Physics-Based Lidar Simulation and Wind Gust Detection and Impact Prediction for Wind Turbines

by

Kai Zhou

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2019 by the
Graduate Supervisory Committee:

Ronald Calhoun, Chair
Kangping Chen
Wenbo Tang
Yulia Peet
Raghavendra Krishnamurthy

ARIZONA STATE UNIVERSITY

May 2019

ABSTRACT

Lidar has demonstrated its utility in meteorological studies, wind resource assessment, and wind farm control. More recently, lidar has gained widespread attention for autonomous vehicles.

The first part of the dissertation begins with an application of a coherent Doppler lidar to wind gust characterization for wind farm control. This application focuses on wind gusts on a scale from 100 m to 1000 m. A detecting and tracking algorithm is proposed to extract gusts from a wind field and track their movement. The algorithm was implemented for a three-hour, two-dimensional wind field retrieved from the measurements of a coherent Doppler lidar. The Gaussian distribution of the gust spanwise deviation from the streamline was demonstrated. Size dependency of gust deviations is discussed. A prediction model estimating the impact of gusts with respect to arrival time and the probability of arrival locations is introduced. The prediction model was applied to a virtual wind turbine array, and estimates are given for which wind turbines would be impacted.

The second part of this dissertation describes a Time-of-Flight lidar simulation. The lidar simulation includes a laser source module, a propagation module, a receiver module, and a timing module. A two-dimensional pulse model is introduced in the laser source module. The sampling rate for the pulse model is explored. The propagation module takes accounts of beam divergence, target characteristics, atmosphere, and optics. The receiver module contains models of noise and analog filters in a lidar receiver. The effect of analog filters on the signal behavior was investigated. The timing module includes a Time-to-Digital Converter (TDC) module and an Analog-to-Digital converter (ADC) module. In the TDC module, several walk-error compensation methods for leading-edge detection and multiple timing algorithms were modeled and tested on simulated signals. In the ADC module, a benchmark (BM) timing algorithm is proposed. A Neyman-Pearson (NP) detector was implemented in the time domain and frequency domain (fast Fourier transform (FFT) approach). The FFT approach with frequency-domain zero-paddings improves the timing resolution. The BM algorithm was tested on simulated signals, and the NP detector was evaluated on both simulated signals and measurements from a prototype lidar (Bhaskaran, 2018).

DEDICATION

Dedicate to my beloved grandparents, Mr. Zhongxing Zhou and Mrs. Su'e Gao.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Professor Ronald Calhoun, for supporting my Ph.D. study and for his enthusiastic encouragement, invaluable advice, and patience. His instrumental guidance did not only benefit my academic research, but also helped me develop scientific thinking and prepared me for my career. I will never forget the lesson he taught me about how to understand a problem - understand the physics. This lesson enlightened my entire research time and will also guide me in my future life.

I would like to extend my sincere thanks to my committee: Dr. Yulia Peet, Dr. Kangping Chen. Dr. Wenbo Tang, and Dr. Raghavendra Krishnamurthy, for their insightful comments and encouragement which widens my research and makes me think deeper and harder.

I also would like to thank Dr. Werner Dahm, Dr. Huei-ping Huang, Dr. Ronald Adrian, Dr. Patrick Phelan, and Dr. Marcus Herrmann, who provided me initial guidance in my coursework and helped me develop my fundamental engineering knowledge.

The ASU lidar group would like to acknowledge the generous support of ONR's Neptune Program (Program Manager: Rich Carlin Ph.D.), ASU Lightworks (Mr. Bill Brandt, Director Strategic Integration, ASU lightworks) and the Electric Power Research Institute (EPRI, Aidan Tuohy Ph.D.).

I would like to acknowledge the help of Dr. Aditya Choukulkar. Both Dr. Aditya Choukulkar and Dr. Raghavendra Krishnamurthy provided insightful suggestions at the beginning of my Ph.D. program, which made my research started smoothly. I also would like to thank all my groupmates, Dr. Nihanth Wagmi Cherukuru, Dr. Sreevatsan Bhaskaran, for the stimulating discussions and the helpful advice on coursework, research and field trips. The days and nights in Tehachapi, California for the li-

dar deployment and in the lab for the prototype lidar testing will forever live in my memory.

I would like to thank my friends Xiaoyu Sun, Ruijin Cang, Hope Yao, Cheng Lv, Xiao Liao, Haotong Wang for the encouragement to go through the tough times and all the fun.

Last but not least, I would like to thank my parents for supporting me spiritually throughout all my life.

TABLE OF CONTENTS

CHAPTER Page

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

BACKGROUND

1.1 Wind Gust Detection and Impact Prediction for Wind Turbines

Rapid changes of wind speed in the atmosphere, also called wind gusts, cause large fatigue loads on wind turbines. These loads reduce the lifetime of wind turbine components. Oscillations or ramping of the generated power can result in fast fluctuations of grid voltage and may pose additional burdens to the electric grid. To investigate the profile of the wind interacting with wind turbines, researchers have conducted measurements of wind field at and behind the rotor-swept area of a wind turbine (Kumer *et al.*, 2015; Smalikho *et al.*, 2013; Aitken *et al.*, 2014; Wang and Barthelmie, 2015; Banta *et al.*, 2015; Zhou and Calhoun, 2016). Additionally, researchers have proposed adaptive and feed-forward control systems, which can adjust wind turbine settings for approaching winds (Schlipf *et al.*, 2012; Scholbrock *et al.*, 2013; Kumar *et al.*, 2015; Kristalny and Madjidian, 2011; Zhou and Calhoun, 2016). A feed-forward control system requires an accurate and fast gust detection system. Our purpose is to provide the information of wind gusts to the control systems.

There are a variety of gust detecting and tracking algorithms in the literature. The international standard IEC 61400-1 specifies standardization of several temporal gust models for wind turbine design. Branlard defined gusts as a short-term wind speed variation within a turbulent wind field (International Electrotechnical Commission *et al.*, 2005; Branlard, 2009). Although different definitions exist, all suggest gusts invoke rapid wind speed changes. As many atmospheric sensors measure winds in relatively small volumes, changes in wind speed have been considered temporally.

Temporal variations of wind speed can be converted to spatial variations using Taylor's frozen turbulence hypothesis. Note that spatial variations in pressure on wind speed also damage buildings (Beljaars, 1987). However, fewer studies directly measure spatial variations. The lack of such studies could be due to the limitation of the available instruments. Anemometers on met masts are the most common instruments for gust studies, but they are limited to measuring wind speed at fixed points. On the other hand, Doppler lidar can address this limitation. A long-range Doppler lidar can provide wind velocity field in a 3D domain up to 10 km with high temporal and spatial resolution.

The scale of a spatial gust is an important factor. Kelley *et al.* (2004) and Chamorro *et al.* (2015) studied the flow-structure interaction between wind turbines and atmospheric coherent structures with a scale ranging from the size of a wind turbine rotor to the thickness of the atmospheric boundary layer. They found that structures primarily in that range have a high correlation with the generated power and can induce strong structural responses. For structures with scales smaller than the size of a wind turbine rotor, the effects of their high-frequency components will be averaged out along the turbine blades and will not propagate to the drive train of a wind turbine and affect the power generation. For atmospheric gusts larger than 1000 m, varying winds in these large-scale gusts can be classified and captured as "meandering of the wind." Effects of the wind meandering on wind turbines are complicated and begin to become relevant for yaw control. Therefore, we believe the gusts with scales between 100 m and 1000 m have the most significant effects on wind turbine performance.

In addition to the definition of wind gusts, the literature contains a variety of ideas for gust detecting and tracking. Mayor *et al.* adapted two computer-vision methods for flow motion estimation: the cross-correlation method and the wavelet-based optical flow method (Mayor *et al.*, 2012, 2013). However, the cross-correlation method has limitations for non-uniform velocity fields, and the optical flow method requires relatively small (few pixels) movement and is computationally demanding. On the other hand, Branlard proposed several detection methods for different gust models in Branlard (2009), but gusts are defined in the time domain. He also estimated the arrival time of the gusts and presented an exponential probability distribution of the gust's spanwise propagation. However, the gust distribution may not be representative, as the turbulence induced by the blades would heavily contaminate the signal collected behind the turbine rotor. The number of measurement points from anemometers is also limited. Therefore, a fast detecting and tracking algorithm and a reliable prediction model that can be used in real-time prediction of spatial gusts need to be developed.

1.2 Time of Flight Lidar

1.2.1 Overview of Time of Flight Lidar Techniques

Ranging sensors

With the emergence and evolution of autonomous driving vehicles in the last five years, many effects have been made to provide an intelligent and safe driving environment. The brain of an autonomous vehicle consists of several core advanced algorithms like perception, localization, path planning, and vehicle control, and examples of the advanced driving assistant system (ADAS) includes departure warning system, collision avoidance system, and self-parking system. All the systems directly or indirectly require sensing of the environment from various perspectives, and the range between the ego car and surrounding objects is one of the indispensable information.

Common ranging sensors include cameras, ultrasonic sensors, radars, and lidars, but each sensor has its advantages and shortcomings under different conditions. Triangulation that uses two cameras as a passive stereoscope is the most common approach of range sensing for cameras, and it has advantages of offering high-resolution depth maps. However, the high dependency on the lighting condition, as for any passive camera-based approaches, make triangulation method subject to noise from ambient light and less reliable on poor lighting conditions. Also, features of small objects at a remote distance are also difficult to be identified by a camera. Even though advanced computer vision algorithms could compensate this drawback, intense computation effort is a demand for computer vision algorithms (Hussmann *et al.*, 2008). Structured-light sensors provide a solution applying artificial structured light to address the lightening issue. However, this approach is limited to the indoor application due to its high sensitivity to environment light (Kadambi *et al.*, 2014). Compared to cameras, ultrasound sensors are alternative sensors of depth measurement, but it has a limited range of detection (a few meters) due to the strong attenuation by air. Moreover, the dependency of sound propagation on humidity and air pressure also makes the measurement less robust to complex atmospheric conditions (Süss *et al.*, 2016; Fersch *et al.*, 2017). Radar is a supplementary for ultrasonic sensors because of its immunity to varying environmental conditions, which also allows sensing of the velocity of objects by using Doppler effect. However, radar has a difficulty of detection of objects with high absorption of transmit frequency band (Fersch *et al.*, 2017). Moreover, since major targets are close to the ground in the application of autonomous vehicles, radars suffer strong ground clutters, which is caused by reflection of slide lobes of a waveform by ground surfaces. In addition to the ground cluster, the low resolution of sensing image also makes radars only useful in applications like collision warning, but less reliable on high-precision functions like object classification or localization. Lidar, short for Laser Imaging, Detection And Ranging, has drawn increasing attention for range sensing from last decade. Compared to ultrasonic sensors, lidar has its advantages thanks to its more robustness to environmental conditions and the independence on the surrounding lighting. Moreover, the narrow laser beams allow a high-resolution depth map than that radar can produce. Fast and adjustable scanning patterns with advanced control systems also allow dense and high-precision sensing of remove objects that could fail a camera, which provides a complementary tool of cameras for perception applications. However, complicated atmospheric conditions, like rain, fog or snow are still a challenge of lidar which could lead to in noisy result or reduced maximum observable distance, and the cost of lidar sensors is also higher than cameras or radars.

Using a lidar sensor, the range between the sensor and an object can be calculated from the time of flight (TOF) of the laser emitted from a lidar, and the TOF measurement can be classified into direct TOF method and indirect TOF method. In the direct TOF method, narrow laser pulses are emitted. After being reflected by an object, the return pulses are received by a photon-receiver (receiver), and the flying time of the pulse until its reception is measured by an electronic counter. On the other hand, indirect TOF method usually utilizes continuous wave laser and the phase and power information of the incident light to obtain the TOF. Depending on if frequency or amplitude of the wave is modulated, the indirect TOF methods can be divided into frequency-modulated continuous wave (FMCW) method and amplitude modulated continuous wave (AMCW) method. Since this dissertation focuses on direct detection TOF lidar using pulsed laser, only the previous work on direct TOF method will be reviewed next, and readers could refer to (Pierrottet *et al.*, 2008; Poulton *et al.*, 2017; Ebben *et al.*, 1995; Adams and Probert, 1996; Adams, 2000; Godbaz *et al.*, 2011; Banks, 2013; Jo *et al.*, 2016; Zhang *et al.*, 2017; Schmidt *et al.*, 2012; Jo *et al.*, 2016) for details of in-direct TOF methods.

Direct TOF method

The direct TOF method can be further classified according to the number of pulses required for the TOF calculation. *Single-shot method* provide a TOF product using a single return, in which linear-mode avalanche photon-detector (APD) is the common choice. Single-shot TOF method has been well developed in academic (Burns *et al.*, 1991; Kilpelä *et al.*, 1998) and are also available in industry, like Velodyne. Velodyne HDL-64E lidar and its similar products have a mechanical scanning system that emits one or a line of laser pulses each time, and rotates the scanner with a certain speed to cover a wide field of view (FOV). On the contrary, solid-state lidars avoid the mechanical scanning components, but direct the beams using optical phase array (Heck, 2017), or spreads the light using an optical diffuser to illuminate the entire scene of interest. The latter one is also referred to as a flash lidar, which is another common lidar design for autonomous driving vehicles. At the receiver end of a flash lidar using the single-shot method, a focal plane array (FPA) with 2D linear-mode APD's is used as a photo-sensor, and behind each pixel, an APD is integrated with a circuit that amplifies and filters the input analog signals and convert the signals to TOF measurements. The flash lidar design provides 2D TOF images each time the light is emitted, rather than a dot or a line as a scanning lidar does. Examples of using such FPA for single-shot method includes the work done by Richmond *et al.* (1996), and commercial products of Voxtel and advanced scientific concepts (ASC), Inc. (McManamon *et al.*, 2017). Both the scanning and flash lidar design using the single-shot measurement method demonstrates the advantage of allowing high-quality TOF products at a single laser event. Moreover, the LM-APD is less sensitive to ambient light compared to Geiger-mode APD's used in multiple-shot method that will be introduced next. One drawback of the single-shot method is that the gain of the photon-detector is relatively low, which requires high laser energy to ensure weak signals beyond the noise level. Additionally, fast electronic clocks and high bandwidth of the detection circuits are also required to achieve a distance resolution of a few centimeters. However, high bandwidth could lead to increased system noise and reduced signal SNR, which makes the weak signal less detectable from noise.

In addition to linear-mode APD's, another widely applied photon-detectors for TOF lidars are Geiger-mode APD's (GM-APD's) (Aull, 2016; Becker, 2015; Niclass *et al.*, 2007; Niclass, 2008; Stoppa and Simoni, 2011; Albota *et al.*, 2002; Beer *et al.*, 2018). Due to the extreme high gain, a GM-APD has the capability to detect extremely weak signals down to single photon level. Therefore, such a photon-detector is also referred to as single-photon avalanche diodes (SPAD's) (with exception that some designs using LM-APD's are also belonged to SPAD's (Kasunic, 2016)). However, the high sensitivity of GM-APD's makes it extremely susceptible to other events than the desired photons reflected from targets, and the undesired events could arise from various sources like ambient light (*i.e.* sunlight), dark current, and backscatter from unintended objects. Therefore, signals from multiple returns have to be averaged to reduced the noise, which is why such approach is also called *multi-shot method*. In addition to the high false positive events, the Geiger-mode avalanching process makes GM-APD's unable to measure the signal intensity, instead, only binary intensity can be provided. In light of these drawbacks, GM-APD's are commonly employed in a group of arrays, which is also called multipixel photon counters (MPPC), and the MPPC is usually applied in a flash lidar design integrated with digital CCD or CMOS timing circuits (Kasunic, 2016). Each MPPC consists a 2D array of GM-APD's and produces a single TOF product (a pixel value), calculated from the measurements of all the GM-APD's, rather than an image has multiple pixels (Kasunic, 2016). At each pixel, the MPPC produces a histogram of incident event against time. The time stamp of the histogram is recorded by a high-precision Time-to-Digital Converter (TDC), and the number of events at each timestamp is obtained by averaging the measurements over multiple returns (Niclass *et al.*, 2013). Then, the TOF is obtained from the time stamp of the peak of the histogram, and a coincidence detection (time-correlation) technique could be applied to reduce the false positives in the measurements (Niclass *et al.*, 2013; Beer *et al.*, 2017; Steinvall *et al.*, 2008). The advantage of using GM-APD's is that the nearly infinite gain enable the detection of extremely weak signal and reduces the requirement of laser energy. However, if it is used in a flash-lidar design, the wide transmitter FOV also causes intensity drops. As will be seen in Section 4.1.2 that, the power drops to the 4-th power with distance. Therefore, to detect signals from remote distance, the laser power is still a major concern for flash lidars using MPPC's.

1.2.2 Overview of TOF Lidar Simulation

With the growth of various lidar technologies, the simulation of those systems have become a valuable tool for the evaluation of the performance under various conditions and the investigation of different components for new system development. In general, a lidar system consists of laser source, transmitter and receiver lens systems, photon-detectors and signal processing units. Since the overall performance of a lidar system relies on the functionality of all the components and the interaction between each other, a full-system lidar simulation that covers all the major components is of importance, rather than analyzing each component individually. Correspondingly, a full-system lidar simulator could be divided into modules that account each stage of a lidar system, and the modules include laser source module, propagation module (including propagation through free space, and reflection by targets), receiver module, optics module (transmitter and receiver) and signal processing module (or timing module). A schematic of a full-system lidar simulation is shown in Figure 1.1. Researches have conducted extensive investigation on the full-system lidar simulation with various emphases. Depending on purposes, some modules are modeled in more depth and the others of less interests are simplified. Previous efforts on different modules of a full-system lidar simulation are reviewed next.

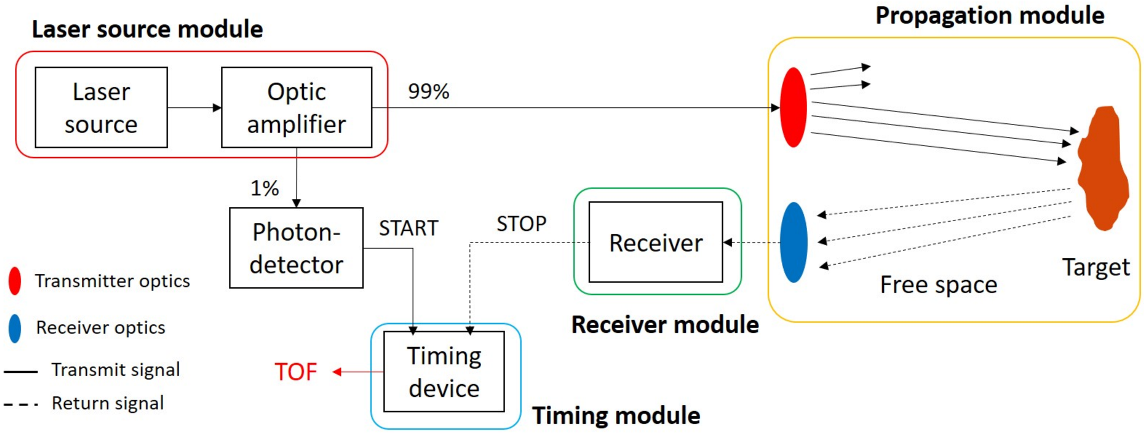


Figure 1.1: Schematic of Full-System Lidar Simulation Modules.

The laser source module of a lidar simulation is to model the temporal and spatial power distribution of a laser pulse generated by a laser source, given laser specification like pulse width, rise time and peak power. Depending on the complicity, a laser intensity can be simulated by a volume distribution (3D) or approximated by a Gaussian curve in 2D neglecting the power distribution of the cross-section (Chevalier and Steinvall, 2009; Al-Temeemy, 2017; Richmond and Cain, 2010). Budge *et al.* (2006) also proposed a sophisticated model which divides the laser beam to sub-samples to account the beam divergence.

Models in the propagation module simulate the absorption, scattering, and refraction by targets with different geometries and particles in the air (*e.g.* dust, water droplets, and ice crystal) under various atmospheric conditions (*e.g.* turbulence). Several atmospheric models have been developed (refer to Section 4.1.3), but they are rarely implemented in a full-system lidar simulator. Among those atmospheric effects, power attenuation by the medium is the most covered effect by existing simulators. Der *et al.* (1997) took the speckle noise and scintillation due to atmospheric turbulence into account, and Chevalier and Steinvall (2009) modeled the laser beam distortion caused by turbulence, including scintillation, beam broadening, and beam wandering on output images. Richmond and Cain (2010) proposed a comprehensive model of laser beam propagation through free space in which speckle noise is taken into account. Broad studies are also conducted on the influence of characteristics of targets on laser propagation. Chevalier and Steinvall (2009) took into account the target size, shape, and bidirectional reflectance distribution function (BRDF) in their model. Steinvall emphasized on a comprehensive study on the distortion of laser shape after reflection by different geometries, which was later included in their full-system lidar model (Steinvall, 2000; Steinvall and Carlsson, 2001). Alternatively, Al-Temeemy (2017) proposed an impulse response function to describe the changes of pulse energy and shape after interaction of laser pulse with targets.

The receiver module simulates the functionality of a receiver, including the conversion from optical power to electrical signals, and noise contamination from various sources. The characteristics of a receiver play an important role in determining the performance of a lidar sensor, and they were simulated by researchers from different aspects (Budge *et al.*, 2006; Der *et al.*, 1997; Burns *et al.*, 1991). Budge *et al.* (2006) introduced transfer functions to describe the propagation of electrical signals and noises through the electrical components, and Burns *et al.* (1991) focused on the modeling of dominant noises including shot noise and thermal noise in the system-wise lidar model. Among all the lidar simulations, noise on the signals is assumed to be white. However, noise occurring periodically or randomly in the time domain, which is also common to real lidar systems, is not taken into consideration. Moreover, no detailed analog filters were included in the aforementioned full-system lidar simulation.

