LEARNING DRAG COEFFICIENT OF BALLISTIC TARGETS USING GAUSSIAN PROCESS MODELING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FIRAT KUMRU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2019

Approval of the thesis:

LEARNING DRAG COEFFICIENT OF BALLISTIC TARGETS USING GAUSSIAN PROCESS MODELING

submitted by **FIRAT KUMRU** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İlkay Ulusoy _____
Head of Department, **Electrical and Electronics Engineering**

Assist. Prof. Dr. Emre Özkan _____
Supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Emre Özkan
Electrical and Electronics Engineering Department, METU _____

Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Gökhan Soysal
Electrical and Electronics Engineering Department, Ankara Uni. _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: FIRAT KUMRU

Signature :

ABSTRACT

LEARNING DRAG COEFFICIENT OF BALLISTIC TARGETS USING GAUSSIAN PROCESS MODELING

Kumru, Fırat
M.S., Department of Electrical and Electronics Engineering
Supervisor : Assist. Prof. Dr. Emre Özkan

September 2019, 92 pages

Ballistic object tracking involves estimating an unknown ballistic coefficient which directly affects the dynamics of the object. In most studies, the ballistic coefficient is assumed to be constant throughout the object's flight. In reality, the ballistic coefficient is a function of the speed of the object and depends on the object's aerodynamic properties. In the literature, the impact point prediction is defined as predicting the position that the object is expected to hit on the ground while the object is still on the fly. The accuracy of the impact point prediction highly depends on the treatment of the ballistic coefficient in the prediction model. In this thesis, we propose a method to learn the unknown function that describes the relationship between the speed and the ballistic coefficient of the object from the observations. Then, the function is used to predict the impact point of the ballistic object. The unknown function is learned via Gaussian process in the Bayesian framework. The proposed and conventional methods are comparatively studied in a realistic simulation environment. Extensive simulation studies are conducted to characterize the performance of the proposed method

and it is shown that the method has a better impact point prediction performance than the conventional ones in terms of the root mean square error.

Keywords: Ballistic Target Tracking, Ballistic Coefficient, Gaussian Process

ÖZ

BALİSTİK HEDEFLERİN SÜRTÜNME KATSAYISININ GAUSSİAN SÜREÇ MODELİYLE ÖĞRENİLMESİ

Kumru, Fırat
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Dr. Öğretim Üyesi Emre Özkan

Eylül 2019, 92 sayfa

Balistik obje takibi, obje dinamiğini doğrudan etkileyen balistik katsayının kestirimini içermektedir. Çoğu çalışmada, bu balistik katsayı obje uçuşu boyunca sabit kabul edilmektedir. Gerçekte, balistik katsayı obje süratinin bir fonksiyonudur ve objenin aerodinamik özelliklerine bağlıdır. Literatürde düşme noktası öngörümü, balistik obje havadayken objenin satıh ile buluşacağı konumun öngörüm problemidir. Öngörüm doğruluğu, balistik katsayının öngörüm modelinde nasıl ele alındığına bağlıdır. Bu tezde, obje sürati ve balistik katsayı arasındaki ilişkiyi tanımlayan fonksiyonu gözlemlere dayalı olarak öğrenmek için bir metot önerilmektedir. Bu öğrenilen fonksiyon daha sonra düşme noktası öngörümü boyunca kullanılır. Bilinmeyen bu fonksiyon Bayesçi yaklaşım altında Gauss süreci vasıtasıyla öğrenilmektedir. Önerilen yöntem ve geleneksel metotlar gerçekçi bir benzetim ortamında karşılaştırmalı olarak çalışılmıştır. Önerilen yöntemi karakterize etmek amacıyla geniş çaplı analizler yürütülmüş, yöntemin karekök ortalama hata ölçütü açısından geleneksel metotlardan

daha iyi bir düşme noktası öngörüm performansı olduğu gösterilmiştir.

Anahtar Kelimeler: Balistik Hedef Takibi, Balistik Katsayı, Gauss Süreci

To my family and my love Ezgi...

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Assist. Prof. Dr. Emre Özkan for accepting me to his work group, his valuable supervision, encouragement and valuable contributions throughout my thesis study.

I would like to thank my family for all their support. I am very grateful to my dearest mother for her endless patience and unconditional love. Also, This thesis was a chance to realize how lucky I am to have a brother like Murat Kumru. His mentorship and unceasing support throughout the thesis was very precious.

I wish to thank love of my life Ezgi for her endless patience, support, care and love. I owe her a debt of gratitude for being the joy of my life and the source of inspiration.

I also would like to thank my Boncuklar, especially to Övünç Sezer, Barış Nasır and Can Apaydın, for their technical and non-technical support and being there whenever I need.

I would like to express my gratitude to my colleagues, especially to Tamer Akça, Emre Altuntaş, Talha İnce and Recep Serdar Acar, at ASELSAN for their patience and support.

I would like to thank to ASELSAN Inc. for providing the opportunity to fulfil my study. I also thank TÜBİTAK (Scientific and Technological Research Council of Turkey) for their financial support (2210-A) during my study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

ABBREVIATIONS

BTT	Ballistic Target Tracking
BO	Ballistic Object
BC	Ballistic Coefficient
BP	Ballistic Parameter
EKF	Extended Kalman Filter
IPP	Impact Point Prediction
GP	Gaussian Process
RMS	Root Mean Square
WGS-84	World Geodetic System of 1984
MLE	Maximum Likelihood Estimation
2D	2-Dimensional
3D	3-Dimensional
CC	Cross-covariance
MC	Monte Carlo
NED	North-East-Down
DoF	Degrees of Freedom

CHAPTER 1

INTRODUCTION

Ballistic target tracking (BTT) involves estimating the kinematics of a ballistic object (BO) in a surveillance region based on a set of measurements collected by a sensor. A BO is a body with momentum which is free to move and subjected to forces such as propulsion, gravity or air drag. The term ballistic is used to describe that there is no guidance which steers the object during its flight (or at least, during most of the flight) [21].

A wide range of objects can be categorized as BO including bullets, unguided bombs, rockets, mortars, and even ballistic missiles. Some BTT methods in the literature propose a generic solution which is applicable to any type of BOs [17], [12] whereas some provide dedicated solutions to a specific type of BOs to improve accuracy [28], [29].

BOs can be broadly categorized into two types; with propulsion, without propulsion. Flight of a BO with propulsion has two phases. In the former, there exists a thrust force which accelerates the BO from launch to the end of propulsion. The latter is called the ballistic phase, and there only exist gravitational and aerodynamic forces on the BO. Some methods focus on the motion of BOs in the ballistic phase only [17], [28], whereas other methods consider both phases [39], [8], [19]. In this study, we consider the ballistic phases of BOs which stay within atmosphere throughout their entire flight.

In BTT literature, various filtering techniques such as Extended Kalman Filter (EKF) [12], Unscented Kalman Filter, Particle Filter [5] are used for estimating object's

state. The performance of these filters are compared in several studies [17], [13], [30], [40]. In this work, we use an EKF for inference.

Atmospheric drag force is one of the forces that are dominant in determining the trajectory of a BO traveling in the atmosphere. Therefore, a realistic modeling of the drag force is essential for tracking accuracy. The drag force depends on the velocity of the object, air pressure at the object's altitude, the cross-sectional area of the object along the wind and the ballistic coefficient (BC) of the object. The BC of a body, which is a measure of its ability to overcome the air resistance during flight, depends on the speed of the body. In general, BC-speed characteristics of BOs are highly non-linear (see Fig. 1.1). BC directly affects the magnitude of the drag force acting on a BO; hence, the estimation accuracy of the BC has a significant impact on the performance of a tracking system.

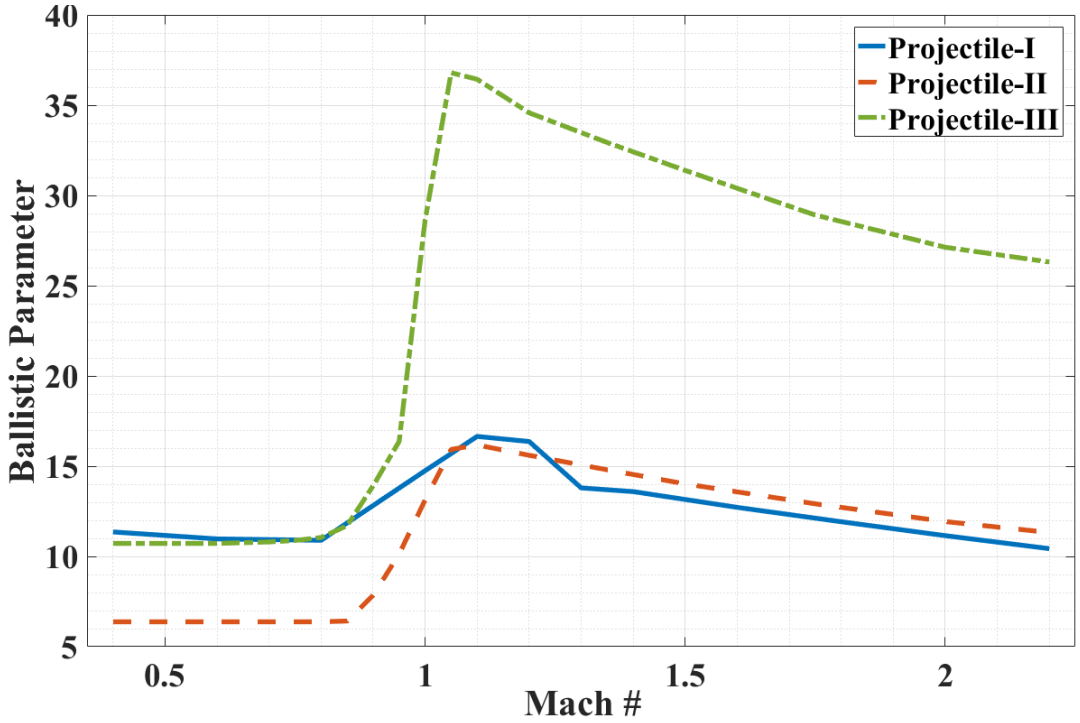


Figure 1.1: Drag coefficients for different projectiles [2]

There exist several studies in the literature which propose different methods of including BC in their models and perform estimation. The methods presented in Farina [13] and Benavoli [3] assume that BC is either known or there exists prior information in

the form of lower and upper bounds on the BC value depending on the object type.

If BC is assumed to be unknown, a standard approach is to augment the state vector with the BC and estimate them concurrently using the aforementioned filters. In most models, the dynamics of the BC is assumed to exhibit an artificial random walk [17], [28], [8].

In some applications, the impact point of a BO is the main interest of tracking. In these applications, an impact point prediction (IPP) is performed while the BO is still in flight. As a simple solution, BC can be assumed to be constant throughout the IPP. This assumption might hold for the BOs which stay in subsonic speed regime. Furthermore, it is critical to correctly estimate the constant but unknown BC to achieve prediction accuracy.

Contrary to the aforementioned assumption, BC is not constant throughout the prediction phase, in general. In such cases, drag templates can be used from a library of drag curves to aid IPP [8]. These methods require the construction of aerodynamic models regarding prospective projectile types and generation of the drag curve database beforehand. Furthermore, the projectile type must be correctly identified online so that an appropriate drag curve can be used. Using an improper drag template may cause significant IPP errors [8].

To aid BTT and IPP, Yuan et al. [39] suggest exploiting the dependency of the BC on the speed of the object. They assume that the BC characteristics of the object is a scaled version of a generic BC-Speed function which is considered to be known in advance. Then, the scale factor is estimated by augmenting it into the state vector. Briefly, this approach is a means of decoupling the projectile specific parameters (mass and cross-sectional area) from speed dependency of the BC. Then, a function of the projectile specific parameters can be treated as the scale factor. This scale factor is estimated concurrently with the kinematic state of the BO.

1.1 Proposed Work and Contributions

In this study, instead of utilizing a drag curve database, BC-Speed characteristic of the target is learned in a Bayesian framework. This learning is performed based on the BC and the velocity estimates of the tracking filter. Then, this learned function is used to determine the value of the BC during the IPP. For learning the characteristic, a Gaussian Process (GP) model [27] is utilized. GP is a modeling tool which has been widely used by the machine learning, statistics, and signal processing communities for identification [37], classification [24] and regression problems [35] due to its tractable posterior computation and attractive analytical properties. With Gaussian Process model, it is convenient to embed any a priori information about shape and smoothness of the BC characteristic of the target in the learning model by using various kernel functions. Furthermore, incorporating the uncertainty of the training data into the regression is possible with GP model [22] .

The proposed method has several advantages. It requires neither any preliminary work on aerodynamic modeling of the targets nor classification of the projectile on the fly. Therefore, it makes IPP against an unexpected target type possible with higher accuracy compared to the traditional methods. Moreover, it has a reasonably low computational load while providing a significant improvement in the IPP performance when compared with conventional methods which do not utilize any learning procedure on the BC-Speed characteristic.

The method presented here has its basis on manipulating the information extracted from the BC and velocity estimates of the filter while predicting the impact point. The phenomenon that makes our method applicable is that a projectile attains the same speed values on both ascending and descending phases, especially when it follows a lofted trajectory. As observed in Fig.1.2, the BOs of interest decelerate at ascending phase and accelerate at descending phase under the effect of gravity. The behaviour that is deduced at the ascending phase can be readily used at the descending phase by the model. To achieve the transfer of the information from one phase to the other, the GP regression model provides the BC characteristic of the target based on the BC and

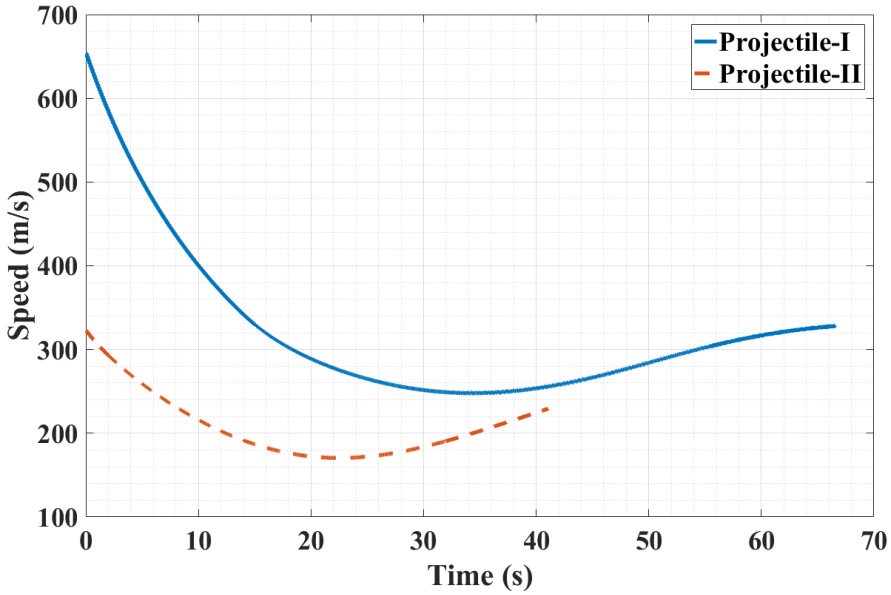


Figure 1.2: Sample projectile speeds vs time

speed estimates of the filter. This characteristic is then used in the system model to enhance the IPP performance. Furthermore, by using an appropriate kernel function, adequate information about the BC characteristic at non-visited speed values can also be inferred based on the BC-speed estimates at visited speeds. The whole method described here is briefly illustrated on a sample trajectory of a projectile in Fig. 1.3.

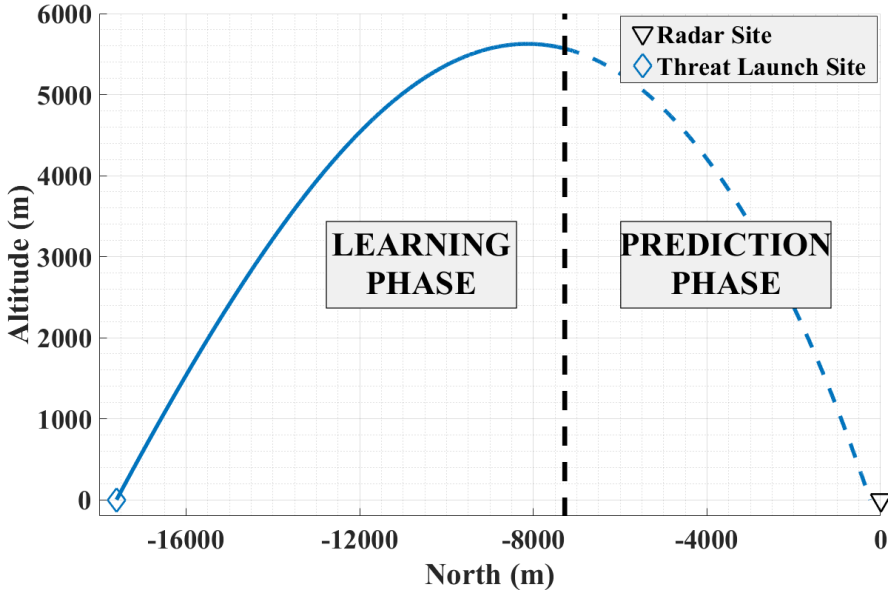


Figure 1.3: Phases of the proposed method on a sample trajectory

CHAPTER 2

BALLISTIC OBJECTS AND DYNAMICS

In this chapter, types of ballistic objects and effective forces acting on them are presented with the intention of informing the reader about the targets of interest.

2.1 Ballistic Objects

A flying object is called BO if there is no self-guidance mechanism to steer the object to a specified target. During the flight, a BO is affected by different forces and moments according to the flight conditions. Furthermore, the stability mechanism of the object is another factor that can initiate additional forces and moments on the object. The trajectory of a BO is divided into multiple phases considering the effective forces on the object. These phases and the stability mechanisms are discussed in more detail in the following subsections.

2.1.1 Trajectory of Ballistic Objects

Trajectory of a BO is divided into multiple phases mainly according to forces acting on the object:

- **Boost Phase:** In the boost phase, there exists a thrust force which accelerates the object starting from the launch till the end of propulsion. This phase usually takes short period of time compared to the time of flight. Among the ballistic object family, only rockets and ballistic missiles use the propulsion, since they

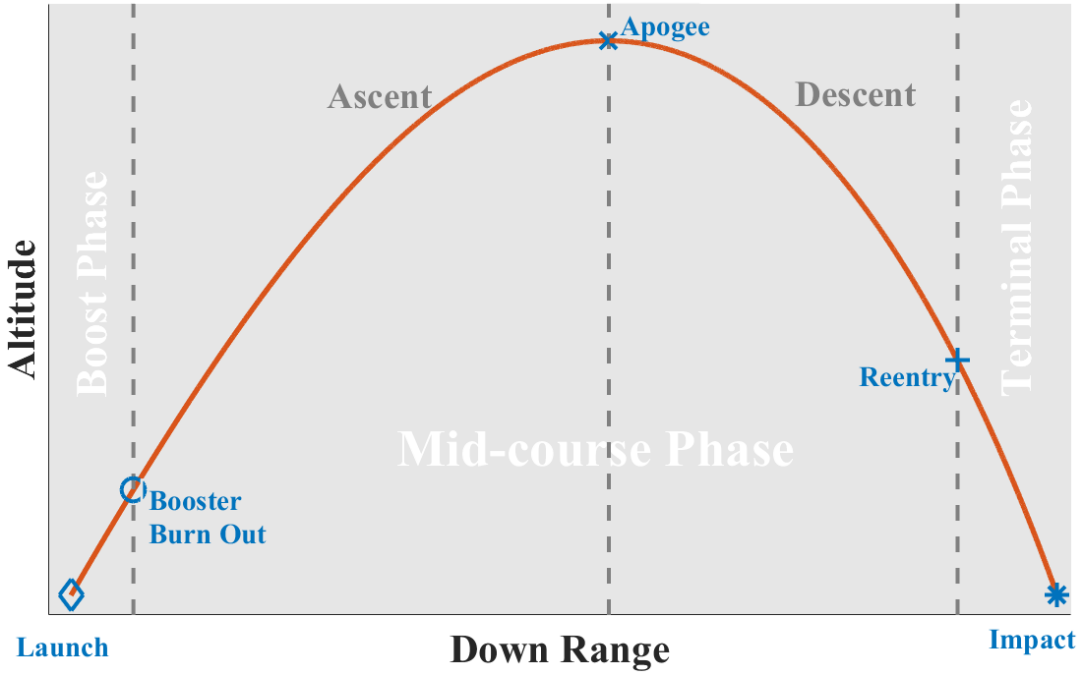


Figure 2.1: Phases of a ballistic trajectory

intend to reach longer distances. In this phase, besides gaining momentum, ballistic missiles may also use a control mechanism such as thrust vector control to steer themselves into predefined trajectory towards their desired impact point.

- **Mid-course (Ballistic) Phase:** The mid-course phase commences when the propulsion expires. This phase is also called ballistic phase to show that there is neither a thrust force to accelerate the object nor a mechanism to control and steer the object. In this phase, long range ballistic missiles are mostly outside the atmosphere. Hence, there is no aerodynamic forces acting on these ballistic missiles. This type of flight is called exo-atmospheric flight.

- **Terminal (Re-entry) Phase:** The terminal phase commences when the ballistic object re-enters the Earth's atmosphere and terminates when the BO hits the ground. The objects which exhibit endo-atmospheric flight do not have the terminal phase since they are in the atmosphere throughout their whole flight.

2.1.2 Stability of Ballistic Objects

To enhance the firing range and targeting accuracy, ballistic objects need to be stabilized throughout the flight. The stabilization means that the object's longitudinal axis tends to point towards the direction of movement [25]. The main factor determining the stabilization of a ballistic object is the placement of center of gravity (c_g) and center of pressure (c_p) along the object's longitudinal axis. c_g is a hypothetical point where whole distribution of mass of an object can be assumed to concentrate on. c_p of an object traveling through the atmosphere can be defined as the point where all of the aerodynamic pressure may be represented by a single force vector with no moment [7].

This stabilization can be achieved by applying one of the two techniques described below.

- **Fin Stabilization:**

  The instability problem mainly arises when c_p is located in front of c_g along the object's longitudinal axis. In that circumstance, a small deviation of the direction of the projectile's nose from the velocity vector results in an overturning moment on the object which increases the deviation. As a result, the object becomes unstable and it eventually tumbles. Fin stabilization aims to stabilize the object by forcing c_p to be located behind c_g by using tail surfaces (fins) so that any yawing in the object's orientation is opposed by the introduced moment. Most projectiles such as mortars, sabots, rockets and missiles use this stability mechanism because of their large size [21].

  Translocation of c_p with translocation of the fins are illustrated in Figure 2.2. In the top figure, the rocket is stable since c_g is in front of c_p. The rocket will return to its stable orientation when any deviation of the orientation occurs. In the middle figure, c_g and c_p are very close to each other. Such a condition is called neutral stability. The rocket may have a stable or unstable flight depending on the forces acting on it. Lastly, in the bottom figure of Figure 2.2, c_p is in front of c_g and the rocket is unstable.

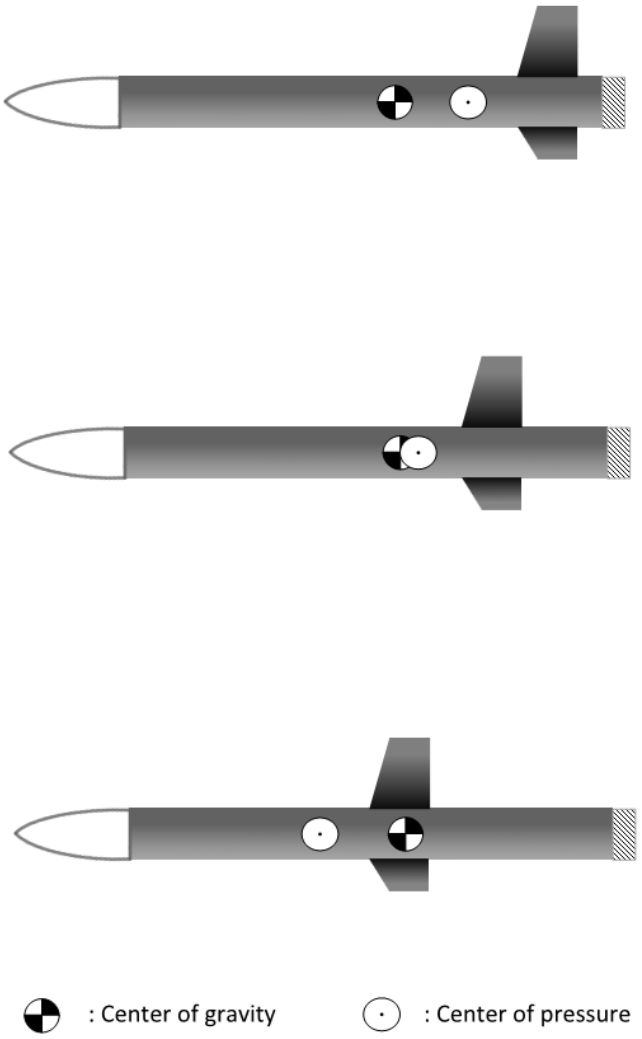


Figure 2.2: (Top) Stability, (Middle) Neutral stability, (Bottom) Instability conditions under fin stabilization.

