\exp\left\{-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{\Sigma}_n|) - \frac{1}{2}\boldsymbol{v}^\top\boldsymbol{\Lambda}_n^{-1}\boldsymbol{v} + \boldsymbol{v}^\top\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{\Lambda}_n\boldsymbol{x}\right\} \tag{A.29}$$

Comparing Equations A.24 and A.29 it should be apparent that the second exponential follows a Gaussian function $t(\boldsymbol{x})$ with covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{n}} = \big(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\big)^{-1} = \big(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\big)^{-1}$ and an expected value vector $\boldsymbol{\mu}_n$ that can be found from the following expression $\boldsymbol{v} = \boldsymbol{\mu}_n \boldsymbol{\Lambda}_n \Rightarrow \boldsymbol{\mu}_n = \boldsymbol{\Lambda}_n^{-1}\boldsymbol{v} = \boldsymbol{\Sigma}_{\boldsymbol{n}}\boldsymbol{v} = (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}\big(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\big)$.

As far as the first exponential goes we have the following terms for it:

$$\begin{aligned}
\exp\Big\{\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 - \boldsymbol{\zeta}_n\Big\} = \exp\Big\{&-\frac{d}{2}\log(2\pi) + \frac{1}{2}\log(|\boldsymbol{\Lambda}_1|) - \frac{1}{2}\boldsymbol{\eta}_1^\top \boldsymbol{\Lambda}_1^{-1}\boldsymbol{\eta}_1 \\
&-\frac{d}{2}\log(2\pi) + \frac{1}{2}\log(|\boldsymbol{\Lambda}_2|) - \frac{1}{2}\boldsymbol{\eta}_2^\top \boldsymbol{\Lambda}_2^{-1}\boldsymbol{\eta}_2 + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2|) + \frac{1}{2}\big(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\big)^\top\big(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\big)^{-1}\big(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\big)\Big\} \\
= \exp\Big\{&-\frac{d}{2}\log(2\pi) + \frac{1}{2}\{\log(|\boldsymbol{\Lambda}_1| + \log(|\boldsymbol{\Lambda}_2|)) - \log(|\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2|)\} \\
&-\frac{1}{2}\boldsymbol{\eta}_1^\top \boldsymbol{\Lambda}_1^{-1}\boldsymbol{\eta}_1 - \frac{1}{2}\boldsymbol{\eta}_2^\top \boldsymbol{\Lambda}_2^{-1}\boldsymbol{\eta}_2 - \frac{1}{2}\big(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\big)^\top\big(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\big)^{-1}\big(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\big)\Big\}. \qquad \text{(A.30)}
\end{aligned}$$

Let us now examine the individual terms. For the determinants we have $|\boldsymbol{AB}| = |\boldsymbol{A}||\boldsymbol{B}| = |\boldsymbol{B}||\boldsymbol{A}|$, therefore:

$$\begin{aligned}
\frac{1}{2}\{\log(|\boldsymbol{\Lambda}_1| + \log(|\boldsymbol{\Lambda}_2|)) - \log(|\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2|)\} &= \frac{1}{2}\{\log(|\boldsymbol{\Lambda}_1| - \log(|\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2|) + \log(|\boldsymbol{\Lambda}_2|))\} \\
&= \frac{1}{2}\{\log(|\boldsymbol{\Lambda}_1(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2)^{-1}\boldsymbol{\Lambda}_2|)\} = -\frac{1}{2}\{\log(|\boldsymbol{\Lambda}_2^{-1}(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2)\boldsymbol{\Lambda}_1^{-1}|)\} = -\frac{1}{2}\{\log(|\boldsymbol{\Lambda}_2^{-1} + \boldsymbol{\Lambda}_1^{-1}|)\} \\
&= -\frac{1}{2}\{\log(|\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2|)\}. \qquad \text{(A.31)}
\end{aligned}$$

As far as the last term is concerned we have:

$$\begin{aligned}
-\frac{1}{2}\boldsymbol{\eta}_1^\top \boldsymbol{\Lambda}_1^{-1}\boldsymbol{\eta}_1 - \frac{1}{2}\boldsymbol{\eta}_2^\top \boldsymbol{\Lambda}_2^{-1}\boldsymbol{\eta}_2 - \frac{1}{2}\big(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\big)^\top\big(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\big)^{-1}\big(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2\big) &= -\frac{1}{2}\Big\{\boldsymbol{\eta}_1^\top\boldsymbol{\Sigma}_1\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2^\top\boldsymbol{\Sigma}_2\boldsymbol{\eta}_2 - (\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2)^\top\boldsymbol{\Sigma}_n(\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2)\Big\} \\
&= -\frac{1}{2}\Big\{\boldsymbol{\eta}_1^\top(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_n)\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2^\top(\boldsymbol{\Sigma}_2 - \boldsymbol{\Sigma}_n)\boldsymbol{\eta}_2 - \boldsymbol{\eta}_1^\top\boldsymbol{\Sigma}_n\boldsymbol{\eta}_2 - \boldsymbol{\eta}_2^\top\boldsymbol{\Sigma}_n\boldsymbol{\eta}_1\Big\} \qquad \text{(A.32)}
\end{aligned}$$

Now from the Woodbury formulas and the inverse identities we obtain the following expressions:

$$\boldsymbol{\Sigma}_n = \big(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\big)^{-1} = \begin{cases}
\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_1\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\Sigma}_1, & \text{(A.33)} \\
\boldsymbol{\Sigma}_2 - \boldsymbol{\Sigma}_2\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\Sigma}_2, & \text{(A.34)} \\
\boldsymbol{\Sigma}_1\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\Sigma}_2, & \text{(A.35)} \\
\boldsymbol{\Sigma}_2\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\Sigma}_1. & \text{(A.36)}
\end{cases}$$

From Equations A.33 and A.34 we obtain:

$$\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}_1\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\Sigma}_1, \qquad \text{(A.37)}$$

$$\boldsymbol{\Sigma}_2 - \boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}_2\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\Sigma}_2. \qquad \text{(A.38)}$$