The time discriminator module is another important module that simulates how to distinguish pulse from noises and how to measure the TOF by time-discrimination algorithms. The commonly modeled timing algorithms in a full-system lidar simulation include peak detection, leading-edge detection, constant fractional discrimination (CFD) and matched filter (Budge *et al.*, 2006; Al-Temeemy, 2017; Chevalier and Steinvall, 2009; Der *et al.*, 1997; Richmond and Cain, 2010). Nevertheless, no further discussion on the performance of detection algorithms was provided. Moreover, the trigger level in a leading-edge detection method and the constant fraction in a CFD are usually chosen arbitrarily without involving any detection theory. Richmond and Cain (2010) illustrated the underlying detection and estimation theory in addition to peak estimator and leading-edge detection and also provided corresponding models. However, since the noise is assumed to be white in the receiver module, behaviors of matched filters on non-white or structured noise in the time domain were not discussed. In addition, the signal input to the detection algorithms in this module is usually assumed to be obtained in advanced. Therefore, how the configuration of a TDC or an ADC affects the resolution, accuracy, and precision of TOF measurements were not further discussed.

1.2.3 Prototype Lidar Design

Due to the importance of lidar on the autonomous industry, we started a TOF lidar project, to first design a long-range, high pulse repetition frequency (PRF), highly tunable lidar at the 'whiteboard' level. The project is to augment a 'camera-first' autonomous car sensor philosophy, that is, the role of a lidar is to 'disambiguate' tough cases, since $5-10\%$ of the most difficult cases cause most accidents. Second, the project is to build and physically test components and a full lidar prototype (Bhaskaran, 2018). Third, the project is aimed at modeling, characterizing and testing various system settings and detection algorithms for this prototype lidar.

1.3 Outline of the Dissertation

In the first part of this dissertation, we focus on spatial wind gusts within a limited range of scales. We propose a practical gust detecting and tracking algorithm with low computational cost (Zhou *et al.*, 2018). The novelty of the algorithm is to utilize dispersion and transport theory to create a practical tool which can provide short-range predictions of probable impact zones downstream for puffs or gusts detected upstream with a long-range Doppler lidar. The tool can provide real-time short-term predictions of impact time and location for gusts approaching a wind farm. The propagation of the wind gusts through the wind farm is not considered in this paper. By taking the gust size into consideration, the accuracy of the prediction can be increased, and valuable wind forecasting information can be provided to the control system.

Secondly, this dissertation proposes a full-system lidar simulation that consists of all the main stages of a lidar system with emphasis on signal processing and time discrimination methods. In the simulator, the parameters of the whole lidar system are adjustable from the laser source to the timing device. Users can also test various analog filters on signals with different types of noise. Additionally, both TDC and ADC timing devices that have tunable specifications and different timing algorithms implemented are also available in the proposed simulator. This full-system simulator was proposed from the aspect of system design, and it is aiming to provide a fast and reliable tool for the evaluation of the performance of the current lidar system and exploration of different system settings for a new lidar design.

This dissertation will describe the proposed gust detecting and tracking algorithm in Chapter 2. Next, the dissertation will introduce the fundamentals of the laser pulse, propagation module, receiver and time discrimination individually in Chapter 3 - 9, followed by application of models to simulated signals or/and experimental data. Specifically, Chapter 3 focuses on the simulation of the laser source. Chapter 4 describes the models applied in the propagation module. Chapter 5 explains the receiver module, followed by the application of different analog filters. Chapter 6 discusses the Time-to-Digital converters and corresponding timing algorithms. Chapter 7 focuses on the key characteristics of the analog-to-digital converter. Chapter 8 and 9 illustrated a new ADC-based timing algorithm and the Neyman-Pearson detector. Finally, Chapter 10 gives the conclusion and future work.

Chapter 2

WIND GUST DETECTION AND IMPACT PREDICTION FOR WIND TURBINES

2.1 Methodology

2.1.1 Definition of Spatial Gusts

The spatial gust studied in the present work is defined mathematically in this section. The scale of the gust of interest ranges from $1D$ to $10D$ (D is the diameter of a wind turbine. $D = 100$ m is used, hereinafter). The magnitude of the wind speed fluctuation of a gust region, $|\boldsymbol{v}'|$, should be 1.5 times larger than the standard deviation σ of the wind speed over the wind field, *i.e.* $|\boldsymbol{v}'| > 1.5\sigma$ and $\boldsymbol{v}' = \boldsymbol{v} - \bar{\boldsymbol{v}}$, where $\boldsymbol{v}$ is the local velocity vector and $\bar{\boldsymbol{v}}$ is the mean wind velocity. The gust regions should also have good temporal coherency and local spatial connectivity around their centers. Gusts are assumed to advect roughly along the mean streamline, and the major structure should preserve during traveling.

2.1.2 Data Information and Wind Field Retrieval

The wind field utilized in this work was retrieved by a new proposed 2D variational analysis method (2D-VAR) from the measurements collected by a Lockheed Martin Coherent Technologies (LMCT) WindTracer® Doppler lidar (Louisville, CO, USA) during 25 - 27 June 2014, at Tehachapi, California (Cherukuru *et al.*, 2017). The specification of the Doppler lidar is listed in Table 2.1. The lidar was located on a hill (1450 m above sea level (ASL)) at a wind farm near Tehachapi City, which is at an altitude of 1220 m (ASL). The data was collected in a horizontal plane at 1453 m

Table 2.1: Specifications of the Doppler Lidar

Parameter	**Value**
Wavelength	1.6 μm
Pulse energy	2 mJ
Pulse repetition frequency	750 Hz
Range resolution	100 m
Blind zone	436 m
Max range	10 km

ASL which included the height of the lidar system (3 m).

The 2D-VAR method used in the present work is based on a variational parameter identification formulation (Cherukuru *et al.*, 2017). The method involves finding the best fit 2D wind velocity vector ($\boldsymbol{X}$) which minimizes a cost function: $J(\boldsymbol{X}) = \frac{1}{\Omega}\int \sum W_i C_i^2 d\Omega$, where $\boldsymbol{W}$ is a pre-defined weight matrix which determines the relative importance of the terms in the cost function. The constrains, C, are functions of the wind vector $\boldsymbol{X}$, and are comprised of radial velocity equation, tangential velocity equation for low elevation angles and the advection equation. And Ω represents the analysis domain. A quasi-Newton method is implemented for the minimization. This retrieval algorithm has the advantage of preserving local structures in complex flows while being computationally efficient with possible real-time applications.

The retrieval algorithm was used to convert the measured radial velocities to two-dimensional wind field in Cartesian coordinate in a domain of size 6 $km \times 4$ km. A sample contour plot of the retrieved wind field is shown in Figure 2.1. The temporal resolution of the retrieved results is 30 s as determined by the lidar scanning pattern, and the size of the spatial grid is 80 $m \times 80$ m as specified by the retrieval algorithm.

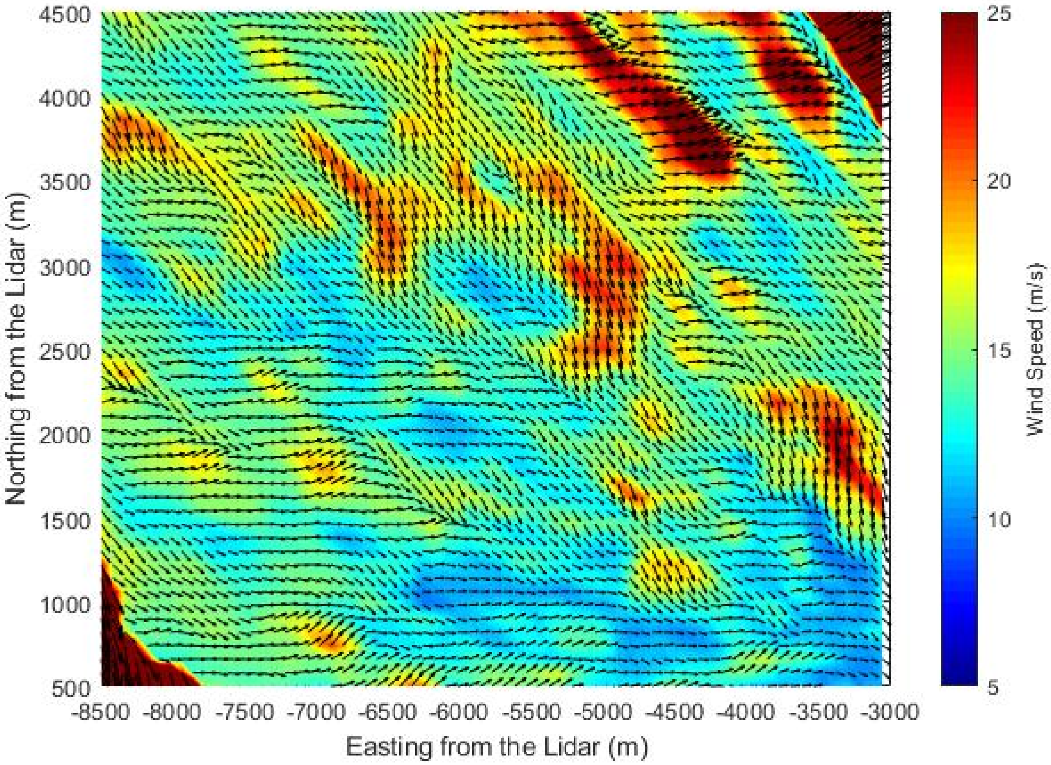


Figure 2.1: Wind Field Retrieved by the 2D Variational Analysis Method (2D-VAR) Algorithm. Background colors indicate the magnitude of the local wind speed, and the arrows show the local wind direction.

Note that even though the 2D-VAR method was used in the present work, the proposed detecting and tracking algorithm can be applied to any 2D wind field retrieved from any algorithm or obtained from any experimental instrument, given sufficient spatial and temporal resolution.

2.1.3 Data Preprocessing

Data quality control was first performed in the wind field. The dark red regions at the northeast and southwest corners in Figure 2.1 were removed. The reason the data at the two corners were removed is two-fold: First, the hills were treated as hard targets, and therefore, due to possible contamination in the range-gates immediately in front of the hard targets, the data at the hills and 1-2 range gates before the hills were removed before running the retrieval algorithm. Second, the retrieval algorithm requires the data to be treated with a Gaussian filter to minimize the effect of noise on the gradients for the advection term in the cost function. Therefore, the Gaussian filter could not be applied at the boundaries of the lidar scan due to the missing data, which caused the artifacts at the corners in Figure 2.1, so the results 1-2 more range gates before the hills were removed.

Additionally, data with very high magnitudes above 30 m/s were rejected since they were judged to be spurious for this dataset. While a small amount of still valid data might be filtered out, we found empirically 30 m/s could be a reasonable trade-off value between removing most of the noises and keeping sufficient valid data points.

Moreover, the spatial resolution of the dataset needs to be considered. The current grid size is 80 m, implying any structures smaller than the grid size will be smoothed. Only very few data points were left for atmospheric structures with scales of a few hundred meters. Because the gust extraction process in the next step (Section 2.1.4 is based on the wind speed at the data points and the size and shape of the gust patch, the truncation process may cause unnatural-looking contours of the extracted gusts. Because the shape of the small-scale gusts is reduced to few pixels connected only at the vertices, some small patches could be easily neglected during the boundary tracing process. Therefore, to avoid losing many valid small-scale patches, linear interpolation using Delaunay triangulation was used to keep the shape of the gust contour to some extent (Amidror, 2002). Delaunay triangulation is a method that triangulates the discrete points in a plane such that no discrete point is inside the circumcircle of any triangle in the triangulation of the points. In the Delaunay triangulation, same weights were assigned to the vertices of the triangular. The wind field after the interpolation is shown in Figure 2.2. However, the interpolated boundaries should not be interpreted as better approximations to true boundaries than the coarse grid. Ideally, one may avoid the interpolation step, if the lidar range-gate size were significantly smaller.

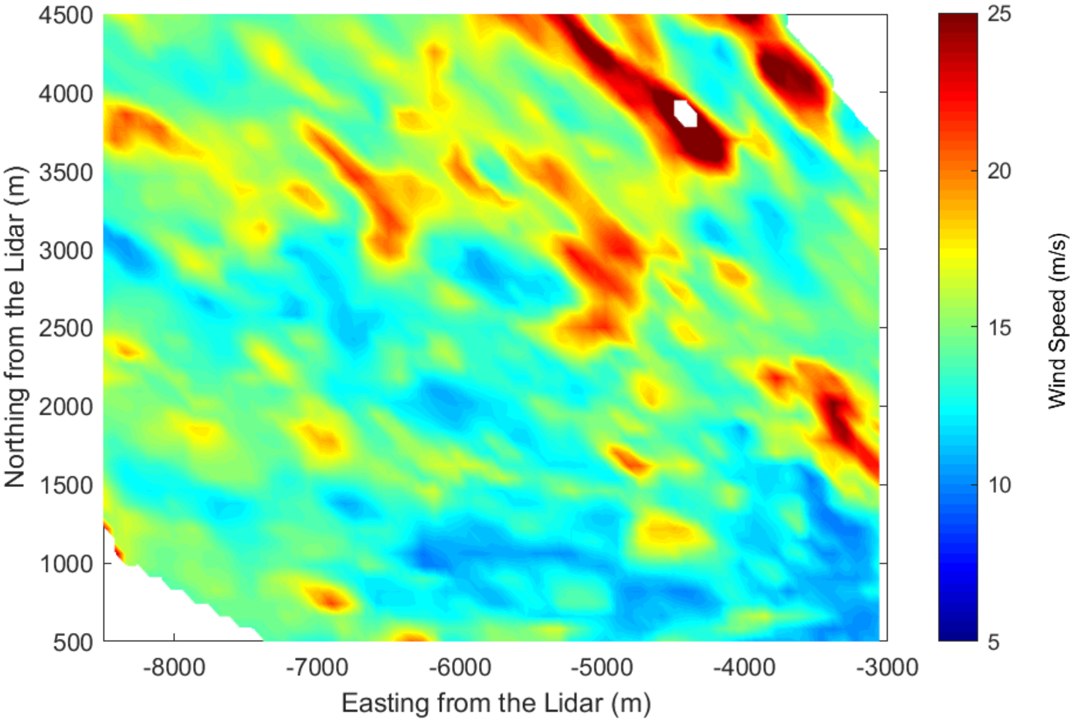


Figure 2.2: Wind Field in Figure 2.1 after Data Preprocessing. Colors indicate the magnitude of the local wind speed.

2.1.4 Detection of Gust Patches

Peak over threshold (POT) method was used to extract gust regions from the wind field (Branlard, 2009). The POT method is to detect any wind event that has an amplitude over a predefined threshold. Here, we used the POT method for both up-crossing and down-crossing the threshold. The POT method removes the data points not satisfying the velocity threshold mentioned in Section 2.1.1 and only keeps the gust regions. The resultant data was converted to binaries to facilitate the boundary tracing algorithm. The extracted and converted gust regions are shown in Figure 2.3, with gust regions in white. Note that no scale filters were applied to the patches presented in Figure 2.3.

Figure 2.3: Extracted Gust Regions in Black-White Scale from the Wind Field in Figure 2.2. The gust regions are in white.

Next, the Moore-Neighbor tracing algorithm with Jacob's termination condition was applied to the binary data to trace the boundaries of the patches (Weisstein, 2019). In the Moore-Neighbor tracing algorithm, an important concept is the Moore neighborhood which is a set of 8 pixels around a target pixel that share a vertex or edge with the target pixel. To track the boundary of a patch, as in Figure 2.3, the tracking algorithm uses any one of the white pixels on the boundary of the patch as the target pixel, and visits (moving clockwise for example) its Moore-neighbor pixels (black pixels) before entering another white pixel. Then, it uses the next white pixel (moving clockwise for example) as the target pixel and repeats the procedure. When it revisits the first white pixel it entered originally, the algorithm stops and all the visited black pixels comprise the boundary of the patch. There are two widely used termination conditions for the algorithm. The original one is to stop the algorithm after reentering the first white pixel for the second time. The other one is called Jacob's stopping condition, which also stops the algorithm after reentering the first white pixel for the second time, but in the same direction one originally enters it. Since concave shapes are common for gust regions as seen in Figure 2.3, Jacob's stopping condition was applied because it is more powerful to trace such shapes than the original stopping condition.

Moreover, since the patches with holes inside due to the above filtering still need to be considered as a whole, only exterior boundaries of the patches were traced. Additionally, patches with scale out of the range from $1D$ to $10D$ were filtered out and the centroids of the patches were calculated during the tracing process. The scale of a patch is defined as the square root of its area calculated by multiplying the grid area ($10\ m \times 10\ m$) by the number of the grid points inside or on the boundary. The retrieved wind field along with detected gust boundaries is shown in Figure 2.4.

2.1.5 Tracking of Gust Patches

To predict the future location of the gust patches, their advective characteristics need to be confirmed at first. Therefore, the next question is how to associate the patches between time frames given the detected regions. Given the formation of the spatial gust is contributed by the atmospheric turbulence, we assume that the gust regions propagate along the mean wind direction and their turbulent properties remain unchanged according to Taylor's frozen turbulence hypothesis. Nevertheless, since the time intervals between two wind fields are integers, multiples of 30 s, they

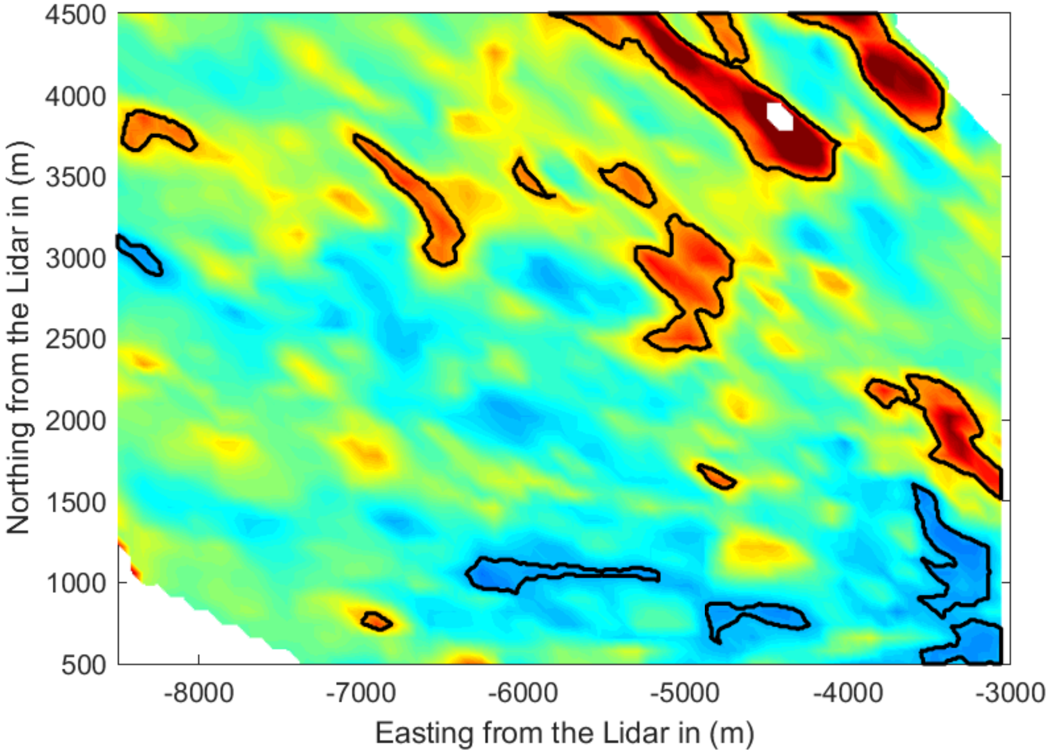


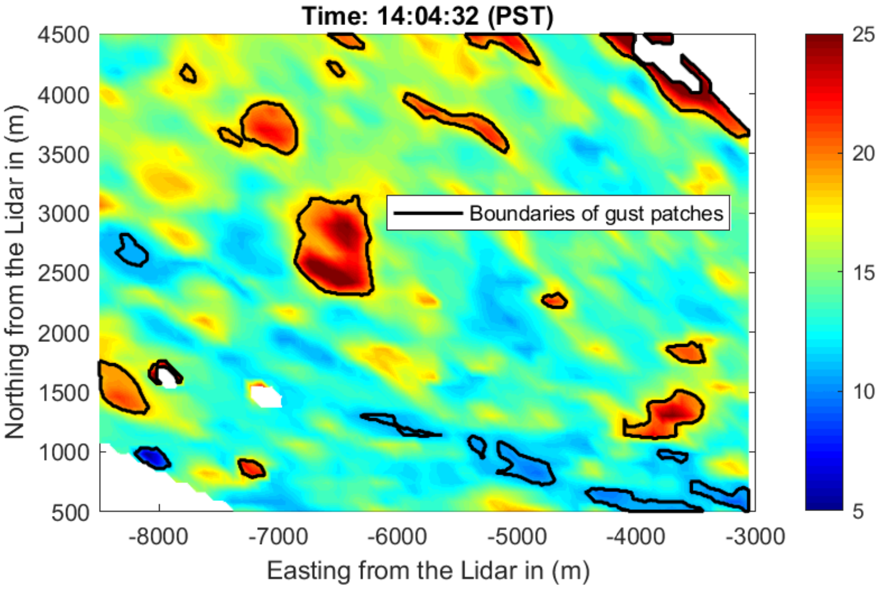
Figure 2.4: Retrieved Wind Field with Detected Boundaries of Gust Regions. Black contours are the boundaries of the detected patches.

are relatively long compared to the eddies in the atmosphere, so Taylor's hypothesis is relaxed to some extent. It means slight changes in wind speed and wind direction, variations of scales, and deformation of the shapes are allowed in the proposed algorithm.

After extracting the gust regions from the wind field at two time frames, the tracking algorithm takes the detected gusts as inputs. The patches at an earlier frame are called 'original patches,' and patches at later frame are called 'candidate patches.' The time interval between the two frames should not be larger than 90 s. Otherwise, the tracking algorithm will fail due to large changes in wind speed or shape caused by the evolution of the wind field.