- **Spin Stabilization:**

  Other types of projectiles such as small arms and artillery shells must deal with the instability in a different way. These projectiles take advantage of the gyroscopic effect to stabilize themselves by spinning around their longitudinal axes. The spin is created by rifling within barrel. Gyroscopic forces exerted on the spinning object give resistance to the object against the destabilizing moment. In this mechanism, the amount of introduced spin rate is significant. Too little spin may result in unstable projectile flight, whereas too much spin

may result in failure in following the curvature of the trajectory [15]. Moreover, the projectile encounters significantly high drag force and it becomes unable to reach the intended range. The projectile trajectories under sufficient spin (stable flight), insufficient spin (unstable flight) and over spin (over stable flight) are illustrated in Figure 2.3.

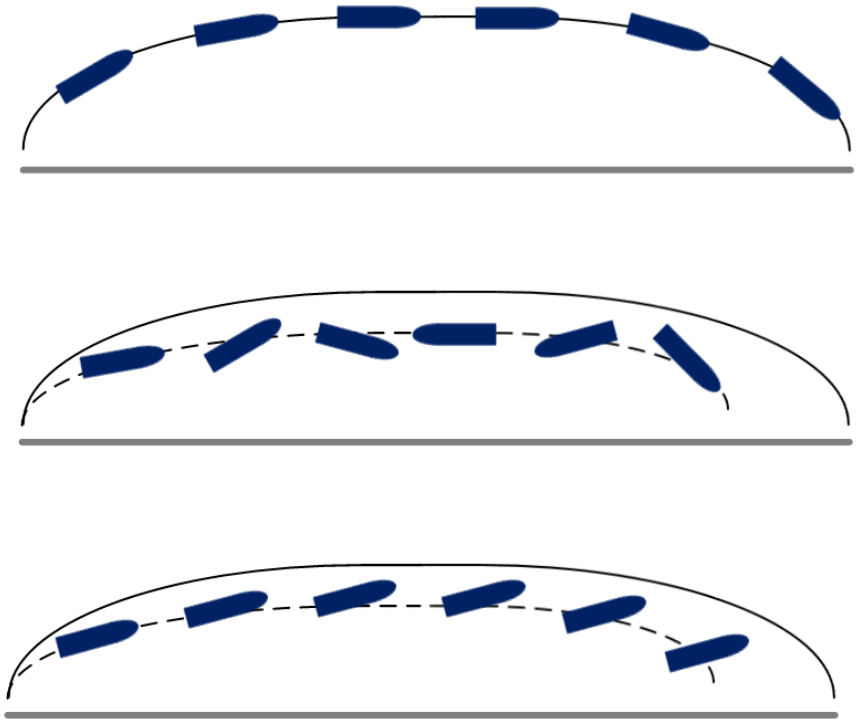


Figure 2.3: (Top) Stable, (Middle) Unstable, (Bottom) Over stable flight trajectories under spin stabilization.

Moreover, spin stabilized projectiles exhibit epicyclic motion which is depicted in Figure 2.4 from head-on view (at top) and side view (at bottom) [15]. As the projectile proceeds, this motion diminishes but does not vanish completely.

The utilized stabilization technique affects the trajectory significantly because additional forces and moments may be introduced depending on the employed technique. Such phenomena will be explained in details in the following subsections.

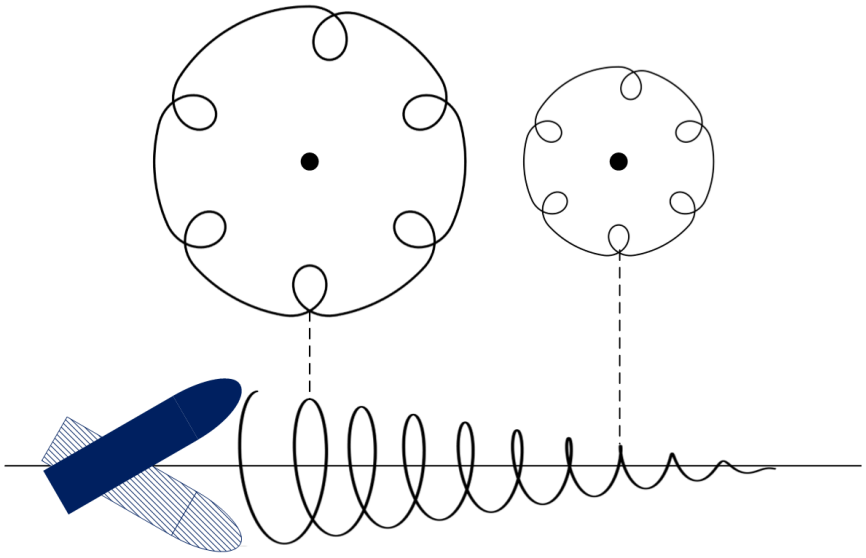


Figure 2.4: Generic epicyclic motion of a spin stabilized projectile

2.2 Dynamics of Ballistic Objects

Due to unguided and non-maneuvering nature of ballistic objects, estimating a ballistic trajectory can be considered to be a trivial task. However, there are various forces and moments acting on the object and estimating them by a filter can be a tedious or even infeasible task without the knowledge of the object type and its behaviour. These forces and moments are introduced in the following subsections.

2.2.1 Aerodynamic Forces

Aerodynamic forces are exerted on a body by the air in which the body is immersed. These are caused by the relative motion between the body and air [7]. In the following subsections, drag, lift and Magnus forces are introduced in details.

2.2.1.1 Atmospheric Drag Force

In fluid dynamics, drag is basically the air resistance; more formally, it is a force acting opposite to the relative velocity of an object with respect to the surrounding

fluid [1].

$$F_{drag} = -\frac{1}{2}\rho s c_d(M, \alpha_t, \zeta) \|\mathbf{v}\| \mathbf{v}. \tag{2.1}$$

In (2.1), ρ is the air density, s is the cross-sectional area of the projectile, c_d is the drag coefficient which is a function of object's Mach Number (M) and total yaw angle (α_t), $\mathbf{v}$ is the object's velocity relative to the air and $\|\cdot\|$ stands for the Euclidean norm of its argument. M is a dimensionless quantity that represents the ratio of object's speed with respect to the local speed of sound [38] and formulated as

$$M = \frac{\|\mathbf{v}\|}{c}. \tag{2.2}$$

In the equation, c is the local speed of sound which is subject to change with the altitude of interest, temperature, density and pressure of the environment. c_d also depends on the object's shape and ζ is included in 2.1 to emphasize this dependency. In order to show this dependency, the drag coefficient against Mach number of three projectiles with different shapes are illustrated in Figure 2.5.

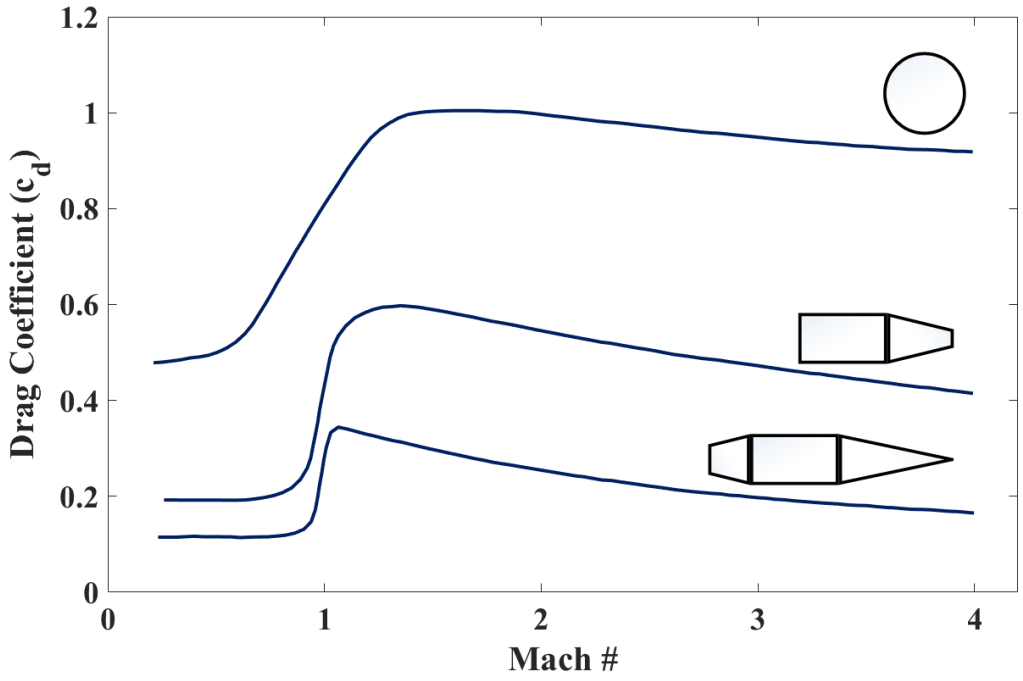


Figure 2.5: Drag coefficients vs Mach number

A closer look to c_d profiles given in Figure 2.5 can reveal significant information about the common characteristics of most BOs' drag coefficient profiles. The outlier profiles given in the figure belong to the projectiles which have the extreme projectile shapes: perfectly spherical and extremely pointy. In fact, a projectile with a different shape has a drag coefficient characteristic in between these extreme profiles. Furthermore, for the majority of the projectiles, the drag coefficient is almost constant below 0.8 Mach and shows a highly nonlinear characteristic around 1 Mach. The characteristic around 1 Mach mainly stems from the formation of shock waves around the object. c_d generally begins to decrease linearly around 1.2 Mach [34]. In general, the Mach number's range is divided into three regions according to different wave formations induced around the objects: below 0.8 Mach is called the subsonic speed regime, above 1.2 Mach is called the supersonic speed regime, and the region between these regimes is called the transonic speed regime.

Another term which affects the drag coefficient is the total yaw angle of the projectile. The term is used to describe any angular motion of the projectile's axis of rotational symmetry relative to the trajectory. In other words, it is a combination of angle of attack (vertical angle) and angle of sideslip (horizontal angle). The total yaw angle (α_t) can be seen in Figure 2.6 with the drag force which is in the opposite direction of the velocity.

c_d's dependency on the total yaw angle can be well approximated by the following equation:

$$c_d = c_{d_0} + c_{d_{\delta 2}} \delta^2, \tag{2.3}$$

where $\delta = \sin \alpha_t$, c_{d_0} is zero-yaw drag coefficient and $c_{d_{\delta 2}}$ is δ-yaw drag coefficient. Both of these coefficients change with Mach number. In addition to that, α_t changes during the flight due to the epicyclic motion of a spin-stabilized projectile (see Figure 2.4). As a result, c_d also varies during the flight since M and α_t vary.

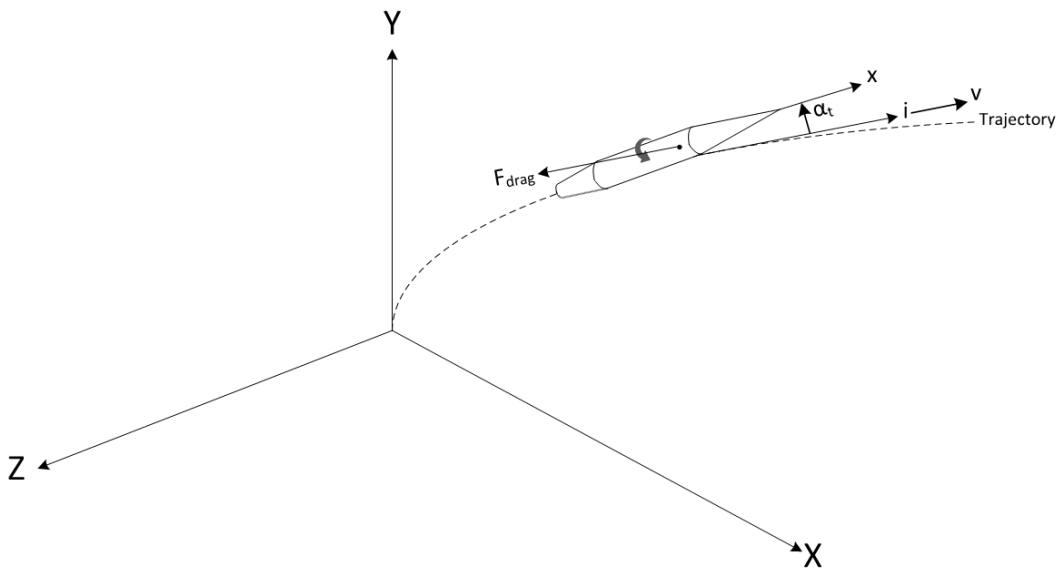


Figure 2.6: Drag force

2.2.1.2 Lift Force

Aerodynamic lift force is defined as the force perpendicular to the tangent of the trajectory (or to the velocity vector). It tends to pull the projectile towards its longitudinal axis [21]. Lift force can be stated in vectorial form as

$$F_{lift} = \frac{1}{2}\rho s c_{L_\alpha} \|\mathbf{v}\|^2 \left(\mathbf{i} \times (\mathbf{x} \times \mathbf{i})\right), \tag{2.4}$$

where c_{L_α} is the lift force coefficient, $\mathbf{i}$ is the unit vector in the direction of the velocity, $\mathbf{x}$ is the unit vector in the direction of the projectile's axis of rotational symmetry and $\times$ represents the cross product of two vectors.

Lift force is perpendicular to the velocity vector and lies in the plane spanned by the velocity vector and the projectile's longitudinal axis. Contrary to what its name implies, lift force may affect in arbitrary directions depending on the velocity vector and the longitudinal axis of the projectile. If the total yaw angle becomes zero, in other words, if the velocity vector and the projectile's longitudinal axis are aligned, lift force vanishes.

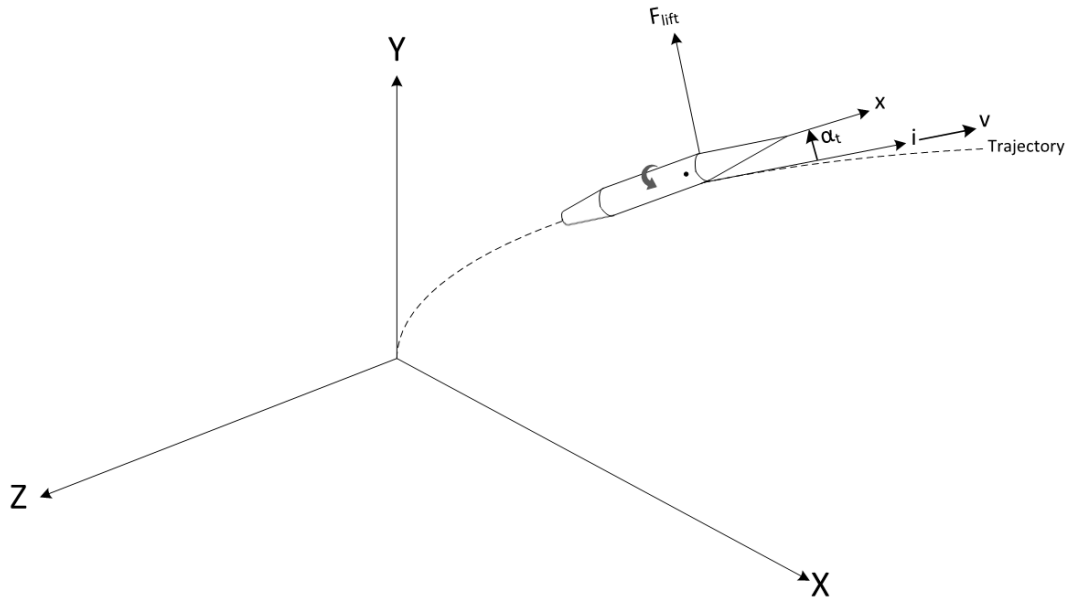


Figure 2.7: Lift force

2.2.1.3 Magnus Force

Magnus force is a side force that affects a spinning object having non-zero total yaw angle. It stems from unequal pressure on vicinity of the spinning object and lies in a direction perpendicular to the plane spanned by the velocity vector and the projectile's longitudinal axis. It can be formulated in vectorial form as

$$F_{magnus} = \frac{1}{2}\rho \left\|\mathbf{v}\right\|^2 s \frac{pd}{\left\|\mathbf{v}\right\|} c_{N_{P_\alpha}} (\mathbf{i} \times \mathbf{x}), \tag{2.5}$$

where p is the axial spin rate, d is the projectile's reference diameter and $c_{N_{P_\alpha}}$ is the Magnus force coefficient which is a small negative quantity. Magnus force vanishes only if spin rate or total yaw angle becomes zero. Figure 2.8 illustrates the Magnus force.

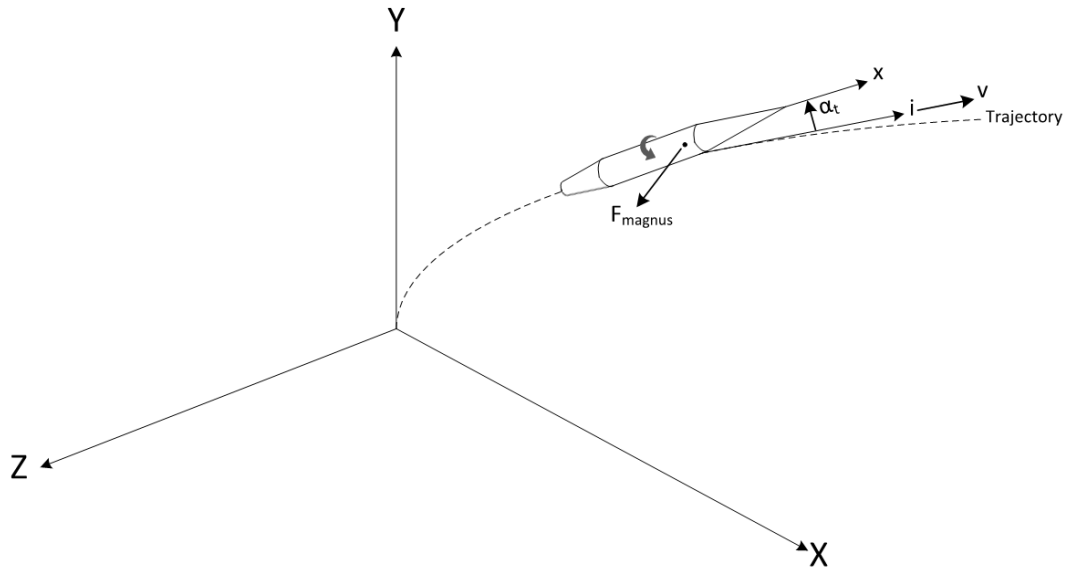


Figure 2.8: Magnus force

2.2.2 Aerodynamic Moments

The aerodynamic forces mentioned in Section 2.2.1 also generate a moment which rotates the object along a direction depending on the positions of its c_p and c_g.

2.2.2.1 Overturning Moment

The overturning moment is induced by the lift force discussed in Section 2.2.1.2. For the majority of spin stabilized projectiles, the overturning moment raises the total yaw angle since c_p is located in front of c_g which may cause instability. The projectiles should spin at sufficient spin rates to overcome this phenomenon. The vectorial form of the moment is

$$M_{\text{overturning}} = \frac{1}{2}\rho s d c_{M_\alpha} \|\mathbf{v}\|^2 (\mathbf{i} \times \mathbf{x}), \tag{2.6}$$

where c_{M_α} is the overturning moment coefficient. The moment is shown in Figure 2.9.

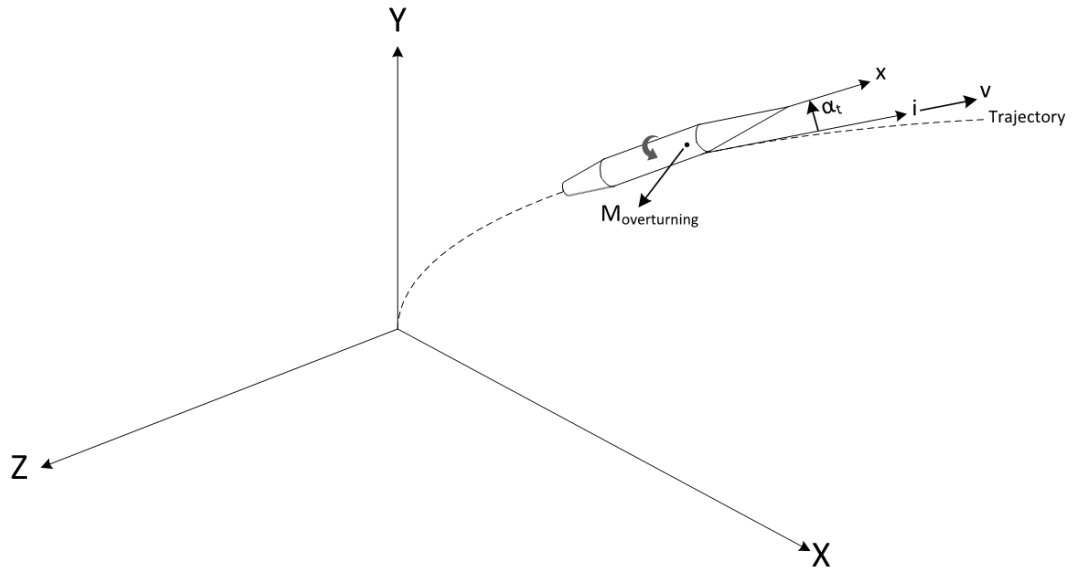


Figure 2.9: Overturning moment

2.2.2.2 Magnus Moment

Magnus force introduced in Section 2.2.1.3 generates a moment which is perpendicular to the projectile's longitudinal axis and lies in the plane spanned by the velocity vector and the longitudinal axis. Its vectorial form is defined as

$$M_{magnus} = \frac{1}{2}\rho d \left\|\mathbf{v}\right\|^2 s \frac{pd}{\left\|\mathbf{v}\right\|} c_{M_{P_\alpha}} \big(\mathbf{x} \times (\mathbf{i} \times \mathbf{x})\big), \tag{2.7}$$

where $c_{M_{P_\alpha}}$ is the Magnus moment coefficient. This coefficient can either be positive or negative depending on the position of c_g, the projectile's shape, the total yaw angle and the value of Mach number.

Magnus moment is the primary moment which determines the stability of the object. Hence, it must be always considered and well represented in calculations.

Besides aforementioned forces and moments, there are other forces and moments acting on the ballistic object, such as pitch damping force, pitch damping moment, spin damping moment, rolling moment (for canted projectiles), Magnus cross force and Magnus cross moment. However, they usually produce such insignificant effects

on the trajectory of the projectile that they can be neglected in most circumstances. Therefore, they are not given in this document.

CHAPTER 3

GAUSSIAN PROCESSES

Gaussian process is a non-parametric stochastic model which specifies a prior over an infinite-dimensional space of functions. It can also be interpreted as a collection of random variables, any finite number of which have a joint Gaussian distribution [14].

An unknown function $f(\cdot)$ which has a GP prior is denoted as

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')),$$

where $\mu(x)$ and $k(x, x')$ are its mean and covariance functions, respectively and x is the argument of the function. A Gaussian process is uniquely defined by its mean and covariance functions. These functions are defined as

$$\mu(x) = \mathbb{E}[f(x)], \tag{3.1a}$$

$$k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))^T]. \tag{3.1b}$$

By GP definition, function values evaluated at a finite number of inputs, $x_1, \ldots, x_L$, are normally distributed,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_L) \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, K), \tag{3.2a}$$

where

$$\boldsymbol{\mu} = \begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_L) \end{bmatrix}, \; K = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_L) \\ \vdots & \ddots & \vdots \\ k(x_L, x_1) & \dots & k(x_L, x_L) \end{bmatrix}. \tag{3.2b}$$

3.1 Kernel Functions

Constructing a correlation structure between the data points is crucial in the context of supervised learning, since the output of a test point is predicted based on the "closeness" of training data to the test point. This closeness can be defined in several ways, e.g., based on the distance of two input points, x and x', in the input space. Kernel function specifies the correlation between data points in GP models. Before examining the common forms of kernel functions, let us first define the stationary and isotropic kernel functions: A kernel function is called stationary if it is a function of $x - x'$. In that case, the kernel function is invariant to translations in the input space. Secondly, a kernel function is called isotropic if it is a function of $\|x - x'\|$, which is the norm distance of two input points. In that case, the direction of the vector difference of the inputs has no importance. For example, in a 2-D input space, the correlation between $f([1, 0])$ and $f([2, 0])$ is the same as the correlation between $f([0, 1])$ and $f([0, 2])$. Stationarity is a more general class of kernels than isotropy, i.e., every isotropic kernel is stationary, but not vice versa.

For the function $f(\cdot)$, it is stated that the function values, evaluated at any number of inputs, are jointly Gaussian and this joint distribution has a covariance matrix which is given in (3.2b). Kernel functions, $k(\cdot, \cdot)$, are used to construct this covariance matrix which specifies the correlation between the function values. Please note that the correlations between the function values are determined by their input values.

In order to understand the significant role of the kernel (and its hyperparameters) on GP modeling, different kernel functions are introduced and discussed in the following subsections.

3.1.1 Exponentiated Quadratic Kernel

Exponentiated Quadratic (EQ) Kernel, which is also called Squared Exponential, is one of the most preferred kernels in GP models. It forms infinitely differentiable functions, i.e., at highest possible smoothness. EQ Kernel is specified with the following expression

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{(\mathbf{x}-\mathbf{x}')^\top \Sigma^{-1} (\mathbf{x}-\mathbf{x}')}{2}\right), \tag{3.3a}$$

$$\Sigma = diag(l_1^2, ..., l_N^2), \tag{3.3b}$$

where the dimension of the input space is N and $diag(\cdot)$ stands for a function which places the given inputs diagonally on a square matrix. The kernel has two types of hyperparameters which determine its characteristic: σ^2 is the variance and it determines the amount of variation in function values. It is also called the scaling parameter since it is a multiplier of the exponential function. Correlation between two points specified by EQ decreases as the distance between the points increases. $l_1, ..., l_N$ are length scale parameters and they specify how fast this correlation decreases with the distance at each corresponding dimension. In that sense, length scale parameters describe the smoothness of the function. A small value of l enables the function to change its values rapidly, whereas a high value results in a smoother function. In the regression framework, the value of the parameter specifies how well the training data is extrapolated on test points.