We also know that $\boldsymbol{\eta_1} = \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 \Rightarrow \boldsymbol{\eta}_{\boldsymbol{1}}^\top = \boldsymbol{\mu}_1^\top\boldsymbol{\Sigma}_1^{-1}$ and $\boldsymbol{\eta_2} = \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 \Rightarrow \boldsymbol{\eta}_{\boldsymbol{2}}^\top = \boldsymbol{\mu}_2^\top\boldsymbol{\Sigma}_2^{-1}$ so continuing from Equation A.32 we obtain:

$$\begin{aligned}
&-\frac{1}{2}\Big\{\boldsymbol{\eta}_1^\top(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_n)\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2^\top(\boldsymbol{\Sigma}_2 - \boldsymbol{\Sigma}_n)\boldsymbol{\eta}_2 - \boldsymbol{\eta}_1^\top\boldsymbol{\Sigma}_n\boldsymbol{\eta}_2 - \boldsymbol{\eta}_2^\top\boldsymbol{\Sigma}_n\boldsymbol{\eta}_1\Big\} \\
&= -\frac{1}{2}\Big\{\boldsymbol{\mu}_1^\top\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_1\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^\top\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_2\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^\top\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_1\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 \\
&\qquad - \boldsymbol{\mu}_2^\top\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_2\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1\Big\} \\
&= -\frac{1}{2}\Big\{\boldsymbol{\mu}_1^\top\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^\top\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^\top\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_2^\top\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\boldsymbol{\mu}_1\Big\} \\
&= -\frac{1}{2}\Big\{\boldsymbol{\mu}_1^\top\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\big(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\big) - \boldsymbol{\mu}_2^\top\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\Big\} \\
&= -\frac{1}{2}\big(\boldsymbol{\mu}_1^\top - \boldsymbol{\mu}_2^\top\big)\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\big(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\big) \qquad \text{(A.39)}
\end{aligned}$$

So eventually, Equation A.30 becomes:

$$\exp\Big\{\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 - \boldsymbol{\zeta}_n\Big\} = \exp\Big\{-\frac{d}{2}\log(2\pi) - \frac{1}{2}\{\log(|\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2|)\} - \frac{1}{2}\big(\boldsymbol{\mu}_1^\top - \boldsymbol{\mu}_2^\top\big)\big(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\big)^{-1}\big(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\big)\Big\} \qquad \text{(A.40)}$$

$$\Rightarrow c = \exp\Big\{\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2 - \boldsymbol{\zeta}_n\Big\} = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2|^{1/2}}e^{-\frac{1}{2}\left(\boldsymbol{\mu}_1^\top - \boldsymbol{\mu}_2^\top\right)\left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\right)^{-1}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)} \qquad \text{(A.41)}$$

Consequently as we see the second exponential has the form of a **Gaussian function** but since it is dependent on a constant vector $\boldsymbol{\mu}_1$ (or alternatively dependent on a constant vector $\boldsymbol{\mu}_2$) it is a constant. So we reached the result that:

$$f(\boldsymbol{x})g(\boldsymbol{x}) = c \cdot t(\boldsymbol{x}) \tag{A.42}$$

with the values calculated as above.

A.6. Integration of the Product of Two Gaussian Functions

From Equation A.42 we obtained the product of two Gaussian functions. Integrating the result we find:

$$\int f(\boldsymbol{x})g(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int c \cdot t(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = c \tag{A.43}$$

The result is this due to the fact that the integral of a probability density function (**such as a Gaussian function**) evaluated from $-\infty$ to ∞ is always equal to 1.

A.7. Sum of Two Independent Random Vectors

Here we follow explicitly the analysis of Susana Vinga shown in [54]. We shall prove how to express the sum of two independent random vectors. Assume therefore such two independent random vectors $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$ and a resulting random vector $\tilde{\boldsymbol{z}} = \tilde{\boldsymbol{x}} + \tilde{\boldsymbol{y}}$. The produced draw $\tilde{\boldsymbol{z}} \sim \mathrm{pdf}_{\tilde{\boldsymbol{z}}}(\boldsymbol{z})$ is dependent by the draws $\tilde{\boldsymbol{x}} \sim \mathrm{pdf}_{\tilde{\boldsymbol{x}}}(\boldsymbol{x})$ and $\tilde{\boldsymbol{y}} \sim \mathrm{pdf}_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})$, in the form that if $\tilde{\boldsymbol{x}} = \boldsymbol{x}$, $\tilde{\boldsymbol{y}} = \boldsymbol{y}$ and $\tilde{\boldsymbol{z}} = \boldsymbol{z}$ then $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y} \Rightarrow \boldsymbol{x} = \boldsymbol{z} - \boldsymbol{y}$. One can see that the $\tilde{\boldsymbol{x}}$ vector is parametrized by the $\tilde{\boldsymbol{y}}$ vector for a given draw $\tilde{\boldsymbol{z}} = \boldsymbol{z}$ of the $\tilde{\boldsymbol{z}}$ vector.

Let us now assume that the random vectors $\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}$ are continuous so they follow their continuous probability distributions $f_{\tilde{\boldsymbol{x}}}(\boldsymbol{x})$ and $g_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})$ and they also have cumulative probability distributions $F_{\tilde{\boldsymbol{x}}}(\boldsymbol{x})$ and $G_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})$. One now may express the probability distribution function of $\tilde{\boldsymbol{z}}$, $h_{\tilde{\boldsymbol{z}}}(\boldsymbol{z})$ as follows:

$$\begin{aligned}
h_{\tilde{\boldsymbol{z}}}(\boldsymbol{z}) &= \frac{\mathrm{d}\Big(\mathbb{P}(\tilde{\boldsymbol{z}} \le \boldsymbol{z})\Big)}{\mathrm{d}\boldsymbol{z}} = \frac{\mathrm{d}\Big(\mathbb{P}(\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{y}} \le \boldsymbol{z})\Big)}{\mathrm{d}\boldsymbol{z}} = \frac{\mathrm{d}\Big(\int \mathbb{P}(\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{y}} \le \boldsymbol{z} | \tilde{\boldsymbol{y}} = \boldsymbol{y}) g_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}\Big)}{\mathrm{d}\boldsymbol{z}} \\
&= \frac{\mathrm{d}\Big(\int \mathbb{P}(\tilde{\boldsymbol{x}} \le \boldsymbol{z} - \boldsymbol{y}) g_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}\Big)}{\mathrm{d}\boldsymbol{z}} = \frac{\mathrm{d}\Big(\int F_{\tilde{\boldsymbol{x}}}(\boldsymbol{z} - \boldsymbol{y}) g_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}\Big)}{\mathrm{d}\boldsymbol{z}} = \int \frac{\mathrm{d}\big(F_{\tilde{\boldsymbol{x}}}(\boldsymbol{z} - \boldsymbol{y})\big)}{\mathrm{d}(\boldsymbol{z} - \boldsymbol{y})} g_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y} \\
&= \int f_{\tilde{\boldsymbol{x}}}(\boldsymbol{z} - \boldsymbol{y}) g_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y} \Rightarrow h_{\tilde{\boldsymbol{z}}}(\boldsymbol{z}) = g_{\tilde{\boldsymbol{x}}} * f_{\tilde{\boldsymbol{y}}}(\boldsymbol{z})
\end{aligned} \tag{A.44}$$