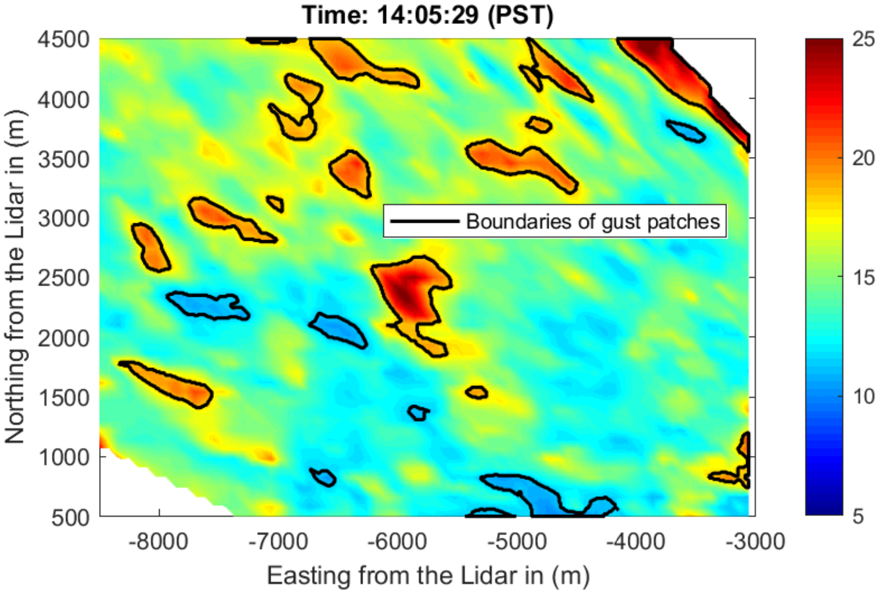
Next, a searching zone is assigned to each original patch. In the searching zone, the algorithm searches for the corresponding target patch on the second frame over all the candidate patches. The candidate patches close to the boundaries of the studied wind field are neglected because of partial observation of the patches. The searching zone is placed downstream of the original patches and oriented along the mean wind direction of the measurement domain. The distance between the original patch and the center of the searching zone is set to the product of the mean wind speed of the measurement domain and the time interval between the frames. The size of the searching zone changes adaptively because a patch could deviate further from the streamline as time elapses. In that case, a large searching zone is needed to cover the possible area that the target patch could reach. The spanwise searching range is set proportional to the product of time and the standard deviation of wind velocity in the spanwise direction, and the same method is applied in the streamwise direction. The spanwise and streamwise velocity components can be calculated by projecting the retrieved x and y velocity components to the streamwise and spanwise directions. The searching zone constrains the tracking algorithm to a small window instead of the whole domain, which avoids unnecessary calculation. Examples of the original-target patch pairs along with their searching zones are presented in Figure 2.5c. The wind fields are shown in Figure 2.5a and 2.5b.

For an original patch, the algorithm searches for the target patch over all the candidate patches. Then, it loops the same procedure over all the other original patches. The target patch of each original patch is chosen from the candidate patches according to predefined criteria, and scores are assigned to the candidate patches for each criterion. The criteria include the changes of wind speed, the directional difference between the mean wind direction and the real moving direction, size difference, and



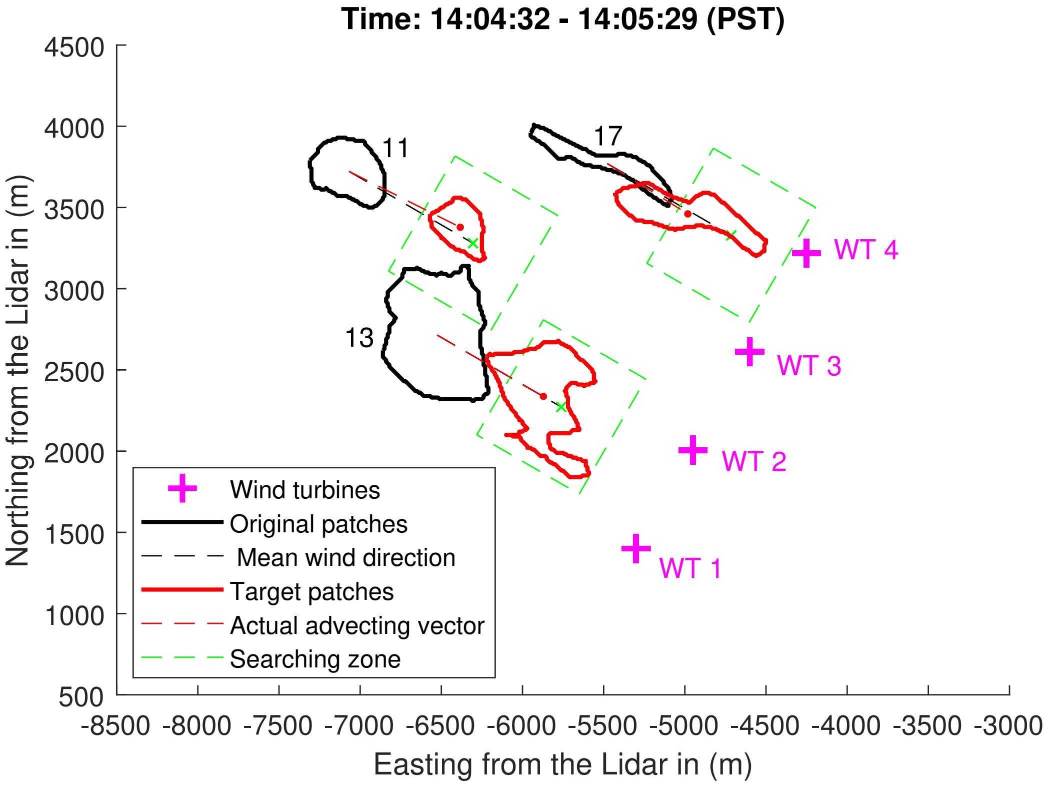
Time: 14:04:32 (PST)
Boundaries of gust patches
Northing from the Lidar in (m)
Easting from the Lidar in (m)
4500
4000
3500
3000
2500
2000
1500
1000
500
-8000
-7000
-6000
-5000
-4000
-3000
25
20
15
10
5

(a)



Time: 14:05:29 (PST)
Boundaries of gust patches
Northing from the Lidar in (m)
Easting from the Lidar in (m)
4500
4000
3500
3000
2500
2000
1500
1000
500
-8000
-7000
-6000
-5000
-4000
-3000
25
20
15
10
5

(b)



(c)

Figure 2.5: (a) Wind Field at the First Time Frame. Colors indicate the wind speed. Black contours are the boundaries of detected patches; (b) Wind Field at the Second Time Frame; (c) Example of Original- Target Patch Pairs. Black patches are the original gust regions extracted from (a), and red patches are the corresponding target gust regions extracted from (b). Green dashed boxes are the searching zones. Black and red dashed lines indicate the moving direction of the streamline and the real movement of the patches, respectively. Pink crosses indicate the locations of a virtual wind turbine array.

shape deformation. For the changes of wind speed and direction, the root-mean-square deviations (RMSD) of the wind speed and direction between the points in the original and candidate patch are calculated using Equation (2.1), and the scores are assigned according to the RMSD's in Equation (2.2).

$$RMSD_p = \sqrt{\frac{\sum_{i=1}^{N}(x_{i,original} - x_{i,target})^2}{N}} \tag{2.1}$$

$$Score_p = \begin{cases} 1 & RMSD_p < X \\ 0 & RMSD_p \geq X \end{cases} \tag{2.2}$$

Here, the subscript p represents the calculated variable name, wind speed or wind direction, and x_i stands for the value of the corresponding variable of the i-th point inside or on the original or candidate patch. The threshold $X = 0.5$ m/s for wind speed, and $X = 10^\circ$ for wind direction. The values of X are determined empirically. In the calculation, we assume that the original and candidate patches share the same centroid. N represents the number of the points that are in both original and candidate patches. If a point is only in either the original or candidate patch, the point is ignored during the calculation.

For size changes and shape deformation, the intersection of union (IoU) value for each original-candidate pair is obtained by calculating the ratio of the number of intersection points to the number of union points. Here, we assume the original and candidate patch share the same centroid as well. The score for the shape change is calculated by Equation (2.3).

$$Score_p = \begin{cases} 0 & \text{IoU} < 0.5 \\ \text{IoU} & \text{IoU} \geq 0.5 \end{cases} \tag{2.3}$$

The threshold value (0.5) in Equation (2.3) is determined empirically. After the individual score for each criterion is obtained, the total score for each candidate patch m, $TotalScore_m$, is calculated by summarizing over all the individual scores.

$$TotalScore_m = Score_{speed} + Score_{direction} + Score_{shape} \tag{2.4}$$

Patches with small difference earn high total scores, and patches with zero total scores are rejected. If multiple candidates have non-zero total scores, the one with the highest total score is chosen as the target. If such a target is identified, we claim that the original patch is advecting, and the target patch is the counterpart of the original at the second frame. Conversely, if no target is detected, the original patch loses its temporal coherency during traveling.

The tracking algorithm was tested on the wind fields shown in Figure 2.5a and Figure 2.5b. Three examples of tracked original-target pairs are shown in Figure 2.5c. The time interval between the original and target patches is 60 s. The other patch pairs are not shown. In Figure 2.5, we can see many small-scale patches shown in Figure 2.5a are not shown inside the corresponding searching zones in Figure 2.5b where they should arrive after 60 s by following Taylors' hypothesis. It provides the evidence that smaller patches are more difficult to track than larger ones. The reason would be that larger-scale patches have strong connectivity inside their internal structures, while the spatial correlation of small patches is weak and unstable. The coherency can be easily destroyed by the stretching and compression of larger-scale turbulent eddies. The destructive effect can reduce the velocity of the patches to a value beyond the threshold, deform or separate the patches, or merge small patches with other structures. All these situations can make the detecting and tracking algorithm ineffective.

2.1.6 Prediction of Impact

After identifying advecting patches, the arrival time and probability of arrival locations of the patches can be predicted.

Prediction of impact time

The impact time of gust patches at a certain location can be predicted using Taylor's hypothesis, that is, dividing the distance between the gust and the location by the mean velocity. The equation is shown as:

$$t = \frac{d_{stream}}{v_{stream}}. \tag{2.5}$$

Both the distance d_{stream} and velocity v_{stream} are the values projected on the mean wind direction. The digression of wind speed along the streamline during traveling to the location of interest is ignored due to the negligible difference for high wind speed (> 15 m/s) (Branlard, 2009). Equation (2.5) provides us with fair estimation of the arrival time, since the gust generally follows Taylor's hypothesis in a short time period without much deviation, and this simple estimation makes the real-time prediction possible. However, spatial gusts do not always travel with the same velocity in reality. Alternatively, one can track the same patch for several time steps, calculate the actual traveling speed, and replace the mean wind speed with the obtained wind speed. Using the actual speed could improve the accuracy of the prediction, but it might also increase the complexity of the algorithm. Since the present work is to provide a fast prediction within a few seconds for wind turbine control, we focus on using the mean wind speed to calculate the arrival time.

Prediction of the probability of impact location

If Taylor's hypothesis holds, the gust region should arrive at a point downstream from the original patch, called the 'predicted point.' However, due to the turbulent nature of the atmosphere, the real arrival point can only be estimated by a statistical probability centered at the predicted point. Here, we treat the gust patches as massless passive particles that are released from a point source, and use the concentration distribution model of plumes to model the spanwise distribution of gust patches dispersing in a turbulent wind field.

Several plume models and puff models have been developed by researchers to investigate the particle distribution in the air, such as Gaussian plume model (Batchelor, 1952; Petersen *et al.*, 1984), K-model, statistical model, and similarity model (Hanna *et al.*, 1982) *etc.*, and more advanced but computationally more expensive models like SCIPUFF (Sykes *et al.*, 1998), RIMPUFF (Thykier-Nielsen *et al.*, 1999), and LODI (Leone Jr *et al.*, 2005). Since this work is aimed at estimating the impact of gusts in a few seconds or minutes in advance of wind turbine control, the fast Gaussian plume model is used.

The reason for using the Gaussian plume model for wind gust studies is explained in the following. First, since the gusts travel in a relatively short distance in this study, we assume the interaction between the large-scale structures and the background turbulence is small. Therefore, we can treat the gusts as passive quantities. Second, we assume the eddy viscosity and eddy diffusivity are of comparable value, *i.e.* the Prandtl number is close to 1. The reason is that, according to the Reynold analysis, under stable or neutral boundary conditions, the turbulent transport of scalar quantities is similar to the transport of momentum, given the transport agents are the same (Li *et al.*, 2015). In our case, the mean wind speed during the studied time period is relatively high (around 15 m/s), so the neutral boundary condition dominated the studied time period (The strong mechanical mixing due to the high wind speed balanced the buoyance-driven turbulence in the atmosphere.). Therefore, we can apply the plume model that was used for passive particles to the transportation of gust structures studied in the present work.

The original Gaussian model (Equation (2.6)) is modified to suit the current case: vertical distribution is neglected since only the deviation of gusts in the horizontal plane is of interest. The velocity term is removed as the probabilities will be normalized by the maximum value, meaning only relative probabilities matter. The modified equation is shown in Equation (2.7).

$$P(y,z) = \frac{1}{2\pi\sigma_y\sigma_z\vec{u}} e^{-y^2/2\sigma_y^2} \times \left[e^{-(z-h)^2/2\sigma_z^2} + e^{-(z+h)^2/2\sigma_z^2}\right] \tag{2.6}$$

$$P(y) = \frac{1}{\sqrt{2\pi}\sigma_y} e^{-y^2/2\sigma_y^2} \tag{2.7}$$

where y refers to the spanwise deviation of a gust patch from the streamline with y equal to zero on the streamline, and z indicates the height above the ground. In Equation (2.6) h is the distance from the source to the ground. P is the probability at the deviation y, and σ_y is the standard deviation of y. We assume the turbulence field driving the gust patches is stationary and homogeneous. Taylors' dispersion theorem provides an expression of σ_y^2 as a function of the releasing time t for stationary turbulence (Hanna *et al.*, 1982).

$$\sigma_y^2 = 2\sigma_{v'_L}^2 \int_0^t \int_0^\tau r(\tau) d\tau dt \tag{2.8}$$

where $\sigma_{v'_L}$ is the standard deviation of the spanwise Lagrangian velocity fluctuation in which the subscript L stands for Lagrangian scale, and r is the autocorrelation of the spanwise velocity fluctuation. Using the random walk theory (Kundu and Cohen, 2010), the standard deviation of the deviation y, σ_y in the Lagrangian system can be approximated by Equation (2.9) when $t \to \infty$

$$\sigma_y = \sqrt{2\sigma^2_{v'_L} T_L t} \tag{2.9}$$

and T_L is the Lagrangian integral time scale.

Here, we assume the gust patches are 'released' for sufficient time ($t \to \infty$) in the atmosphere, so the gust structures only advect but not disperse (grow) during traveling. It indicates that the diffusivity is constant, *i.e.* $K \propto \sigma^2_{v'} T_L$. As the diffusivity is defined as $K = d\sigma^2_y/2dt$, Equation (2.9) can be deduced from the diffusivity. On the other hand, because of the stationary and homogenous turbulence assumption, the temporal correlation of the spanwise velocity fluctuation is approximated by instantaneous spatial correlation. The standard deviation of the Lagrangian velocity fluctuation $\sigma_{v'_L}$ was replaced by the standard deviation of Eulerian velocity fluctuation $\sigma_{v'}$, due to instantaneous velocities being studied in Price (2005). The spanwise velocity fluctuation and its standard deviation were obtained from the given wind field. Equation (2.9) is transformed to

$$\sigma_y = \sqrt{2\sigma^2_{v'} T_L t}, \tag{2.10}$$

where the Lagrangian time scale T_L=200 s, adapted from the work of Dosio *et al.* (2005). The reason of the adoption is that, we did not have sufficient statistical measurements for the time scale calculation in the present work, since the time interval of our lidar measurements is relatively long (30 s) and we did not have any other supplementary device installed locally. Therefore, instead of calculating the time scale directly, we adapted the result from Dosio *et al.*'s simulation of a convective boundary layer. The data used in this paper is between 3 *pm* and 6 *pm* (local time), so the atmospheric flow may be expected to contain some convective activity. Therefore, we believe the selected value could be considered as the best available approximation of the time scale during the studied period. However, we suggest to use meteorological measurements (like a met mast) if available to calculate the time scale, which would improve the estimations.

Additionally, we assume the atmospheric characteristics, with respect to the meandering of gusts, are preserved during the studied period so that the Lagrangian time scale remains constant. Furthermore, the random walk theory used above is only for particles. The effect of the scales of the gusts needs to be considered in Equation (2.10) since we are modeling gusts. The details will be illustrated in Section 2.2, and the modified random walk theory is given in Equation (2.11). The modified random walk theory (Equation (2.11)) and the Gaussian model (Equation (2.7)) comprise the impact prediction model for advecting gust patches with different sizes.

2.2 Results and Discussion

2.2.1 Preliminary Results

Three hours of retrieved wind fields (during 3 - 6 *pm* PST on 26 June) were processed. A time interval of 60 s between original patches and target patches was used. The original patches and corresponding target patches were extracted and associated using the detecting and tracking algorithm. Original random walk theory, Equation (2.10) was used for prediction. The spanwise deviation (L_{real}) of the target patches was measured. The spanwise deviation is defined as the deviation of the centroid of the gust from the streamline of the wind field, in the direction perpendicular to the mean wind direction of the measurement domain. Then, the spanwise deviation was normalized by the rotor diameter, *i.e.* $l_{real} = L_{real}/D$, and l_{real} is the dimensionless spanwise deviation. The histogram of the distribution probability of the dimensionless deviation is shown in Figure 2.6. The probability is normalized by the value at the 'predicted point'. The predicted point is assumed to have the maximum probability. Figure 2.6 presents a Gaussian shape distribution of the gust spanwise deviation. It demonstrates our assumption that the spanwise distribution of advecting patches can be represented by a Gaussian model, rather than the exponential shape found by Branlard (2009).

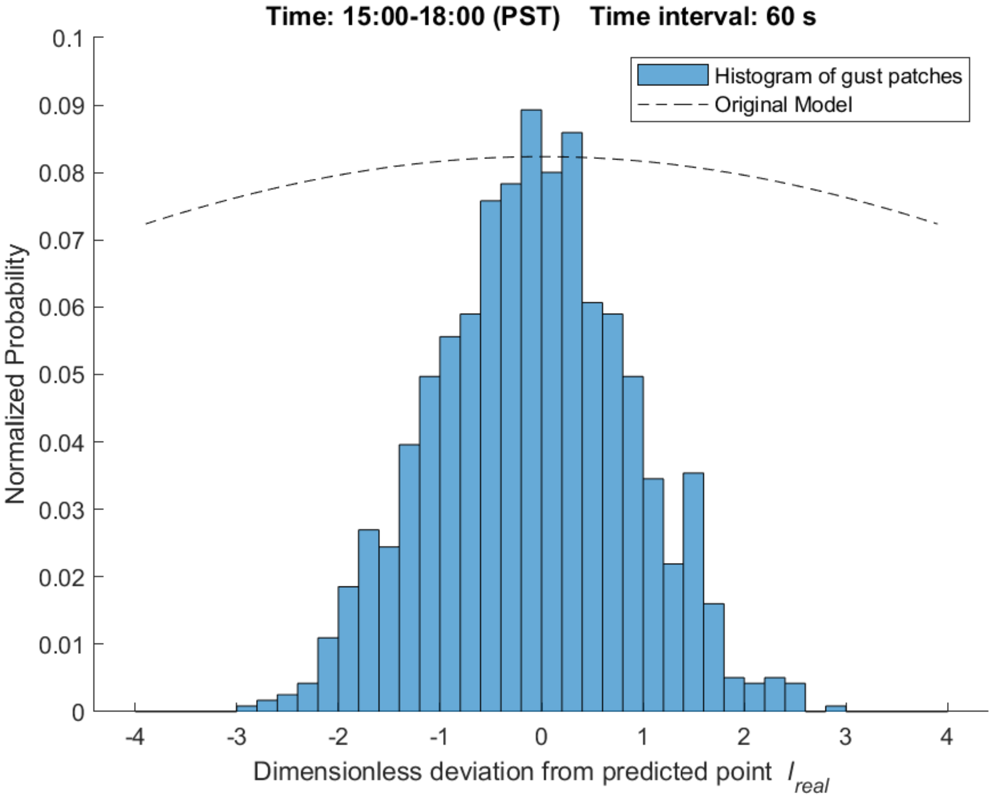


Figure 2.6: Normalized Distribution Probability of the Patches Spanwise Deviation from the Predicted Points. The dashed line represents the normalized probability modeled by Equation (2.10). The histogram shows the normalized probability of the real spanwise deviations.

However, a large discrepancy was observed between the modeled distribution of the gust spanwise deviation (dashed line) and the real deviation using the detecting and tracking algorithm (histogram). The reason is that the original random walk theory does not consider the size of the particles, as the random walk theory assumes that the size difference between particles is negligible. However, the assumption does not hold for gust patches with scales in the order larger than 100 m. The size of the gusts significantly affects their movement. Small gusts are easy to move because many large-scale eddies in the atmosphere move them around, which makes their movement relatively random. On the other hand, large-scale gusts that need even larger-scale eddies to move them are more likely to flow along the streamline of the wind field. Therefore, size dependency of the gust patches needs to be considered.

2.2.2 Size Dependency

To study the effect of the size of the gusts on the spanwise deviation, the dimensionless deviation l_{real} was further normalized by the square root of the time interval, *i.e.* $l_{real,\Delta t} = l_{real}/\sqrt{\Delta t}$, to remove the dependency of the deviation on time. The usage of the square root is according to the relationship between the deviation and time presented in Equation (2.10). The patch scale (L_{pat}) was normalized by the rotor diameter, *i.e.* $l_{pat} = L_{pat}/D$, and l_{pat} is the dimensionless scale. A scatter plot of the normalized deviation $l_{real,\Delta t}$ of all the detected patches with the dimensionless scale l_{pat} is shown in Figure 2.7. Then, the dimensionless deviations were grouped to bins according to their sizes, and inside each bin, the deviations and the dimensionless scales were averaged, respectively. The averaged values were fitted by a power function. The results and the fitting curve are plotted in Figure 2.8. Figures 2.7 and 2.8 present a negative dependency of the deviation on the patch scale. The relationship can be well expressed by a function $l_{real,\Delta t} = \alpha {l_{pat}}^{-\beta}$, where α and β are fitting parameters, and $\alpha = 1.0$ and $\beta = -0.36$ is found for the current dataset.