In its general form, the kernel is stationary but not isotropic. The isotropic form, which is used more frequently, is

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{(\mathbf{x}-\mathbf{x}')^\top (\mathbf{x}-\mathbf{x}')}{2l^2}\right). \tag{3.4}$$

The isotropic form has only two hyperparameters the length scale parameter for all dimensions of the input and the scaling parameter. Using the same length scale at

all dimensions is a reasonable assumption, since the correlation of data at different dimensions is the same in most regression problems.

3.1.2 Exponential Kernel

Exponential kernel generates continuous but non-differentiable functions. It is formulated as

$$k(r) = \sigma^2 \exp\left(-\frac{r}{l}\right), \tag{3.5}$$

where $r = \sqrt{(\mathbf{x} - \mathbf{x}')^\top (\mathbf{x} - \mathbf{x}')}$ is the Euclidean distance between the input points, σ^2 is the scaling parameter and l is the characteristic length scale.

3.1.3 Matern Kernel

Matern Kernel is a broad class of kernels whose members are distinguished by a parameter ν. For $p = \nu - 1/2$, the kernel can be written as a product of an exponential and a polynomial of order p. Most popular choices of ν are $3/2$ and $5/2$ which correspond to Matern-3 and Matern-5, respectively. The expressions for these kernels are given below:

$$k_{3/2}(r) = \sigma^2 \left(1 + \frac{\sqrt{3}r}{l}\right) \exp\left(-\frac{\sqrt{3}r}{l}\right), \tag{3.6a}$$

$$k_{5/2}(r) = \sigma^2 \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}r}{l}\right). \tag{3.6b}$$

A Gaussian process with Matern kernel generates sample functions that are p times differentiable. Matern-3 and Matern-5 generates one time and two times differentiable functions. Furthermore, as ν approaches to ∞, Matern kernel converges to EQ kernel given in (3.4). For $\nu = 1/2$, Matern kernel becomes Exponential kernel given in (3.5). Stein [33] states that infinitely differentiable kernel functions are unrealistic for physical processes and proposes to use Matern function with finite values of ν.

In Figure 3.1, aforementioned kernels are depicted by keeping the hyperparameters at the same values.

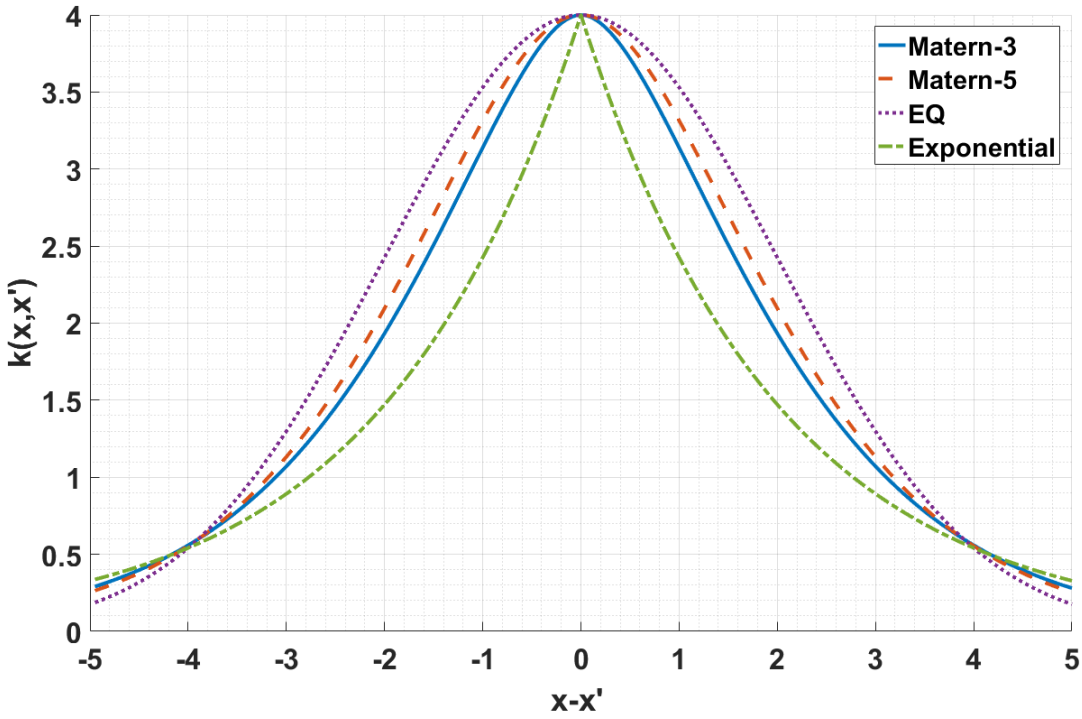


Figure 3.1: EQ, Exponential and Matern Kernels (σ=2, l=2)

To better understand the effect of hyperparameters, functions are sampled from GP priors which have covariance matrices constructed by Matern-3 kernel having different values of hyperparameters which are given in Table 3.1.

Table 3.1: $\{\sigma, l\}$ pairs

$\{2\ ,\ 0.5\}$,	$\{2\ ,\ 2\}$,	$\{2\ ,\ 15\}$
$\{5\ ,\ 0.5\}$,	$\{5\ ,\ 2\}$,	$\{5\ ,\ 15\}$

For an explicit illustration, the dimension of the input space is selected as 1. Finite number of inputs are generated with equal distance in the input space. Then, the covariance matrix in (3.2b) is constructed by calculating the value of the kernel $k(x_i, x_j)$ for $i = 1, ..., L$ and $j = 1, ..., L$. Then, the multivariate normal distribution having zero mean and the covariance matrix is sampled for multiple Monte Carlo runs. The Matern-3 kernels having these values of hyperparameters are shown in Figure 3.2.

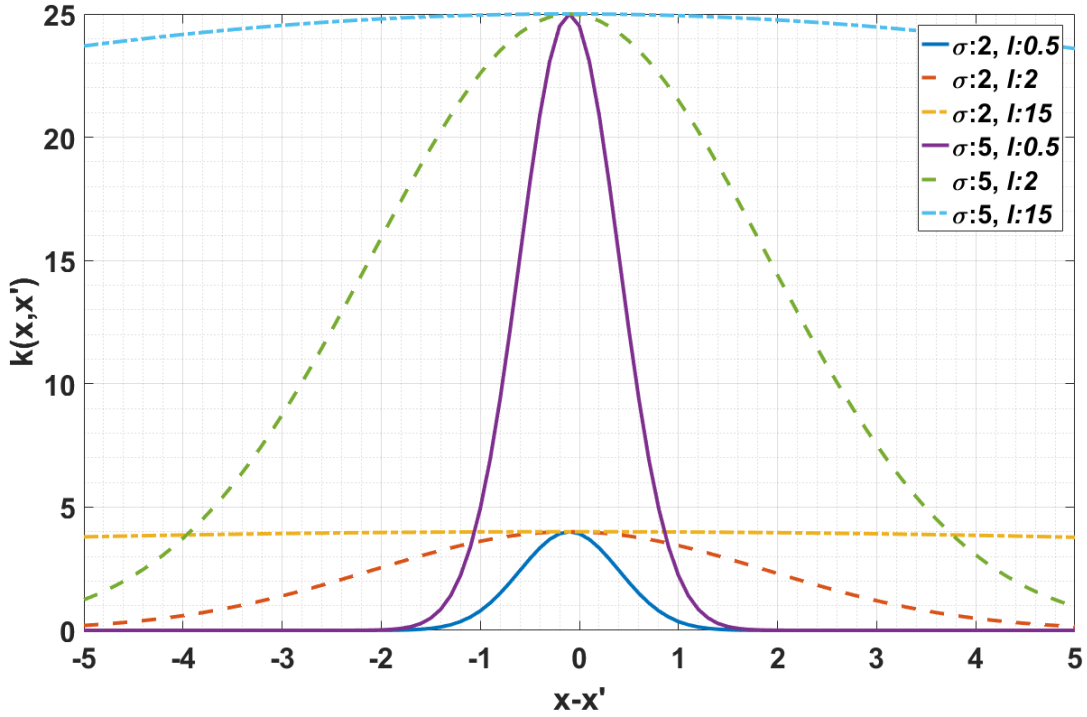


Figure 3.2: Matern-3 Kernel with various $\{\sigma, l\}$ pairs

In Figure 3.3, sample functions are shown. By comparing the figures from top to bottom, the effect of scaling parameter, σ^2, can be examined. With an increase in σ, the amplitude of the fluctuation in sample functions increases. Similarly, by comparing the figures from left to right, the effect of length scale, l, can be understood. With an increase in l, cross-correlation among function values increases and sample functions become smoother. With a small value of length scale (l=0.5), sample functions change rapidly; at the other extreme, with an excessively high value of length scale (l=15), functions become almost constant.

3.1.4 Periodic Kernel

A periodic kernel specifies the correlation based on a period of distance in addition to the closeness of input points with each other. By means of this kernel, functions

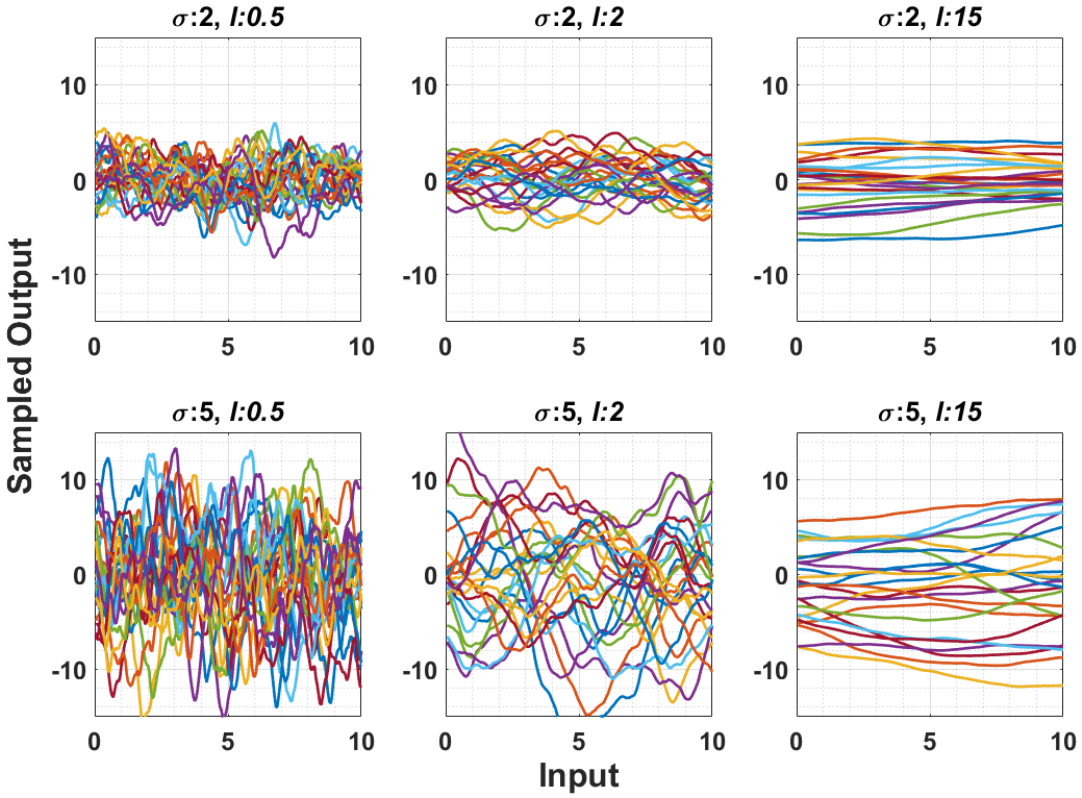


Figure 3.3: Sample functions

which repeat themselves can be generated. A periodic kernel can be defined as

$$k(r) = \sigma^2 \exp\left(-\frac{2}{l^2}\sin^2\left(\frac{\pi r}{p}\right)\right), \tag{3.7}$$

where p is the period. Being able to represent periodic functions makes GP a very powerful tool. For instance, Wahlström and Özkan [35] utilize this functionality for learning the radial extent (which is periodic) of symmetric vehicles.

3.1.5 Linear Kernel

A linear kernel is used to model linear functions. In fact, it corresponds to Bayesian linear regression in a less efficient way [10]. However, this kernel is favorable especially when it is combined with other kernels such as EQ or a periodic which is

explained in Section 3.1.6. Linear kernel is defined as

$$k(x, x') = \sigma^2(x - c)(x' - c), \tag{3.8}$$

where c is an input offset. The offset determines the point on x coordinate that all sample functions go through. Linear kernel is not stationary and the value of the kernel is determined by input points rather than the relative positions. Sample functions are generated for different values of σ by setting the value of c to 2 and illustrated in Figure 3.4. Notice that all sample functions exactly pass through the point (2,0). Furthermore, it is seen that an increase in σ results in more scattered sample functions.

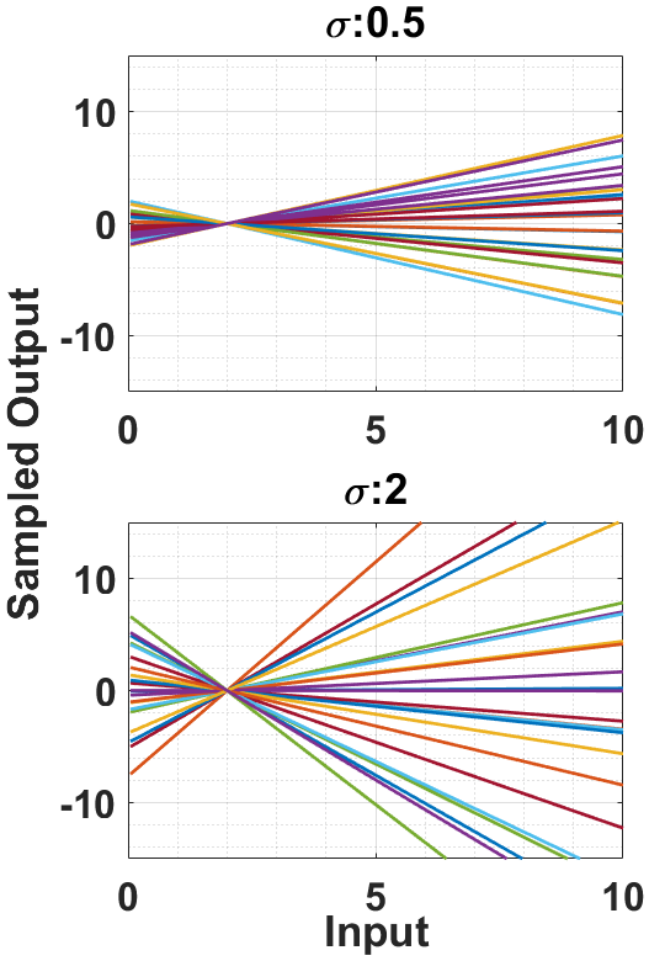


Figure 3.4: Functions sampled from a linear kernel ($\sigma_d = 0$)

Linear kernels are so strict that functions are compelled to pass through the point (c, 0). To bring flexibility to this model, this kernel is used with an additional constant

variance term, σ_d^2, which determines how much the model differs from linear model by specifying a prior on it [10]. In that case, the kernel becomes

$$k(x, x') = \sigma_d^2 + \sigma^2(x - c)(x' - c). \tag{3.9}$$

Sample functions, generated with the new form of the kernel by taking σ_d as 1, are shown in Figure 3.5.

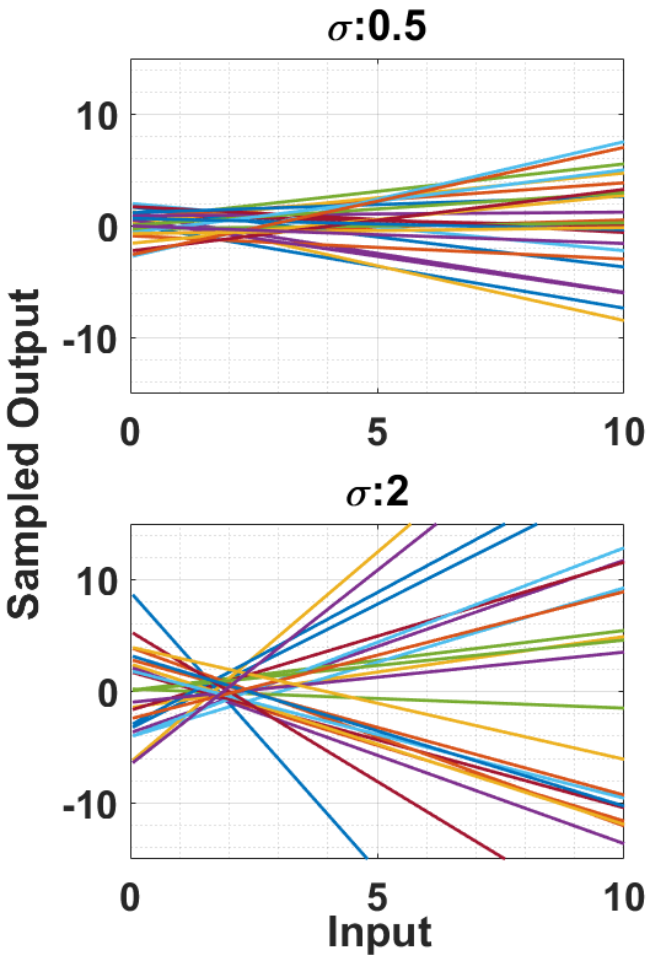


Figure 3.5: Functions sampled from a linear kernel ($\sigma_d = 1$)

3.1.6 Combining Different Kernels

Completely new kernels can be built by combining different classes of kernels. By doing so, the model can utilize the features of different kernels at the same time and more complex systems can be conveniently modeled by means of this combination. Most preferred methods of combining the kernels are summation and multiplication of two or more. These methods will be explained in details with some examples:

- **Summation:**

  The summation of two kernels $k_a(x, x')$ and $k_b(x, x')$ results in a valid kernel $k_s(x, x')$. In general, any number of kernels can be summed in this way to combine their features. The kernels which are summed can be at different forms or at the same form with different hyperparameters. As an example, the sum of a linear and a periodic kernel and sample functions, drawn from a GP prior having a covariance matrix constructed by this new kernel, are depicted in Figure 3.6.

  The periodic kernel's hyperparameters are chosen as σ=1, l=2, p=2 while the linear kernel's ones are σ=0.5, σ_d=0.5, c=2. Please note that the sample functions are locally periodic with a linear rate of change in general.

- **Multiplication:**

  Similar to summation, multiplication of two or more kernels yields a valid kernel $k_m(x, x')$.

  One of the functionalities of multiplication is to restrict the support of a kernel by multiplying with a kernel whose support is narrower. For instance, EQ, Exponential or Matern kernels are all local kernels which specify the correlation based on closeness of two input points; whereas periodic kernel specifies periodicity all over the input space. Multiplying a periodic kernel by a local kernel produces local periodic characteristic. Moreover, characteristic of a kernel can be leveraged as input points move away from input offset c. This can be achieved by multiplication of a kernel by linear kernel function, since absolute

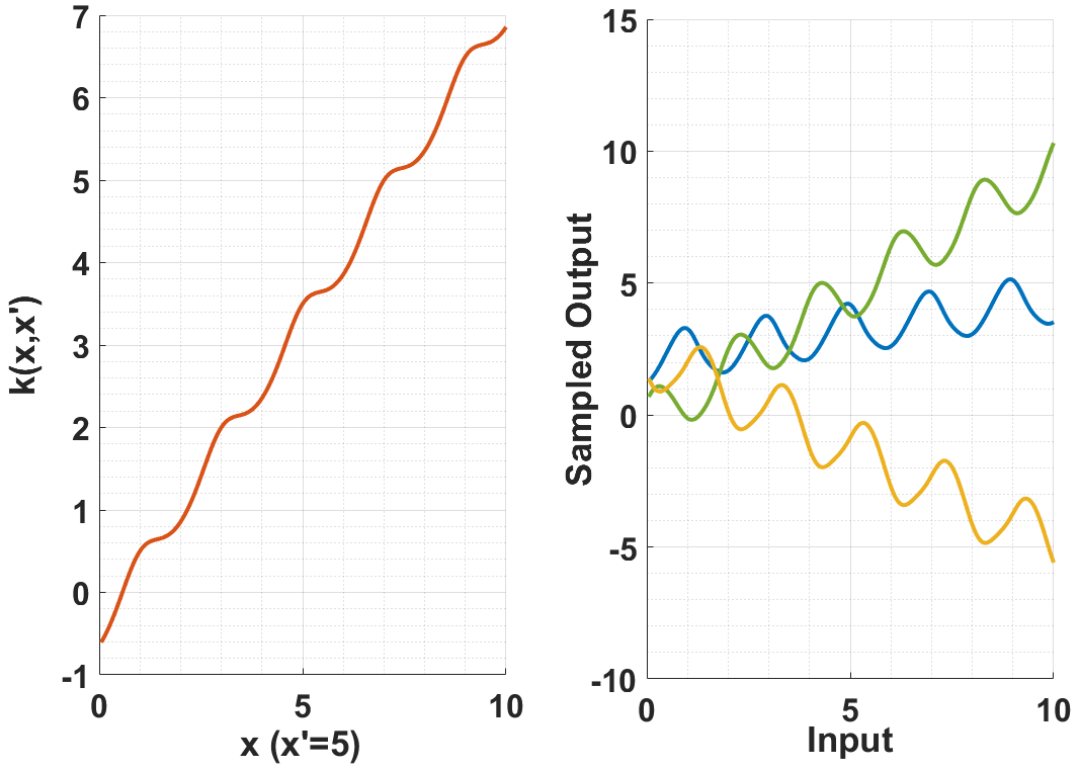


Figure 3.6: A linear + a periodic kernel and sample functions

value of linear kernel grows away from the point c. These applications of kernel multiplication are illustrated in Figures 3.7 and 3.8, respectively.

Figure 3.7 is produced with kernel parameters of σ=5, l=2 and p=0.5 for periodic kernel and σ=5 and l=2 for EQ. The kernel form implies that the correlation decreases as the distance between input points increases with the help of EQ and locally periodic by means of a periodic kernel. Besides, sample functions are locally periodic and their characteristics slowly changes thanks to EQ.

Figure 3.8 is produced with kernel parameters of σ=1, l=2 and p=0.5 for periodic kernel and σ=1, σ_d=0.5 and c=2 for linear kernel. As the input moves away from point c, the amplitude of the periodicity in sample functions grows.

It should be noted that there are numerous kernel forms and plenty of hyperparameters to be set especially if the input dimension is high and the kernel is non-isotropic. Although some of these kernels can be considered similar with

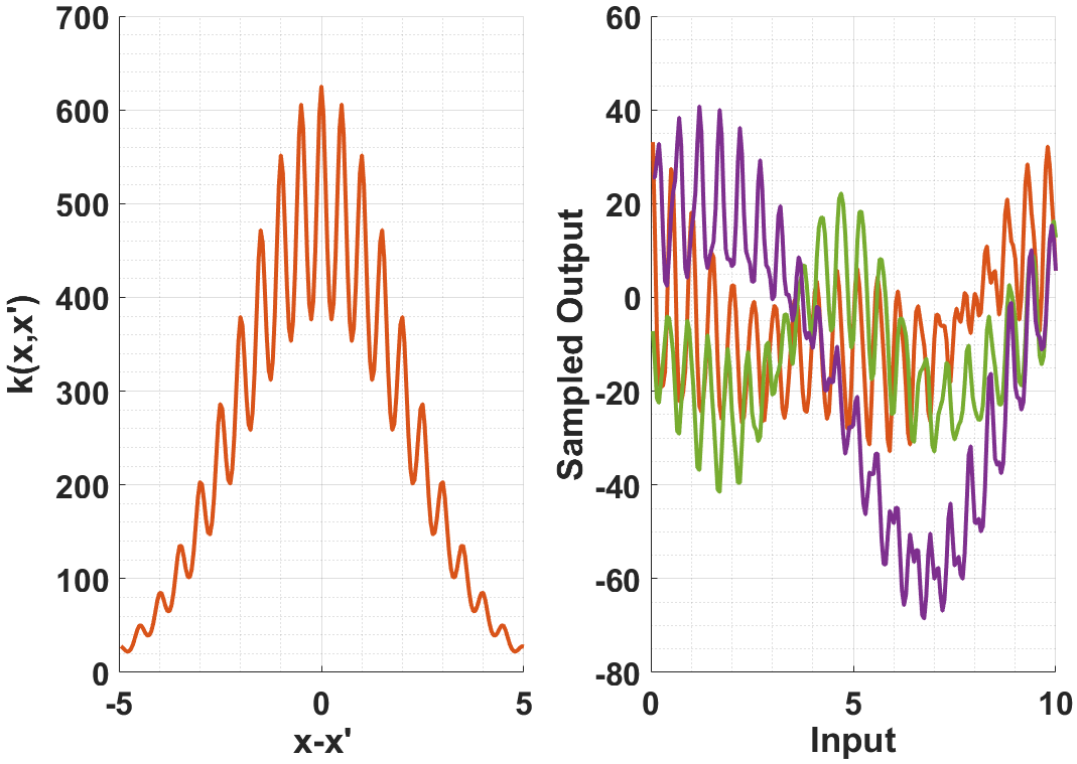


Figure 3.7: A periodic $\times$ EQ kernel and sample functions

respect to their functional forms (3.1), their effects on GP prior substantially differ. Furthermore, the characteristic of the modeled function, varies significantly with the different values of the hyperparameters of the same kernel. Moreover, more complex and specialized kernels can be developed as introduced at Section 3.1.6 and there are also other methods of combining different kernels which are not mentioned in this thesis, such as defining different kernels for different pieces of input space known as combination by changepoints method [10]. Because of this diversity, determining the most appropriate kernel and designating its hyperparameters is an essential but challenging task in Gaussian process modeling. Later, this task will be discussed in details at Section 3.3.1.