This result shows that the pdf of the vector $\tilde{\boldsymbol{z}}$ is related to the convolution of the probability density functions of $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$, if the latter are independent. As we shall see below, the convolution integral is tractable for the case of multivariate Gaussian functions.

A.8. Convolution of Two Gaussian Multivariate Variables

The findings of this section follow directly from the discussion in Appendix Sections A.4 to A.7.

Here we follow along the analysis of Susana Vinga shown in [54]. For multivariate Gaussian random vectors we have $\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{x};\ \boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}) = f_{\tilde{\boldsymbol{x}}}(\boldsymbol{x})$ and $\tilde{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{y};\ \boldsymbol{\mu}_{\tilde{\boldsymbol{y}}}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}) = g_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})$. We remind the reader that as noted above $\boldsymbol{x} = \boldsymbol{z} - \boldsymbol{y}$. Now let us expand the above term $f_{\tilde{\boldsymbol{x}}}(\boldsymbol{z} - \boldsymbol{y})$:

$$f_{\tilde{\boldsymbol{x}}}(\boldsymbol{z} - \boldsymbol{y}) = f_{\tilde{\boldsymbol{x}}}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}) = \mathcal{N}(\boldsymbol{z} - \boldsymbol{y}; \boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}) \tag{A.45}$$

$$= \frac{1}{(2\pi)^{n/2}|{\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}}^{1/2}|} e^{-\frac{1}{2}\left(\boldsymbol{z}-\boldsymbol{y}-\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}\right)^{\top}\left(\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}\right)^{-1}\left(\boldsymbol{z}-\boldsymbol{y}-\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}\right)} = \frac{1}{(2\pi)^{n/2}|{\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}}^{1/2}|} e^{-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{z}+\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}\right)^{\top}\left(\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}\right)^{-1}\left(\boldsymbol{y}-\boldsymbol{z}+\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}\right)} \tag{A.46}$$

$$= \mathcal{N}(\boldsymbol{y} - \boldsymbol{z}; -\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}) \tag{A.47}$$

All these above expressions are *always* equivalent. Let us now notice that $h_{\tilde{\boldsymbol{z}}}(\cdot)$ takes values for *specific evaluations* $\boldsymbol{z}$. This means that the term $\boldsymbol{z}$ inside the integral of the Equation A.44 is a constant, a determinate vector. Therefore this leads to the following result:

$$h_{\tilde{\boldsymbol{z}}}(\boldsymbol{z}) = \int f_{\tilde{\boldsymbol{x}}}(\boldsymbol{z} - \boldsymbol{y}) g_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y} = \int \mathcal{N}(\boldsymbol{y}; \boldsymbol{z} - \boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}) \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{\tilde{\boldsymbol{y}}}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}})\mathrm{d}\boldsymbol{y} = \int k_{\tilde{\boldsymbol{y}}}(\boldsymbol{y}) g_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}, \tag{A.48}$$

where $k_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})$ is a function of $\boldsymbol{y}$ that is however not the pdf of the random vector $\tilde{\boldsymbol{y}}$; this is unique and has been determined as $g_{\tilde{\boldsymbol{y}}}(\boldsymbol{y})$. Nevertheless, it is an alternative mapping on the random vector $\tilde{\boldsymbol{y}}$ and since they both are Gaussian functions of $\boldsymbol{y}$, from the findings of the previous sections, we know how to compute the integral. The new integrand that will come out of the integral in the end will be dependent on $\boldsymbol{z}$ and this was also the desirable outcome as the term $h_{\tilde{\boldsymbol{z}}}(\boldsymbol{z})$ dictates. The result therefore will be:

$$h_{\tilde{\boldsymbol{z}}}(\boldsymbol{z}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}} + \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}|^{1/2}} e^{-\frac{1}{2}\left(\boldsymbol{z}-(\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}+\boldsymbol{\mu}_{\tilde{\boldsymbol{y}}})\right)^{\top}\left(\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}}+\boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}\right)^{-1}\left(\boldsymbol{z}-(\boldsymbol{\mu}_{\tilde{\boldsymbol{x}}}+\boldsymbol{\mu}_{\tilde{\boldsymbol{y}}})\right)}, \tag{A.49}$$

where as one can see this is a Gaussian function based on the $\boldsymbol{z}$ vector alone. Therefore it defines the Gaussian distribution of this vector and can be written as:

$$\tilde{\boldsymbol{z}} \sim h_{\tilde{\boldsymbol{z}}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z};\ \boldsymbol{\mu}_{\tilde{\boldsymbol{x}}} + \boldsymbol{\mu}_{\tilde{\boldsymbol{y}}}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}} + \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}}) \tag{A.50}$$

which gives answer to what happens when two Gaussian multivariate variables of the same dimensionality are added together.