After the size dependency was empirically derived, Equation (2.10) was modified to include the derived relationship by multiplying it by the derived power function.

$$\begin{aligned} \widetilde{\sigma}_{y,mod} &= \widetilde{\sigma}_y \cdot \alpha(l_{pat})^{\beta} \\ \widetilde{\sigma}_y &= \sqrt{2\sigma_{v'}^2 T_L t}/D \\ l_{pat} &= L_{pat}/D \end{aligned} \tag{2.11}$$

where $\widetilde{\sigma}_y$ is the dimensionless standard deviation of the deviation y, modeled by the original random walk theory.

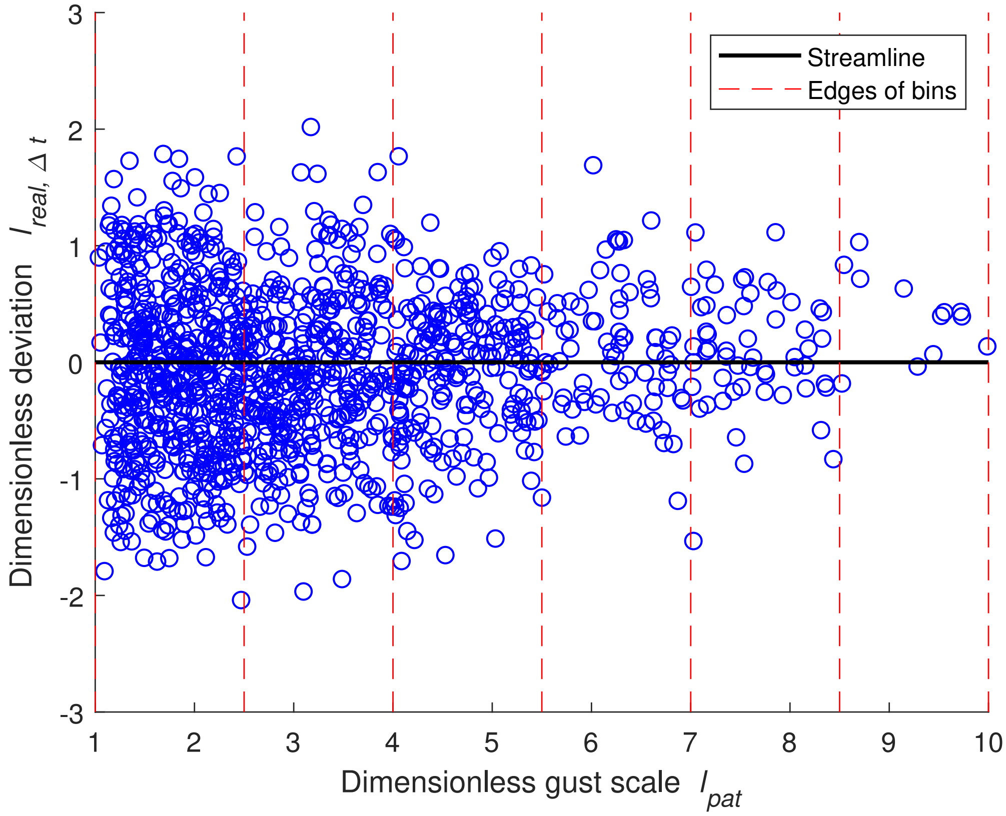


Figure 2.7: Scatter Plot of Normalized Deviations vs Patch Scales. The streamline overlays the line with the deviation $l_{real,\Delta t}$ equal to zero. The red dashed lines imply the edges of the bins. The average dimensionless scales at the bins are 1.75, 3.25, 4.75, 6.25, 7.75, and 9.25.

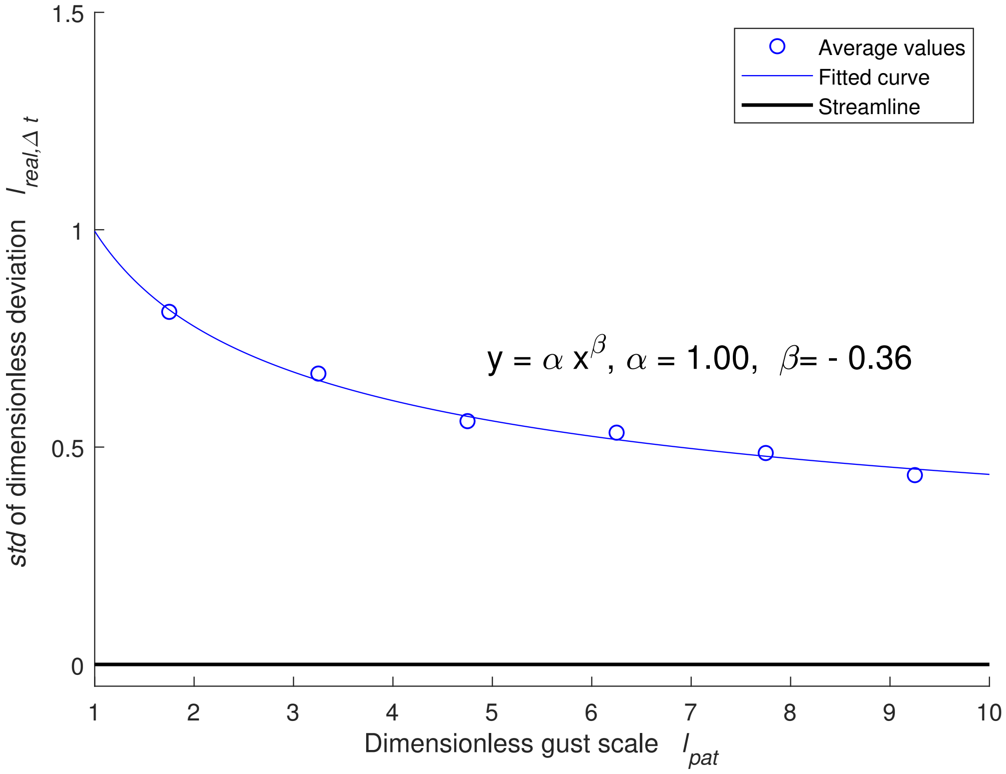
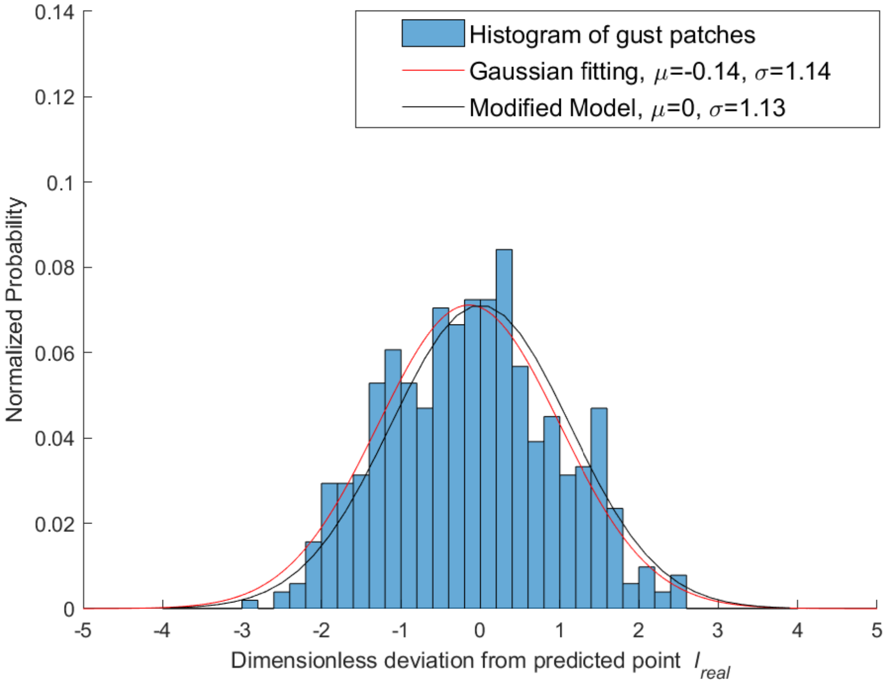


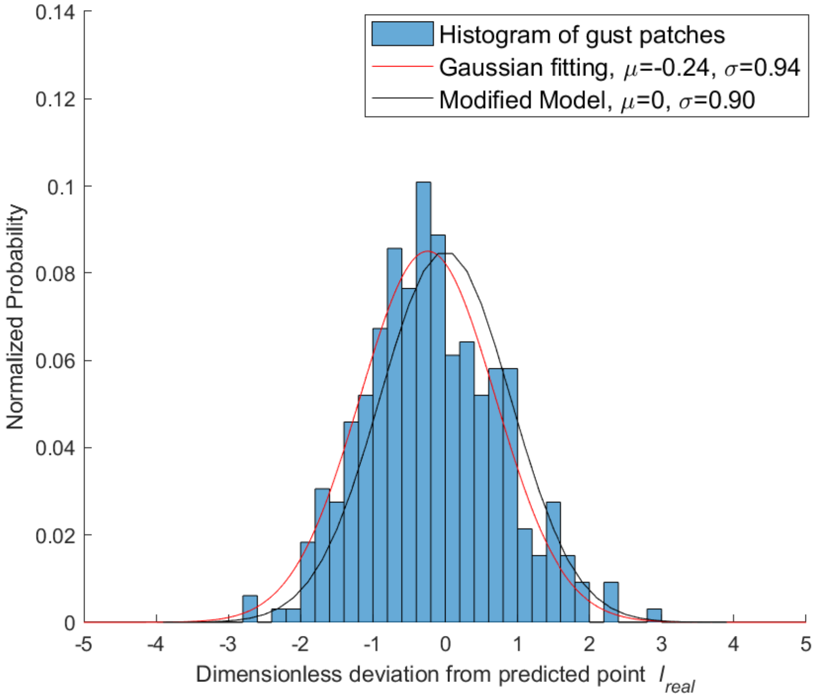
Figure 2.8: The Relationship between Dimensionless Deviations and Dimensionless Gust Scales. Circles are the average dimensionless deviations at different average gust scales, and the solid line is the fitting curve.

2.2.3 Improved Results using the Modified Prediction Model

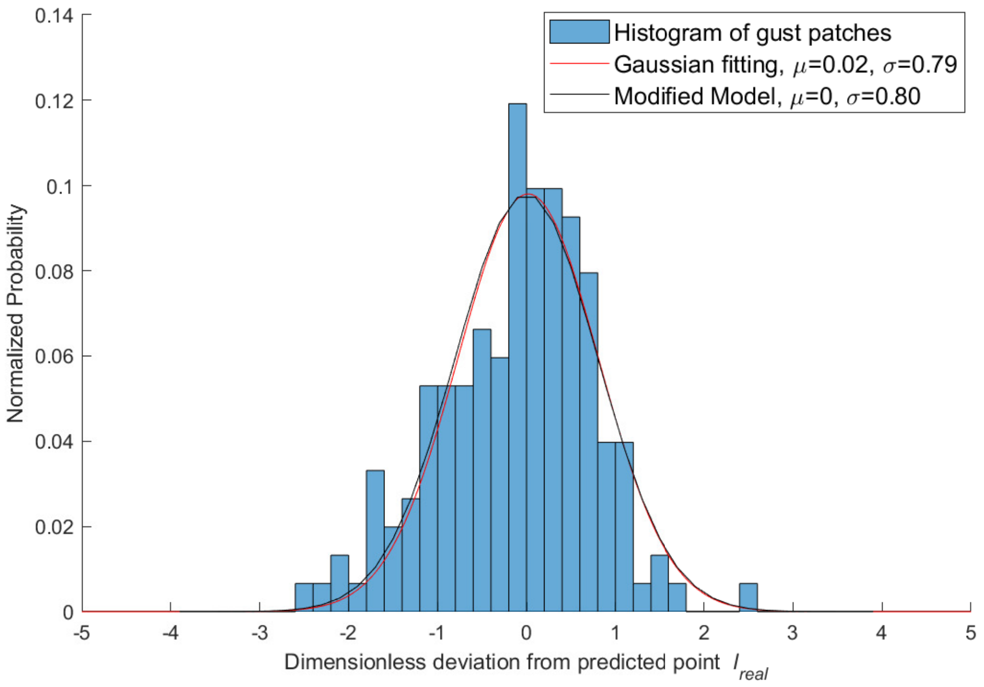
After the size dependency correction was added to Equation (2.10), the modified model was retested on the data presented in Figure 2.6. The patches were separated into three groups according to their sizes: $1D$ - $2.5D$, $2.5D$ - $4D$, and $4D$ - $5.5D$. Patches larger than $5.5D$ are not included due to insufficient sample points. The results are shown in Figure 2.9. Gaussian functions (red lines) were fitted to the histograms. The modeled distribution (black lines) were computed using Equations (2.7) and (2.11). A good match for different sizes of patches are shown in the comparison



(a) Size: $1D$ - $2.5D$



(b) Size: $2.5D$ - $4D$



(c) Size: $4D$ - $5.5D$

Figure 2.9: Normalized Distribution Probability of the Modeled Deviation (Black Lines) and Real Deviation (Histogram) of the Patches. The red lines are the fitting curves to the histograms. The same dataset was used as in Figure 2.6. The patches were separated into three groups: (a) $1D$ - $2.5D$, (b) $2.5D$ - $4D$, and (c) $4D$ - $5.5D$.

between the real measured distribution and the modeled distribution. It illustrates that the distribution probability of gust spanwise deviation can be well represented by the proposed prediction model at least for the current dataset.

2.2.4 Application of the Modified Prediction Model

By applying the detecting and tracking algorithm to the retrieved wind field, we monitored the movement of the gusts during the three-hour period. Next, we positioned a virtual wind turbine array in the wind field shown in Figure 2.5a, and used the three original gusts in Figure 2.5c as examples, to illustrate what the time of impact and probable arrival locations of the gusts are for the hypothetical turbines. Here, the impact time and locations were calculated by the prediction model. The wind turbine array consists of four turbines that align perpendicular to the mean wind direction. The distance between two adjacent wind turbines is 700 m. The distance between the gusts and the wind turbine locations were measured, and the time taken by the gusts to reach the turbines was estimated using Equation (2.5). The real arrival time was also obtained by applying the detecting and tracking algorithm to the wind field at different time frames. Additionally, the spanwise distribution probabilities of the gusts at different wind turbines were predicted and are shown in Figure 2.10. Both predicted and real arrival time and positions are listed in Table 2.2. Patch 11 disappeared before reaching any wind turbine, so the real arrival time and position were left blank in Table 2.2 and the real arrival position is not shown in Figure 2.10. This fact also verifies that small patches are more difficult to track. Table 2.2 shows a good match between the predicted and real arrival time and locations within reasonable tolerances, which validates the proposed prediction model. However, a further quantitative validation is needed in the future work.

2.3 Conclusion

Due to the importance of wind gusts for wind turbine performance, spatial wind gusts with a scale from 100 m to 1000 m are studied using a Doppler lidar. A detecting and tracking algorithm for the spatial gusts is introduced. The proposed algorithm is computationally inexpensive and can provide real-time tracking and prediction. However, it is sensitive to the temporal and spatial resolution of the dataset. The algorithm was performed on a three-hour wind field data obtained from the measurements of a WindTracer Doppler lidar. The results reveal a Gaussian shape

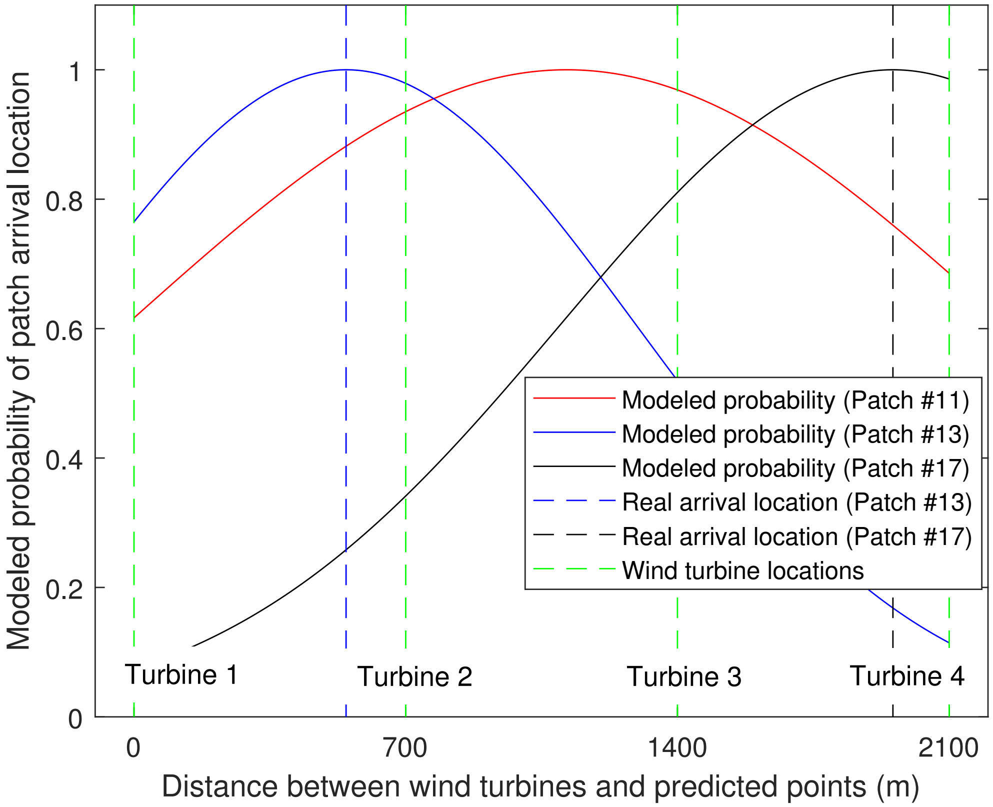


Figure 2.10: Real and Predicted Arrival Positions, and Predicted Normalized Spanwise Distribution Probability of the Gusts. The patch numbers correspond to the gust numbers in Figure 2.5c. Green dash lines indicate the locations of the wind turbines relative to Turbine 1, and the distance is measured along the wind turbine array. The blue and black dash lines represent the real arrival positions of Patch 13 and 17, respectively. Since Patch 11 disappeared before arriving any wind turbine, no line indicates its real arrival location in the figure.

Table 2.2: Predicted and Real Arrival Time and Position of Patches

Patch Number	**#11**	**#13**	**#17**
Impact wind turbine	Turbine 3	Turbine 2	Turbine 4
Predicted arrival time	183.03 s	116.37 s	92.9 s
Real arrival time	-	~119.54 s	~119.54 s
Spanwise distance between predicted arrival position and wind turbine of impact	285.2 m [1]	153.7 m [1]	144.9 m [1]
Spanwise distance between real arrival position and wind turbine of impact	-	106.16 m [1]	87.48 m [1]

distribution of gust spanwise deviation from the streamline and a negative power-law dependency of the deviation on gust scales. Furthermore, a prediction model using the Gaussian plume model and random walk theory was introduced and corrected by the size dependency. The prediction model was tested on the same dataset, and the results further demonstrate the power law relationship. The prediction model was applied to a virtual wind turbine array to estimate the impact time and locations of the gusts on the turbines. The model was validated in a limited set of conditions, due to the lack of the dataset with a longer time period. Therefore, more quantitative validation of the model is suggested for future work. However, the major value of this work is to provide a novel approach for the study of large-scale wind gusts using a long-range Doppler lidar. In real-life applications, the proposed algorithm could be efficiently integrated to feed-forward wind turbine control systems given 2D wind fields in time series, and the wind field could be provided by a long-range Doppler lidar. Given the knowledge of the movement of potential hazardous gusts, the control

[1] The measurement is to the left of the wind turbine of impact.

system would have sufficient leading time to adjust wind turbines to optimal settings. In this case, the power generation could remain stable at the desired value, and the loads on the turbine components could be reduced. Consequently, the impact on the electric grid due to the power fluctuation could be mitigated.

Chapter 3

LASER SOURCE

3.1 Fundamentals of Laser Source

The laser source is the heart of a lidar system. Laser pulses are transmitted to free space and return to be detected by photon-detectors. Laser sources can be divided into different categories according to the gain medium where stimulated emission of photons and optical gains take place. Examples of different types of laser sources are semiconductor lasers, solid-state lasers, fiber lasers, and gas lasers. Kasunic (2016) reviewed the characteristics and applications of different laser sources. This section will focus on the fundamentals and modeling of the laser source for a TOF lidar.

3.1.1 Characteristics

Key characteristics of a laser source include temporal and spatial coherence, wavelength, pulse width, rise time, pulse repetition frequency (PRF), energy and power.

Temporal and spatial coherence

Temporal coherence of a laser source describes the average correlation period between two wavelengths over which they become completely out of phase. The temporal coherence can be specified by a *coherence time* τ_c and the corresponding *coherent length* $d_c = c\tau_c$ (Kasunic, 2016):

$$\tau_c = \frac{1}{2c}\frac{\lambda_0^2}{|\Delta\lambda|}, \tag{3.1}$$

where c is the speed of light, λ_0 is the center wavelength of a laser and $\Delta\lambda$ is the *linewidth* of a laser. Even though a single-mode laser should ideally emit at a single wavelength, the gain medium or quantum-mechanical always broadens the wavelength. The wavelength difference from the center wavelength in a laser spectrum is called *linewidth.* The temporal coherence of a laser tells the degree of monochromaticity of a laser, *i.e.* a laser with larger average linewidth starts to decorrelate faster. A laser with low temporal coherence is more likely subject to *chromatic aberration.* A lens may, therefore, be unable to focus all wavelengths at a convergence point. Specifically, if a laser has a wide spectrum, the beam could disperse after the focusing lens due to different refractive indices for different wavelengths of light. Minimizing the chromatic aberration is important for a TOF lidar, since large dispersion of the laser requires a large detector which could introduce more unwanted ambient light. However, speckle noise is a drawback of a temporally coherent laser, because the reflection of highly correlated beams from rough surfaces causes stable interference. The interference could produce obvious speckle patterns which affects the power of return pulses and the image quality of a flash lidar.