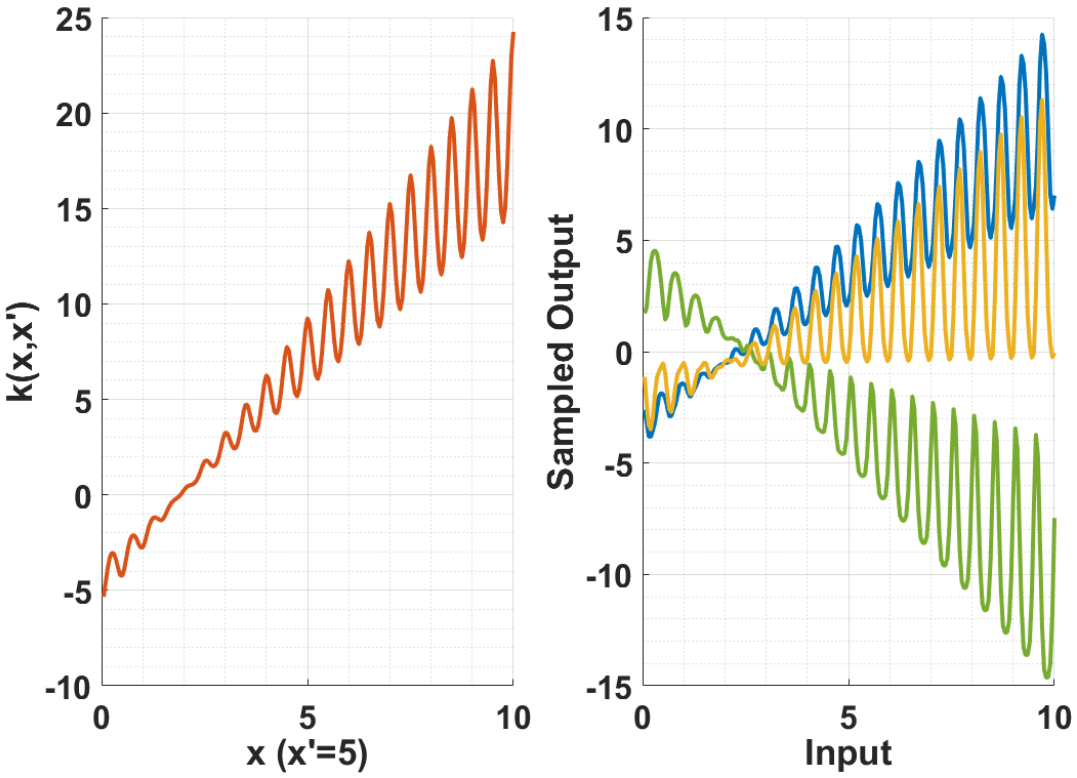


Figure 3.8: A linear $\times$ a periodic kernel and sample functions

3.2 Gaussian Process Regression

An observation y can be formulated as a deviation from a GP model with an additive noise such that

$$y = f(x) + e, \quad e \sim \mathcal{N}(0, \sigma_r^2). \tag{3.10}$$

The primary objective of a GP model is to estimate the function values evaluated at some inputs considering the observations. Let's denote the values of the latent function, which are intended to be estimated, as $\mathbf{f} \triangleq [f(x_1^{\mathbf{f}}) \ \dots \ f(x_T^{\mathbf{f}})]^\top$ at the test inputs $\mathbf{x}^{\mathbf{f}} \triangleq [x_1^{\mathbf{f}} \ \dots \ x_T^{\mathbf{f}}]^\top$. For that purpose, a set of measurements represented by $\mathbf{y} \triangleq [y_1 \ \dots \ y_L]^\top$ are collected from the latent function with corresponding inputs $\mathbf{x} \triangleq [x_1 \ \dots \ x_L]^\top$. By using (3.2) and (3.10), the joint distribution of the measure-

ments and the function values at the test points can be written as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{x},\mathbf{x}) + R & K(\mathbf{x},\mathbf{x}^{\mathbf{f}}) \\ K(\mathbf{x}^{\mathbf{f}},\mathbf{x}) & K(\mathbf{x}^{\mathbf{f}},\mathbf{x}^{\mathbf{f}}) \end{bmatrix}\right), \tag{3.11a}$$

where R is $L \times L$ diagonal matrix whose diagonal elements are σ_r^2s and

$$K(\mathbf{x},\mathbf{x}^{\mathbf{f}}) = \begin{bmatrix} k(x_1, x_1^{\mathbf{f}}) & \dots & k(x_1, x_T^{\mathbf{f}}) \\ \vdots & & \vdots \\ k(x_L, {x_1}^{\mathbf{f}}) & \dots & k(x_L, x_T^{\mathbf{f}}) \end{bmatrix}. \tag{3.11b}$$

Notice that, the underlying GP model given in (3.11) is selected to have a zero-valued mean function.

From the joint distribution, $p(\mathbf{y},\mathbf{f})$, the conditional distribution $p(\mathbf{f}|\mathbf{y})$ can be computed as

$$p(\mathbf{f}|\mathbf{y}) \sim \mathcal{N}(S\mathbf{y}, P), \tag{3.12a}$$

$$S = K(\mathbf{x}^{\mathbf{f}},\mathbf{x}){K_{\mathbf{y}}}^{-1}, \tag{3.12b}$$

$$P = K(\mathbf{x}^{\mathbf{f}},\mathbf{x}^{\mathbf{f}}) - SK(\mathbf{x},\mathbf{x}^{\mathbf{f}}), \tag{3.12c}$$

$$K_{\mathbf{y}} = K(\mathbf{x},\mathbf{x}) + R. \tag{3.12d}$$

In the study, smoothed ballistic parameter and Mach number estimates are used as observation and input to GP, respectively. It enables the estimation of the BP value by using predicted Mach number as a test point on impact point prediction phase.

3.3 Further Discussions on Gaussian Processes

3.3.1 Kernel Selection and Hyperparameter Optimization

Broad set of kernels used in Gaussian Process models are introduced in Section 3.1. In this section, determination of the form and hyperparameters of kernel function based on a training data set is discussed in details. In most of the real world applications, choosing the most appropriate kernel for the phenomena and determining its hyperparameters are challenging problems. The challenge stems from the fact that there are

numerous kernels and plenty of hyperparameters that need to be set. Whereas some parameters of a kernel are easy to specify based on the characteristics of the modeled phenomenon, some parameters are hard to be set since these parameters cannot be easily associated with the modeled function. For example, stationarity of the kernel function can be easily determined by considering the phenomenon; however, determining the values of the length scale parameters is troublesome, especially when the input dimension is high.

In the literature, approaches to the kernel selection problem can be categorized in three classes and they are based on one of the followings: the probability of the model given the training data, the prediction error and bounding the prediction error [27].

In this study, evaluation and comparison of different kernels and their hyperparameters are done in a Bayesian framework. The probability of the model given the data is computed in Section 3.3.1.1 and Maximum Likelihood Estimation (MLE) is utilized to determine the optimal kernel and its hyperparameters in Section 3.3.1.2.

3.3.1.1 Calculation of Marginal Likelihood

For an observation set $\mathbf{y}$ and the corresponding input set $\mathbf{x}$, the marginal likelihood of the observations can be formulated in logarithmic form as

$$\log(p(\mathbf{y}|\mathbf{x},\Theta)) = -\frac{1}{2}\mathbf{y}^\top K_\mathbf{y}^{-1}\mathbf{y} - \frac{1}{2}\log(|K_\mathbf{y}|) - \frac{N}{2}\log(2\pi), \tag{3.13}$$

where $|\cdot|$ stands for the determinant of its input argument, $K_\mathbf{y}$ is the covariance matrix of the observation set calculated in 3.12d, N is the input dimension and Θ is the set of hyperparameters. The terms in the log likelihood given in (3.13) can be interpretable as follows: The first term, $-\frac{1}{2}\mathbf{y}^\top K_\mathbf{y}^{-1}\mathbf{y}$, is the only term which involves observations and it qualifies how well the observations fit the model. The second term, $-\frac{1}{2}log(|K_\mathbf{y}|)$, depends on the inputs and the kernel of the model and it is the negative of the complexity penalty. The last term is a normalization constant and depends only on the dimension of the problem. Evolution of the first two terms as the

length scale of the kernel changes is depicted in Figure 3.9 to understand the terms more clearly.

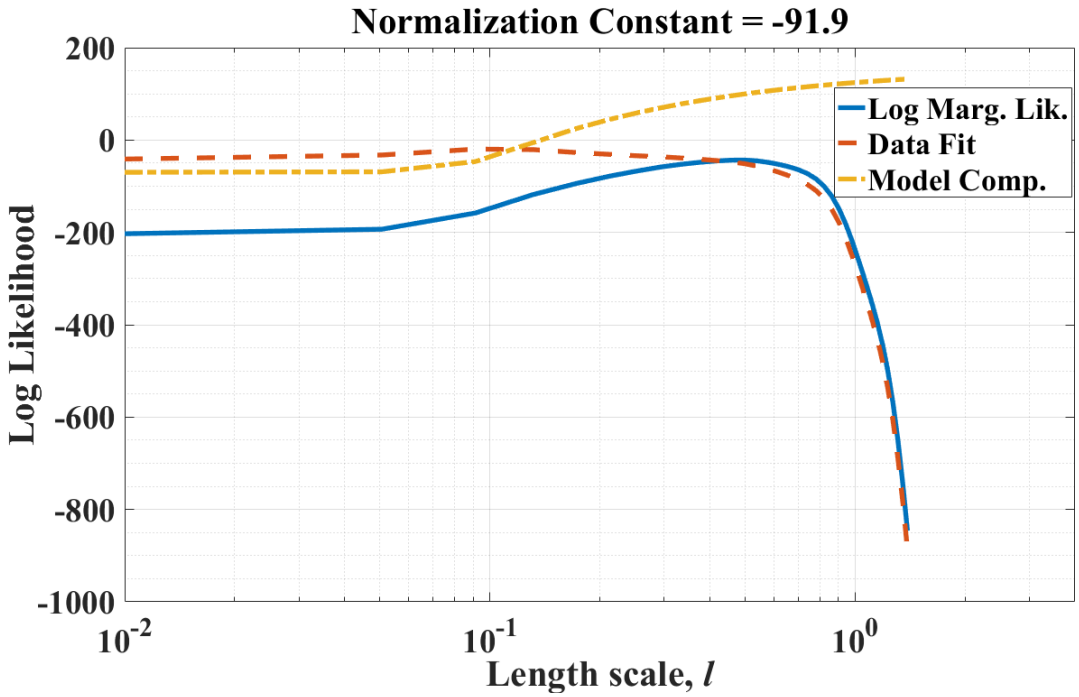


Figure 3.9: Log marginal likelihood and its constituent terms

Log marginal likelihood given in Figure 3.9 belongs to the data sampled from a GP having a covariance matrix constructed by EQ kernel with the hyperparameters of $\sigma = 2$ and $l = 0.5$ with an additive observation noise having a standard deviation of $\sigma_d = 0.2$. While generating Figure 3.9, σ and σ_d are kept at their reference values. In the figure, the marginal likelihood reaches its maximum at $l_{opt} = 0.497$ which is very close to the reference value. The data fit term decreases with increasing length scale because the model loses its flexibility to fit the data. Moreover, the negative of the complexity penalty increases with the length scale, since the model loses its complexity as the length scale increases. For the values of the length scale greater than the optimal value, the marginal likelihood decays rapidly, because as the length scale grows, the model becomes so strict that it loses its ability to model the data.

3.3.1.2 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) is a technique used for estimating the parameters of a given distribution via maximizing the likelihood function, so that under the assumed statistical model the observed data is most probable. To use a gradient-based optimization technique, the partial derivative of the log marginal likelihood with respect to any hyperparameter Θ_j is obtained as

$$\frac{\partial}{\partial \Theta_j} log(p(\mathbf{y}|\mathbf{x}, \Theta)) = \frac{1}{2}\mathbf{y}^\top K_\mathbf{y}^{-1} \frac{\partial K_\mathbf{y}}{\partial \Theta_j} K_\mathbf{y}^{-1}\mathbf{y} - \frac{1}{2} tr\Big(K_\mathbf{y}^{-1}\frac{\partial K_\mathbf{y}}{\partial \Theta_j}\Big), \tag{3.14a}$$

$$= \frac{1}{2} tr\Big((\boldsymbol{\gamma}\boldsymbol{\gamma}^\top - K_\mathbf{y}^{-1})\frac{\partial K_\mathbf{y}}{\partial \Theta_j}\Big), \tag{3.14b}$$

where $tr(\cdot)$ stands for the trace of its input argument and $\boldsymbol{\gamma} = K_\mathbf{y}^{-1}\mathbf{y}$ [27].

Rasmussen and Williams state that the marginal likelihood may suffer from multiple local optima. However, in practice, with large data sets, the global optimum is of the orders of magnitude more probable than any other local optima. In general, using a local optimum may result in an undesired representation of the data, hence it should be avoided [27].

3.3.2 Some Approximations of GP

As mentioned earlier, GP provides an elegant way of realizing Bayesian nonparametric regression. However, its non-parametric nature gives rise to significant computational problems especially for large training sets. More specifically, the training procedure scales with N^3 where N is the number of the data points. This is simply due to the inversion of the covariance function performed to determine the hyperparameters maximizing the marginal likelihood. Besides, the predictive distribution for a single test input has $\mathcal{O}(N^2)$ computational complexity which is dominated by the calculation of the predictive variance. The storage demands of prediction, on the other hand, scale with $\mathcal{O}(N^2)$ which might hinder the utilization of GP models with large data sets.

To ease these unfavourable computational aspects of GP regression, there has been developed a great body of literature tackling with the issues from various angles. The first branch of the studies suggests algebraic improvements to the conventional settings and investigates approximations to the matrix inversion included in the predictive covariance [27]. Another approach suggests to partition the complete data set into subsets and then to learn local models for each subset [26]. The outputs of the local models are combined to obtain the whole resulting prediction. An alternative line of algorithms relies on the selection of a subset of the complete training set [31], [6], [18]. These, however, generally lack a systematic solution to reliably determine the hyper-parameters as the selection of the data set interferes with the correct values for these parameters [31]. Considering the difficulties engaged with the mentioned approaches, we prefer to employ a pseudo-input method named Sparse Pseudo-input Gaussian Processes (SPGP) which was first proposed in [32].

3.3.2.1 Sparse Pseudo-input Gaussian Processes

SPGP essentially provides an efficient approximate GP regression based on the joint optimization of the active input point locations and the hyper-parameters by a gradient based optimization scheme. The complexity of regression with SPGP for each test case is $\mathcal{O}(M^2)$ where M is the number of pseudo data points. Before proceeding to the details of the sparse model, let us rewrite the predictive density with a slightly modified notation then the one utilized in Section 3.2.

$$p(y|\mathbf{x}, \mathcal{D}, \boldsymbol{\theta}) = \mathcal{N}\left(y|\mathbf{k}_{\mathbf{x}}^{\top}\left(\mathbf{K}_N + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y}, K_{\mathbf{xx}} - \mathbf{k}_{\mathbf{x}}^{\top}\left(\mathbf{K}_N + \sigma^2\mathbf{I}\right)^{-1}\mathbf{k}_{\mathbf{x}} + \sigma^2\right) \tag{3.15}$$

$\mathcal{D}$ is the data set with input vectors $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ and corresponding measurements $\mathbf{y} = \{y_n\}_{n=1}^N$. The covariance matrices are defined as follows: $[\mathbf{k}_{\mathbf{x}}]_{\boldsymbol{n}} = K(\mathbf{x}_{\boldsymbol{n}}, \mathbf{x})$ and $K_{\mathbf{xx}} = K(\mathbf{x}, \mathbf{x})$ where $K(\cdot, \cdot)$ is the kernel or covariance function.

Notice that the likelihood given in (3.15) has mean and variance which are both functions of the test point, $\mathbf{x}$. Additionally, these functions could be interpreted to be parameterized by the training data set, $\mathbf{X}$ and $\mathbf{y}$. SPGP proposes a probabilistic model

considering this likelihood to be parameterized by a pseudo data set, $\bar{\mathcal{D}}$. As the number of points included in the pseudo data set, M, is selected to be smaller than that of the original data set, it provides an effective way of regression. $\overline{\mathbf{X}} = \{\overline{\mathbf{x}}_m\}_{m=1}^{M}$ and $\overline{\mathbf{f}} = \{\overline{f}_m\}_{m=1}^{M}$ indicate the input and output pairs in the pseudo data set, respectively. Regarding this new set, one can write the predictive distribution as

$$p(y|\mathbf{x}, \overline{\mathbf{X}}, \overline{\mathbf{f}}) = \mathcal{N}\left(y|\mathbf{k}_{\mathbf{x}}^{\top}\mathbf{K}_M^{-1}\overline{\mathbf{f}}, K_{\mathbf{xx}} - \mathbf{k}_{\mathbf{x}}^{\top}\mathbf{K}_M^{-1}\mathbf{k}_{\mathbf{x}} + \sigma^2\right) \tag{3.16}$$

where $[\mathbf{K}_M]_{mm'} = K\left(\overline{\mathbf{x}}_m, \overline{\mathbf{x}}_{m'}\right)$ and $[\mathbf{k}_{\mathbf{x}}]_m = K\left(\overline{\mathbf{x}}_m, \mathbf{x}\right)$, for $m = 1, ..., M$.

Subsequently, the likelihood of the observed data set can be written as

$$p(\mathbf{y}|\mathbf{X}, \overline{\mathbf{X}}, \overline{\mathbf{f}}) = \prod_{n=1}^{N} p\left(y_n|\mathbf{x}_n, \overline{\mathbf{X}}, \overline{\mathbf{f}}\right) = \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{NM}\mathbf{K}_M^{-1}\overline{\mathbf{f}}, \mathbf{\Lambda} + \sigma^2\mathbf{I}\right) \tag{3.17}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda}), \lambda_n = K_{nn} - \mathbf{k}_n^{\top}\mathbf{K}_M^{-1}\mathbf{k}_n,$ and $[\mathbf{K}_{NM}]_{nm} = K\left(\mathbf{x}_n, \overline{\mathbf{x}}_m\right)$

By selecting a Gaussian prior on the pseudo targets, i.e., $p(\overline{\mathbf{f}}|\overline{\mathbf{X}}) = \mathcal{N}\left(\overline{\mathbf{f}}|\mathbf{0}, \mathbf{K}_M\right)$, the posterior distribution is obtained as

$$p(\overline{\mathbf{f}}|\mathcal{D}, \overline{\mathbf{X}}) = \mathcal{N}\left(\overline{\mathbf{f}}|\mathbf{K}_M\mathbf{Q}_M^{-1}\mathbf{K}_{MN}\left(\mathbf{\Lambda} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y}, \mathbf{K}_M\mathbf{Q}_M^{-1}\mathbf{K}_M\right) \tag{3.18}$$

where $\mathbf{Q}_M = \mathbf{K}_M + \mathbf{K}_{MN}\left(\mathbf{\Lambda} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{K}_{NM}$.

Hence for a new test point the predictive distribution can be attained as

$$p\left(y_*|\mathbf{x}_*, \mathcal{D}, \overline{\mathbf{X}}\right) = \int \mathrm{d}\overline{\mathbf{f}} p\left(y_*|\mathbf{x}_*, \overline{\mathbf{X}}, \overline{\mathbf{f}}\right) p(\overline{\mathbf{f}}|\mathcal{D}, \overline{\mathbf{X}}) = \mathcal{N}\left(y_*|\mu_*, \sigma_*^2\right) \tag{3.19}$$

where $\mu_* = \mathbf{k}_*^{\top}\mathbf{Q}_M^{-1}\mathbf{K}_{MN}\left(\mathbf{\Lambda} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y}$ and $\sigma_*^2 = K_{**} - \mathbf{k}_*^{\top}\left(\mathbf{K}_M^{-1} - \mathbf{Q}_M^{-1}\right)\mathbf{k}_* + \sigma^2$. Notice that the inversion of $(\mathbf{\Lambda} + \sigma^2\mathbf{I})$ does not impose a computational problem as it is a diagonal matrix. By this formulation, the complexity of a prediction is set to be $\mathcal{O}(M^2)$ (by utilizing some precomputations), [32].

Finally, the last step is to find the pseudo-input locations, $\overline{\mathbf{X}}$, and the hyper-parameters, $\mathbf{\Theta} = \{\boldsymbol{\theta}, \sigma^2\}$. To this end, the following form of marginal likelihood is maximized with respect to $\overline{\mathbf{X}}$ and $\mathbf{\Theta}$ using a gradient-based optimization scheme.

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \overline{\mathbf{X}}, \mathbf{\Theta}) &= \int \mathrm{d}\overline{\mathbf{f}} p(\mathbf{y}|\mathbf{X}, \overline{\mathbf{X}}, \overline{\mathbf{f}}) p(\overline{\mathbf{f}}|\overline{\mathbf{X}}) \\ &= \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}_{\mathrm{NM}}\mathbf{K}_M^{-1}\mathbf{K}_{MN} + \mathbf{\Lambda} + \sigma^2\mathbf{I}\right) \end{aligned} \tag{3.20}$$

CHAPTER 4

TRACKING AND SMOOTHING FOR BALLISTIC OBJECTS

This chapter provides detailed information about the ballistic target tracking (BTT) and construction of smoothed state estimates and covariance matrices. It begins with the motion and measurement models utilized in BTT with the emphasize on ballistic coefficient. Effective forces captured by the motion model are discussed and utilized Earth and gravity models are introduced by discussing the alternatives. Then, the state estimation is described. We conclude this chapter by introducing smoothed state estimates and their covariance matrices which are obtained by a Rauch, Tung and Striebel (RTS) Smoother.

4.1 Ballistic Target Tracking Method

A tracking algorithm aims to extract the maximum information about the target state from available observations. To achieve this purpose, mathematical models of target's kinematics and sensor are utilized. A tracking algorithm is responsible for jointly using the target motion and sensor models to provide accurate state estimates.

4.1.1 Motion Model

A motion model of a ballistic object is to be derived first to utilize in the tracking algorithm. Let the state vector be

$$\mathbf{x}_k \triangleq \begin{bmatrix} \mathbf{p}_k \\ \mathbf{v}_k \\ \alpha_k \end{bmatrix}, \tag{4.1a}$$

$$\mathbf{p}_k = \begin{bmatrix} p_k^x \\ p_k^y \\ p_k^z \end{bmatrix}, \quad \mathbf{v}_k = \begin{bmatrix} v_k^x \\ v_k^y \\ v_k^z \end{bmatrix}, \tag{4.1b}$$

where the position $\mathbf{p}_k$ and the velocity $\mathbf{v}_k$ at time k are expressed in 3D local Cartesian coordinates which its origin is at the sensor location and α_k denotes the ballistic parameter (BP) at time k that is inversely proportional to the BC. BP is a slowly changing parameter and its exact dynamics cannot be expressed in a closed form. In most studies, a crude approximation is made by assuming BP to be a random walk throughout the entire flight. In that sense, the state space model (SSM) can be formulated in discrete form as

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + G\mathbf{a}_k(\mathbf{x}_k) + B\mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(0, Q_k), \tag{4.2a}$$

$$F \triangleq \begin{bmatrix} I_{3\times3} & TI_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & 1 \end{bmatrix}, \quad G \triangleq \begin{bmatrix} \frac{T^2}{2}I_{3\times3} \\ T_{3\times3} \\ 0_{1\times3} \end{bmatrix}, \quad B \triangleq \begin{bmatrix} G & 0_{6\times1} \\ & 1 \end{bmatrix}, \tag{4.2b}$$

where T is the update period, $\mathbf{a}_k(\mathbf{x}_k)$ is the acceleration at time k and $\mathbf{w}_k$ is white-Gaussian process noise with covariance matrix Q_k. This model can be interpreted as a variant of the well-known constant velocity model when the acceleration is regarded to be an input to the system.

In the motion model described in (4.2), acceleration at time k can be computed by

considering the net effective force acting on the object as

$$\mathbf{a}_k = \frac{F_k^{net}}{m} \tag{4.3}$$

where m is the mass of the object. The effective forces on the ballistic objects are discussed in Section 2.2.1 in details. BTT introduced here is intended to be applicable against ballistic targets which stay within atmosphere throughout their entire flight. Consequently, the aerodynamic drag force is effective throughout the whole BTT. Furthermore, the prospective projectiles have small total yaw angle along most of their trajectories; as a result, assuming the total yaw angle to be zero is feasible. Under this assumption, lift and magnus forces mentioned in Section 2.2.1 as well as overturning and magnus moments mentioned in Section 2.2.2 become negligible. Moreover, the prospective projectiles are assumed not to have any propellant which accelerates the projectile during the boost phase of their flight. For thrusting projectiles, on the other hand, it is assumed that the burn-out occurs before the BTT begins. This is a feasible assumption, since the boost phase of the thrusting projectiles generally lasts in a short period of time. Thus, thrust force is assumed to be zero. Under these assumptions, the net force acting on the projectile is determined by the drag and gravitational forces only. By using (2.1), the net force can be written at time k as

$$F_k^{net} = F_k^{drag} + F_k^{gravity}, \tag{4.4a}$$

$$= -\frac{1}{2}\rho S c_d(\mathbf{v}_k) \left\|\mathbf{v}_k\right\| v_k + m\mathbf{g}, \tag{4.4b}$$

where $\mathbf{g}$ is gravitational acceleration.