A.9. Completing the Square for Symmetric Matrices

Assume a quadratic expression for a vector $\boldsymbol{x}$:

$$\frac{1}{2}\boldsymbol{x}^{\top}\boldsymbol{C}\boldsymbol{x} + \boldsymbol{b}^{\top}\boldsymbol{x} + a \tag{A.51}$$

Here we also assume the matrix C is symmetric. We set $\boldsymbol{m} = -\boldsymbol{C}^{-1}\boldsymbol{b}$ and $v = a - \frac{1}{2}\boldsymbol{b}^{\top}\boldsymbol{C}^{-1}\boldsymbol{b}$. Then we have:

$$\frac{1}{2}\boldsymbol{x}^{\top}\boldsymbol{C}\boldsymbol{x} + \boldsymbol{b}^{\top}\boldsymbol{x} + a = \frac{1}{2}\boldsymbol{x}^{\top}\boldsymbol{C}\boldsymbol{x} + \boldsymbol{b}^{\top}\boldsymbol{x} + v + \frac{1}{2}\boldsymbol{b}^{\top}\boldsymbol{C}^{-1}\boldsymbol{b} = \frac{1}{2}\boldsymbol{x}^{\top}\boldsymbol{C}\boldsymbol{x} + \frac{1}{2}\boldsymbol{b}^{\top}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^{\top}\boldsymbol{b} + v + \frac{1}{2}\boldsymbol{b}^{\top}\boldsymbol{C}^{-1}\boldsymbol{b} \tag{A.52}$$

$$= \frac{1}{2}\boldsymbol{x}^{\top}\boldsymbol{C}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^{\top}\boldsymbol{C}\boldsymbol{C}^{-1}\boldsymbol{b} + \frac{1}{2}\boldsymbol{b}^{\top}\boldsymbol{x} + v + \frac{1}{2}\boldsymbol{b}^{\top}\boldsymbol{C}^{-1}\boldsymbol{b} \tag{A.53}$$

$$= \frac{1}{2}\left\{\boldsymbol{x}^{\top}\boldsymbol{C}\left(\boldsymbol{x} + \boldsymbol{C}^{-1}\boldsymbol{b}\right) + \boldsymbol{b}^{\top}\left(\boldsymbol{x} + \boldsymbol{C}^{-1}\boldsymbol{b}\right)\right\} + v = \frac{1}{2}\left(\boldsymbol{x}^{\top}\boldsymbol{C} + \boldsymbol{b}^{\top}\right)\left(\boldsymbol{x} - \boldsymbol{m}\right) + v \tag{A.54}$$

We notice that $\boldsymbol{m} = -\boldsymbol{C}^{-1}\boldsymbol{b} \Rightarrow \boldsymbol{m}^{\top} = -\boldsymbol{b}^{\top}\left(\boldsymbol{C}^{-1}\right)^{\top} = -\boldsymbol{b}^{\top}\left(\boldsymbol{C}^{\top}\right)^{-1} = -\boldsymbol{b}^{\top}\boldsymbol{C}^{-1}$ because it is always valid that $\left(\boldsymbol{C}^{-1}\right)^{\top} = \left(\boldsymbol{C}^{\top}\right)^{-1}$ and $\boldsymbol{C}^{\top} = \boldsymbol{C}$ because it is symmetric. Therefore we have the final form deduced as:

$$\frac{1}{2}\left(\boldsymbol{x}^{\top}\boldsymbol{C} + \boldsymbol{b}^{\top}\right)\left(\boldsymbol{x} - \boldsymbol{m}\right) + v = \frac{1}{2}\left(\boldsymbol{x}^{\top}\boldsymbol{C} + \boldsymbol{b}^{\top}\boldsymbol{C}^{-1}\boldsymbol{C}\right)\left(\boldsymbol{x} - \boldsymbol{m}\right) + v = \frac{1}{2}\left(\boldsymbol{x}^{\top} + \boldsymbol{b}^{\top}\boldsymbol{C}^{-1}\right)\boldsymbol{C}\left(\boldsymbol{x} - \boldsymbol{m}\right) + v \tag{A.55}$$

$$= \frac{1}{2}\left(\boldsymbol{x}^{\top} - \boldsymbol{m}^{\top}\right)\boldsymbol{C}\left(\boldsymbol{x} - \boldsymbol{m}\right) + v = \frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{m}\right)^{\top}\boldsymbol{C}\left(\boldsymbol{x} - \boldsymbol{m}\right) + v \tag{A.56}$$

A.10. Marginal Probability of a Gaussian Distribution

The findings of this section follow directly from the discussion in Appendix Sections A.1 to A.3 and A.9.

Here we follow explicitly the analysis of Thomas B. Schon and Fredrik Lindsten shown in [47]. We shall now prove the Equation of the marginal probability of a multivariate Gaussian distribution. Suppose a random vector $\tilde{\boldsymbol{x}}$ with two partitions of it $\tilde{\boldsymbol{x}}_A$ and $\tilde{\boldsymbol{x}}_B$ so that the results for the expected value vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ are the following:

$$\begin{bmatrix}\tilde{\boldsymbol{x}}_A \\ \tilde{\boldsymbol{x}}_B\end{bmatrix} \qquad \begin{bmatrix}\boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B\end{bmatrix} \qquad \begin{bmatrix}\boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_{BB}\end{bmatrix} \tag{A.57}$$

For the marginal probability the following Equation is valid:

$$f_{\tilde{\boldsymbol{x}}_A}(\boldsymbol{x}_A) = \int f_{\tilde{\boldsymbol{x}}_A,\tilde{\boldsymbol{x}}_B}(\boldsymbol{x}_A, \boldsymbol{x}_B)\mathrm{d}\boldsymbol{x}_B \tag{A.58}$$

where $f_{\tilde{\boldsymbol{x}}_A,\tilde{\boldsymbol{x}}_B}(\boldsymbol{x}_A, \boldsymbol{x}_B) = f_{\tilde{\boldsymbol{x}}}(\boldsymbol{x})$ so because we assume that vector $\boldsymbol{x}$ follows a multi-variate Gaussian distribution we have:

$$f_{\tilde{\boldsymbol{x}}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{A.59}$$

Since the terms $(2\pi)^{-n/2}|\boldsymbol{\Sigma}|^{-1/2}$ are a constant they come out of the integral of Equation A.74. We proceed to analyze the context of the exponential term. Therefore we have:

$$-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Lambda}(\boldsymbol{x}-\boldsymbol{\mu}) \tag{A.60}$$

where we defined $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$. We now continue to split the matrix and the vectors according to the partition. So we have:

$$\begin{aligned}
-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\boldsymbol{x}-\boldsymbol{\mu}) &= -\frac{1}{2}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A) - \frac{1}{2}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B) \\
&\quad - \frac{1}{2}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top \boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A) - \frac{1}{2}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top \boldsymbol{\Lambda}_{BB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B) \\
&= -\Big\{\frac{1}{2}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A) + (\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B) + \frac{1}{2}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top \boldsymbol{\Lambda}_{BB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)\Big\}, \quad \text{(A.61)}
\end{aligned}$$