Spatial coherence describes the phase correlation across the wavefront of a laser beam. For a spatially coherent beam, the irradiance distribution profile is an ideal Gaussian profile. However, in practice, due to unavoidable misalignment between the mirrors of a laser cavity, the distribution profile becomes a near-Gaussian profile superposed by non-Gaussian beams which results in phase variation, and the laser becomes spatially incoherent. The degraded spatial coherence due to the phase distortion results in (1) an increase of divergence angle at which the beam exits the laser, and (2) an increase of the spot size when the beam is focused with a lens. Therefore, spatially coherent beams are also called diffraction-limited beams. The spatial coherence of a laser beam can be quantified by the beam quality factor M^2. Specifically, for a diffraction-limited beam the beam quality $M^2 = 1$, while for the beam that is not diffraction-limited, $M^2 > 1$. The spatial coherence is also crucial for a TOF lidar.

One one hand, a large beam size due to large divergence of a beam at a long distance limits the resolvable size of a target and makes the edges less resolvable. A large beam size can also cause multiple-path effects (Adams, 2000). On the other hand, when the return beam is focused on a detector, the increasing spot size requires a larger detector to cover the whole profile, which allows more background light incident on the detector and increases electrical noise of the system. However, similar to the temporal coherence, the speckle noise is also a disadvantage of a spatial-coherent laser. The standard deviation of the speckle noise (Richmond and Cain, 2010) is:

$$\sigma_{sp} = \frac{iG}{M}. \tag{3.2}$$

It has the largest value for an ideal laser that is temporally and spatially coherent (*i.e.* $M^2 = 1$), compared to a non-diffraction-limited laser. The iG in Equation (3.2) is the signal current output from a photon-detector with a gain of G. Therefore, a laser with broadband spectrum and low spatial coherence is less subject to speckle noise.

Wavelength

The wavelength is another important factor of a laser source. The most common wavelength bands for the TOF lidar application are near infrared band (NIR, $0.7-1.1\ \mu m$) and shortwave infrared band (SWIR, $1.1-3.0\ \mu m$). Two commercial examples for each band are 905 nm and 1550 nm. Each band of wavelength has its own advantages and weaknesses from different perspectives. Wojtanowski *et al.* (2014) studied the vulnerability of the 905 nm and 1550 nm laser to different environmental conditions with respect to normal atmospheric conditions and conditions with high water content. The results show at normal atmospheric conditions, the atmospheric extinction coefficient for 1550 nm is two times lower than that for 905 nm, while the water extinction coefficient γ_{water} for the former one is two orders of magnitude higher than latter one (Figure 3.1). The comparison indicates that 1550 nm could provide farther measurable distance at normal conditions, but at conditions with increasing humidity, rains, fogs, or wet targets, the maximum detectable distance for 1550 nm drops dramatically. Moreover, the water absorption of the wavelength is only a portion of the atmospheric effects on the laser. The other influence of the atmosphere on the laser include Mie-scattering of fog, specular reflection of rain droplets, and complex reflection mechanisms at a wet surface (Wojtanowski *et al.*, 2014), which will be briefly discussed in Section 4.1.3.

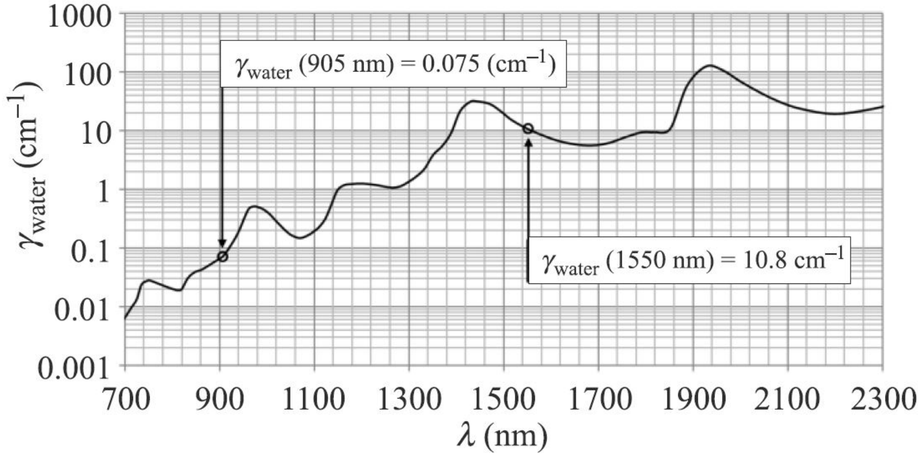


Figure 3.1: Water Extinction Coefficient Spectrum (Wojtanowski *et al.*, 2014).

In addition to the effect of atmosphere, the eye-safety of a laser also needs to be considered when choosing a wavelength. The maximum permissible exposure (MPE) for 905 nm is much lower than 1550 nm (Wojtanowski *et al.*, 2014). It is because 905 nm wavelength is close to the visible light range and can be focused to a tiny spot on the retina, while 1550 nm wavelength rarely reaches the retina due to high water absorption of the interior of human eyes, and it is less hazardous. Therefore, the laser safety standards allow higher power output at 1550 nm which provides a positive impact on the maximum measurable distance. The selection of wavelength is also depended on the selection of photon-detectors, so the performance of photon-detectors under different conditions should be considered. More details of the behavior of photon-detectors will be discussed in Chapter 5.

Pulse width and rise time

Another key parameter for a laser is the pulse width which is the time duration of a pulse, and is usually approximated by the Full width half maximum (FWHM) for a Gaussian pulse. Another parameter of importance is the rise time defined as the time required for a pulse to rise from 10% to 90% of its peak value. The relation between the rise time t_r and the pulse width Δt_w can be related using Equation (3.8) and Equation (3.9), $t_r \approx 0.7164\Delta t_w$. The rise time of a laser pulse affects the timing accuracy of a TOF lidar which will be discussed further in Chapter 6.

Pulse repetition rate

The pulse repetition rate, PRF, is defined as the number of pulse transmitted by a laser source per second. The PRF represents the number of products per second for a scanning lidar assuming one product per pulse, and the frame rate for a flash lidar (assuming one image frame is generated from one flash of the source). In addition, the PRF also limits the maximum achievable range of a laser: $R_{max} = c/2PRF$. An explanation is that, when a laser emits a first pulse, the timing device associates the returns with the first pulse only before the second pulse is emitted. When the second pulse is emitted, the returns from the first pulse will be considered as the returns of the second pulse. Therefore, there is a trade-off between the number of points per second and the maximum distance a lidar can achieve.

Power and energy

Pulse energy E is the power integrated over the time duration of a pulse, and can be calculated using the pulse width and peak power for a rectangular pulse

$$E = \int P(t)dt = P_0 \cdot \Delta t_w. \tag{3.3}$$

We can also use Equation (3.3) to approximate the pulse energy of a Gaussian pulse under the assumption that the area under a Gaussian curve is equal to the area of a rectangular defined by the pulse width and the peak power, as shown in Figure 3.2. Thus, a useful form of the peak power of a pulse is derived as

$$P_0 = \frac{E}{\Delta t_w}. \tag{3.4}$$

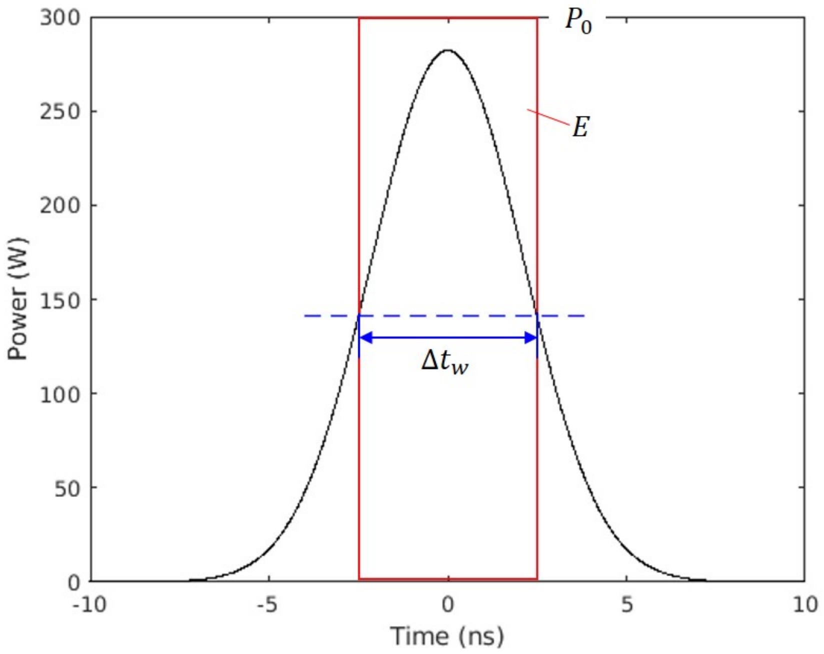


Figure 3.2: Approximation of Pulse Energy of a Gaussian Pulse. The area of the red rectangular represents the approximated pulse energy.

The peak power is a key parameter of a laser source, which represents the maximum optical power occurs during the pulse duration and significantly affects the maximum achievable range of a laser. In addition to the peak power, we also concern the average power P_{aveg} of a laser which is defined as the power averaged over an entire period between the emission of two successive pulses $T = 1/PRF$:

$$P_{aveg} = \frac{E}{T} = E \cdot PRF. \tag{3.5}$$

The peak power and average power can be related by the *duty cycle* of a pulse:

$$Duty\ cycle = \frac{\Delta t_w}{T} = \frac{P_{aveg}}{P_0}. \tag{3.6}$$

3.2 Transmit Pulse Model

In this work, the laser pulse is assumed to have a Gaussian distribution in time (Richmond and Cain, 2010; Budge *et al.*, 2006):

$$P_t(t) = \frac{E}{\sigma\sqrt{2\pi}} e^{-\frac{-t^2}{2\sigma^2}} \tag{3.7}$$

where E is the pulse energy of the laser, and σ is the standard deviation of the Gaussian distribution. The relation between the standard deviation, the pulse width Δt_w and the rise time Δt_r are

$$\Delta t_w = 2\sigma\sqrt{2 \ln 2} \approx 2.355\sigma \tag{3.8}$$

$$\Delta t_r \approx 1.687\sigma, \tag{3.9}$$

assuming the area under the Gaussian curve is equal the area of a rectangular defined by the pulse width and the peak power. The derivation is given in Appendix B. The peak power can be found as

$$P_0 = \frac{E}{\Delta t_w} = \frac{E}{2\sigma\sqrt{2 \ln 2}} \tag{3.10}$$

Inserting Equation (3.10) to Equation (3.7), the Gaussian pulse model used in this work is obtained

$$P_t(t) = 2\sqrt{\frac{\ln 2}{\pi}} P_0 e^{-4\ln 2 \frac{t^2}{\Delta t_w^2}} \approx 0.94 P_0 e^{-4\ln 2 \frac{t^2}{\Delta t_w^2}}. \tag{3.11}$$

Equation (3.11) also describes the relation between the height of the Gaussian pulse and the peak power: $h_{pulse} \approx 0.94P_0$. Using Equation (3.11), one can obtain the temporal power distribution of the transmit pulse given the peak power and the pulse width of a laser source. The corresponding pulse energy, average power, duty cycle and rise time can be obtained using Equation (3.3), (3.5), (3.6) and (3.9). One example of the transmit pulse is shown in Figure 3.3, and the peak power and pulse width is obtained from the specification (Table A.1) of a prototype lidar developed by our team. The other characteristics of a laser source is given in Table 3.1.

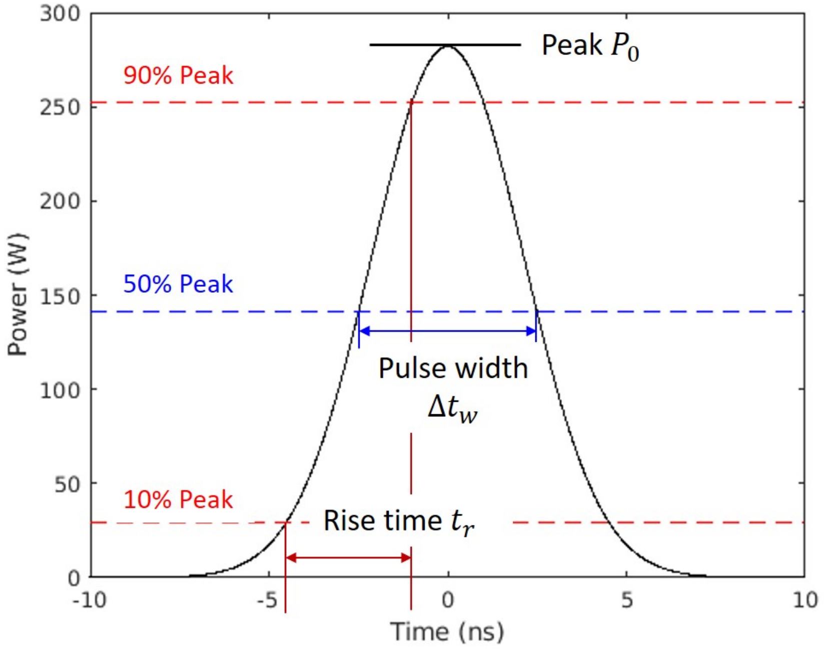


Figure 3.3: Example of the Modeled Transmit Pulse.

Table 3.1: Parameters of Transmit Pulse Model

Parameter	**Symbol**	**Value**
Peak power	P_0	300 W
PRF	-	1 MHz
Pulse energy	E	1.5 μJ
Average power	P_{aveg}	1.5 W (31.76 dbm)
Pulse width	Δt_w	5 ns
Rise time	t_r	3.52 ns
Duty cycle	-	0.005

3.3 Sampling Rate

Since we utilized discrete signals to simulate continuous signals in the pulse model, the sampling rate of the discrete signals needs to be considered. Wahab *et al.* (2016) summarized research on the selection of sampling-rate for chromatographic separations with a focus on accurate description peak height, pulse area, and pulse shape. They discussed the effects of the sampling rate on noise modeling, taking account of the response time of detection devices. For the simulation of continuous signals, a proper representation of a pulse with respect to the peak, area, and shape is critical, but additional factors also need to be considered to fully describe the nature of a continuous signal and the response of the electrical system to the signal. The considerations are listed as following:

- accurately represent the pulse characteristics with respect to heights, area, and shape;

- have no effect on the precision of arrival time measurement by any timing algorithm using any timing device;

- have no effect on the resolution of any timing device;
- accurately represent noise and the effect of the noise on the time measurement;
- low computational cost.

We used the above considerations as metrics to determine the sampling rate of the pulse model.

3.3.1 Simulation

In this study, we tested several sampling rates of 1 GHz, 10 GHz, 100 GHz, 1000 GHz, and 10000 GHz to simulate continuous signals. Those signals is referred to as original 'continuous' signals. To properly describe the shape of a signal, we followed the Nyquist sampling theorem that the sampling rate of the signal should be at least twice the bandwidth of the signal. Since the bandwidth of the transmit signal is around 0.2 GHz as shown in Figure 3.4, all the sampling rates satisfy the first metric and also meet the suggestions proposed by previous work (Kelly and Horlick, 1973; Dyson, 1999; Vandeginste *et al.*, 1998; Scheeren *et al.*, 1985; Kromidas and Kuss, 2009; Hinshaw, 2003). Next, the return signals were calculated in the

Table 3.2: Simulation Parameters

Parameter	**Value**
Target Distance R (m)	10, 15, 20, 25
SNR (dB)	32.99, 25.95, 20.95, 17.07
Reflectivity ρ (%)	10
Visibility V (km)	10

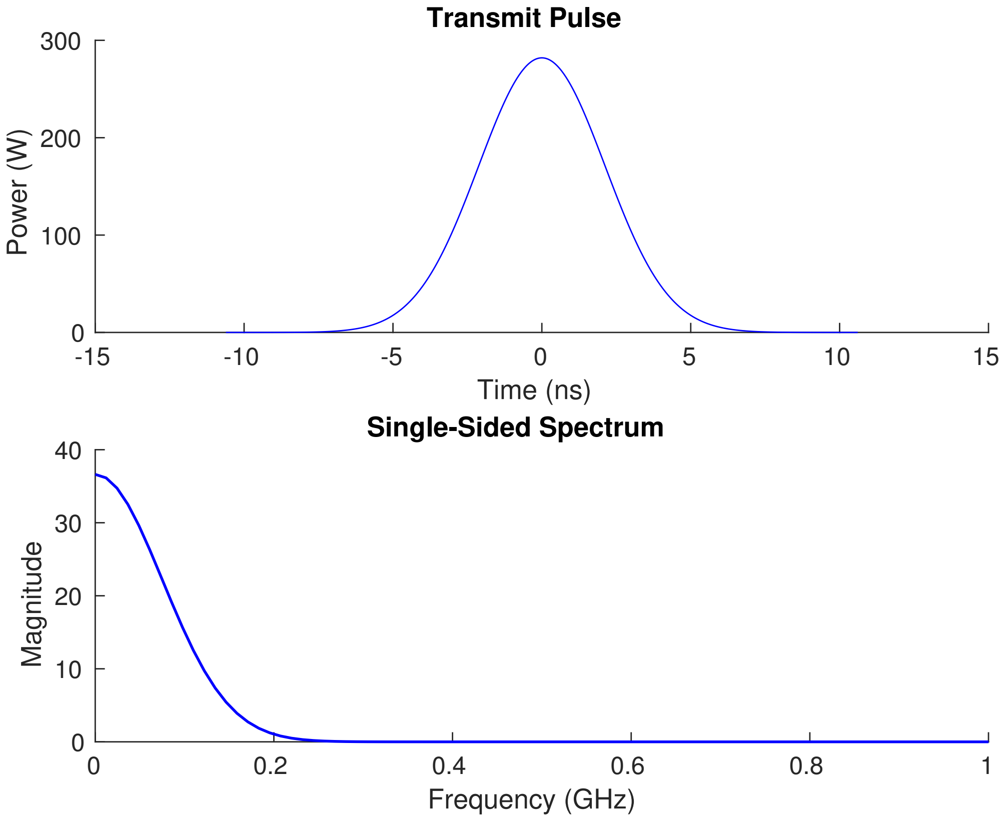


Figure 3.4: Transmit Pulse (Top) and Its Single-Sided Power Spectrum (Bottom).

propagation module (Chapter 4) and targets were placed at different distances. Noise on the signals was calculated in the receiver module (Chapter 5) and added to each data sample. The tested target distances and the corresponding SNR's are listed in Table 3.2, as well as the key parameters used in the propagation module and the receiver module. Other variables in those modules were kept unchanged.

A leading-edge timing method using a TDC was applied. The TDC model and the leading-edge detection method is described in details in Section 6.2. Here, a brief description of the TDC model is provided. The TDC model first takes 'analog signals' and produces transient signals (signals being processed by a comparator). The transient signal is smoother than the initial 'analog signal' due to the propagation delay (bandwidth) of the TDC. Then, the transient signals are processed by the leading-edge timing method. The TDC resolution was set to 1 ps (AMS, 2017), and the trigger of the leading-edge detection was set at 3x the noise-floor. It should be noticed that it is impossible to represent infinite many modes of noise using finite sampling rate, but it is not necessary since the propagation delay of the TDC limits the highest frequency of the noise that needs to be modeled. Examples of the original 'continuous' signals and the corresponding transient signals using different sampling rates are shown in Figure 3.6.