Gravitational force can be modeled in different ways depending on the Earth model which induces gravity. Most common alternatives of earth models are summarized below.

- **Flat Earth Model:** Flat Earth model considers the Earth as flat and non-rotating [11]. In the flat Earth model, gravity is always in the direction of a down component of a local frame. Depending on the application, it may vary

with the height of the object or can be assumed to be constant if the height of the object is not too high.

- **Spherical Earth Model:** Spherical Earth model considers the Earth as a sphere by neglecting the oblateness of the Earth. The model is usually used to take the curvature of the Earth into account without increasing the model complexity too much. In spherical Earth model, gravity always directs towards the center of the Earth and usually varies with the altitude of the object of interest. The variation with the altitude is usually performed by Newton's law of universal gravitation which is

$$\mathbf{g} = -\frac{\mu}{\|\mathbf{r}\|^3}\mathbf{r}, \tag{4.5}$$

  where μ is the standard gravitational parameter of the Earth and $\mathbf{r}$ is the vector from the center of the Earth to the object of interest.

- **Ellipsoidal Earth Model:** Ellipsoidal Earth model considers the Earth as an oblate spheroid which is an ellipsoid with two equal semi-diameters and it is wider at its equatorial plane than along its axis of rotational symmetry. The ellipsoid exhibits rotational symmetry about the axis passing through its poles and mirror symmetry over the equatorial plane [16]. The direction of the gravity does not point to the center of the Earth; instead, it directs perpendicular to the ellipsoid at any point. Its magnitude varies with the altitude of the object as well as the latitude of the object due to the oblateness of the Earth. There are standard ellipsoid models which defines a standard on the geometry of the Earth, gravity and the rotation of the Earth. World Geodetic System of 1984 (WGS-84) is the most commonly used model among many [23]. For air applications, it is standard practice to use an empirical model of the surface gravity and apply a simple scaling law to calculate the variation with altitude [16]. The WGS-84 provides a simple model of the gravity at the ellipsoid as a function of latitude.

- **A Geoid Model:** Unlike the ellipsoidal Earth model's assumption, the real world is not homogeneous. There are mass differences between lands and

oceans. Moreover, even under the surface, there is inhomogeneous material distribution. This mass differences and inhomogeneities result in variations in the gravity field. To account for these variations, a constant gravity potential surface is defined around the physical surface of the Earth. On the defined surface, magnitude of the gravity is equal and gravity is perpendicular to the surface at all points. This definition of the Earth surface is called the geoid. As the Earth's gravity field varies with location, the geoid can differ from the ellipsoid by up to 100 m. The geoid also represents mean sea level and the gravity is calculated according the height from mean sea level.

There are standard geoid models as well, where the most common and up to date is Earth Gravitational Model 2008 (EGM 08) which defines the geoid height (from the surface of the ellipsoid) and gravitational potential as a spherical harmonic function of latitude and longitude.

Within the context of this thesis, ballistic targets which exhibit endo-atmospheric flight are the main concern. Maximum range of endo-atmospheric ballistic targets are approximately less than 50 km. For a spherical Earth model, height of an object changes around 200 m at range of 50 km due to the curvature of the Earth. At these ranges, curvature of the Earth can be neglected. Furthermore, variation in the magnitude of the gravity with the altitude can be neglected for an endo-atmospheric flight. The objects in endo-atmospheric flight reach up to 30 km of altitude. When the magnitude of gravity is assumed to be 9.81 m/s^2 at the surface of the Earth and the radius of the Earth is assumed to be 6400 km, the magnitude of gravity at altitude of 30 km is around 9.72 m/s^2 according to (4.5). Under these assumptions, flat Earth model is utilized and gravity is modeled as constant throughout the trajectory such that

$$\mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} m/s^2 \tag{4.6}$$

in a North-East-Down (NED) local frame.

While modeling the drag force, BC which is a measure of an object's ability to over-

come air resistance in flight is defined as

$$\beta(\mathbf{v}_k) = \left(\frac{sc_d(\mathbf{v}_k)}{m}\right)^{-1}, \tag{4.7}$$

where β stands for BC. In this work, we introduce a new parameter, called ballistic parameter (BP), so that the resulting state-space representation will easily lend itself to the filtering techniques based on linearization of the system dynamics such as EKF. The definition of the BP is

$$\alpha_k(\mathbf{v}_k) = \frac{1}{A\beta(\mathbf{v}_k)}. \tag{4.8}$$

where α_k denotes the BP at time k. In this equation, A is called the numerical stabilizer and it ensures that the values of BP is at the same order of magnitude with other state variables since it is augmented into the state vector to be estimated. The dynamic of BP is modeled as a random walk in state-space representation as given in (4.2).

By combining (4.4) with (4.6)-(4.8), acceleration of the projectile at time k can be obtained as

$$\mathbf{a}_k(\mathbf{x}_k) = -\frac{1}{2}\rho(p_k^z)A\alpha_k(\mathbf{v}_k)\left\|\mathbf{v}_k\right\|\mathbf{v}_k + \mathbf{g}. \tag{4.9}$$

We further model the air density, $\rho(p_k^z)$, as exponentially decreasing with altitude of the projectile such that

$$\rho(p_k^z) = 1.227e^{1.0931\times10^{-4}p_k^z}, \tag{4.10}$$

where p_k^z is the negative of the altitude of the projectile. By plugging $\mathbf{a}_k(\mathbf{x}_k)$ into (4.2a), the dynamical model can be finally obtained as

$$\mathbf{x}_{k+1} = \underbrace{F\mathbf{x}_k + G\Big(-\frac{1}{2}\rho(p_k^z)A\alpha_k(\mathbf{v}_k)\left\|\mathbf{v}_k\right\|\mathbf{v}_k\Big) + G\mathbf{g}}_{\mathbf{f}(\mathbf{x}_k)} + B\mathbf{w}_k. \tag{4.11}$$

4.1.2 Measurement Model

The measurement equation of our state space model is given by

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\nu}_k, \qquad \boldsymbol{\nu}_k \sim \mathcal{N}(0, R_k), \tag{4.12}$$

where ν_k is white-Gaussian measurement noise with covariance matrix R_k. As a common practice, the measurement sensor is assumed to be a 3D radar which acquires the position of the target in spherical coordinates. Hence, the measurement vector, $\mathbf{z}_k$ is comprised of range, azimuth and elevation of the target,

$$\begin{bmatrix} z_k^R \\ z_k^\theta \\ z_k^\phi \end{bmatrix} = \begin{bmatrix} \sqrt{p_k^{x2} + p_k^{y2} + p_k^{z2}} \\ \tan^{-1}\left(\frac{p_k^y}{p_k^x}\right) \\ \sin^{-1}\left(\frac{p_k^z}{\sqrt{p_k^{x2}+p_k^{y2}}}\right) \end{bmatrix} + \boldsymbol{\nu}_k. \tag{4.13}$$

As a result, both the motion and measurement equations are non-linear and they must be handled properly at inference.

4.1.3 Inference via Extended Kalman Filter (EKF)

Considering the nonlinearities included in both process and measurement equations, we employed an EKF. EKF basically computes the posterior distribution by propagating its mean estimate $\hat{\mathbf{x}}_{k|k}$ and covariance matrix $P_{k|k}$.

4.1.3.1 Time Update

In the prediction step, predicted state estimate $\hat{\mathbf{x}}_{k+1|k}$ and predicted covariance matrix $P_{k+1|k}$ are calculated as follows [4]:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{f}(\hat{\mathbf{x}}_{k|k}), \tag{4.14a}$$

$$J_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k}}, \tag{4.14b}$$

$$P_{k+1|k} = J_k P_{k|k} J_k^\top + BQB^\top \tag{4.14c}$$

where J_k is the Jacobian of the state transition function $\mathbf{f}(\cdot)$ evaluated at the latest state estimate $\hat{\mathbf{x}}_{k|k}$.

4.1.3.2 Measurement Update

In measurement update step; first, innovation $\tilde{\mathbf{z}}$, innovation covariance matrix S and Kalman gain K are computed; then, updated state estimate $\hat{\mathbf{x}}_{k+1|k+1}$ and covariance matrix $P_{k+1|k+1}$ are calculated as

$$\tilde{\mathbf{z}}_{k+1} = \mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}), \tag{4.15a}$$

$$H_{k+1} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k+1|k}}, \tag{4.15b}$$

$$S_{k+1} = H_{k+1} P_{k+1|k} H_{k+1}^{\top} + R, \tag{4.15c}$$

$$K_{k+1} = P_{k+1|k} H_{k+1}^{\top} S_{k+1}^{-1}, \tag{4.15d}$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1} \tilde{\mathbf{z}}_{k+1}, \tag{4.15e}$$

$$P_{k+1|k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k}. \tag{4.15f}$$

Time and measurement updates are repeated recursively throughout tracking.

4.2 Smoothing Method

Before starting to construct the GP model based on the estimates of the tracking filter, a smoothing operation needs to take place. This is due to the low observability of the ballistic parameters by position measurements at low/medium update rates (0.1-10 Hz). In particular, BP estimates of the tracking filter tend to be significantly noisy and time delayed; therefore, they are to be pre-processed before being fed as input to GP training. For this purpose, a smoother which uses all available measurements to obtain a state estimate at an arbitrary time instant k is used.

Smoothed state estimate $\hat{\mathbf{x}}_{k|1:N}$ and covariance matrix $P_{k|1:N}$ are calculated by the following set of equations

$$\hat{\mathbf{x}}_{k|1:N} = \hat{\mathbf{x}}_{k|k} + C_k[\hat{\mathbf{x}}_{k+1|1:N} - \hat{\mathbf{x}}_{k+1|k}], \tag{4.16a}$$

$$P_{k|1:N} = P_{k|k} + C_k[P_{k+1|1:N} - P_{k+1|k}]C_k^\top, \tag{4.16b}$$

$$C_k = P_{k|k}J_k^\top P_{k+1|k}^{-1} \tag{4.16c}$$

for $k = N-1, \ldots, 1$ where the final time instant is denoted by N. In (4.16), updated and predicted state estimates are denoted by $\hat{\mathbf{x}}_{k|k}$ and $\hat{\mathbf{x}}_{k+1|k}$, respectively while corresponding covariance matrices are represented by $P_{k|k}$ and $P_{k+1|k}$. A Rauch, Tung and Striebel (RTS) smoother is preferred due to its lower requirements in terms of memory storage and computational power [9]. In RTS smoother, forward Kalman filter is executed as standard tracking filter up to current time. Throughout forward filtering, smoother gain, C_k, is calculated as given in (4.16c). Note that the smoother is initialized from the latest estimate of the forward filter at time N; then, smoothed estimates are iterated backwards in time as revealed in (4.16).

Filtering and smoothing phases are summarized in 4.1.

Algorithm 4.1 Summary of Filtering and Smoothing Phases

1: Filtering Phase:
2: Initialize the state $\hat{\mathbf{x}}_0$, the covariance matrix P_0, process noise covariance matrix Q and measurement noise covariance matrix R
3: **for time** $k = 1, \ldots, N$ **do**
4: Calculate $\hat{\mathbf{x}}_{k|k-1}$ and $P_{k|k-1}$ (time update)
5: Calculate $\hat{\mathbf{x}}_{k|k}$ and $P_{k|k}$ (measurement update)
6: Calculate smoother gain C_k
7: **end for**
8: Smoothing Phase:
9: Initialize the smoothed state $\hat{\mathbf{x}}_{k|1:N}$ and the smoothed cov. matrix $P_{k|1:N}$ at $k = N$
10: **for time** $k = N-1, \ldots, 1$ **do**
11: Calculate $\hat{\mathbf{x}}_{k|1:N}$ and $P_{k|1:N}$
12: **end for**

CHAPTER 5

BALLISTIC PARAMETER LEARNING VIA GAUSSIAN PROCESS AND IMPACT POINT PREDICTION

The proposed method includes a GP regression step which uses smoothed Mach number and ballistic parameter estimates. In this chapter, we first describe non-standard cases in GP regression including noisy inputs, and correlated input-output pairs. Then, we introduce the modified form of GP regression. Later, how BP predictions of GP are utilized in motion model is described. Lastly, the impact point prediction problem is defined and how the modified motion model is employed in the solution to IPP problem is introduced.

5.1 Modifications on Standard GP Model for Ballistic Parameter Estimation

5.1.1 Cross-correlation Among GP Observations

In a standard GP regression framework, observation noise is commonly modeled as independent and identically distributed. However in our problem, GP observations are smoothed estimates of the same filter and they are not independent in time. Hence, their cross-correlation must be considered to prevent double counting of the same information. To calculate the correlation among state estimates of $N_c = 3$ recent time

instants, state vector and state space model are extended as

$$\mathbf{x}_k^c \triangleq \begin{bmatrix} \mathbf{x}_{k-2} \\ \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{bmatrix}, \qquad \mathbf{x}_{k+1}^c = F^c \mathbf{x}_k^c + G^c \mathbf{a}_k(\mathbf{x}_k) + B^c \mathbf{w}_k, \tag{5.1a}$$

$$F^c \triangleq \begin{bmatrix} 0_{7\times7} & I_{7\times7} & 0_{7\times7} \\ 0_{7\times7} & 0_{7\times7} & I_{7\times7} \\ 0_{7\times7} & 0_{7\times7} & F \end{bmatrix} G^c \triangleq \begin{bmatrix} 0_{7\times3} \\ 0_{7\times3} \\ G \end{bmatrix} B^c \triangleq \begin{bmatrix} 0_{7\times4} \\ 0_{7\times4} \\ B \end{bmatrix}. \tag{5.1b}$$

where F, G and B matrices are defined in (4.2b). By augmenting state vectors of consecutive time instants, the temporal correlation among estimates can be computed throughout filtering and smoothing procedures. The extended form in (5.1) is given for three recent time instants; however, N_c can be increased at the expense of an increase in computational load.

As a result, tracking and smoothing are done with this extended form of the state vector to calculate the temporal correlation among the estimates and Algorithm 5.1 summarizes these procedures.

5.1.2 Construction of GP Input and Observation Sets

The proposed GP model aims to represent the BP-Mach number profile of the target based on the smoothed estimates. To this end, first, Mach number estimates must be obtained from the velocity estimates. Mach number at time k can be defined as

$$M_k = \frac{\|\mathbf{v}_k\|}{c}, \tag{5.2}$$

where c is the speed of sound which is approximately $340\ m/s$. The magnitude of a vector whose components are uncorrelated, normally distributed with zero mean and equal variance is characterized by a Rayleigh distribution [20]. In our problem, on the other hand, the velocity estimates are always nonzero and the corresponding uncertainties are two orders of magnitude smaller than the estimates. This situation

Algorithm 5.1 Filtering and Smoothing in the Extended Form

Filtering Phase:

Initialize the state $\hat{\mathbf{x}}_0$, the covariance matrix P_0, process noise covariance matrix Q and measurement noise covariance matrix R

for time $k = 1, \ldots, N_c$ **do**

    Calculate $\hat{\mathbf{x}}_{k|k-1}$ and $P_{k|k-1}$ (time update)

    Calculate $\hat{\mathbf{x}}_{k|k}$ and $P_{k|k}$ (measurement update)

end for

Initialize the extended state $\hat{\mathbf{x}}^c_k$ and the extended cov. matrix P^c_k at $k = N_c$

for time $k = N_c + 1, \ldots, N$ **do**

    Calculate $\hat{\mathbf{x}}^c_{k|k-1}$ and $P^c_{k|k-1}$

    Calculate $\hat{\mathbf{x}}^c_{k|k}$ and $P^c_{k|k}$

    Calculate smoother gain C^c_k

end for

Smoothing Phase:

Initialize the extended smoothed state $\hat{\mathbf{x}}^c_{k|1:N}$ and cov. matrix $P^c_{k|1:N}$ at $k = N$

for time $k = N - 1, \ldots, 1$ **do**

    Calculate $\hat{\mathbf{x}}^c_{k|1:N}$ and $P^c_{k|1:N}$

end for

enables Mach number to be well approximated by a normal distribution such that

$$M_k \sim \mathcal{N}(\hat{M}_{k|1:N}, P_{M_{k|1:N}}), \tag{5.3a}$$

$$\hat{M}_{k|1:N} \approx \frac{\left\|\hat{\mathbf{v}}_{k|1:N}\right\|}{c}. \tag{5.3b}$$

This approximation can be validated by examining the probability density function for a Mach estimate at an arbitrary time given in Figure 5.1. In the figure, the histogram is generated for the samples of a mach number estimate which are calculated from the samples of a velocity estimate. As it can be seen, the normal distribution completely covers the empirical Mach number samples.

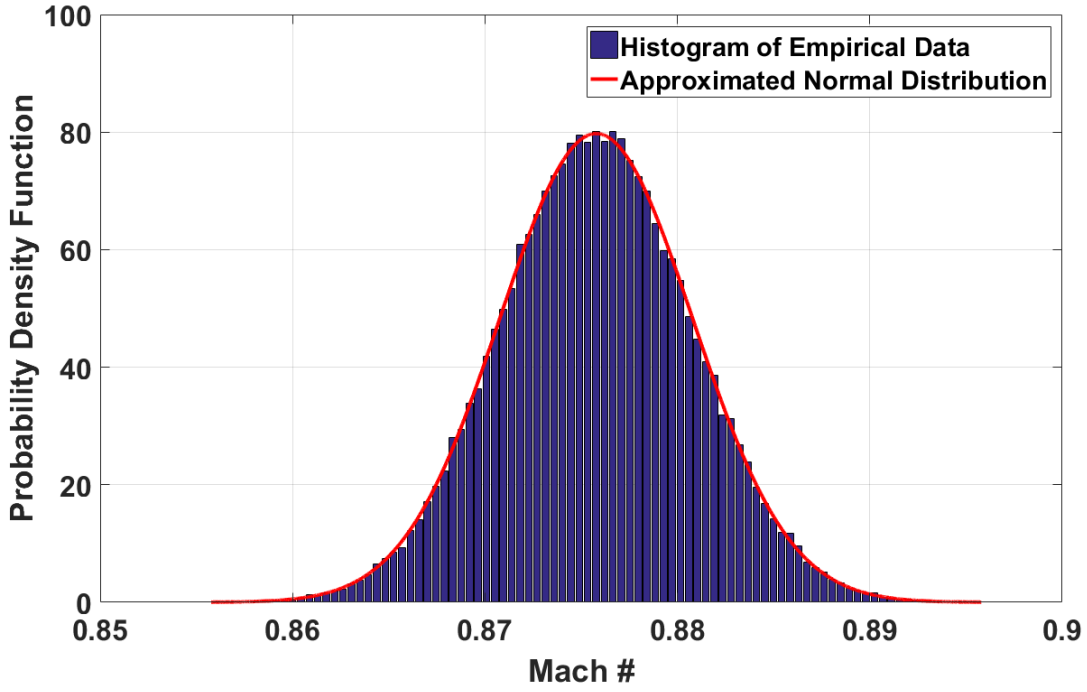


Figure 5.1: Probability density function for a Mach number estimate

The covariance matrix $P_{M_{k|1:N}}$ can be approximately calculated by linearizing (5.2) around the velocity estimate as follows.

$$P_{M_{k|1:N}} \approx L_k P_{\mathbf{v}_{k|1:N}} L_k^\top, \tag{5.4a}$$

$$L_k = \frac{1}{c}\left[\frac{\partial\|\mathbf{v}\|}{\partial v^x} \quad \frac{\partial\|\mathbf{v}\|}{\partial v^y} \quad \frac{\partial\|\mathbf{v}\|}{\partial v^z}\right]\Bigg|_{\hat{\mathbf{v}}_{k|1:N}}, \tag{5.4b}$$

$$= \frac{1}{c}\frac{\hat{\mathbf{v}}_{k|1:N}}{\left\|\hat{\mathbf{v}}_{k|1:N}\right\|}. \tag{5.4c}$$

where $P_{\mathbf{v}_{k|1:N}}$ is smoothed velocity covariance matrix at time k. Furthermore, to be able to calculate the temporal correlation among Mach number estimates, $P_{M_{k|1:N}}$ is calculated in the extended form as

$$P_{M^c_{k|1:N}} = L^c_k P_{\mathbf{v}^c_{k|1:N}} {L^c_k}^\top, \tag{5.5a}$$

$$L^c_k = diag(L_{k-2}, L_{k-1}, L_k) \tag{5.5b}$$

for $k \geq N_c$ which is chosen as 3 as an example.

The GP input set $\mathbf{M}^s$ comprises of target's Mach number at all time instants such that

$$\mathbf{M}^s = \begin{bmatrix} M_1 & \dots & M_N \end{bmatrix}^\top. \tag{5.6}$$

The input set can be written as a normal distribution

$$\mathbf{M}^s \sim \mathcal{N}(\hat{\mathbf{M}}^s, P_{\mathbf{M}^s}), \tag{5.7a}$$

$$\hat{\mathbf{M}}^s = \begin{bmatrix} \hat{M}_{1|1:N} & \dots & \hat{M}_{N|1:N} \end{bmatrix}^\top. \tag{5.7b}$$

Construction of the covariance matrix $P_{\mathbf{M}^s}$ with the local covariance matrices $P_{M^c_{k|1:N}}$ for $k = N_c, \dots, N$ is illustrated in Figure 5.2 for $N_c = 3$. Notice that the consecutive local covariance matrices overlap with each other; nevertheless, the overlapped elements have the same values at both matrices.

Equations (5.7) and Figure 5.2 completely define the input set of the GP model.

The observation set of the GP model, $\boldsymbol{\alpha}^s$, is constituted in a similar manner. The set is a collection of BP at all time instants such that

$$\boldsymbol{\alpha}^s = \begin{bmatrix} \alpha_1 & \dots & \alpha_N \end{bmatrix}^\top. \tag{5.8}$$

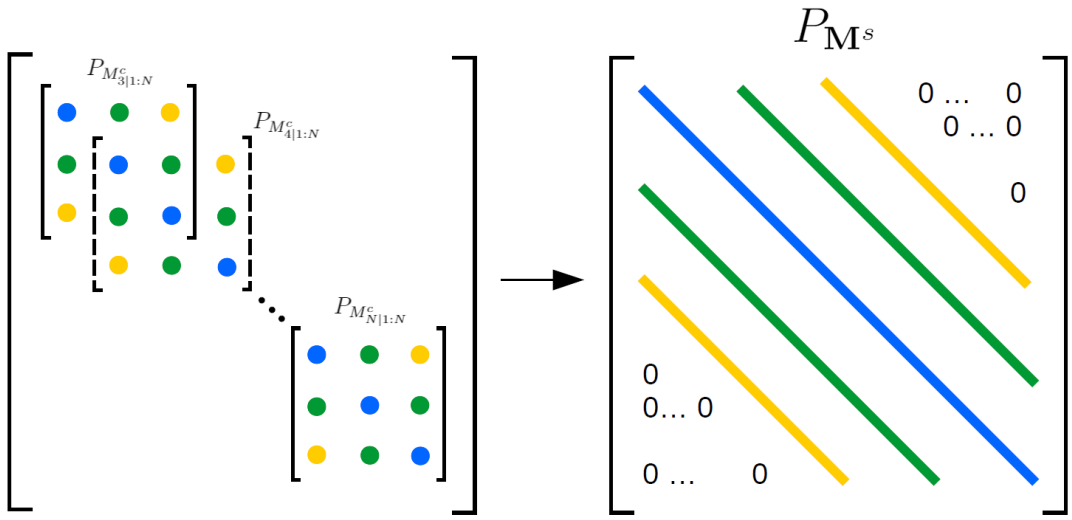


Figure 5.2: Construction of $P_{\mathbf{M}^s}$

The observation set has a normal distribution of

$$\boldsymbol{\alpha}^s \sim \mathcal{N}(\hat{\boldsymbol{\alpha}}^s, P_{\boldsymbol{\alpha}^s}), \tag{5.9a}$$

$$\hat{\boldsymbol{\alpha}}^s = \begin{bmatrix} \hat{\alpha}_{1|1:N} & \dots & \hat{\alpha}_{N|1:N} \end{bmatrix}^\top. \tag{5.9b}$$

Please note that the covariance matrix $P_{\boldsymbol{\alpha}^s}$ is constructed with the local covariance matrices $P_{\alpha^c_{k|1:N}}$ in the same way of $P_{\mathbf{M}^s}$ as illustrated in Figure 5.2. Therefore, the temporal correlations among BP estimates are preserved and will be taken into consideration in GP regression.