where above we noticed that $\Big\{(\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)\Big\}^\top = (\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top \boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)$ and $\big(\boldsymbol{\Lambda}_{AB}\big)^\top = \boldsymbol{\Lambda}_{BA}$. For Equation A.61 we shall proceed to complete the square as explained in the previous subsection. We set $\boldsymbol{C} = \boldsymbol{\Lambda}_{BB} \Rightarrow \boldsymbol{C}^{-1} = \boldsymbol{\Lambda}_{BB}^{-1}$, the vector $\boldsymbol{x}_B - \boldsymbol{\mu}_B$ acts as the $\boldsymbol{x}$ vector, $\boldsymbol{b}^\top = (\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AB} \Rightarrow \boldsymbol{b} = \boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top$ and lastly $a = \frac{1}{2}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)$. Therefore we have:

$$\boldsymbol{m} = -\boldsymbol{C}^{-1}\boldsymbol{b} = -\boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A) \qquad \text{and} \tag{A.62}$$

$$v = a - \frac{1}{2}\boldsymbol{b}^\top \boldsymbol{C}^{-1}\boldsymbol{b} = \frac{1}{2}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A) - (\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AB}\boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A) \Rightarrow \tag{A.63}$$

$$v = (\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top (\boldsymbol{\Lambda}_{AA} - \boldsymbol{\Lambda}_{AB}\boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA})(\boldsymbol{x}_A-\boldsymbol{\mu}_A) \tag{A.64}$$

$$\Rightarrow v = (\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Sigma}_{AA}^{-1}(\boldsymbol{x}_A-\boldsymbol{\mu}_A) \tag{A.65}$$

where in A.65 we used the Schur complement identity of Section A.1. For the rest of the quadratic form which we will name w, we have:

$$w = \frac{1}{2}\Big((\boldsymbol{x}_B-\boldsymbol{\mu}_B) + \boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)\Big)^\top \boldsymbol{\Lambda}_{BB}\Big((\boldsymbol{x}_B-\boldsymbol{\mu}_B) + \boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)\Big) \tag{A.66}$$

$$= \frac{1}{2}\Big(\boldsymbol{x}_B - \big(\boldsymbol{\mu}_B - \boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)\big)\Big)^\top \boldsymbol{\Lambda}_{BB}\Big(\boldsymbol{x}_B - \big(\boldsymbol{\mu}_B - \boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)\big)\Big) \tag{A.67}$$

Therefore inside the exponential there are two terms $-v$ and $-w$. Now we have as before:

$$f_{\tilde{\boldsymbol{x}}_A,\tilde{\boldsymbol{x}}_B}(\boldsymbol{x}_A,\boldsymbol{x}_B) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}}e^{(-v-w)} \Rightarrow$$

$$\int f_{\tilde{\boldsymbol{x}}_A,\tilde{\boldsymbol{x}}_B}(\boldsymbol{x}_A,\boldsymbol{x}_B)\mathrm{d}\boldsymbol{x}_B = \int \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}}e^{(-v-w)}\mathrm{d}\boldsymbol{x}_B = \int \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\big(-w\big)\mathrm{d}\boldsymbol{x}_B\exp\big(-v\big) \tag{A.68}$$

where $-v$ is not dependent on $\boldsymbol{x}_B$. We analyze the term $\int\exp\big(-w\big)\mathrm{d}\boldsymbol{x}_B$:

$$\int \exp\Big\{-\frac{1}{2}\Big(\boldsymbol{x}_B - \big(\boldsymbol{\mu}_B - \boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)\big)\Big)^\top \boldsymbol{\Lambda}_{BB}\Big(\boldsymbol{x}_B - \big(\boldsymbol{\mu}_B - \boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)\big)\Big)\Big\}\mathrm{d}\boldsymbol{x}_B \tag{A.69}$$

Since the term $\boldsymbol{\mu}_B - \boldsymbol{\Lambda}_{BB}^{-1}\boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)$ is a vector then this integral is the integral of a non-normalized Gaussian distribution therefore it is equal to:

$$\int \exp\big(-w\big)\mathrm{d}\boldsymbol{x}_B = (2\pi)^{n_B/2}|\boldsymbol{\Lambda}_{BB}^{-1}| \tag{A.70}$$

where n_B is the dimension of the $\boldsymbol{x}_B$ vector. Therefore, one obtains:

$$\begin{aligned}
\int f_{\tilde{\boldsymbol{x}}_A,\tilde{\boldsymbol{x}}_B}(\boldsymbol{x}_A,\boldsymbol{x}_B)\mathrm{d}\boldsymbol{x}_B &= \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}}(2\pi)^{n_B/2}|\boldsymbol{\Lambda}_{BB}^{-1}|\exp\big(-v\big) = \frac{(2\pi)^{n_B/2}|\boldsymbol{\Lambda}_{BB}^{-1}|}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\big(-v\big) \\
&= \frac{|\boldsymbol{\Lambda}_{BB}^{-1}|}{(2\pi)^{n_A/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\big((\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Sigma}_{AA}^{-1}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)\big) \qquad \text{(A.71)}
\end{aligned}$$

Considering the determinant property $|\boldsymbol{\Sigma}| = |\boldsymbol{\Sigma}_{AA}||\boldsymbol{\Sigma}_{BB} - \boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{AA}^{-1}\boldsymbol{\Sigma}_{AB}|$ and from the Schur complement identity in Section A.1 we have $\boldsymbol{\Lambda}_{BB}^{-1} = \boldsymbol{\Sigma}_{BB} - \boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{AA}^{-1}\boldsymbol{\Sigma}_{AB}$ so we finally derive:

$$f_{\tilde{\boldsymbol{x}}_A}(\boldsymbol{x}_A) = \int f_{\tilde{\boldsymbol{x}}_A,\tilde{\boldsymbol{x}}_B}(\boldsymbol{x}_A,\boldsymbol{x}_B)\mathrm{d}\boldsymbol{x}_B = \mathcal{N}(\boldsymbol{x}_A;\, \boldsymbol{\mu}_A, \boldsymbol{\Sigma}_{AA}) \tag{A.72}$$

A.11. Conditional Probability of a Gaussian Distribution

The findings of this section follow directly from the discussion in Appendix Sections A.1 to A.3 and A.9 to A.10