A Monte-Carlo simulation was performed on 100 observations of signals for each sampling rate at each distance. Meanwhile, the arrival time of corresponding noise-free signals was measured by the same technique and used as references. The root-mean-square (RMS) error of the distance measurement Δ_d was calculated at each distance (SNR). The RMS error is defined as:

$$\Delta_d = \sqrt{\frac{1}{N}\sum_{0}^{N-1}(d_{meas,i} - d_{noise-free})^2}, \tag{3.12}$$

where d_{meas} is the distance measurement of the i-th observation, $d_{noise-free}$ is the distance measurement of the noise-free signal, and N is the total number of observations. The results are shown in Figure 3.5. From the results, we observe that when the sampling rate is higher than 10 GHz, the RMS error becomes independent on the sampling rate. Therefore, sampling rates equal or larger than 10 GHz satisfy the second metric. The reason for the independence is that the arrival time was measured on 'smoothed' transient signals. Even though the original 'continuous' signal was sampled at very high frequency, the fluctuation of frequencies higher than the bandwidth of the TDC comparator was smoothed out by the comparator. In other words, the RMS error is determined by the natural of noise on the transient signals (independent on the sampling rate). Another concern of the sampling rate is the resolution of the

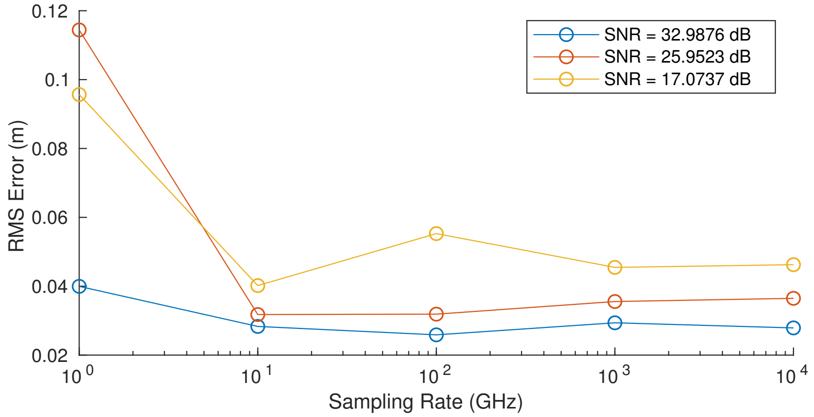


Figure 3.5: Distance RMS Error at Different Sampling Rates and SNR's.

timing device (the third metric). Since the TDC resolution was achieved by interpolating the 'smoothed' transient signal to the TDC resolution in the TDC model, the sampling rate of the original signal has no effect on the TDC resolution. For an ADC, oversampling increases the bit resolution according to (Nakhostin, 2017):

$$F = 4^n F_s,$$

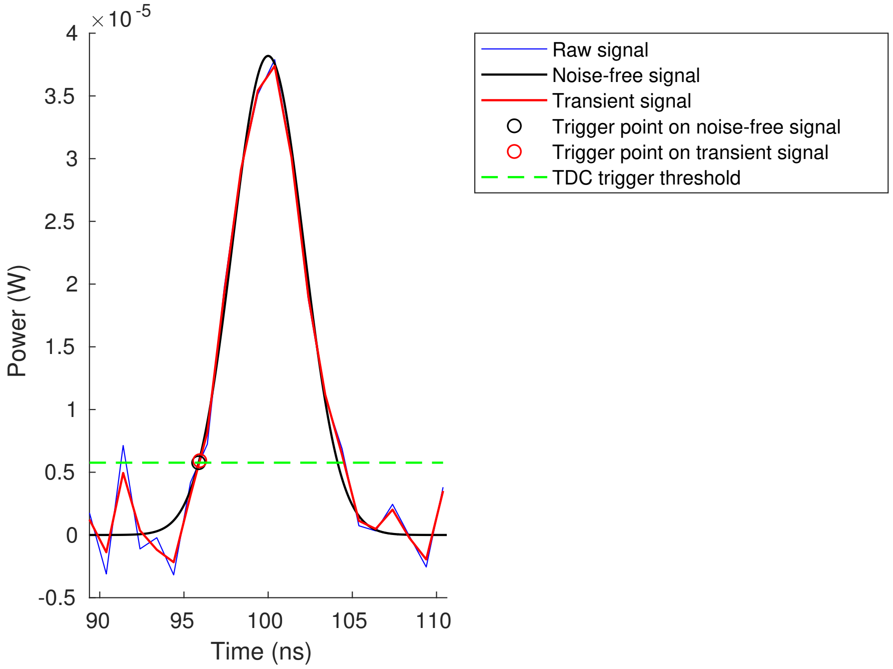
where F_s is the original ADC sampling frequency, F is the oversampling frequency, and n is the number of additional bits. The increased bit resolution reduces the quantization noise (refer to Chapter 7 for details). Thus, signals are usually oversampled before being fed into an ADC to achieve a higher amplitude resolution (Nakhostin, 2017). A sampling rate a few orders higher than the ADC sampling frequency is recommended to model the oversampled 'continuous' signals. The sampling frequency of the ADC used in this study is 2.5 GHz for the two-channel mode, so the sampling rates equal to and higher than 100 GHz satisfy the third metric, among which 100 GHz has a lower computational cost (the fourth metric).

Consequently, after evaluating the different sampling rates according to the metrics, the sampling rate of 100 GHz is selected for the 'continuous' signal simulation in the pulse model without loss of generality for different types of timing devices. This selection is under the assumption that the smoothing effect of the comparator and the linear interpolation are implemented in the TDC model. The recommendations for the sampling rate selection for each metric is given in Table 3.3.

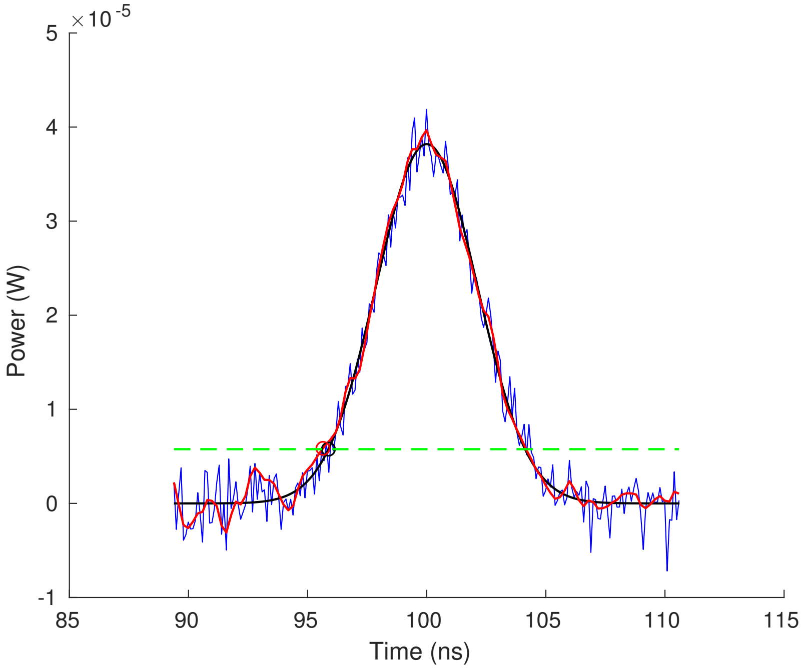
Table 3.3: Recommendations for the Sampling Rate Selection

Consideration	**Requirement for the sampling frequency**
Pulse characteristics	$\geq$ the Nyquist frequency of the signal
Resolution of timing devices	If the resolution is achieved in the TDC/ADC model, no constraints. If not, $\geq$ the sampling frequency of the TDC, and a few orders higher than the sampling frequency of the ADC.
Measurement precision and noise	$\geq$ the bandwidth [1] of the timing device.
Computational cost	Low frequency

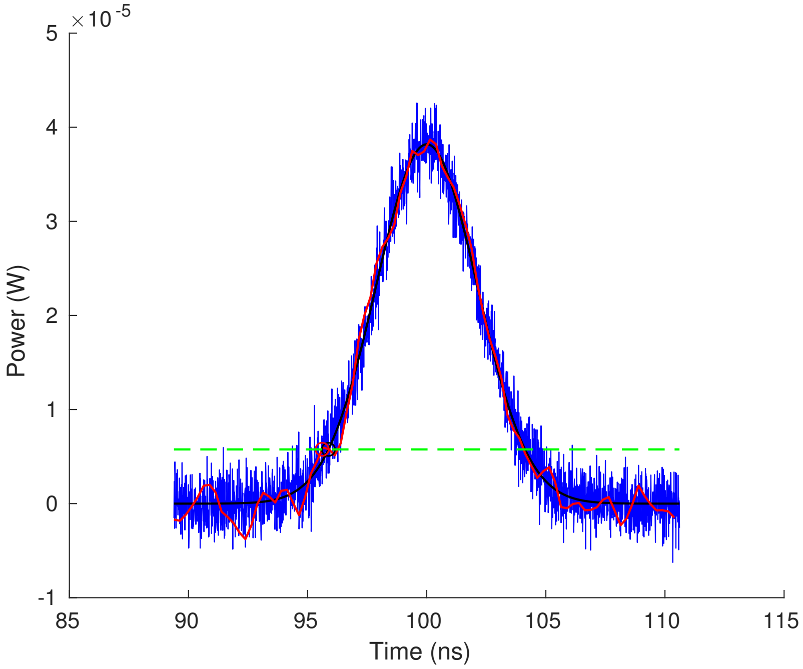
[1] The bandwidth is defined by the system propagation delay.



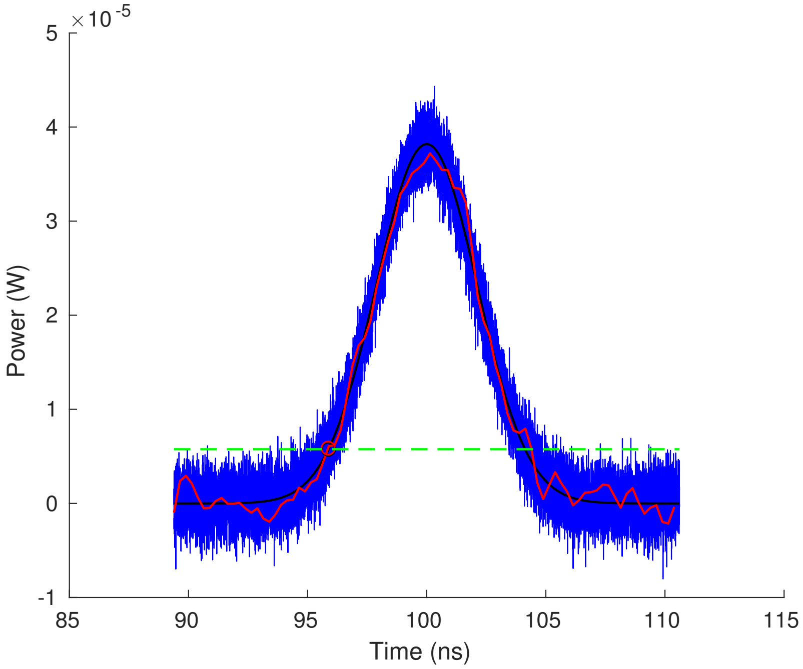
(a) Sampling rate: 1 GHz.



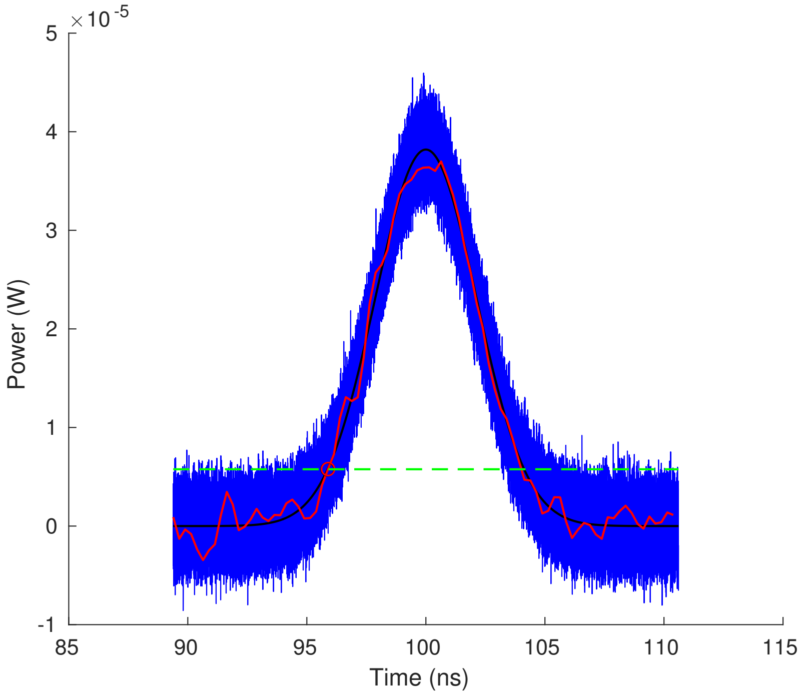
(b) Sampling Rate: 10 GHz.



(c) Sampling Rate: 100 GHz.



(d) Sampling Rate: 1000 GHz.



(e) Sampling Rate: 10000 GHz

Figure 3.6: Original 'Continuous' Signals, Signals with Noise after being 'Smoothed' by the Comparater of a TDC (transient signals), and the Trigger Events at Different Sampling Frequencies for an SNR of 25.95 dB. No sub-sampling was performed on the original signal in (a) because the sampling rate is smaller than twice the TDC bandwidth.

Chapter 4

LASER PROPAGATION

4.1 Radiometry of Laser Propagation

After a laser pulse is emitted from a laser diode, it propagates through the optical system of a transmitter and travels through a free-space environment. After hitting a target, a portion of the backscattered energy will reach the receiver and be sensed by a photon-detector. The pulse is significantly altered through its pathway, including, for example, shape, amplitude and optical noises. This section will discuss the change of energy content of a laser pulse at each stage of the propagation process, and the influence of the optical system, atmospheric conditions, and target characteristics on the behavior of a laser pulse. A schematic of the laser propagation process is given in Figure 4.1. The energy quantities at each stage will be discussed next, focused on the peak power of the laser pulse for simplification.

- Location 0: at the laser source. The peak power of an emitted pulse is equal to P_0.

- Location 1: after the lens system of the transmitter. The laser power $P_1 = P_0 \cdot \eta_{lens,transmitter}$, where $\eta_{opt,transmitter}$ is the overall efficiency of the transmitter optical system, such as lens, scanner, and circulator (refer to Section 4.1.4).

- Location 2: position just before striking targets at distance of R. After the laser pulse passes through free space to the target, the power is:

$$P_2 = P_1 \cdot \eta_{atm,R},$$

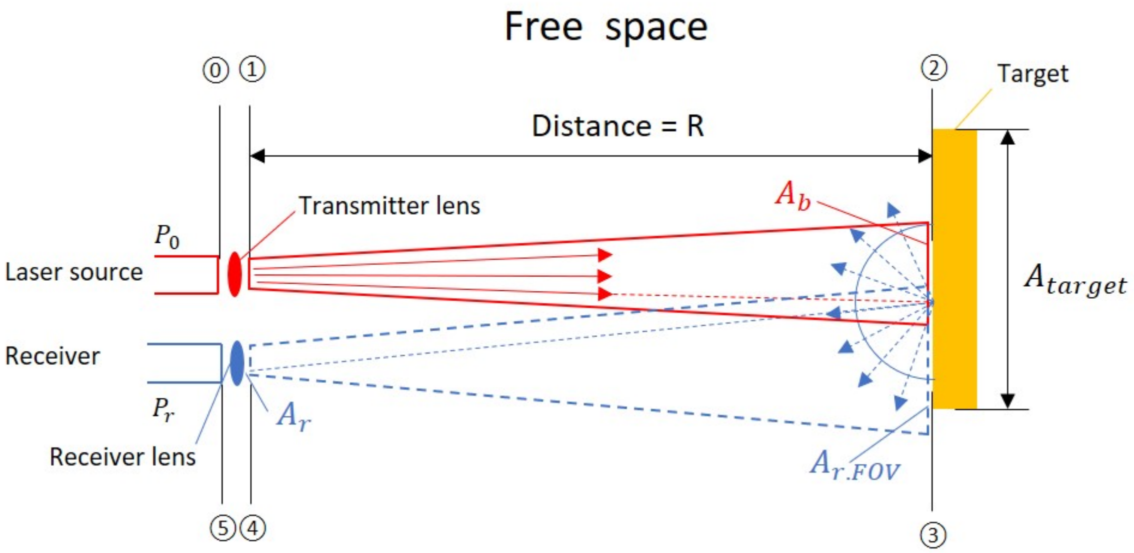


Figure 4.1: Schematic of Laser Propagation Process for a Bi-Static Lidar.

and the radiance (units of $Wm^{-2}sr^{-1}$)

$$L_2 = \frac{P_2}{A_b \pi}.$$

The area A_b is the size of the laser beam at Location 2 (refer to Section 4.1.1) and $\eta_{atm,R}$ is the atmospheric transmission through distance R (refer to Section 4.1.3).

- Location 3: right after a laser beam hits a target. A portion of the backscattered energy falls into receiver's FOV. Assuming the target has Lambertian reflection (*i.e.* the radiance is constant seen from different look-angles), the irradiance (units of Wm^{-2}) seen by the receiver

$$E_{e3} = L_2 \cdot \Omega_{r,\theta} \cdot \rho = L_2 \cdot \frac{A_x}{R^2} \cdot \rho.$$

The variable ρ is the target reflectivity. If the receiver's look-direction and the normal of the target surface have an angle θ, only a projected area that is perpendicular to the receiver's look-direction can be seen by the receiver, and the area is referred as A_x. Consequently, the solid angle subtended by A_x at the receiver can be expressed by $\Omega_{r,\theta} = \frac{A_x}{R^2}$. The area A_x is determined by the minimum value among the beam size A_b, target size A_{target} and receiver's FOV ($A_{r,FOV}$) at Location 2. Details will be illustrated in Section 4.1.2 and Figure 4.2.

- Location 4: at the surface of the receiver lens. After the backscattered laser travels through the distance R and reaches the receiver, the irradiation can be expressed as

$$E_{e4} = E_{e3} \cdot \eta_{atm,R}.$$

- Location 5: at the photon-detector. After passing the optical system of the receiver with an overall efficiency $\eta_{opt,receiver}$, the irradiation sensed by the photon-detector

$$E_{e5} = E_{e4} \cdot \eta_{opt,receiver} \ ,$$

and the optical power of the return pulse

$$P_r = P_5 = E_{e5} \cdot A_r \ .$$

Here, we assume the laser beam has a circular cross-section and the detector has a diameter D_r and an area

$$A_r = \pi D_r^2/4 \ . \tag{4.1}$$

Combining all the stages together, we can derive an equation for the return power, which is referred to as Lidar Equation:

$$\begin{aligned} P_r &= P_0 \cdot \eta_{opt,transmitter} \cdot \eta_{atm,R} \cdot \frac{1}{A_b \pi} \cdot \rho \cdot \frac{A_x}{R^2} \cdot \eta_{atm,R} \cdot \eta_{opt,receiver} \cdot A_r \\ &= P_0 \cdot (\eta_{opt,transmitter} \cdot \eta_{opt,receiver}) \cdot \frac{A_x A_r}{\pi R^2 A_b} \cdot \eta_{atm,2R} \cdot \rho. \end{aligned} \tag{4.2}$$

Here, the two atmospheric transmission $\eta_{atm,R}$ are combined to $\eta_{atm,2R}$. Equation (4.2) illustrates how the laser radiation quantities are calculated at each propagation process. Next, we will further discuss how the characteristics of the laser pulse changes at each stage individually.

4.1.1 Laser Beam Divergence

As shown in Section 3.1, a fiber laser can produce diffraction-limited laser beams. However, the divergence angle is still too large at the exit of the fiber to produce small spots at remote distance. The divergence is usually reduced by placing a collimator after the fiber exit and the divergence angle can be reduced to the order of 1 $mrad$. However, even in this case, a beam of size of 7 mm can still be broadened to $\approx 10\ cm$ at 100 m. The diverged beam reduces laser irradiance, and the relative size between the laser beam and targets can also affect the backscattered energy. Thus, it is beneficial to calculate the beam size at the target distance. The beam area (A_b) at distance R can be obtained by

$$A_b = \pi\omega(R)^2, \tag{4.3}$$

where $\omega(R)$ is the far-field Gaussian $1/e^2$ beam radius (in units of mm) at the distance R. The far-field beam radius can be expressed by (Kasunic, 2016):

$$\omega(R) = \omega_c\sqrt{1 + \left(\frac{\lambda R}{\pi\omega_c^2}\right)^2},$$

where λ is the wavelength, ω_c is the Gaussian $1/e^2$ beam radius after a collimator. The beam radius after a collimator is related to the divergence angle of the collimator θ_{div} by $\theta_{div} = \lambda/\pi\omega_c$ (Kasunic, 2016). Diffraction-limited beam is assumed in the above calculation. The wavelength and the divergence angle are usually available in the specification of a collimator.

4.1.2 Target Characteristics

In Equation (4.2), the target reflectivity ρ is an important variable for the return power calculation. The reflectivity is defined as the fraction of the incident optical power that can be reflected by the target. Reflectivity of an object depends on many factors, including material properties, surface roughness, wavelength of the incident light, and polarization of light. In addition to the reflectivity, the geometry of the target could also change the beam shape and the amount of energy that reaches the receiver (Steinvall, 2000; Al-Temeemy, 2017). In this section, we will provide a brief introduction on how different characteristics of a target can affect the backscatter of a laser beam.

Materials

The roughness of the object surface plays an important role in determining the type of reflection by an object. If the surface of a target (*e.g.* metal or mirror) is smooth (the roughness is small compared to a wavelength of a laser beam), all the incident light is reflected into a single direction following the law of reflection. Such reflection is called specular reflection. On the other hand, if the surface of a target is rough compared to a wavelength (*e.g.* concrete wall and paper), the light is reflected in all directions, and this type of reflection is named 'diffuse reflection'. In this study, we used Lambertian reflection model to approximate the diffuse reflection, which assumes the reflecting surface of objects has zero roughness and the radiance of the light reflected by an object surface is the same seen from different look-angles in the semi-sphere adjacent to the surface. The reason for the approximation is that the intensity distribution of reflected light varies with the roughness of an object for diffuse reflection in reality, but the roughness is not always available for the modeling. Therefore, we ignore the effect of the roughness on the intensity distribution and assume the diffuse reflection is perfect. However, more advanced models for diffusive reflection could also be applied such as Oren–Nayar reflectance model. It has been shown to provide more accurate prediction of natural surfaces, but the albedo and the roughness of the surface have to be known (Oren and Nayar, 1994). In addition to the roughness, the composition and structure of the material of an object also lead to different absorption of light with different wavelengths. Thus, the reflectivity also varies accordingly. Therefore, the reflectivity of the object needs to be determined carefully for the laser power calculation.

Geometry

The geometry of the object includes object orientation, size, and edges of the surface. Figure 4.2 shows a case where the normal of the object surface points to the laser source, but it does not align with the look-direction. In this case, the area of the object subtended to the receiver's FOV, A_x, is equal to $A_{x,normal} \cos\theta$, and $A_{x,normal}$

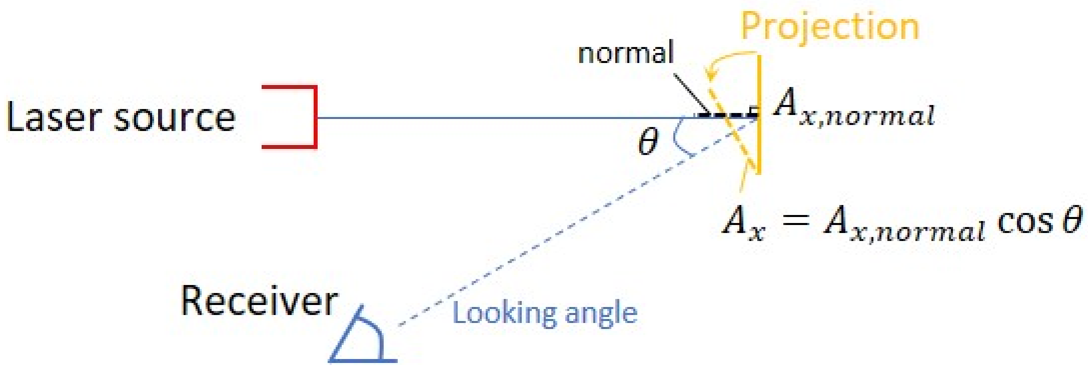


Figure 4.2: The Normal of the Target Surface Has an Angle θ to the Receiver's Look-Direction. The area subtended to the receiver becomes $A_{normal,cos\theta}$ which is a projection of $A_{x,normal}$ to the plane perpendicular to the receiver's look-direction.

is the area subtended to the receiver when the normal of the target surface is aligned with the look-direction of the receiver. Moreover, if sharp edges on the surface are illuminated by a laser beam or the object surface does not face the incident light perpendicularly, the light could be multi-reflected, and the pulse shape could be modified (Richmond and Cain, 2010).