5.1.3 Regression with Noisy Input

In most of the Gaussian Process applications, the input of the GP is modelled as noise-less and the only noise is at observations as in (3.10). However, in our problem, the input is smoothed Mach number estimate; in other words, it is a noisy measurement of the true Mach number, $\tilde{M}$, such that

$$M = \tilde{M} + e_M, \quad e_M \sim \mathcal{N}(0, P_M). \tag{5.10}$$

By considering (5.10), (3.10) can be rewritten where true Mach number, $\tilde{M}$, is the input of GP and smoothed ballistic parameter estimate is the observation, as

$$\alpha = f(\tilde{M}) + e_\alpha, \qquad e_\alpha \sim \mathcal{N}(0, P_\alpha), \tag{5.11a}$$

$$= f(M - e_M) + e_\alpha. \tag{5.11b}$$

In the equations, e_M and e_α stand for input and observation noises, respectively. The posterior distribution based on (5.11) is intractable. To overcome the intractability, Taylor expansion of the measurement equation about the observed input, M, can be written as

$$f(M - e_M) = f(M) - e_M^\mathsf{T} \frac{\partial f(M)}{\partial M} + HOT, \tag{5.12}$$

where HOT denotes higher order terms. In the equation, derivative of the GP is another Gaussian Process, and distribution over Taylor expansion must be considered as stated by McHutchon and Rasmussen [22]. It is also stated that analytical calculation of first and second moments of Taylor expansion of GP is tractable but requires high computational load. However, it provides no significant improvement over the quicker approximate method which is taking derivative of the predictive mean of the GP function with respect to the input and evaluating it at an input point. Differentiating the mean function corresponds to ignoring the uncertainty about the derivative. In that case, (5.11) becomes

$$\alpha = f(M) + e_\alpha + e_M^\mathsf{T} \partial_{\overline{f}} \tag{5.13}$$

by expanding up to the first order terms, where $\partial_{\overline{f}}$ is derivative of predictive mean evaluated at the input point, M. The equation can be interpreted as treating the input as deterministic and adding a corrective term to the output noise.

In GP regression, the covariance matrix of the observation set K_y in (3.12d) can be rewritten with the input set of smoothed Mach number estimates $\hat{\mathbf{M}}^s$, observation noise covariance matrix $P_{\boldsymbol{\alpha}^s}$, and the addition of the last term due to input noise (if the input and observation noises are independent) such that

$$K_y = K(\hat{\mathbf{M}}^s, \hat{\mathbf{M}}^s) + P_{\boldsymbol{\alpha}^s} + \Delta_{\mathbf{M}^s} P_{\mathbf{M}^s} \Delta_{\mathbf{M}^s}^\mathsf{T} \tag{5.14}$$

where $\Delta_{\mathbf{M}^s}$ is an $N \times N$ matrix whose diagonal elements are $\partial_{\overline{f}}$ evaluated at corresponding input points and $P_{\mathbf{M}^s}$ is the input set covariance matrix.

Taking the input noise into account is crucial especially when the uncertainty in target speed estimates is high and the target is at transonic speed regime where BP rapidly changes with Mach number. For instance, $20\ m/s$ uncertainty in the target speed estimate at transonic speed regime may correspond to $3\ m^2/kg$ uncertainty in the ballistic parameter for a generic target.

5.1.4 Cross-correlation Among Input-Observation Pairs

In the calculation of covariance matrix of observations given in (5.14), the input and observation noises are assumed to be independent; however, in our problem, the input and output are smoothed Mach number and ballistic parameter estimates, and their noises are correlated with each other since BP and target velocity are estimated concurrently by the same filter. Due to the cross-correlation between input and observation noises, an additional term is added to the equation and it becomes

$$K_y = K(\hat{\mathbf{M}}^s, \hat{\mathbf{M}}^s) + P_{\boldsymbol{\alpha}^s} + \Delta_{\mathbf{M}^s} P_{\mathbf{M}^s} \Delta_{\mathbf{M}^s}^{\mathsf{T}} + 2\Delta_{\mathbf{M}^s} P_{\mathbf{M}^s \boldsymbol{\alpha}^s} \tag{5.15}$$

where $P_{\mathbf{M}^s \boldsymbol{\alpha}^s}$ is the cross-covariance (CC) matrix and it is constructed in the same way as $P_{\mathbf{M}^s}$ and $P_{\boldsymbol{\alpha}^s}$. First, the local CC matrix $P_{M\alpha^c_{k|1:N}}$ is calculated in the extended form as

$$P_{M\alpha^c_{k|1:N}} = L^c_k P_{\mathbf{v}\alpha^c_{k|1:N}} \tag{5.16}$$

where L^c_k is defined in (5.5b) and $P_{\mathbf{v}\alpha^c_{k|1:N}}$ stands for the cross-covariance matrix between the velocity and BP. Then, the local CC matrices for $k = N_c, \ldots, N$ are placed similar to $P_{\mathbf{M}^s}$ as illustrated in Figure 5.2 to form global CC matrix $P_{\mathbf{M}^s \boldsymbol{\alpha}^s}$.

5.1.5 Modified Gaussian Process Regression

For input-observation sets $\hat{\mathbf{M}}^s$-$\hat{\boldsymbol{\alpha}}^s$, a test point M_* and a GP prediction α_*; the modified GP regression is given in (5.17) by considering the input noise, the temporal cross-correlation among inputs, the temporal cross-correlation among observations

and cross-correlation between input-observation sets.

$$\alpha_* = S(M_*)\boldsymbol{\alpha}^s + e_*, \quad e_* \sim \mathcal{N}(0, E(M_*)), \tag{5.17a}$$

$$S(M_*) = K(M_*, \hat{\mathbf{M}}^s)K_y^{-1}, \tag{5.17b}$$

$$E(M_*) = k(M_*, M_*) + \partial_{M_*}^2 P_{M_*} - S(M_*)K(\hat{\mathbf{M}}^s, M_*) \tag{5.17c}$$

where K_y is given in (5.15). In regression, a test input M_* is assumed noisy as well. As given in (5.13), the noise in the test input is conveyed to the observation, and $\partial_{M_*}^2 P_{M_*}$ is added to the test output covariance matrix in (5.17c) where the derivative term ∂_{M_*} is scalar.

5.2 Modified Motion Model

While predicting the impact point of the target, BP is dropped from the state vector and modeled with Gaussian process such that it is a function of Mach number of the target rather than modeling with random walk. Modified GP Regression, which uses smoothed ballistic parameter-Mach number pairs up to current time instant as the training dataset, can be used to predict BP at time k. By replacing α_* and M_* in (5.17) by α_k and M_k respectively, and using GP predicted α_k in (4.11), state space model involving Gaussian process regression of ballistic parameter is obtained as

$$\begin{aligned}\tilde{\mathbf{x}}_{k+1} &= \overbrace{\tilde{F}\tilde{\mathbf{x}}_k + \tilde{G}(-\frac{1}{2}\rho(p_k^z)AS(M_k)\boldsymbol{\alpha}^s \left\|\mathbf{v}_k\right\| \mathbf{v}_k) + \tilde{G}\mathbf{g}}^{\mathbf{f}^m(\tilde{\mathbf{x}}_k)} + \\ &\quad \underbrace{\tilde{G}\,(-\frac{1}{2}\rho(p_k^z)Ae_k \left\|\mathbf{v}_k\right\| \mathbf{v}_k + \tilde{\mathbf{w}}_k)}_{\overline{\mathbf{w}}_k}, \end{aligned} \tag{5.18a}$$

$$= \mathbf{f}^m(\tilde{\mathbf{x}}_k) + \tilde{G}\overline{\mathbf{w}}_k, \qquad \overline{\mathbf{w}}_k \sim \mathcal{N}(0, \overline{Q}_k), \tag{5.18b}$$

where the process noise covariance matrix is

$$\overline{Q}_k = \zeta_k E(M_k)\zeta_k^{\mathsf{T}} + \tilde{Q}, \tag{5.19a}$$

$$\zeta_k = -\frac{1}{2}\rho(p_k^z)A \left\|\mathbf{v}_k\right\| \mathbf{v}_k. \tag{5.19b}$$

Since the ballistic parameter is dropped from the state vector, the last row of the state vector $\mathbf{x}$, the process noise vector $\mathbf{w}$ and G matrix as well as the last row and column of F and B matrices in state transition function (4.2) are wiped out. The modified forms of these vectors and matrices are represented by tilde symbol over them in (5.18). The same notation is valid for the process noise covariance matrix Q in (5.19a) as well.

Using the modified state space model involving GP regression of the ballistic parameter given in (5.18)-(5.19) results in more accurate predictions of target state during the impact point prediction phase.

5.3 Impact Point Prediction

The impact point prediction problem can be defined as predicting the position that the target is expected to hit on the ground while the target is still on the fly.

On the IPP phase, EKF introduced in Section 4.1.3 is employed to compute the predictive distribution with modified state transition function $\mathbf{f}^m(\cdot)$. Note that only time update in Section 4.1.3.1 is employed since there is no available measurement in the IPP phase. This prediction step can be formulated as

$$\tilde{\mathbf{x}}_{k+1|N} = \mathbf{f}^m(\tilde{\mathbf{x}}_{k|N}), \tag{5.20a}$$

$$J_k^m = \left.\frac{\partial \mathbf{f}^m}{\partial \tilde{\mathbf{x}}}\right|_{\tilde{\mathbf{x}}_{k|N}}, \tag{5.20b}$$

$$\tilde{P}_{k+1|N} = J_k^m \tilde{P}_{k|N} {J_k^m}^\top + \tilde{G}\overline{Q}_k\tilde{G}^\top \tag{5.20c}$$

for time $k \geq N$. The prediction continues until the target hits the ground. The impact point prediction using GP model is summarized in Algorithm 5.2.

Algorithm 5.2 Summary of IPP using GP

Construction of GP Input Set:
for time $k = 1, \ldots, N$ **do**
    Calculate smoothed Mach number estimate $\hat{M}_{k|1:N}$ according to (5.3b)
    Calculate smoothed cov. matrix $P_{M^c_{k|1:N}}$ according to (5.4b) and (5.5)
end for
Construct GP input set $\hat{\mathbf{M}}^s$ according to (5.7b)
Construct GP input noise covariance matrix $P_{\mathbf{M}^s}$ according to Figure 5.2
Construction of GP Observation Set:
Construct GP observation set $\hat{\boldsymbol{\alpha}}^s$ according to (5.9b)
Construct GP observation noise cov. matrix $P_{\boldsymbol{\alpha}^s}$ according to Figure 5.2
Construction of Cross-covariance (CC) Matrix Between Input-Observation Sets:
Calculate CC between Mach number-BP estimates $P_{M\alpha^c_{k|1:N}}$ according to (5.16)
Construct CC between GP input-observation sets $P_{\mathbf{M}^s\boldsymbol{\alpha}^s}$ according to Figure 5.2
IPP Phase:
Initialize $\tilde{\mathbf{x}}_{N|N}$ and $\tilde{P}_{N|N}$
for time $k = N, \ldots, \infty$ **do**
    Predict α_k at test point M_k according to (5.17)
    Calculate $\tilde{\mathbf{x}}_{k+1|N}$ and $\tilde{P}_{k+1|N}$ according to (5.20)
    if $\hat{p}^z_{k+1|N} \geq 0$ **then** (Target hits the ground)
        Return $\tilde{\mathbf{x}}_{k+1|N}$ and $\tilde{P}_{k+1|N}$
    end if
end for

CHAPTER 6

SIMULATION RESULTS

In this section, the performance of the proposed method is evaluated and compared with the existing methods in the literature through simulations. In this context, target models which generate the reference target data and scenarios are described in Section 6.1. To emphasize the importance of smoothing, the characteristics of filtered and smoothed estimates are studied in Section 6.2. Then to illustrate the learning performance of the method, GP regressions for all targets are depicted and examined in Section 6.3. To evaluate impact point prediction performance, change of the ballistic parameter in time in the prediction phase is plotted for all scenarios and Root Mean Square (RMS) impact point prediction error is given in Section 6.4. Lastly, hyperparameter optimization and SPGP which is an approximation of GP are discussed in Section 6.5.

6.1 Simulation Scenarios

In the simulation environment, three different ballistic targets having different shapes and ballistic characteristics are modeled and target trajectories having different ranges are generated [2]. These projectiles are selected as 122 mm rocket, 155 mm artillery shell and 120 mm mortar. The rocket and mortar are fin-stabilized projectiles, whereas the shell is spin-stabilized. To generate the trajectories of the projectiles, a 6-DoF model has been used, i.e., three axes translations in 3D space and three axes orientations of the projectiles are calculated while generating the trajectories. Also, the model consists of forces and moments which are not captured by the motion model

of the filter completely. Simulated trajectories are depicted in Figure 6.1 with radar detection and IPP instants. Corresponding speed profiles are also given in Figure 6.2.

Sensor model generates measurements in 3D polar coordinate composed of range, bearing and elevation angles with 100 Hz update rate. Measurement error is assumed to be unbiased and has a standard deviation of 20 m in range and 0.2° in angle measurements. A measurement is sum of the reference target position in polar coordinates and a measurement error.

The simulation is performed for 100 monte carlo runs with different realizations of the measurement noise at each run. The presented results in this chapter are the average of the Monte Carlo runs.

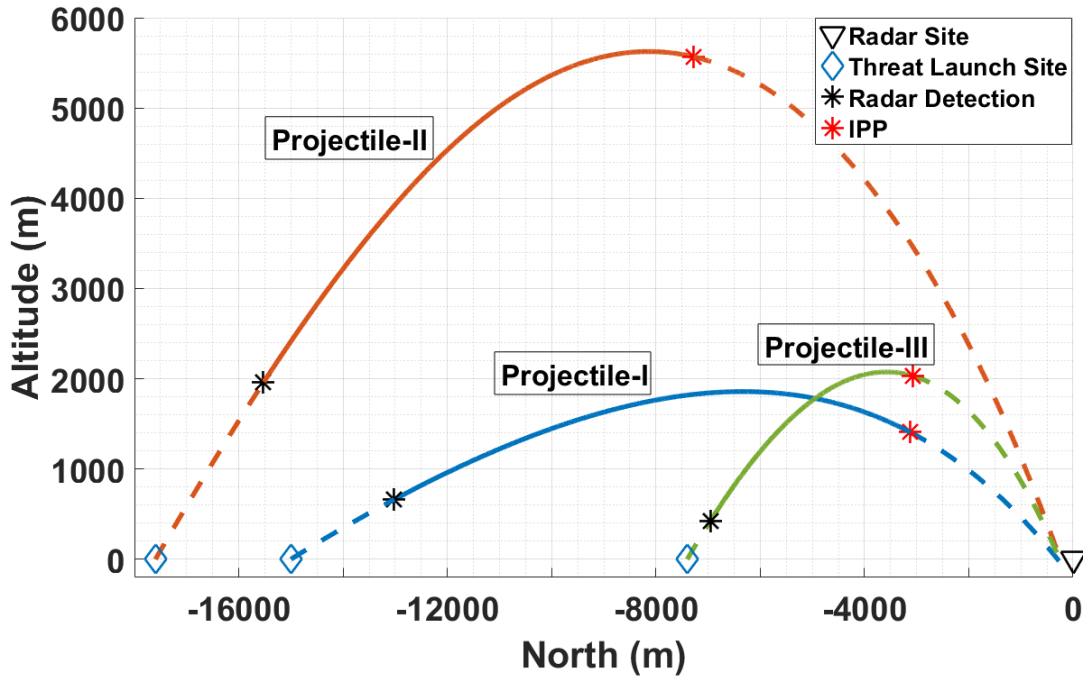


Figure 6.1: Projectile trajectories

6.2 Ballistic Parameter Estimation

Before examining Gaussian process regression and its contribution to impact point prediction, tracking and smoothing phases must be studied first. A tracking filter

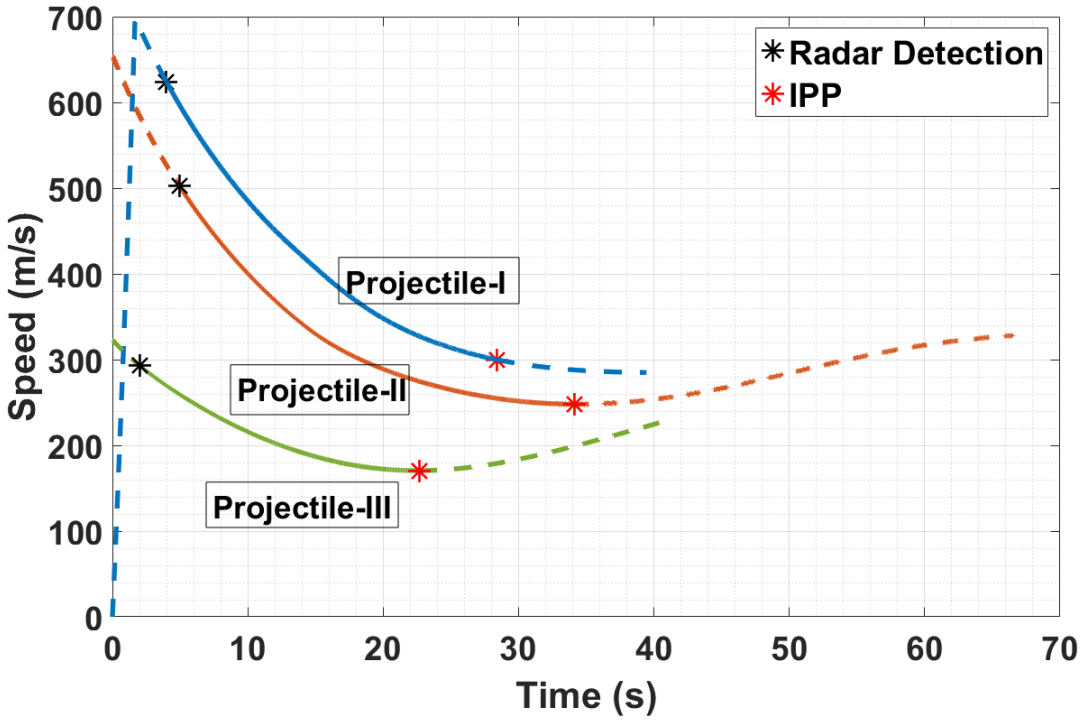


Figure 6.2: Projectile speed profiles

is employed against Projectile-I with the state transition and measurement equations defined in (4.2)-(4.12). BP estimates of the filter are depicted in Figure 6.3 as a function of time for a specific Monte Carlo run. It is seen that BP estimates of the filter pursue the reference ballistic parameter with some delay, and the estimates are noisy.

At IPP instant, ballistic parameter, position and velocity estimates are smoothed by RTS smoother as described in Section 4.2, smoothed ballistic parameter estimate is shown in Figure 6.3 by the dashed line. By utilizing a smoother, the noise in the estimate, as well as time delay, mostly vanish. Nonetheless, there is still some discrepancy from the reference BP value, especially at the beginning and at the end of the processing interval where filter and smoother have not converged yet, respectively. Since this discrepancy is a common situation regardless of the target type, smoothed BP estimates at these predefined time intervals are not used in GP regression to prevent misleading GP learning.

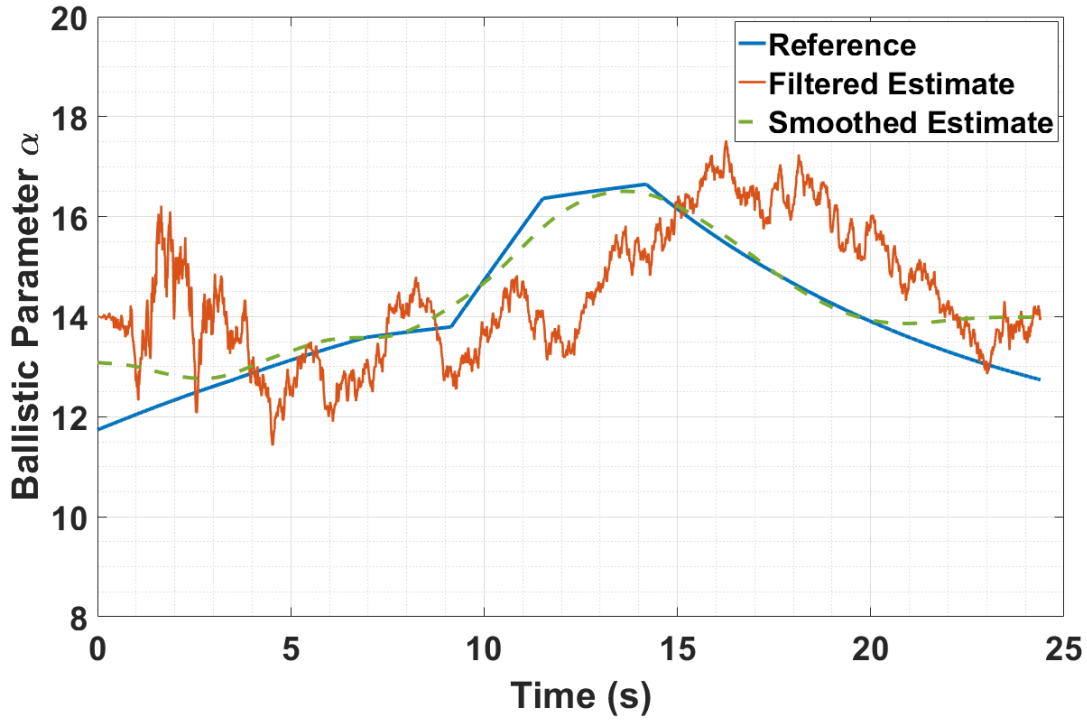
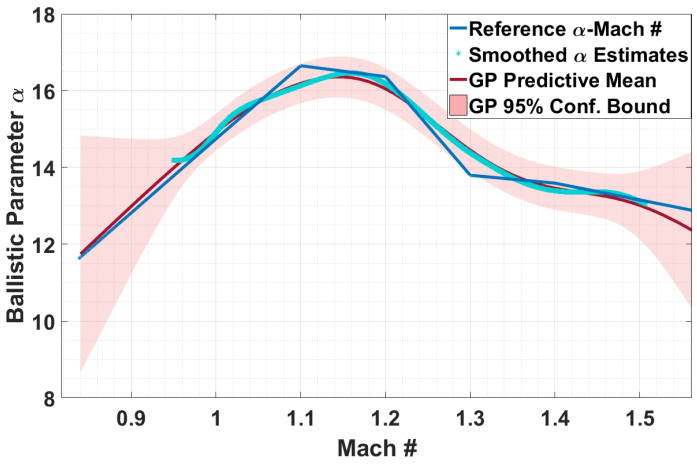


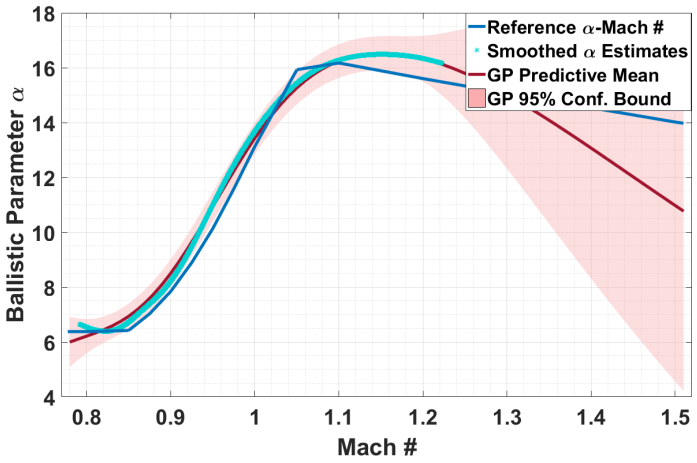
Figure 6.3: Ballistic parameter estimates for Projectile-I

6.3 Performance of GP Regression

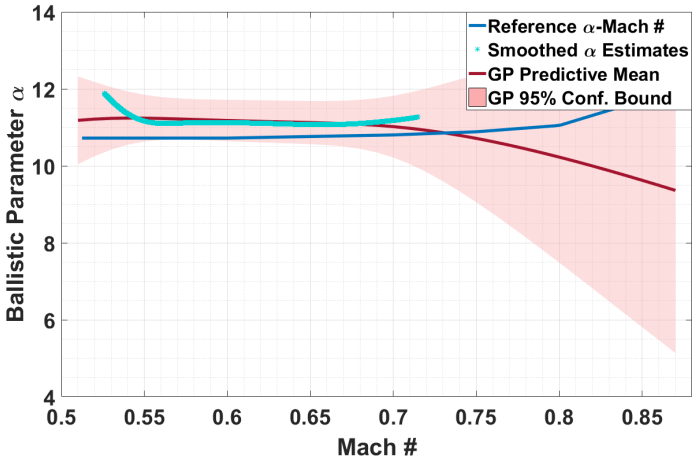
To show the learning performance of the method, GP predictive mean and confidence bound calculated by (5.17) are illustrated in Figure 6.4 with smoothed BP-Mach number pair estimates for three projectiles. It is seen that GP predictive means smoothly fit to the estimates for all projectiles. In the figures, when a test point moves away from the smoothed estimates, the uncertainty of the GP prediction enlarges. Furthermore, predictive means for projectiles II and III may diverge from the reference at unobserved Mach numbers; however, it has no significant effect since the targets do not reach these speed values through their remaining trajectories. The values of hyperparameters are determined in a way that the predictions can be done without overfitting the observations, and the reference BP-Mach number function mostly stays in the confidence bound for all projectiles.