Here we follow explicitly the analysis of Thomas B. Schon and Fredrik Lindsten shown in [47]. We shall now prove the Equation of the conditional probability of a multivariate Gaussian distribution. Suppose a random vector $\tilde{\boldsymbol{x}}$ with two partitions of it $\tilde{\boldsymbol{x}}_A$ and $\tilde{\boldsymbol{x}}_B$ so that the results for the expected value vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ are the following:

$$\begin{bmatrix} \tilde{\boldsymbol{x}}_A \\ \tilde{\boldsymbol{x}}_B \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_{BB} \end{bmatrix}. \tag{A.73}$$

For the conditional probability the following Equation is valid:

$$f_{\tilde{\boldsymbol{x}}_a}(\boldsymbol{x}_A | \tilde{\boldsymbol{x}}_B = \boldsymbol{x}_B) = \frac{f_{\tilde{\boldsymbol{x}}_A, \tilde{\boldsymbol{x}}_B}(\boldsymbol{x}_A, \boldsymbol{x}_B)}{f_{\tilde{\boldsymbol{x}}_B}(\boldsymbol{x}_B)}, \tag{A.74}$$

where $f_{\tilde{\boldsymbol{x}}_A, \tilde{\boldsymbol{x}}_B}(\boldsymbol{x}_A, \boldsymbol{x}_B) = f_{\tilde{\boldsymbol{x}}}(\boldsymbol{x})$ so because we assume that vector $\boldsymbol{x}$ follows a multi-variate Gaussian distribution we have:

$$f_{\tilde{\boldsymbol{x}}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}. \tag{A.75}$$

Marginalizing for $\tilde{\boldsymbol{x}}_B$ we obtain:

$$f_{\tilde{\boldsymbol{x}}_B}(\boldsymbol{x}_B) = \frac{1}{(2\pi)^{n_B/2} |\boldsymbol{\Sigma}_{BB}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top \Sigma_{BB}^{-1} (\boldsymbol{x}_B-\boldsymbol{\mu}_B)} \tag{A.76}$$

Therefore we obtain:

$$f_{\tilde{\boldsymbol{x}}_A}(\boldsymbol{x}_A | \tilde{\boldsymbol{x}}_B = \boldsymbol{x}_B) = \frac{|\boldsymbol{\Sigma}_{BB}|^{1/2}}{(2\pi)^{n_A/2} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}\left\{(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) - (\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top \Sigma_{BB}^{-1} (\boldsymbol{x}_B-\boldsymbol{\mu}_B)\right\}} \tag{A.77}$$

From the determinant formulas and from the Schur complement identity in Section A.1 as explained above we obtain: $|\boldsymbol{\Sigma}| = |\boldsymbol{\Sigma}_{BB}||\boldsymbol{\Sigma}_{AA} - \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}\boldsymbol{\Sigma}_{BA}|$ and $\boldsymbol{\Lambda}_{AA}^{-1} = \boldsymbol{\Sigma}_{AA} - \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}\boldsymbol{\Sigma}_{BA}$, therefore we have:

$$f_{\tilde{\boldsymbol{x}}_A}(\boldsymbol{x}_A | \tilde{\boldsymbol{x}}_B = \boldsymbol{x}_B) = \frac{1}{(2\pi)^{n_A/2} |\boldsymbol{\Lambda}_{AA}|^{-1/2}} e^{-\frac{1}{2}\left\{(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) - (\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top \boldsymbol{\Sigma}_{BB}^{-1} (\boldsymbol{x}_B-\boldsymbol{\mu}_B)\right\}}. \tag{A.78}$$

Now, we observe the expression, let it be named w, inside the exponential term. Then we have:

$$\begin{aligned} w = -\frac{1}{2}\Big\{(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) - (\boldsymbol{x}-\boldsymbol{\mu}_B)^\top \boldsymbol{\Sigma}_{BB}^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_B)\Big\} & \\ = -\frac{1}{2}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A) - \frac{1}{2}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B) & \\ - \frac{1}{2}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top \boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A) - \frac{1}{2}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top (\boldsymbol{\Lambda}_{BB} - \boldsymbol{\Sigma}_{BB}^{-1})(\boldsymbol{x}_B-\boldsymbol{\mu}_B) & \end{aligned} \tag{A.79}$$

From the Schur complement relationships we know that $\boldsymbol{\Sigma}_{BB}^{-1} = \boldsymbol{\Lambda}_{BB} - \boldsymbol{\Lambda}_{BA}\boldsymbol{\Lambda}_{AA}^{-1}\boldsymbol{\Lambda}_{AB}$ so we end up with:

$$w = -\left\{\frac{1}{2}(\boldsymbol{x}_A-\boldsymbol{\mu}_A)^\top \boldsymbol{\Lambda}_{AA}(\boldsymbol{x}_A-\boldsymbol{\mu}_A) + (\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top \boldsymbol{\Lambda}_{BA}(\boldsymbol{x}_A-\boldsymbol{\mu}_B) + \frac{1}{2}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)^\top \boldsymbol{\Lambda}_{BA}\boldsymbol{\Lambda}_{AA}^{-1}\boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)\right\} \tag{A.80}$$

We shall proceed with 'completing the square'. Here we set $\boldsymbol{C} = \boldsymbol{\Lambda}_{AA} \Rightarrow \boldsymbol{C}^{-1} = \boldsymbol{\Lambda}_{AA}^{-1}$ and $\boldsymbol{b} = \boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B - \boldsymbol{\mu}_B)$. We see then that a has the form of $\frac{1}{2}\boldsymbol{b}^\top \boldsymbol{C}\boldsymbol{b}$, therefore $v = \frac{1}{2}\boldsymbol{b}^\top \boldsymbol{C}\boldsymbol{b} - a = 0$. So for the expected value we set $\boldsymbol{m} = -\boldsymbol{C}^{-1}\boldsymbol{b} = -\boldsymbol{\Lambda}_{AA}^{-1}\boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B - \boldsymbol{\mu}_B)$ and we obtain:

$$w = -\frac{1}{2}\Big(\boldsymbol{x}_A - \boldsymbol{\mu}_A + \boldsymbol{\Lambda}_{AA}^{-1}\boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)\Big)^\top \boldsymbol{\Lambda}_{AA}\Big(\boldsymbol{x}_A - \boldsymbol{\mu}_A + \boldsymbol{\Lambda}_{AA}^{-1}\boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)\Big) \tag{A.81}$$

$$= -\frac{1}{2}\Big(\boldsymbol{x}_A - \big(\boldsymbol{\mu}_A - \boldsymbol{\Lambda}_{AA}^{-1}\boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)\big)\Big)^\top \boldsymbol{\Lambda}_{AA}\Big(\boldsymbol{x}_A - \big(\boldsymbol{\mu}_A - \boldsymbol{\Lambda}_{AA}^{-1}\boldsymbol{\Lambda}_{AB}(\boldsymbol{x}_B-\boldsymbol{\mu}_B)\big)\Big) \tag{A.82}$$