Furthermore, as mentioned in Section 4.1.1, the beam size expands with distance, and it could become greater than the size of a target at Location 2, in which case, only a portion of the beam that hits the target can be reflected. On the other hand, if the lateral FOV of the receiver $A_{r,FOV}$ is smaller than the beam size and the target size, the area that can be seen by the receiver is limited by the receiver's FOV. A schematic of the relative size between the beam and the lateral FOV of the receiver at the object distance and the target size are given in Figure 4.3. A mono-static optical path is used and the target surface is assumed perpendicular to the look-angle of the transmitter and receiver. From the figure, we can see the relative size determines the value of A_x, and thus, the return power. Specifically,

- Case 1: if $A_{r,FOV} > \min(A_b, A_{target})$ and $A_{target} \geq A_b$, the entire laser beam hits the target and receivers' FOV is sufficient large. Thus, $A_x = A_b{\cdot}\cos\theta$ (projection of the reflected beam front to the plane perpendicular to the receiver's look-direction).

- Case 2: if $A_{r,FOV} > \min(A_b, A_{target})$ and $A_{target} < A_b$, the target size is smaller than the beam size and receivers' FOV is sufficient large. In this case, only the portion of energy that hits the target is reflected and the rest of the energy keeps traveling. Thus, $A_x = A_{target} \cdot \cos\theta$.

- Case 3: if $A_{r,FOV} \leq \min(A_{target}, A_b)$, the receivers' FOV is the limiting factor, since no matter how much energy is reflected by the target, only the energy falling into the receiver's FOV can be detected. Thus, $A_x = A_{r,FOV} = \pi\beta_r^2 R^2/4$, where β_r is angular FOV of the receiver. In this case, the angle between the receivers' look-direction and normal of the target surface does not matter any more.

Mathematically,

$$A_x = \begin{cases} A_b \cdot \cos\theta & \text{if } A_{r,FOV} > \min(A_b, A_{target}) \text{ and } A_{target} \geq A_b \\ A_{target} \cdot \cos\theta & \text{if } A_{r,FOV} > \min(A_b, A_{target}) \text{ and } A_{target} < A_b \\ A_{r,FOV} = \frac{\pi\beta_r^2 R^2}{4} & \text{if } A_{r,FOV} \leq \min(A_{target}, A_b) \end{cases} \tag{4.4}$$

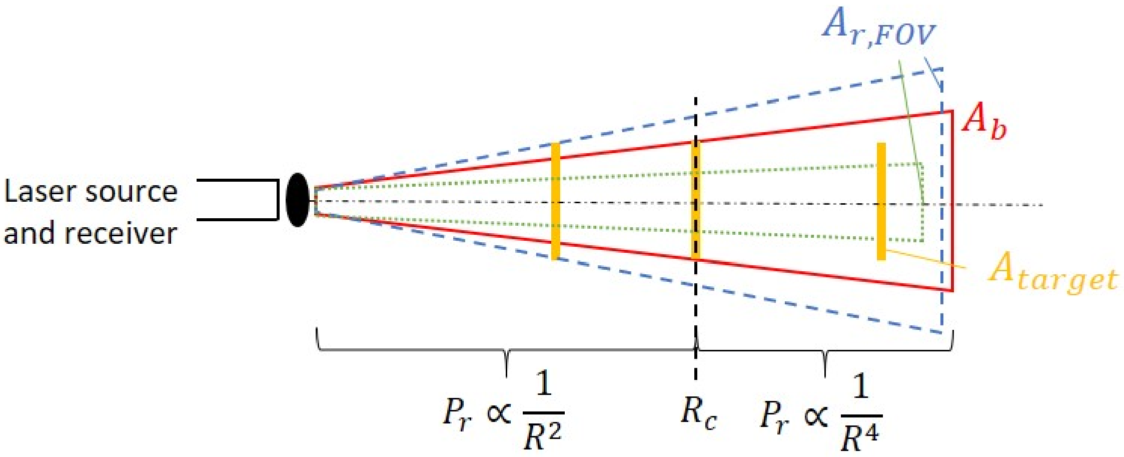

$A_{r,FOV}$
A_b
Laser source and receiver
A_{target}
$P_r \propto \frac{1}{R^2}$
R_c
$P_r \propto \frac{1}{R^4}$

Figure 4.3: Relative Size between $A_{r,FOV}$, A_{target} and A_b. In Case 3 (receiver's FOV is in green), the return power is always a function of $1/R^2$. However, in Case 1 and 2 (the receiver's FOV is in blue), as target distance increases to R_c (critical distance), the target size A_{target} becomes smaller than beam size A_b. Thus, only a portion of the laser beam can be reflected, resulting in a power drop of $1/R^4$. (Figure is adapted from Kasunic (2016)).

Plugging Equation (4.4) into Equation (4.2), we obtain:

$$P_r \propto \begin{cases} \frac{A_b}{R^2 A_b} \propto \frac{1}{R^2} & \text{if } A_{r,FOV} > \min(A_b, A_{target}) \text{ and } A_{target} \geq A_b \\ \frac{A_{target}}{R^2 A_b} \propto \frac{1}{R^4} & \text{if } A_{r,FOV} > \min(A_b, A_{target}) \text{ and } A_{target} < A_b \\ \frac{A_{r,FOV}}{A_b R^2} \propto \frac{1}{R^2} & \text{if } A_{r,FOV} \leq \min(A_{target}, A_b) \end{cases} \tag{4.5}$$

The relationship given in Equation (4.5) is important to understand how power decays with distance. We observe that the return power reduces rapidly with distance ($1/R^4$) when the target distance is larger than a critical distance R_c (target size smaller than beam size). This is frequently the case for flash lidar, which emit light with a wide FOV, given targets can only reflect a portion of the light. Consequently, the maximum achievable distance is dramatically reduced as distance increases. On the other hand, a scanning lidar usually has a narrow beam, so the power decay against distance is only $1/R^2$, which allows detection of targets at further distance.

4.1.3 Atmospheric Effects

During the propagation of laser beam through free space, the light interacts with atmospheric turbulence and molecules with different sizes and geometries. These interactions modify the beam shape, wavefront, and attenuate optical power. Moreover, ambient light from the surroundings inject noise into the lidar system.

Background light

The receiver of a lidar system does not only accept backscattered laser pulse but also light from environment, *e.g.* sunlight. The incident ambient light can be from ambient light backscattered by objects or from unwanted light directly shooting into the receiver (*e.g.* the receiver accidentally 'looking' into the sun). The power of incident background light can be calculated by Equation (4.2) with minor modifications.

First, we assume the background light is parallel light, and we replace the transmit laser power by the spectral radiance of the background light source.

In the first situation, since the background light only travels a single trip (distance R), the atmospheric attenuation is reduce to half, and no transmission coefficient of the transmitter lens is needed. A schematic of the background light propagation is given in Figure 4.4a, and a more specific description is given as following:

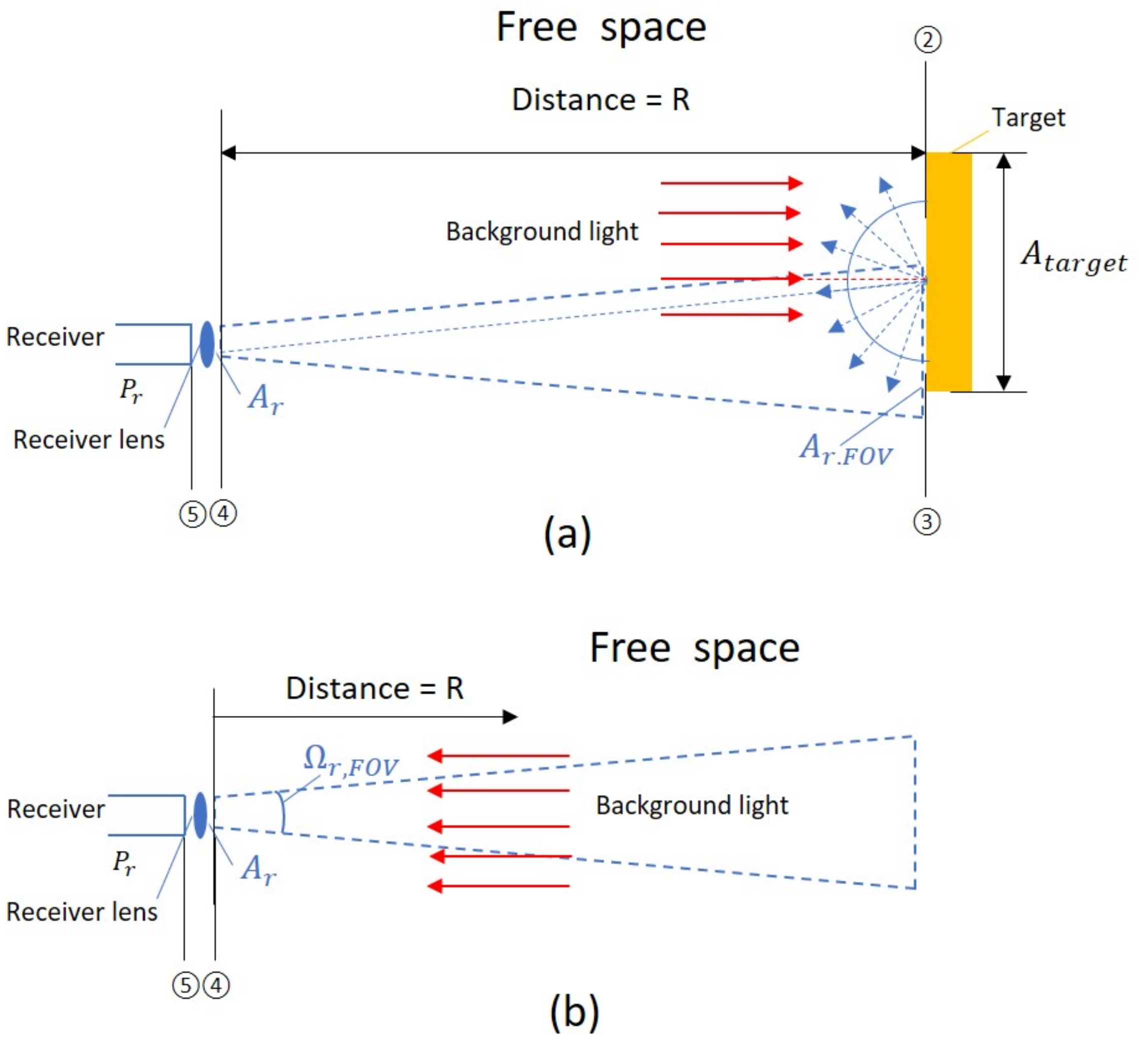


Figure 4.4: Schematic of Propagation of Background Light. (a) Background light is reflected by targets. (b) Background light directly shoots into the receiver.

- Location 2: position right before the background light reaches a target. The radiance of the background light $L_2 = L_\lambda \Delta_\lambda$, where L_λ is the spectral radiance of the background light (units of $W \cdot m^{-2} \cdot sr^{-1} \cdot nm^{-1}$) and $\Delta\lambda$ (units of nm) is the bandwidth of the spectral filter of the photon-receiver. Take the sunlight for example, the solar spectral radiance can be found in (ASTMG159-98, 1998) (Figure 4.5). Moreover, since background light usually has a wide spectrum, a spectral filter is applied to remove wavelengths beyond the spectrum of interests (1550 nm in this study). In this case, much background noise can be removed.

- Location 3: right after background light hits a target. The backscattered radiance

  $$E_{e3} = L_2 \cdot \rho \cdot \Omega_{r,\theta} = L_2 \cdot \rho \cdot \frac{A_x}{R^2}\ .$$

  In this case, the 'beam' size A_b is infinite large, so we only need to compare the receiver's lateral FOV and the target size:

  - If $A_{target} \leq A_{r,FOV}$, $A_x = A_{target} \cdot \cos\theta$.
  - If $A_{target} > A_{r,FOV}$: $A_x = A_{r,FOV} = \pi(\beta_r/2 \cdot R)^2$

- Location 4-5: the same as the discussion at the beginning of Section 4.1.

In summary, the optical power of backscattered background light received by a photon-detector can be obtained by:

$$\begin{aligned} P_{amb} &= (L_\lambda \Delta_\lambda) \cdot \rho \cdot \frac{A_x}{R^2} \cdot \eta_{atm,R} \cdot \eta_{opt,receiver} \cdot A_r \\ &= L_\lambda \Delta_\lambda \cdot \eta_{opt,receiver} \cdot \frac{A_x A_r}{R^2} \cdot \eta_{atm,R} \cdot \rho \end{aligned} \tag{4.6}$$

where

$$A_x = \begin{cases} A_{target} \cdot \cos\theta & \text{if } A_{target} \leq A_{r,FOV} \\ A_{r,FOV} = \pi(\beta_r/2 \cdot R)^2 & \text{if } A_{target} > A_{r,FOV} \end{cases} \tag{4.7}$$

In the second situation (Figure 4.4b), we assume the incident background light has infinite large beam. Thus, $A_x = A_{r,FOV}$ at Location 4, and the irradiation can be expressed as

$$E_{e4} = (L_\lambda \Delta_\lambda) \cdot \Omega_{r,FOV} = (L_\lambda \Delta_\lambda) \cdot \frac{A_{r,FOV}}{R^2}. \tag{4.8}$$

The optical power of direct-shooting background light received by a photon-detector (Location 5) can be obtained by:

$$\begin{aligned} P_{amb} &= E_{e4} \cdot \eta_{opt,receiver} \cdot A_r = (L_\lambda \Delta_\lambda) \cdot \frac{A_{r,FOV}}{R^2} \cdot \eta_{opt,receiver} \cdot A_r \\ &= (L_\lambda \Delta_\lambda) \cdot \pi(\beta_r/2)^2 \cdot \eta_{opt,receiver} \cdot A_r. \end{aligned} \tag{4.9}$$

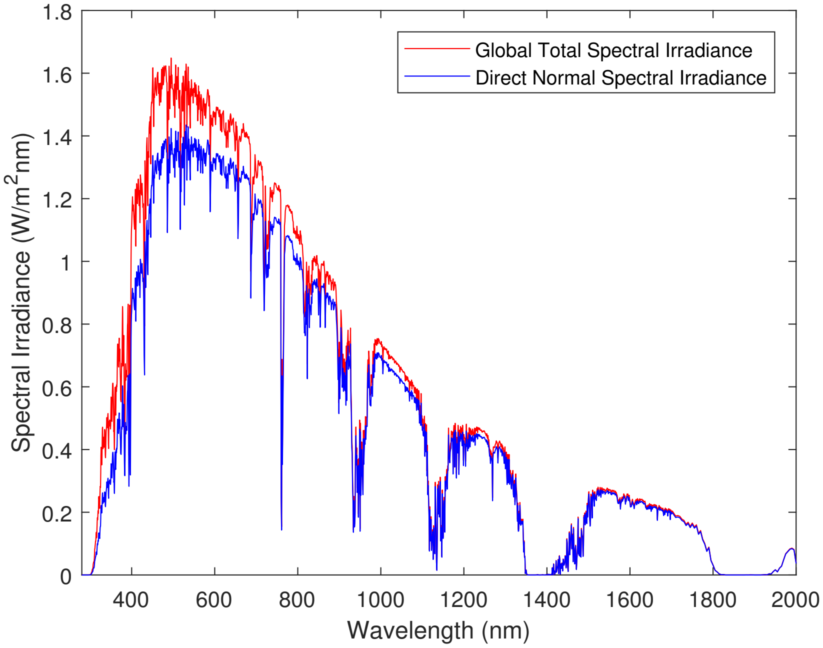


Figure 4.5: Solar Spectral Irradiance Measured at Air Mass 1.5 and 37 Degree Titled Surface (ASTMG159-98, 1998).

Power attenuation

During the propagation through the atmosphere, a portion of the laser beam is scattered and absorbed by molecules, fog, dust, and other aerosols, while the rest of the power is passed through the atmosphere. The scattering and absorption together contribute to the power attenuation and the ratio of attenuation can be obtained as $1 - \eta_{atm}$, in which η_{atm} is the atmospheric transmittance. Opposite to attenuation, the transmittance is defined as the ratio of the transmission to the incident light, and by following Beer's law, the transmittance through a distance $2R$ can be modeled as (Richmond and Cain, 2010)

$$\eta_{atm,2R} = e^{-2\int_0^R \alpha(\lambda,R)dR} \ ,$$

where α is attenuation coefficient in units of length. In principle, the transmission should be calculated by integrating the coefficient α over a range of $2R$. However, since the maximum achievable distance is less than 200 m for a TOF lidar, the attenuation coefficient can be assumed to be constant through the maximum observable distance. Thus, the transmission is reduced to

$$\eta_{atm,2R} = e^{-2\alpha(\lambda)R}. \tag{4.10}$$

The contribution to the power attenuation from absorption can be obtained from the transmittance spectra of atmosphere given the wavelength of interest. However, the scattering mechanics for particles of different sizes are different and thus, the power attenuation due to scattering varies accordingly. When the size of molecules is much smaller than a wavelength, Rayleigh scattering is the dominant scattering mechanism, in which case the attenuation coefficient varies as λ^{-4}. As the particle size approaches the wavelength (*e.g.* fog, haze or dust) of the laser beam, Mie scattering occurs primarily, and the attenuation coefficient follows different functions of wavelength. In this case, to model the laser propagation through different atmospheric conditions, the particle size is required, which, however, is usually not readily available. Alternatively, Kim *et al.* (2001) related atmospheric conditions to the visibility which can be easily obtained from weather stations or airports, and proposed an model that describes the attenuation coefficient is a function of visibility and wavelength:

$$\alpha = \frac{3.91}{V}(\frac{\lambda}{550\ nm})^{-q}$$
$$q = \begin{cases} 1.6 & V > 50 \text{ km} \\ 1.3 & 6 \text{ km} < V < 50 \text{ km} \\ 0.16V + 0.34 & 1 \text{ km} < V < 6 \text{ km} \\ V - 0.5 & 0.5 \text{ km} < V < 1 \text{ km} \\ 0 & V < 0.5 \text{ km} \end{cases} \tag{4.11}$$

where α is in units of km^{-1}, V is the visibility (units of km), λ is the wavelength in units of nm, and q is the size distribution of the scatting particles which is obtained empirically from experiments (Kruse *et al.*, 1962; Kim *et al.*, 2001).

The scattering mechanism of rain droplets is different from fog or haze, as the size of rain droplets is a few orders larger than a wavelength. In this case, the scattering is independent of wavelength, and the laws of geometric optics are usually used to describe the interaction of light with particles (Kim *et al.*, 2001). Moreover, the attenuation coefficient of rain becomes sensitive to droplet size and the rainfall rate (Middleton, 1952; Brazda *et al.*, 2012; Rahman *et al.*, 2008). Guo *et al.* (2015) pointed out that the scattering event increases with rainfall rate and propagation distance. As a result, the trajectory of light deviates further from straight lines, resulting in greater attenuation of laser power and delayed arrival time. Moreover, the arrival of multiple reflections by rain droplets on the detector also elongates the falling edge of the laser pulse and distorts the pulse shape. The interaction between rain and the laser beam is complicated, and advanced optical simulation may be necessary to obtain accurate results.

Speckle and scintillation

The coherency of a laser source is important to achieve a diffraction-limited beam. However, high temporal coherence could also lead to speckle noise. After a coherent laser beam is scattered by molecules in the atmosphere or by a rough surface of a target, the phase of the wavefronts is not identical. Wavefronts with different phases will interfere constructively and destructively, and result in bright and dark patterns on an imager (speckles). An imager is a photon-detector with multiple pixels, which is commonly used in a flash lidar. Speckle causes intensity variance across adjacent pixels, which degrades the image quality of a flash lidar. The speckle noise could also alter the return power seen by a scanning lidar.

Turbulence in the atmosphere also affects the laser propagation. Turbulence causes temperature variations and thus, variations of air density. Since the refractive index varies with air density, the refractive index fluctuates when a laser beam propagates through the turbulent atmosphere. The fluctuation in the refractive index is referred to as optical turbulence. Optical turbulence alters the phase, power and center position (meandering) of laser beams (Lane *et al.*, 1992). Meandering of laser beams is mainly due to large scale turbulence and it can be neglected because of the short length scale studied in this work. Phase distortion is a major effect among all those mentioned. The changes of optical index distort the phase of wavefronts, resulting in a break of diffraction-limited beam into speckles, and subsequently, intensity fluctuation across wavefronts. The intensity fluctuation observed by a receiver is called *scintillation.* Since turbulence occurs all the time, the intensity also varies with time. The variance σ_ϕ^2 of phase changes across the aperture of a receiver with a diameter D_r can be quantified by (Fried, 1966):

$$\sigma_\phi^2 = 1.03(\frac{D_r}{r_0})^{\frac{5}{3}},$$

where r_0 is the Fried parameter, defined as the diameter of an area of which the standard deviation of phase aberration equal to 1 radian. In other words, the Fried parameter defines the diameter of a bright or dark spot of the speckle, within which the phase variation (intensity variation) is negligible. That is why people usually compare the diameter of a receiver aperture D_r with the Fried parameter. If D_r is equal to or less than r_0, the image captured by the aperture is inside a bright/dark spot. Thus, the intensity variation is small and the effect of scintillation can be ignored. On the other hand, if the aperture is larger than r_0, more than one spots appear on the aperture, resulting in intensity variation. In this case, the effect of scintillation becomes significant.