(a) Projectile-I



(b) Projectile-II



(c) Projectile-III

Figure 6.4: Learned ballistic parameter-Mach number for different projectiles

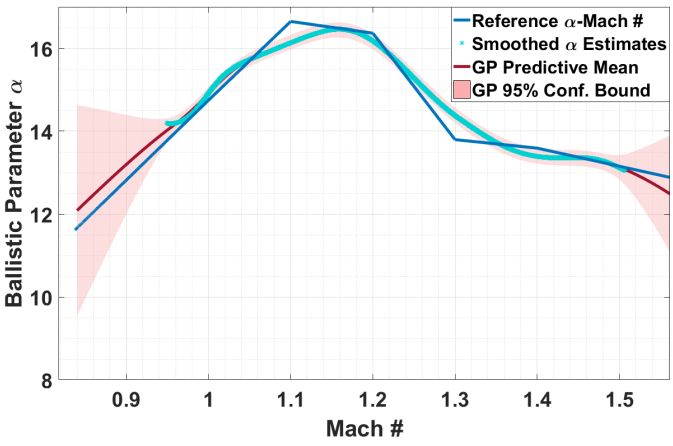
6.3.1 Effect of Cross-correlation Among GP Observations

As discussed in Section 5.1.1, the temporal cross-correlation among smoothed ballistic parameter estimates is calculated at a certain level by using concatenated state vector and state space model defined in (5.1). To understand the effect of considering cross-correlation among BP estimates, GP regressions for three cases are shown in Figure 6.5: (a) without considering cross-correlation, (b) cross-correlation between 5 consecutive time instants and (c) cross-correlation between 25 consecutive time instants are considered with the same GP hyperparameters. In case (a), it is seen that the reference BP profile cannot be covered by the confidence bound, i.e., the bound is over-confident. From case (a) to case (c), the uncertainty of GP predictions increases with the number of time instants. The reason of this phenomenon is that ballistic parameter estimates are highly correlated in time and taking cross-correlation into account prevents double counting of the same information. In other words, from case (a) to case (c), the estimates become more and more decorrelated. Hence, the confidence bound of GP predictions enlarges. In case (c), the correlation coefficients between estimates of different time instants change from 1 to 0.75 for Projectile-I at an arbitrary time instant. These values of the correlation coefficient indicate the high correlation among the estimates. To achieve full decorrelation, the number of time instants should be increased up to 80. In our study, the number of time instants is determined so that the decorrelation is sufficient while its computational cost is still endurable.

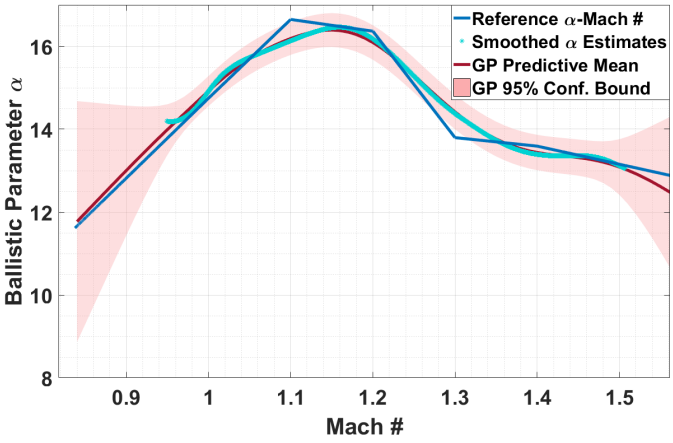
6.4 IPP Performance using GP Model

At impact point prediction phase, BP predictions of GP are used in the state transition function as described in (5.18) and (5.19). In the literature, conventional methods in IPP use the most recent ballistic parameter estimate of the filter as a constant throughout the prediction [28, 19].

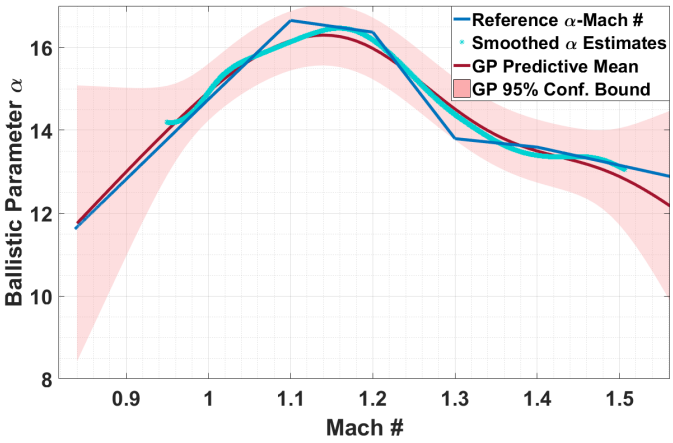
In the analysis, the impact point predictions are done when the target speeds start to increase for Projectile II and III (see Figure 6.2). For Projectile I, on the other



(a) Cross correlation is ignored.



(b) Cross correlation between 5 time instants is considered.



(c) Cross correlation between 25 time instants is considered.

Figure 6.5: Effect of considering cross correlation among observations on GP regression for Projectile-I

hand, the prediction is done at the same range with Projectile II (see Figure 6.1). The IPP instants are 24.4, 29.2 and 20.8 s for the projectiles, respectively. In Figure 6.6, filtered and smoothed BP estimates up to the IPP instants, BP predictions up to the impact instants and the reference ballistic parameters are plotted for three projectiles.

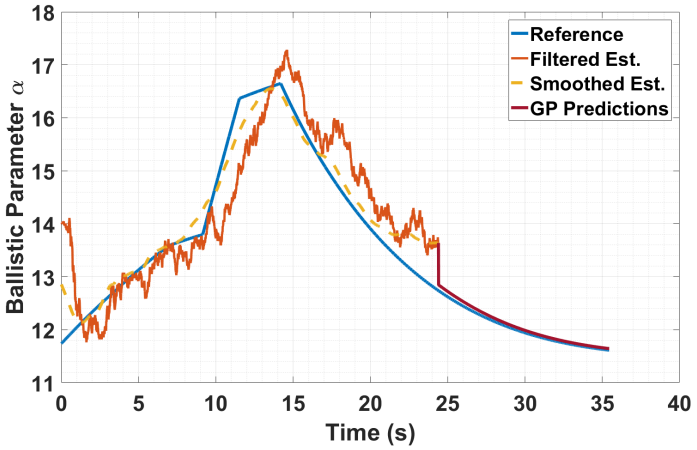
When the modified motion model is invoked at the IPP instant, there is a discontinuity in ballistic parameter since GP predicted BP could be different from the BP estimate. Furthermore, for Projectile I and II, the reference BP quite varies in the prediction phase and predicted BP is able to follow the reference ballistic parameter with an acceptable error. For Projectile III, predicting BP by Gaussian process just outperforms the conventional approach which uses the recent BP estimate throughout the prediction, since the reference ballistic parameter has already been constant throughout the prediction phase. To be able to quantify the effectiveness of the proposed method on the IPP problem, RMS IPP errors

$$\epsilon_{imp}^{RMS} = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}\left\|\hat{\mathbf{p}}_{imp}^{i} - \mathbf{p}_{imp}^{ref}\right\|^2} \tag{6.1}$$

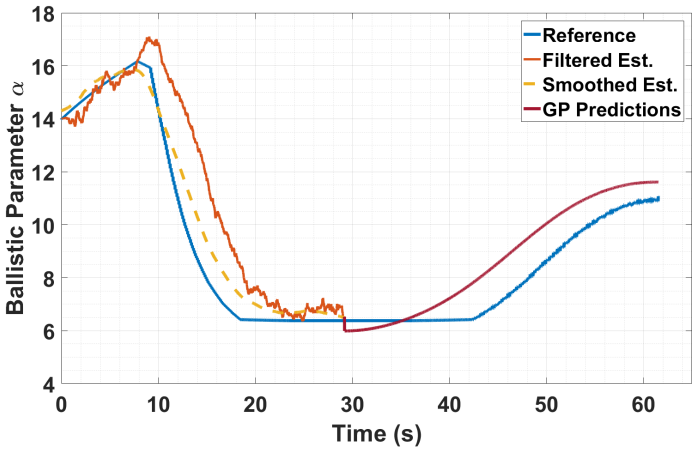
where $\mathbf{p}_{imp}^{ref}$ is true impact point, $\hat{\mathbf{p}}_{imp}^{i}$ is predicted impact point at i^{th} MC run and N is the total run number, are calculated for all scenarios.

Furthermore, to be able to show the asymptotic performance of perfectly learning ballistic parameter, the third model called True BP Model is constructed. For this model, the reference ballistic parameter profile is known and true BP value at the predicted Mach number is used in the IPP.

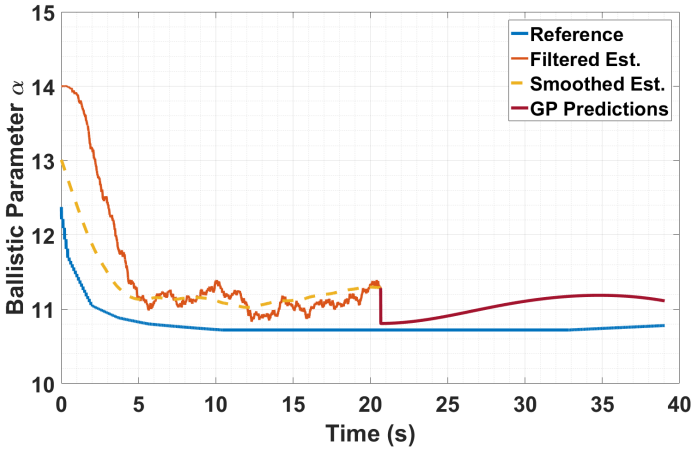
While comparing these methods, the conventional approach which uses constant ballistic parameter represents the primitive (since there is no learning at all) and True BP Model represents the asymptotic cases for a method which aims to learn ballistic parameter function. The IPP errors for these three models are given in Table-6.1. When the table is examined, it is seen that the proposed BP-GP method outperforms the constant BP method with regards to RMS IPP error against all projectiles. For all methods, the RMS IPP error against projectile-II is much higher than other projectiles. The reason is that projectile-II is spin-stabilized and lateral Magnus force



(a) Projectile-I



(b) Projectile-II



(c) Projectile-III

Figure 6.6: Ballistic parameter-time

acting on this projectile has a significant effect on its impact point. However, this force is not taking into account in the system model. This can be achieved by using Multiple Model Kalman Filter as [28] suggests; however, it is not implemented in this study, since the primary objective of this study is showing how to model the ballistic parameter via Gaussian process and to utilize this model in the prediction phase.

Table 6.1: RMS IPP errors of different models (m)

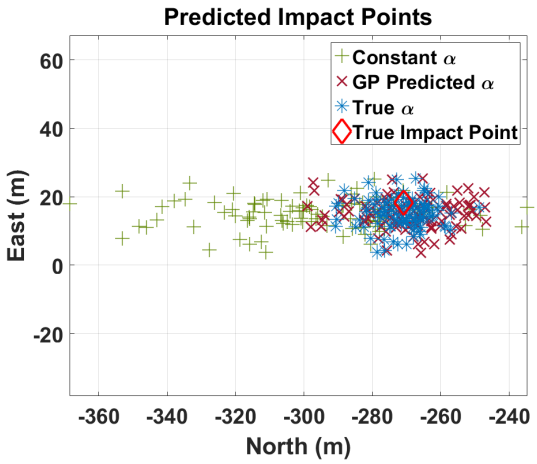
Projectile	Constant BP	GP	True BP
I	39.47	15.48	10.56
II	267.76	188.39	179.46
III	41.90	19.68	15.78

Predicted impact points of these models are shown in Figure 6.7 for different MC runs with the true impact point. For constant BP model, BP estimates of the filter are highly noisy and the BP estimate at the IPP instant has a large variance. As a result, the impact point predictions are extremely scattered around the true impact point. In the scenarios, the projectiles are launched from South to North and the impact point predictions are mainly scattered in this direction since the drag force is aligned with the target velocity. By using BP-GP model, the variance of the impact point predictions decreases since the prediction of BP using a GP regression improves the accuracy of BP predictions significantly. The error of the impact point predictions of Projectile-II can be seen in Figure 6.7b. This lateral bias arises from excluding the effective lateral force in the motion model as discussed earlier. Although BP-GP method cannot handle this bias, as it does not intend to do so, it significantly enhances the IPP performance for all projectiles.

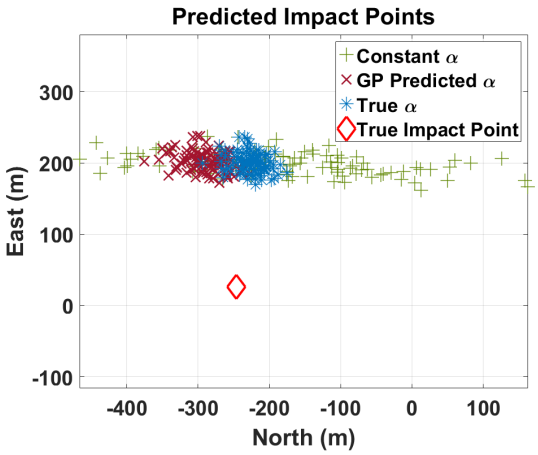
6.5 Further Discussions on Gaussian Processes

6.5.1 Performance of Kernel Selection and Hyperparameter Optimization

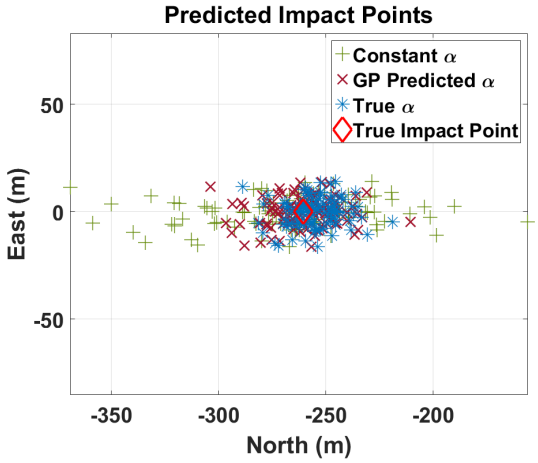
As mentioned in Section 3.3.1, the model selection and hyperparameter optimization comprise of choosing the most suitable kernel among different kernel families and



(a) Projectile-I



(b) Projectile-II



(c) Projectile-III

Figure 6.7: Impact point predictions

setting its hyperparameters. For that purpose, the log likelihood function given in (3.13) is maximized by MLE described in Section 3.3.1.2. In a standard application, the optimization is done over the hyperparameters of the kernel and the variance of the observation noise. This variance is assumed to follow an independent and identically distributed Gaussian distribution with zero mean. However, in this study, the observations are the smoothed estimates of the tracking filter and the observation noise can be modeled as a Gaussian distribution with zero mean and variance of the smoothed estimates as given in (5.11). In other words, the variance of the observation noise is not a parameter to be optimized in our application.

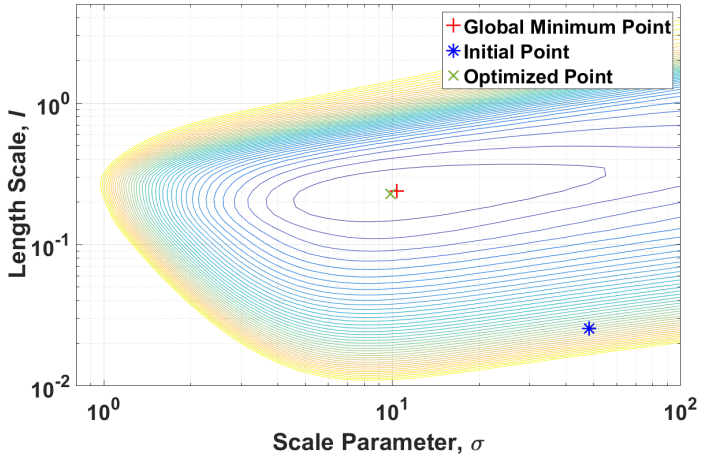
MLE is performed by a constrained optimization which utilizes the interior point algorithm [36] to maximize the log likelihood function. The constraints are defined to limit the range of the hyperparameters at reasonable non-zero values. Figure 6.8 illustrates an example of contour plots showing the minus log likelihood as a function of the length scale and the scale parameter for three kernels. The log likelihoods are based on a realization of the smoothed BP estimates for Projectile-I. In the figure, the randomly initialized, optimized and global minimum points are marked as well. As seen in the figure, the optimized solutions are very close to the global minimum points of the minus log likelihood function for all kernels. These results verify that the optimization procedure can be done correctly.

The optimized values of the hyperparameters and the maximized values of the likelihood functions are given in Table 6.2 for the kernels.

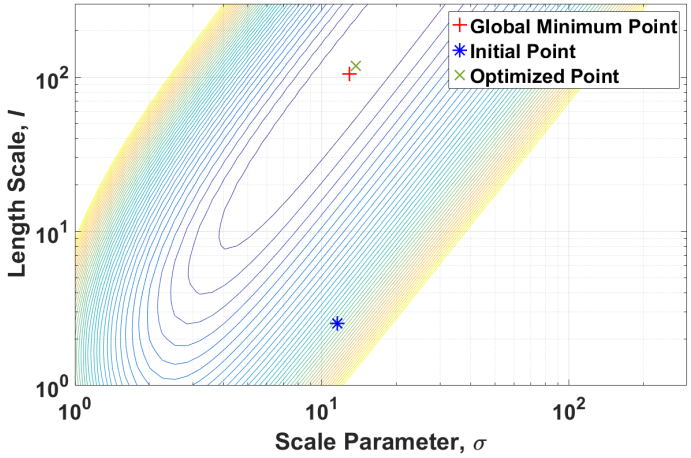
Table 6.2: Optimization results of different kernels

Kernel	σ	l	Log Lik. Value
EQ	9.8644	0.2287	-1516.47
Exp.	13.7	118.7266	-1529.16
Matern-3	11.0424	1.2448	-1513.88

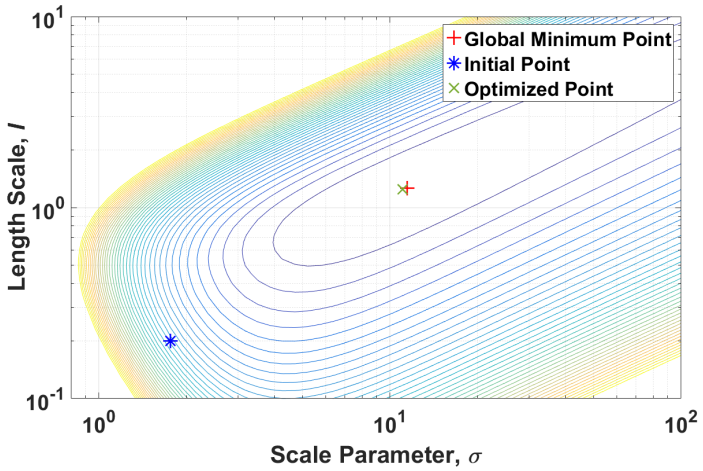
For kernel selection, the kernel which gives the highest likelihood value is preferred, which is the Matern-3. The solution can be confirmed by looking at the GP predictive distributions with the optimized hyperparameters in Figure 6.9.



(a) Exponentiated Quadratic

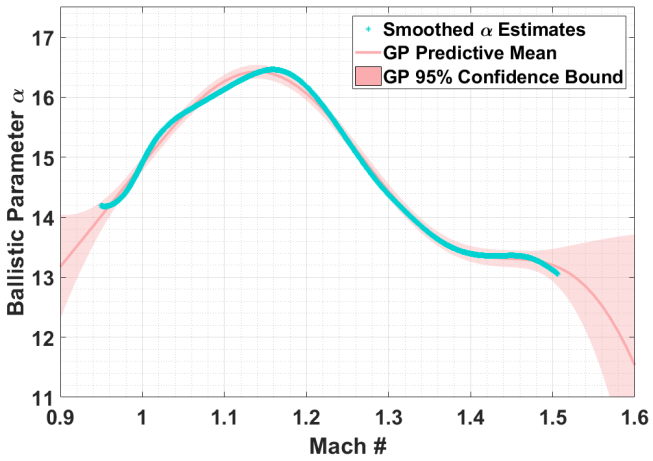


(b) Exponential

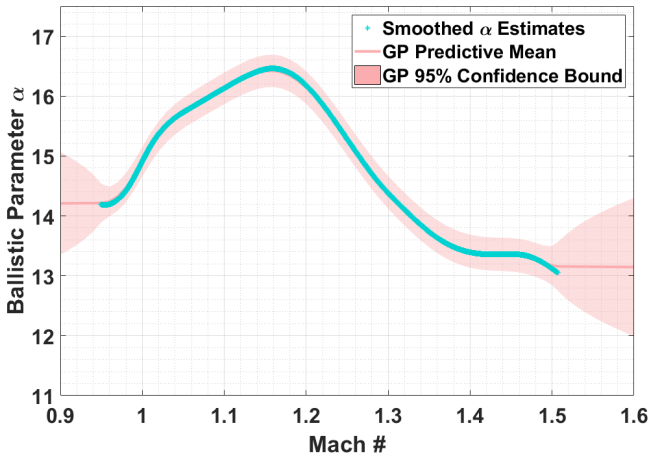


(c) Matern-3

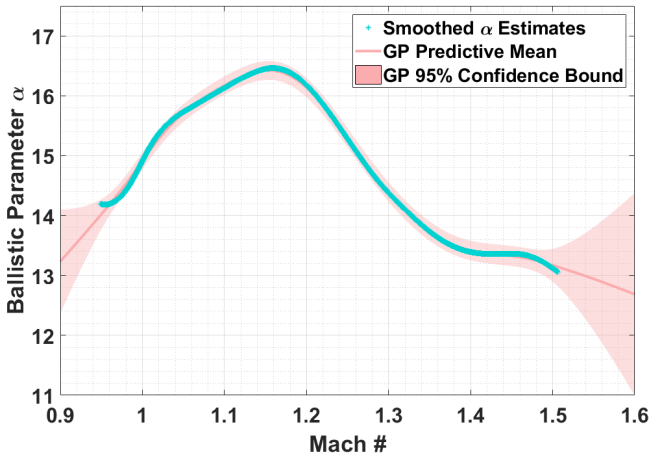
Figure 6.8: Contour plots of minus log likelihood function



(a) Exponentiated Quadratic



(b) Exponential



(c) Matern-3

Figure 6.9: Predictive distributions with several kernels

The predictive mean with EQ kernel given in Figure 6.9a, does not cover the nonlinear characteristic of the smoothed BP estimates entirely. The predictive mean with exponential kernel given in Figure 6.9b, on the other hand, is overfitting the estimates and this is not a desired characteristic to have. Moreover, at the outside of the estimate range, the predictive mean stays constant without carrying any information about the estimates and the confidence bound becomes unnecessarily large. Finally, the predictive distribution with Matern-3 kernel given in Figure 6.9c shows the best characteristic with respect to the mean and confidence bound among the discussed kernels. In conclusion, the selection of the Matern-3 kernel is legitimate.

The optimization can be done as a part of the algorithm so that GP regression is done with the hyperparameters which are optimized on-the-fly at the expense of an increase in computation load of the algorithm. When 100 realizations of the smoothed BP estimates for all projectiles are considered, it is seen that the optimized values of the hyperparameters are very close to each other. That is due to the fact that the characteristics, e.g., the extent of nonlinearity, of the estimates do not vary significantly among different realizations and projectiles. The IPP errors in case of the on-the-fly optimization of hyperparameters are given in Table 6.3 for all projectiles. The table also includes the IPP errors with constant hyperparameters which is given previously in Table 6.1. Both models utilize Matern-3 kernel.

Table 6.3: RMS IPP errors of GPs with fixed and optimized hyperparameters (m)

Projectile	GP with fixed hyp.	GP with opt. hyp.
I	15.48	14.81
II	188.39	186.46
III	19.68	18.29

Table 6.3 shows that employing GP with on-the-fly optimization does not enhance the IPP performance significantly.

6.5.2 Performance of SPGP Regression

Figure 6.10 illustrates the initial and optimized pseudo-inputs based on a realization of smoothed BP estimates for Projectile-I. The initial values of the pseudo-inputs are generated randomly from a uniform distribution in the range of smoothed BP estimates. As SPGP method suggests, the hyperparameters of Matern-3 kernel are optimized in conjunction with the pseudo-inputs. As a constrained optimization method, the interior point algorithm [36] is utilized to maximize logarithm of the likelihood function given in (3.18). The optimization is done over 10 pseudo-inputs and 2 hyperparameters. In Figure 6.10, the initial pseudo-inputs which mostly gather in the range of 1.3 and 1.5 $Mach$ are presented at the top, and the optimized pseudo-inputs are shown at the bottom as an illustration. The figure clearly reveals that the optimized pseudo-inputs are scattered over the range of smoothed BP estimates at almost equal distances. This is in fact, a general outcome which is independent of the initial conditions and the number of pseudo-inputs for a continuous and smooth set of BP estimates. From realization to realization, the positions of the pseudo-inputs may be shifted by keeping the distance between each other almost equal. Besides, the value of the likelihood function does not change significantly. The optimized hyperparameters of the kernel are $\sigma_{opt} = 11.0161$ and $l_{opt} = 1.3088$. For all realizations, the optimized values of the hyperparameters are almost the same.

The predictive distribution of SPGP is illustrated in Figure 6.11, by plotting the predictive mean and 95% confidence bound around the mean. It is seen that hyperparameters and pseudo-inputs are optimized such that the predictive mean of the SPGP exhibits the basic characteristics of the smoothed BP estimates without overfitting on them. Furthermore, the confidence bound of SPGP completely covers the smoothed BP estimates without being excessively large. It indicates that the GP predictions are consistent with the estimates.

To show that the optimization is independent of the initial values of the pseudo-inputs and hyperparameters, GP regressions under different initializations are plotted in Figure 6.12. In the figure, the pseudo-inputs are equally placed along the left and right half sides of the range of smoothed BP estimates, respectively.