Noticing from Equation A.7 that $\boldsymbol{\Sigma}_{BB} = \boldsymbol{\Delta}_{\Lambda_{AA}}^{-1}$ and $\boldsymbol{\Sigma}_{AB} = -\boldsymbol{\Lambda}_{AA}^{-1}\boldsymbol{\Lambda}_{AB}\boldsymbol{\Delta}_{\Lambda_{AA}}^{-1}$ we finally derive a normal distribution with:

$$f_{\tilde{\boldsymbol{x}}_A}(\boldsymbol{x}_A | \tilde{\boldsymbol{x}}_B = \boldsymbol{x}_B) = \mathcal{N}(\boldsymbol{x}_A;\ \boldsymbol{\mu}_{A|B}, \boldsymbol{\Sigma}_{A|B}) \quad \text{with} \quad \boldsymbol{\mu}_{A|B} = \boldsymbol{\mu}_A + \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}(\boldsymbol{x}_B - \boldsymbol{\mu}_B) \quad \text{and} \quad \boldsymbol{\Sigma}_{A|B} = \boldsymbol{\Sigma}_{AA} - \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}\boldsymbol{\Sigma}_{BA} \tag{A.83}$$

A.12. Affine Transformation of a Gaussian Random Vector

From the analysis of Thomas B. Schon and Fredrik Lindsten shown in [47] we know that for a Gaussian random vector $\tilde{\boldsymbol{x}}$ with expected value $\mathbb{E}(\tilde{\boldsymbol{x}}) = \boldsymbol{\mu}$ and variance $\text{Var}(\tilde{\boldsymbol{x}}) = \mathbb{E}\big((\tilde{\boldsymbol{x}} - \boldsymbol{\mu})(\tilde{\boldsymbol{x}} - \boldsymbol{\mu})^\top\big)$ for an affine transformation of the form:

$$\tilde{\boldsymbol{y}} = \boldsymbol{A}\tilde{\boldsymbol{x}} + \boldsymbol{b} \tag{A.84}$$

we obtain again a Gaussian random vector. Its expected value and variance are the following:

$$\mathbb{E}(\tilde{\boldsymbol{y}}) = \mathbb{E}(\boldsymbol{A}\tilde{\boldsymbol{x}} + \boldsymbol{b}) = \boldsymbol{A}\mathbb{E}(\tilde{\boldsymbol{x}}) + \boldsymbol{b} = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b} \qquad \text{and} \tag{A.85}$$

$$\operatorname{Cov}(\tilde{\boldsymbol{y}}) = \mathbb{E}\big((\boldsymbol{A}\tilde{\boldsymbol{x}} + \boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b})(\boldsymbol{A}\tilde{\boldsymbol{x}} + \boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu} - \boldsymbol{b})^\top\big) = \mathbb{E}\big(\boldsymbol{A}(\tilde{\boldsymbol{x}} - \boldsymbol{\mu})(\tilde{\boldsymbol{x}} - \boldsymbol{\mu})^\top \boldsymbol{A}^\top\big) = \boldsymbol{A}\operatorname{Var}(\tilde{\boldsymbol{x}})\boldsymbol{A}^\top \tag{A.86}$$

Therefore we obtain the Gaussian random vector:

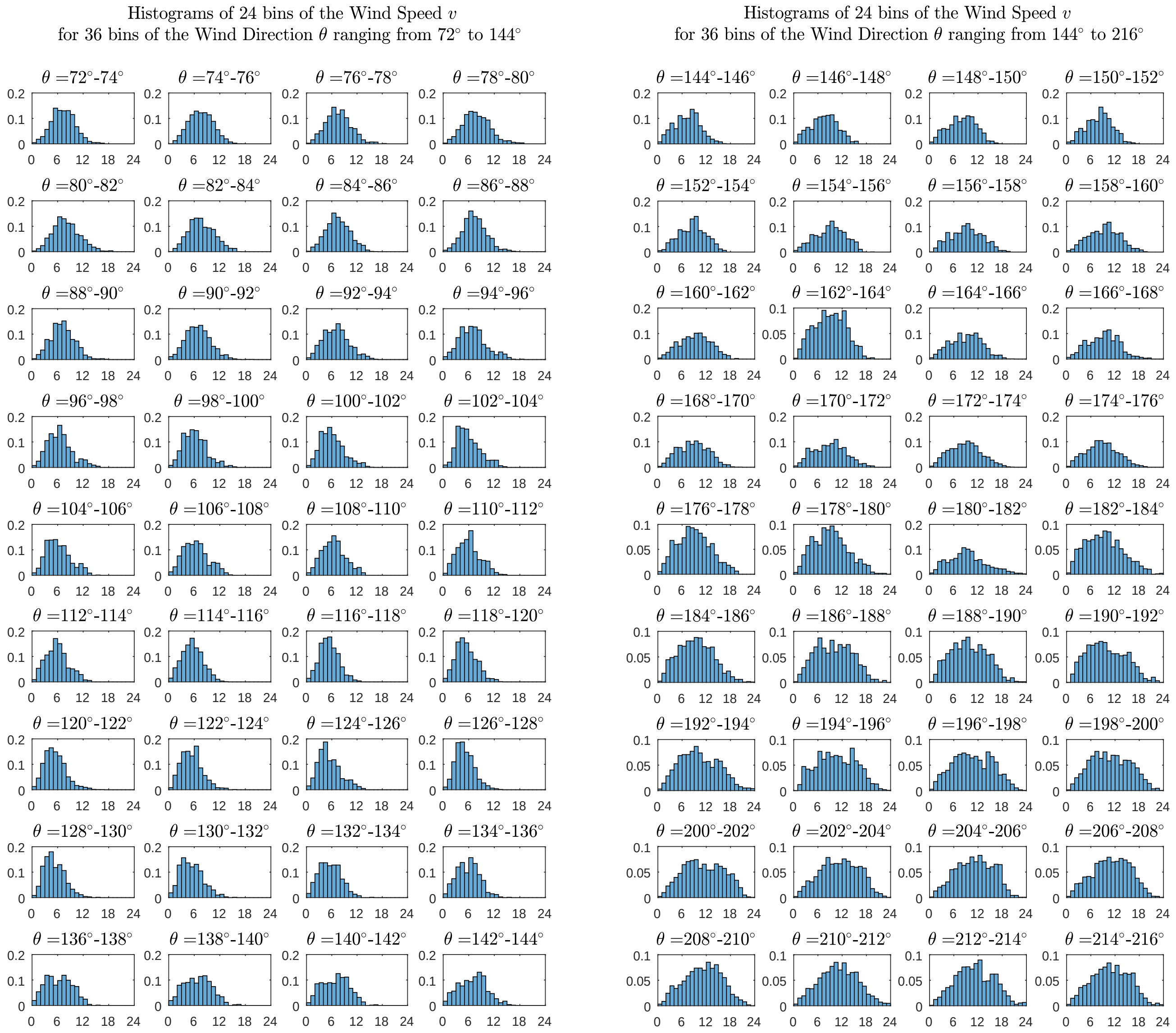
$$\tilde{\boldsymbol{y}} \sim \mathcal{N}(\boldsymbol{y};\ \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{A}\operatorname{Var}(\tilde{\boldsymbol{x}})\boldsymbol{A}^\top) \tag{A.87}$$