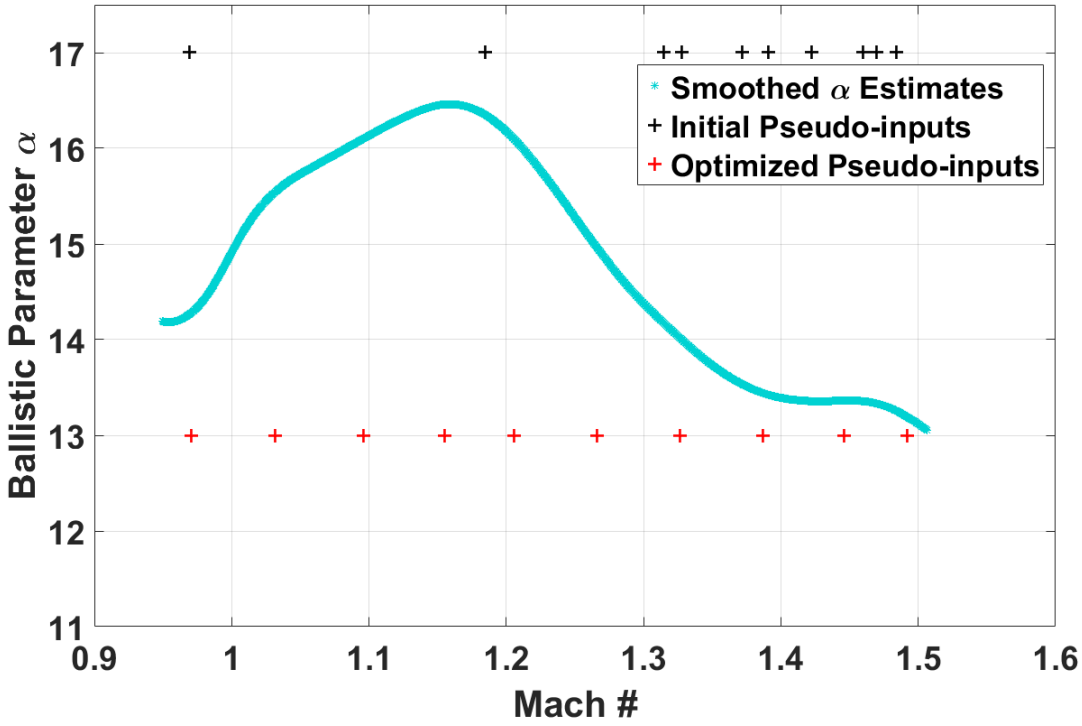


Figure 6.10: Optimization of pseudo-inputs

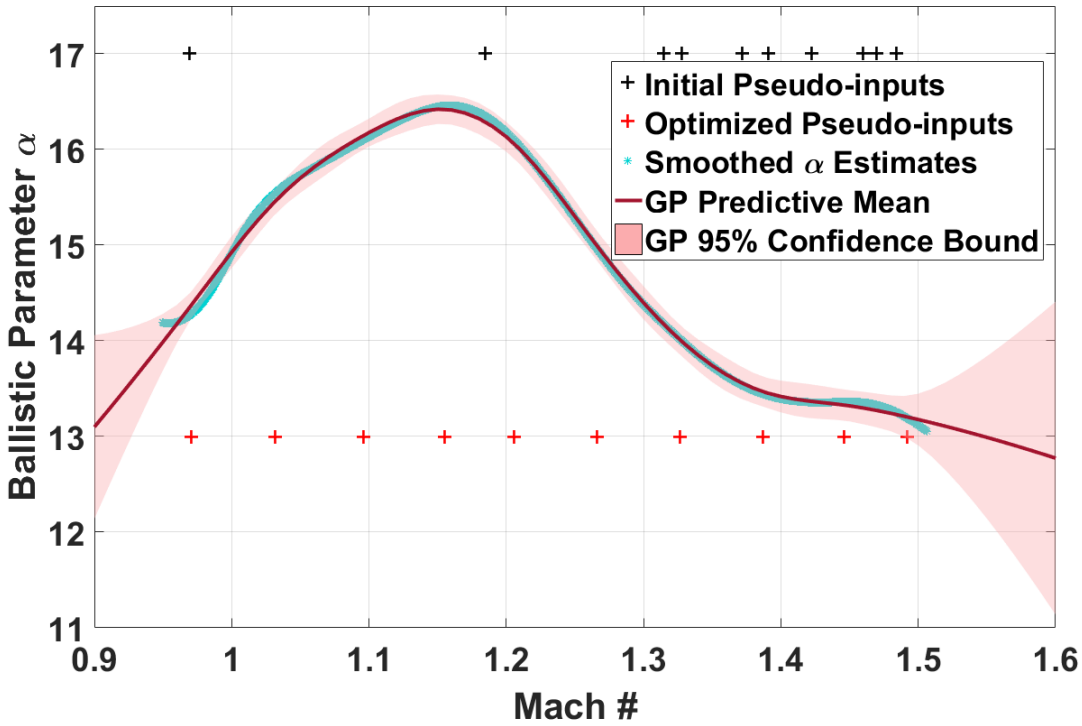
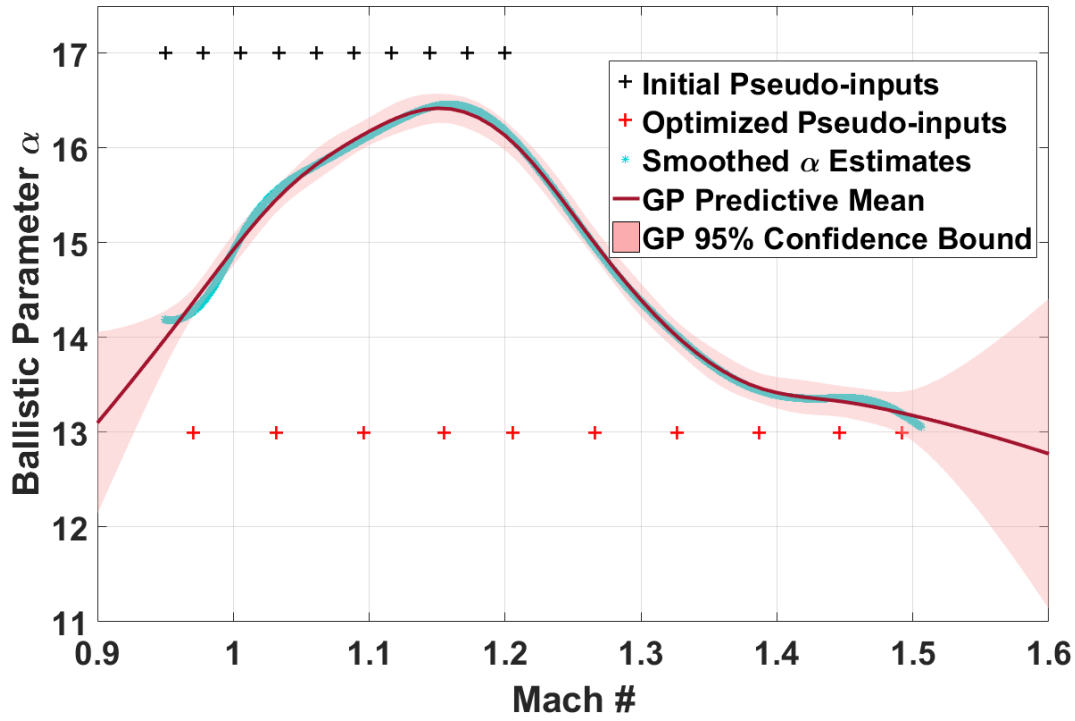
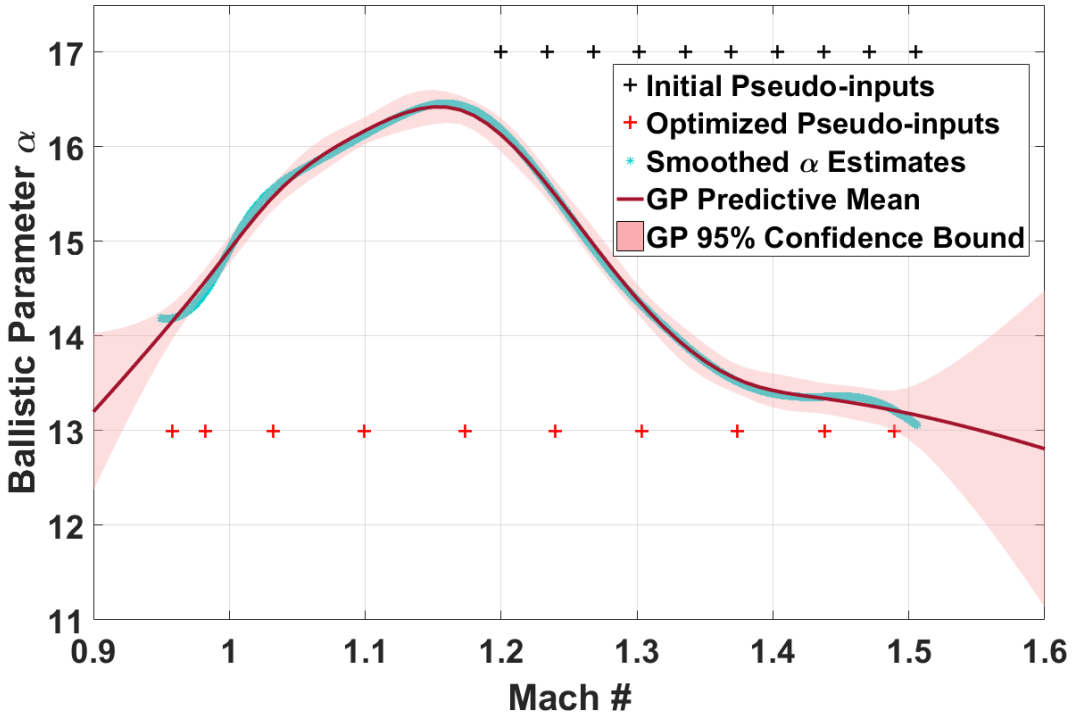


Figure 6.11: Predictive distribution of SPGP



(a) Equally placed along left half side



(b) Equally placed along right half side

Figure 6.12: Predictive distributions of SPGP under various initializations

Figure 6.11-6.12 show that the pseudo-inputs are spread over the range of smoothed BP estimates at almost equal distances regardless of their initial values. Moreover, even though the positions of pseudo-inputs differ slightly in Figure 6.12b, GP regressions under three different initializations are very similar. For these initializations, the optimized values of the hyperparameters and the maximized value of the log likelihood are given in Table-6.4.

Table 6.4: Optimization results of the different initializations of pseudo-inputs

Initialization	σ	l	Lik. function
from an uniform dist.	11.0161	1.3088	-1514.5
at left half side	11.0153	1.3087	-1514.5
at right half side	11.0093	1.3011	-1514.4

The table shows that the values of the optimized hyperparameters and the likelihood functions at different initializations do not differ significantly. Hence, it is concluded that the optimization can be satisfactorily performed without depending on the initialization.

To evaluate the performance of SPGP method, predictive distributions of optimized full GP and optimized SPGP are illustrated in Figure 6.13 for Projectile-I.

When compared with full GP regression, both predictive means and confidence bounds are very close to each other. The minor inconsistency only exists at the points, outside the estimate range.

In the SPGP method, the only parameter, which has to be designated before the execution of the method, is the number of pseudo-inputs. This parameter should be specified by taking the range of input space and the degree of the nonlinearity of the smoothed α estimates into account. To show the effect of the number of pseudo-inputs on the regression, predictive distributions with different numbers of pseudo-inputs are illustrated in Figure 6.14, respectively. They are all initialized at the left half side of the estimate range and optimized in the same manner.

In the case of 5 pseudo-inputs in Figure 6.14a, the predictive mean is underfitting

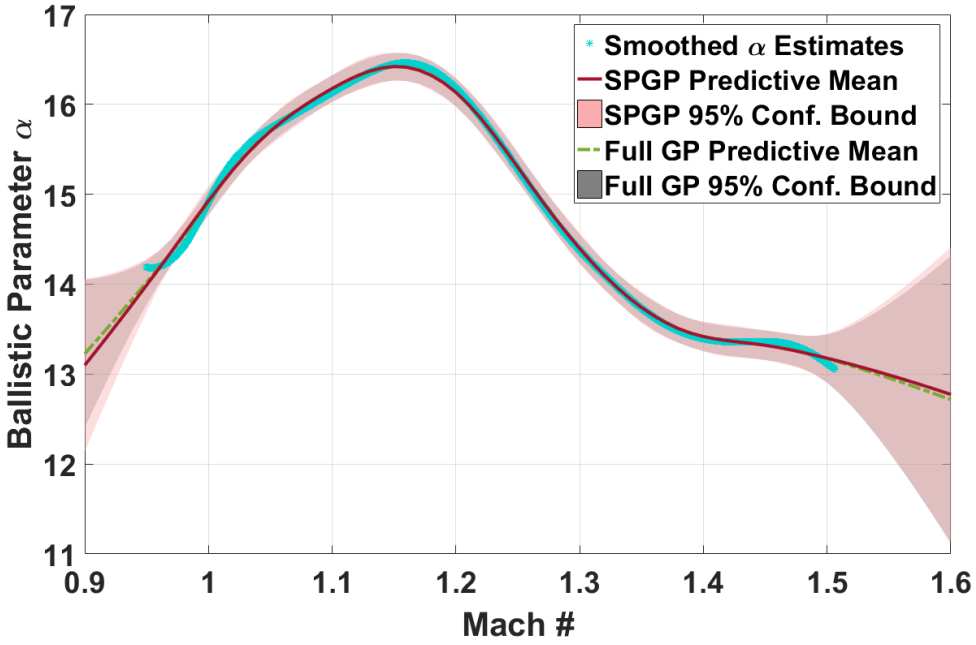
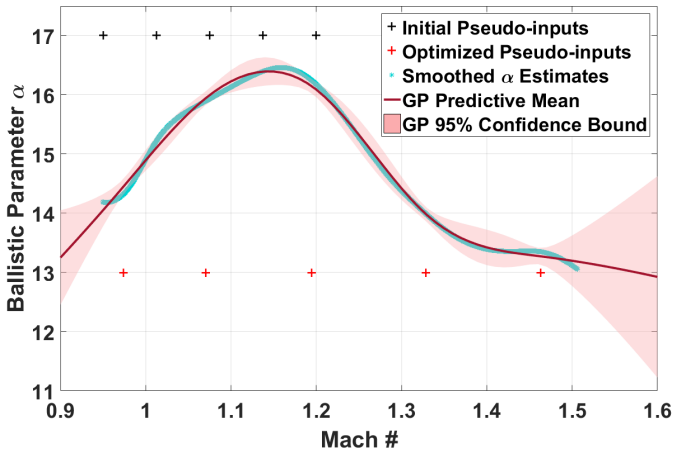
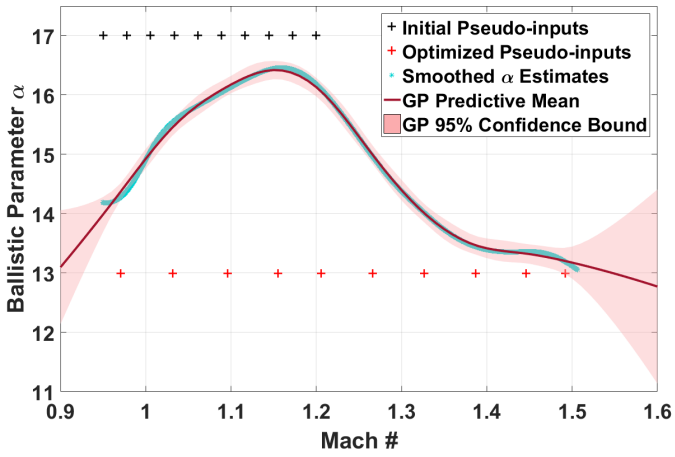


Figure 6.13: Predictive distributions of SPGP and full GP

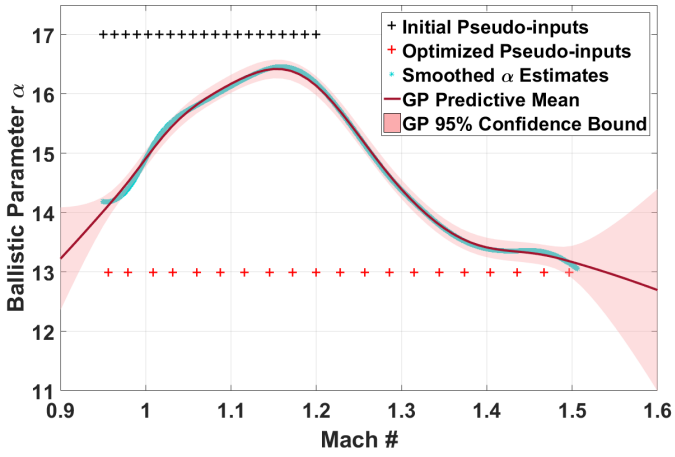
the estimates. Also, the confidence bound enlarges unnecessarily when the point moves away from the pseudo-inputs. On the contrary to 5 pseudo-inputs, GP regressions based on 10 and 20 pseudo-inputs better fits to the estimates without overfitting. However, using 20 pseudo-inputs does not enable the regression to represent the characteristics of the estimates better than the 10 pseudo-inputs. To conclude, using a few pseudo-inputs may result in an inadequate representation of the estimates; whereas, using too many pseudo-inputs yields an increase in the computational load without providing any significant improvement on the performance at all.



(a) 5 pseudo-inputs



(b) 10 pseudo-inputs



(c) 20 pseudo-inputs

Figure 6.14: Effect of the number of pseudo-inputs on SPGP regression

CHAPTER 7

CONCLUSION

In this thesis, a novel approach for modeling ballistic parameter of ballistic targets in the system model is proposed. This new model is then used in the prediction phase of the IPP problem to enhance the prediction accuracy. The method utilizes Gaussian process to model the relationship between the ballistic parameter and Mach number of a target. In the method, GP conditions on the smoothed ballistic parameter and Mach number estimates for regression of BP values in the prediction phase. This regression enables the prediction of BP more accurately and the more accurate predictions result in a significant enhancement in IPP accuracy as demonstrated by the simulations.

The main contribution of the method requires conditioning on the filter estimates which is troublesome. For example, GP observations have correlations in time which has to be taken into account for a proper regression. Also, GP inputs are all noisy and they have correlations in time similar to the observations. There are also cross correlations among the inputs and observations. These circumstances violate the fundamental assumptions of standard GP method and creates a special need to modify it. In this study, these challenges are addressed and handled in elegant and efficient ways.

In the simulation environment, firstly, characteristics of the filtered and smoothed ballistic parameter estimates are studied to have an insight. It appears that the smoother eliminates most of the time delay and noise in the filtered estimates. Then, the aforementioned modifications on standard GP model are discussed by examining their effects on GP regression mostly. Later, ballistic parameter prediction performance

of the proposed model is illustrated and the IPP performance is compared by conventional and reference models for three projectiles. It has been concluded that the proposed model outperforms the conventional one in terms of RMS IPP error and has a slightly worse performance than the reference model.

After validating the method and demonstrating its performance on the IPP problem, enhancements on the method to improve the accuracy and the efficiency are studied. The online optimization of the hyperparameters of the kernel is performed to get more accurate GP regression. In the analysis, it is shown that the online optimization makes a limited contribution to the IPP performance whereas its computation is an exhaustive task. It is demonstrated that using the same predetermined hyperparameters for all projectiles shows a comparable performance with the optimized hyperparameters. It is due to the fact that the characteristic of the ballistic parameter profiles of the most projectiles are very similar; therefore, there is no need for the hyperparameters to be optimized on-the-fly. Moreover, in order for the method to be more efficient, an approximation to GP method, namely SPGP, is implemented on our problem. This method requires the predetermining the number of the pseudo-inputs and the online optimization of the positions of them. It is concluded that the optimization results in pseudo-inputs at equal distances apart in the range of the estimates. It stems from the fact that the estimates are continuous and have similar nonlinear characteristics at every point in their range. Therefore, we could say that every portion of the function deserve to be represented with the same number of pseudo-inputs. Note that, a function having a rapid change at a point requires to be represented with more pseudo-inputs around that point to capture this rapid change [32]. As a result, in our problem, it is advised to pass the optimization over and to position the pseudo-inputs in the range of the estimates at equal distances apart.

Learning ballistic parameter-Mach number relationship based on estimates of the tracker and using this information in the prediction phase is a novel approach at BTT literature, and can have several areas of usage besides enhancing the IPP performance. For instance, using this information not only at the prediction phase but also at filtering can be possible. By doing so, tracking accuracy as well as the IPP accuracy can

be improved. Moreover, this learned information is target specific and can be very beneficial for the classification of the target type. Besides, the method can be easily extendable to multiple targets such that GP regression can be based on BP and Mach number estimates of multiple targets belonging to the same projectile type. Such a method results in learning BP characteristic more accurately at the expense of clustering data of different targets to gather the same type of target and taking the risk of misclassification. These studies are planned as future work of the current study.

REFERENCES

[1] J. D. J. Anderson. *Introduction to Flight*. McGraw Hill Higher Education, 2000.

[2] Arrow Tech Associates Inc. PRODAS V3, Software Tool for the Ballistics Professional. `http://www.prodas.com`. Accessed: 2019-23-09.

[3] A. Benavoli, L. Chisci, and A. Farina. Tracking of a ballistic missile with a-priori information. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):1000–1016, 2007.

[4] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Norwood, MA, USA: Artech House, 1999.

[5] M. G. S. Bruno and A. Pavlov. A density assisted particle filter algorithm for target tracking with unknown ballistic coefficient. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2005.

[6] J. Q. Candela. *Learning with uncertainty-Gaussian processes and relevance vector machines*. PhD thesis, Technical University of Denmark, 2004.

[7] L. Clancy. *Aerodynamics*. Pitman Aeronautical Engineering Series. Wiley, 1975.

[8] S. Conover, J. C. Kerce, G. Brown, L. Ehrman, and D. Hardiman. Impact point prediction of small ballistic munitions with an interacting multiple model estimator. In *Proc. Acquisition, Tracking, Pointing, and Laser Systems Technologies XXI*, 2007.

[9] J. L. Crassidis and J. L. Junkins. *Optimal estimation of dynamic systems*. Chapman & Hall/CRC, 2011.

[10] D. Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

[11] B. Etkin. *Dynamics of atmospheric flight*. Courier Corporation, 2012.

[12] A. Farina, D. Benvenuti, and B. Ristic. Estimation accuracy of a landing point of a ballistic target. In *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2002.

[13] A. Farina, B. Ristic, and D. Benvenuti. Tracking a ballistic target: Comparison of several nonlinear filters. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):854–867, 2002.

[14] R. Frigola-Alcade. Bayesian time series learning with Gaussian processes. *University of Cambridge*, 2015.

[15] H. Goyder. Lecture notes in fundamental of ballistics, October 2017.

[16] P. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*. Artech House, 2013.

[17] D. F. Hardiman, J. C. Kerce, and G. C. Brown. Nonlinear estimation techniques for impact point prediction of ballistic targets. In *Proc. Signal and Data Process. of Small Targets*, 2006.

[18] R. Herbrich, N. D. Lawrence, and M. Seeger. Fast sparse Gaussian process methods: The informative vector machine. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2003.

[19] J.-K. Jung and D.-H. Hwang. The novel impact point prediction of a ballistic target with interacting multiple models. In *Proc. IEEE Int. Conf. on Control, Autom. and Syst. (ICCAS)*, 2013.

[20] K. Krishnamoorthy. *Handbook of statistical distributions with applications*. Chapman and Hall/CRC, 2016.

[21] R. L. McCoy. *Modern Exterior Ballistics : The Launch and Flight Dynamics of Symmetric Projectiles*. Schiffer Pub., 2012.

[22] A. McHutchon and C. E. Rasmussen. Gaussian process training with input noise. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[23] National Geospatial-Intelligence Agency. World Geodetic System 1984 (WGS 84), 2019.

[24] R. M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. *arXiv preprint physics/9701026*, 1997.

[25] R. Nennstiel. How do bullets fly? *AFTE Journal*, 28(2):104–143, 1996.

[26] D. Nguyen-Tuong, J. R. Peters, and M. Seeger. Local Gaussian process regression for real time online model learning. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2009.

[27] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[28] V. C. Ravindra, Y. Bar-Shalom, and P. Willett. Projectile identification and impact point prediction. *IEEE Transactions on Aerospace and Electronic Systems*, 46(4):2004–2021, 2010.

[29] F. Reali, G. Palmerini, A. Farina, A. Graziano, and L. Timmoneri. Tracking a ballistic target by multiple model approach. In *Proc. IEEE Aerospace Conf.*, 2009.

[30] B. Saulson and K. C. Chang. Nonlinear estimation comparison for ballistic missile tracking. *Optical Engineering*, 43(6):1424–1439, 2004.

[31] M. Seeger, C. Williams, and N. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. Technical report, 2003.

[32] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2006.

[33] M. L. Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.

[34] T. A. Talay. *Introduction to the Aerodynamics of Flight*. National Aeronautics and Space Administration, 1975.

[35] N. Wahlström and E. Özkan. Extended target tracking using Gaussian processes. *IEEE Transactions on Signal Processing*, 63(16):4165–4178, 2015.

[36] R. Waltz, J. Morales, J. Nocedal, and D. Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical Programming*, 107(3):391–408, 2006.

[37] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.

[38] D. Young, B. Munson, T. Okiishi, and W. Huebsch. *A Brief Introduction To Fluid Mechanics*. John Wiley & Sons, 2010.

[39] T. Yuan, Y. Bar-Shalom, P. Willett, E. Mozeson, S. Pollak, and D. Hardiman. A multiple IMM estimation approach with unbiased mixing for thrusting projectiles. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):3250–3267, 2012.

[40] Z. Zhao, H. Chen, G. Chen, C. Kwan, and X. R. Li. Comparison of several ballistic target tracking filters. In *Proc. IEEE American Control Conf.*, 2006.