B

Appendix B – Meteorological Data

In this Appendix we demonstrate the continuation of the meteorological data as first presented in Section 6.4 in Figure 6.8.

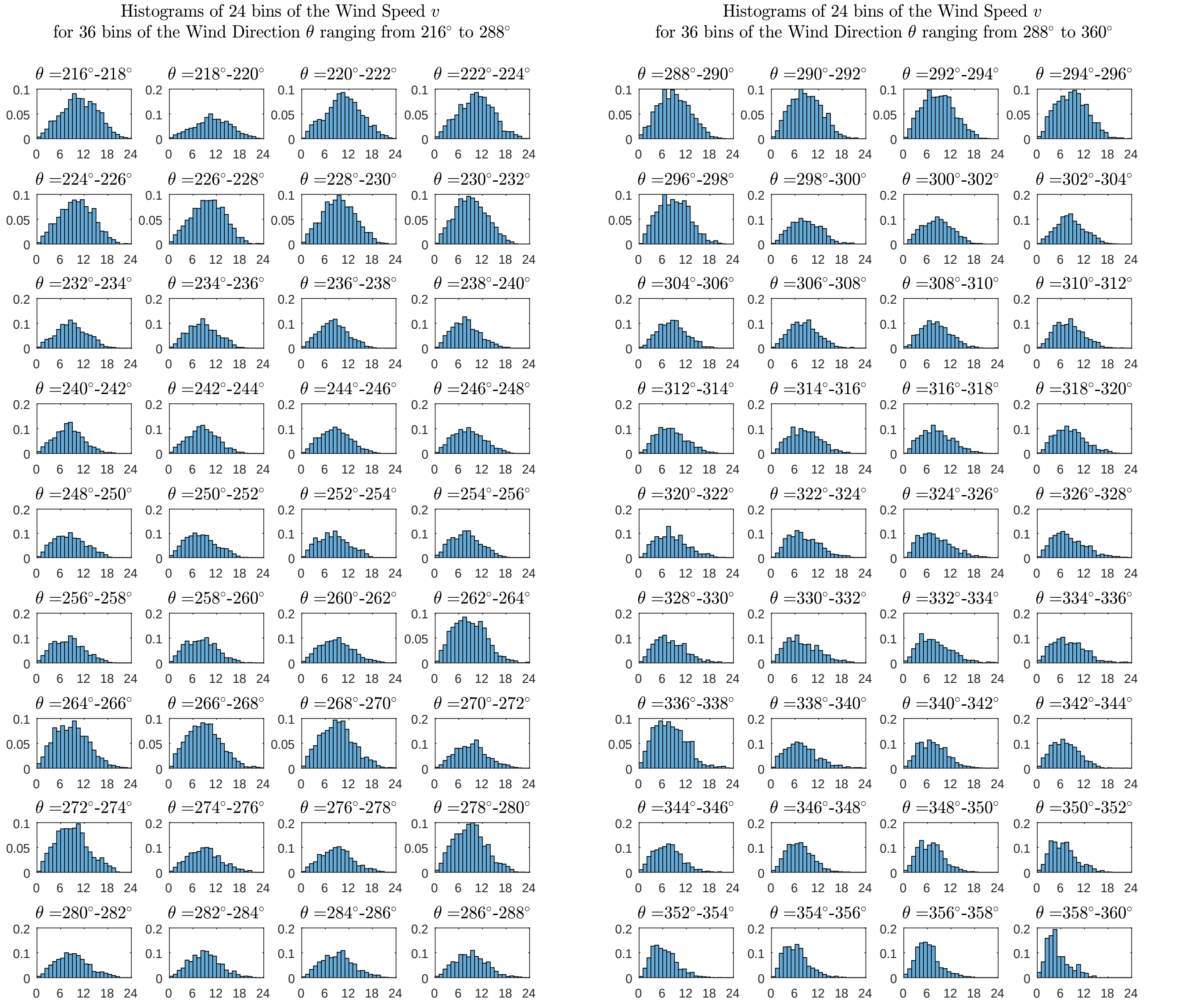
B.1. Meteorological Data - Histograms of Wind Speed



(a) Histograms of wind speed v arranged in 36 bins for given range of wind direction θ ranging from 72° to 144°.

(b) Histograms of wind speed v arranged in 36 bins for given range of wind direction θ ranging from 144° to 216°.

Figure B.1: Presentation of the probability distributions of the meteorological data of our study's wind farm



(a) Histograms of wind speed v arranged in 36 bins for given range of wind direction θ ranging from 216° to 288°.

(b) Histograms of wind speed v arranged in 36 bins for given range of wind direction θ ranging from 288° to 360°.

Figure B.2: Presentation of the probability distributions of the meteorological data of our study's wind farm