The Fried parameter is a function of turbulence strength and propagation distance (Fried, 1966) as illustrated by:

$$r_0 = \left[0.423k^2 \sec\zeta \int_R C_n^2(z)dz\right]^{-3/5}, \tag{4.12}$$

where $k = 2\pi/\lambda$ is the wavenumber, λ is the wavelength in units of meter, and ζ is the zenith angle. The turbulence strength is denoted by C_n^2, which is a function of altitude z and can be obtained from experiments. Since, we consider more about the horizontal propagation of a laser beam in the application of autonomous vehicles, we assume the turbulence strength is constant through a range of R. The turbulence strength C_n^2 near the ground in hot weathers ranges from 10^{-14} to 10^{-12} $m^{-2/3}$, and 10^{-12} $m^{-2/3}$ is considered to be very turbulent conditions (Andrews *et al.*, 2001). After plugging the values of C_n^2, maximum achievable distance (300 m) and the operational wavelength (1550 nm) into Equation (4.12), we obtained $r_0 \approx 16$ cm when $C_n^2 = 10^{-14}$ $m^{-2/3}$, and $r_0 \approx 1$ cm when $C_n^2 = 10^{-12}\ m^{-2/3}$. Since the diameter of the receiver used in this work is 2.54 cm, the scintillation effect is negligible in most conditions. However, if the temperature and turbulence strength becomes extremely high, the scintillation effect should be taken into account.

4.1.4 Optics

Optics in a lidar system also affect the return power of a laser beam. The efficiencies $\eta_{opt,transmitter}$ and $\eta_{opt,receiver}$ in Equation (4.2) are actually a multiplication of optical efficiencies (*e.g.* $\eta_{opt,transmitter} = \eta_{mirror} \cdot \eta_{lens,transmitter}$, and $\eta_{opt,receiver} = \eta_{filter} \cdot \eta_{lens,receiver} \cdot \eta_{FOV}$, where η_{lens} is the transmittance of the transmitter lens or receiver lens, η_{mirror} is the efficiency of a mirror of a scanning lidar, η_{filter} is the transmittance of the spectral filter, and η_{FOV} is the overlap ratio of the FOV of the transmitter and the receiver). The lens of the transmitter and receiver only pass a portion of energy, and the spectral filter of a receiver that removes wavelengths out of the spectrum of interest also reduces pulse energy.

Generally, there are two types of optical system: mono-static system and bi-static system. In a mono-static system, the transmitter and receiver share the same optical pathway, and a circulator is used to separate transmit signals and return signals (Figure 4.6). Oppositely, in a bi-static optical system the transmitter and receiver are located separately, and because of the separation (Figure 4.7), a bi-static system has a blind zone in front of the system. The blind zone means within a short distance from the lidar, the receiver and transmitter have zero overlapping FOV's. In other words, the receiver cannot capture any light emitted from the transmitter. As distance increases, the FOV of the receiver and the transmitter begin to overlap partially. They fully overlap with each other after a sufficient distance. The partially overlapping zone reduces laser power until the fully overlapping zone. The blind, partial overlapping and fully overlapped zone are also shown in Figure 4.7.

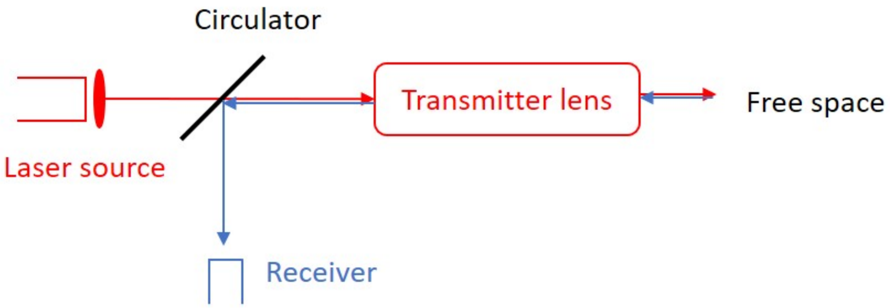


Figure 4.6: Schematic of a Mono-Static Optical System.

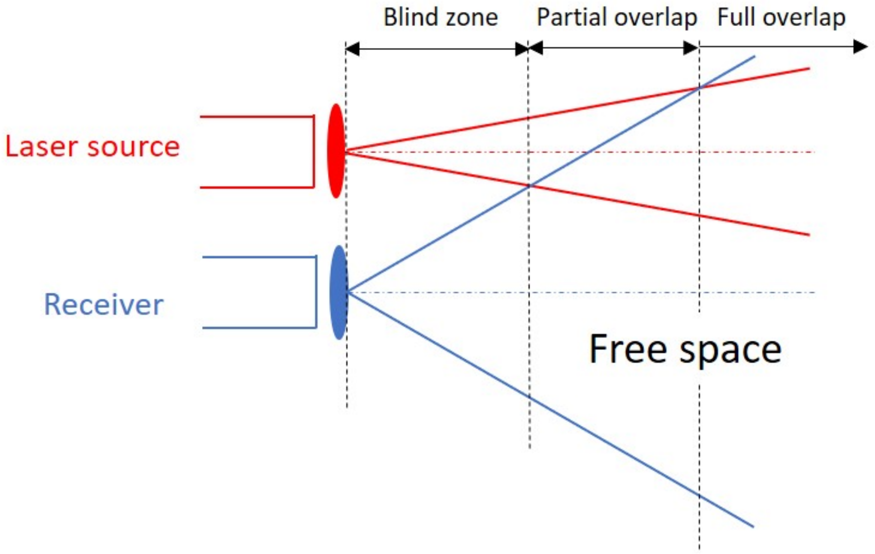


Figure 4.7: Schematic of a Bi-Static Optical System and Illustration of FOV Overlap.

4.2 Modeling Pulse Propagation

4.2.1 Elements in Propagation Model

The propagation model proposed in this dissertation follows the Lidar Equation (Equation (4.2)) and the models used to calculate the variables in the Lidar Equation are listed below:

- Beam divergence: Equation (4.3);
- Area seen by a receiver with a look-angle θ to the normal of target surface: Equation (4.4);
- atmospheric attenuation: Equation (4.10) and Equation (4.11);
- Background light: Equation (4.6) and Equation (4.7);
- Receiver area: Equation (4.1);
- Transmission of transmitter lens and FOV overlap between transmitter and receiver are taken into account in $\eta_{opt,transmitter}$;
- Optical filter and transmission of receiver lens are accounted in $\eta_{opt,receiver}$.

4.2.2 Validation

The propagation model was validated by comparing the peak voltage of return signals between modeled results and measurements from a prototype lidar (Bhaskaran, 2018). For the propagation model, the peak power was first calculated by the Lidar Equation and then converted to voltage using $V_r = P_r \cdot R_0 \cdot M \cdot R_{TIA}$, where R_0 is the responsivity of a photon-detector, M is the APD gain, and R_{TIA} is the gain of a transimpedance amplifier (TIA) (Kasunic, 2016). The values of the parameters are given in Table A.1 and Table 4.1. In the propagation model, we used the same value of the parameters in the prototype lidar specifications (if available). The transmissions in Table 4.1 that is not available were from Burns *et al.* (1991) who built a similar pulsed range-finding lidar. We also assumed clear environmental condition. Thus, the visibility was set as 10 km. For comparison, we fired the prototype lidar to targets at different distances and used an oscilloscope to collect the return signals.

Table 4.1: Parameters of Propagation Model

Parameter	Symbol	Value
Transmission of transmitter lens	$\eta_{lens,transmitter}$	0.9
Transmission of receiver lens	$\eta_{lens,receiver}$	0.9
Visibility	V	10 km
Transmission of spectral filter	η_{filter}	0.7

Aluminum foils were used as targets. A schematic of the experimental setup is given in Figure 4.8.

The peak power at different distances from the propagation model and experiments are given in Figure 4.9. From the comparison, we can see the results from the propagation model and experiments mainly follow the same trend. The discrepancy could be due to that the reflection at the aluminum foil targets which is not pure diffuse and orienting the surface to the receiver's look-angle was difficult. When the surface is not perpendicular to the receiver's look-angle, A_x is reduced, and part of energy was lost. When the target was moved from one distance to another, the orientation of the target may be changed, which could also cause variation of the return power.

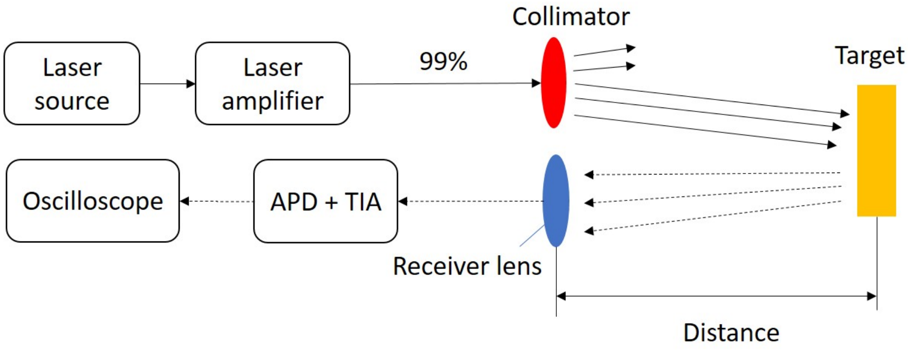


Figure 4.8: Experimental Setup for the Validation of the Propagation Model.

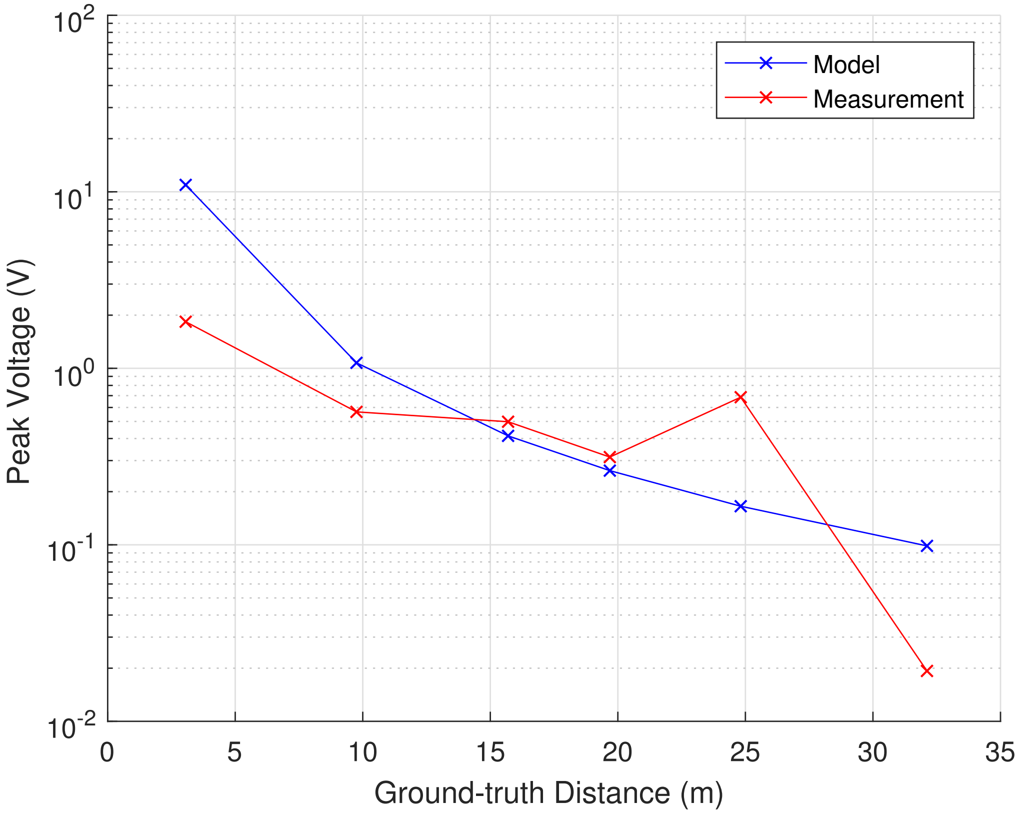


Figure 4.9: Peak Power of Modeled Signals and Experimental Signals Measured by an Oscilloscope.

Chapter 5

PHOTON-RECEIVER

5.1 Photon-detectors and System Noise

Photon-receivers (or receivers) are systems that receive laser beams and convert the incident photons into electrical signals. Generally, a photon-receiver consists of lens, photon-detectors, amplifiers and analog front ends (AFE's). The function of a photon-detector is to convert photons that are collected by the receiver lens to electrical currents. Amplifiers convert the output current to voltage and amplify it to exceed the noise level. Transimpedance amplifiers (TIA's) are one of the commonly used amplifiers. The amplified signals could be noisy due to noise from various sources, so AFE's are usually placed after an amplifier to truncate the noise spectrum and increase the signal-to-noise ratio (SNR) of the signal. Depending on the purpose, different types of AFE's could be applied. This section will introduce the fundamentals of the receivers, and the application of AFE's to various types of noise follows.

5.1.1 Characteristics of Photon-detectors and Amplifiers

A photon-detector is a device that has a *p*-*n* junction that can absorb photons and generate electron-hole pairs when the energy of the absorbed photons exceeds the band-gap of the materials. The flow of the electrons creates an electrical current, which is also called photo-current. Photon-detectors can be classified into different types according to their stimulation mechanism, and in this study, we focus on semiconductor photon-detectors (photon-detector here after).

Quantum efficiency

In a photon-detector, the incident photons do not always generate electron-hole pairs, and the activated electrons are not always collected at the respective electrodes to produce photo-current. The quantum efficiency of a photon-detector is a ratio of the average number of photons required by a photon-detector to produce one electron of photo-current. It is given by:

$$\eta = \frac{n_e}{n_{ph}}, \tag{5.1}$$

where n_e and n_{ph} are the number of electrons of the photo-current and the number of incident photons, respectively. The quantum efficiency depends on the photon wavelength, doping materials and the structure of photon-detector.

Responsivity

Responsivity is another measure of the capability of a photon-detector to convert optical power to electrical current. It is defined as the ratio of the incident optical power (P_{ph}) to photo-current (I_e)

$$R_0 = \frac{I_e}{P_{ph}}, \tag{5.2}$$

and it is in units of A/W. Applying the relation between optical power and number of photons per second ($P_{ph} = n_{ph}h\nu/\Delta t$) and between current and number of electrons per second ($I_e = n_e q/\Delta t$), the relationship between the responsivity and the quantum efficiency can be derived as

$$R_0 = \frac{\eta q}{h\nu} = \frac{\eta q \lambda}{hc} = \frac{\eta \lambda(\mu m)}{1.24} \tag{5.3}$$

where q is the charge of electron, h is the Planck's constant, λ is the wavelength, and ν is the center frequency of the photons and $\nu = c/\lambda$. Equation (5.3) presents a linear dependency of the responsivity on the photon wavelength for a constant quantum efficiency. It is as expected since one watt of optical power has more photons for pulses with a longer wavelength ($n_{ph} = \frac{1(W)\cdot\Delta t}{(hc/\lambda)}$), which consequently produces a larger photo-current if the quantum efficiency remains the same.

Different materials of a photon-detector have their own operational ranges of wavelength due to their band-gap energy, and the responsivity also varies with the wavelength within the range. Therefore, the selection of laser wavelength determines the material used in a photon-detector. Figure 5.1 schematically illustrates the relationship between the responsivity and the wavelength of different materials. We can see that silicon has its optimal operation at NIR, but it is transparent at SWIR. On the other hand, InGaAs has much higher responsivity at SWIR.

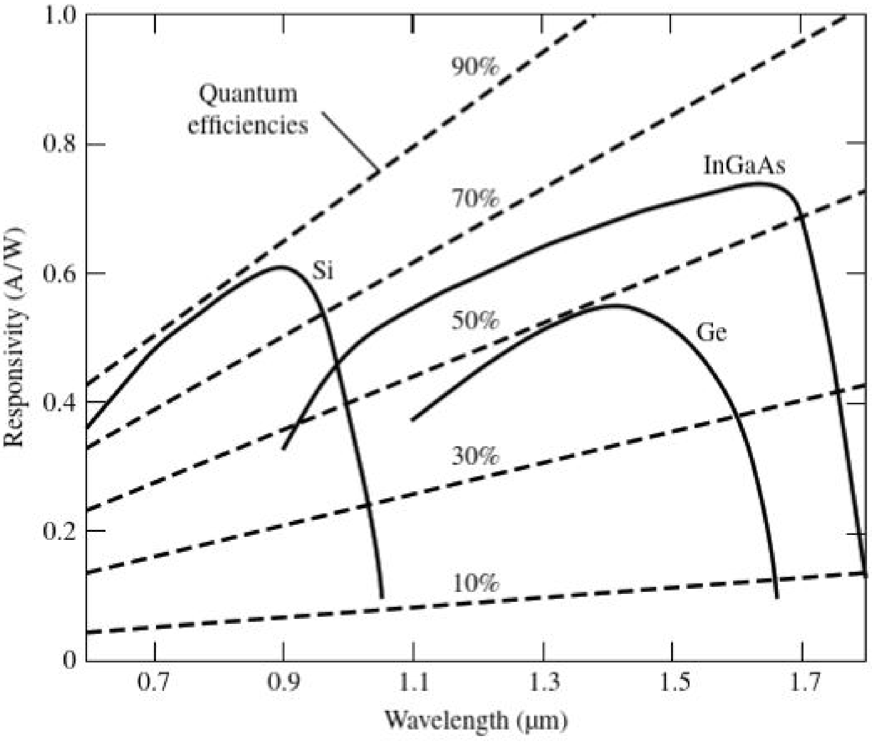


Figure 5.1: Schematic of Responsivity vs Wavelength for Different Materials (Kumar and Deen, 2014).

Response time and bandwidth

A fast response time of the photon-detector and amplifier are of importance for a receiver to reproduce the temporal characteristics (*e.g.* rise time) of the transmit signal. The response time τ_t of an electrical device is determined by the intrinsic junction capacitance (C) and load resistance (R):

$$\tau_t = RC = \frac{1}{2\pi \cdot B}, \tag{5.4}$$

where B is the bandwidth of the system, which is related to the rise time of the output signal by (Ziemer and Tranter, 2006)

$$t_r = \frac{0.35}{B}. \tag{5.5}$$

Equation (5.4) and Equation (5.5) illustrate that to capture a fast rise time, a wide bandwidth is necessary which requires a photon-detector and amplifier that has a small RC time response.

Linearity

The linearity is another key characteristic of a photon-detector. It describes how linear the conversion is from incident optical power to photo-current within the operational range of the photon-detector and conversion from photo-current to voltage by an amplifier. The operational range of a photon-detector is bounded on lower limit by the noise equivalent power (or voltage) of the photon-detector and on the upper limit by the saturation power (or voltage), beyond which the photon-detector won't response further. The noise equivalent power is defined as the signal power that is equal to the standard deviation of noise, *i.e.* SNR = 1. The linearity of a photon-detector and an amplifier is significant to maintain the shape of the incident laser pulses. Otherwise, a distorted signal could lead to errors in TOF measurements.

5.1.2 Type of Photon-detectors

Common photon-detectors include PIN photodiodes and avalanche photodiodes (APD's). PIN photodiodes are photodiodes that have an undoped intrinsic region in between the p-doped and n-doped regions, and they convert optical power to photo-current with unit gain. In contrast to PIN's, APD's are photodiodes that have internal gain greater than one where one photon can produce more than one electron, which makes it widely used for detection of weak signals.

The internal gain of the APD's results from the avalanche multiplication process. In the process, when one incident photon activates several free electron-hole pairs, the charge carriers are accelerated by a strong electric field to a high enough energy, so that they can knock out more free carries. This process is referred to as impact ionization and a chain of impact ionization results in a multiplication of the number of charge carriers in the avalanche process, and therefore, an larger photo-current. In an APD, the high electric field is provided by a sufficiently high reverse-bias voltage. If the reverse-bias voltage is less than the breakdown voltage of the semiconductor, the APD is said to be working in a linear mode, in which case the resultant photo-current is linear with the incident optical power and the internal gain is of the order of ten or a few hundred depending on the material. On the contrary, if the reverse-bias voltage exceeds the breakdown voltage, the number of produced carriers is extremely large, and the output current is no longer linear with the optical power. Such types of APD's are referred to as Geiger-mode APD's (GM-APD's). Compared to linear-mode APD's, Geiger-mode APD's are much more susceptible to other events that can initialize the avalanche process other than return pulses, which makes the output current much noisier. Additionally, GM-APD's are not capable of counting the number of photons. Therefore, more than one GM-APD is usually used for the application of signal detection. One example is the multipixel photon counter (MPPC).

5.1.3 System Noise

When a photon-detector converts optical power to electrical signals (current or voltage), noise is also added. Here, we mainly focus on electrical noise in APD's, and for simplicity, all the noise mentioned in this section is current. Major noise in a photon-detector includes shot noise and thermal noise.

Shot noise

Shot noise results from the discrete nature of photons and electrons, and it results in variation of signal magnitude. Shot noise occurs on all kinds of signals including return laser pulses, ambient light, and also inherent dark current in an electrical system. Those shot noises are referred to as signal-induced shot noise (σ_{ss}), background-induced shot noise (σ_{bs}) and dark current-induced shot noise (σ_{ds}), respectively. Shot noise follows a Poisson distribution at each event, and the standard deviation of shot noise current can be expressed by:

$$\begin{aligned}\sigma_{ss} &= \sqrt{2qI_{sig}B \cdot M^2 F} \\ \sigma_{bs} &= \sqrt{2qI_{bc}B \cdot M^2 F} \\ \sigma_{ds} &= \sqrt{2qI_{dc}B \cdot M^2 F}\end{aligned} \tag{5.6}$$

where the current I is the mean current without avalanche process, and the subscript sig, bc and dc stand for pulse-induced current, ambient-light-induced current and dark current. The average pulse-induced current and ambient-light-induced current can be calculated using the optical power P_{return} and the responsivity of the photon-detector by $I_{sig} = P_{return}R_0$. The dark current can be measured from electrical devices. The bandwidth B in Equation (5.6) is inverse proportional to the integration time (τ_{int})

of a photon-detector: $B = 1/2\tau_{int}$. The term $2qIB$ is due to the random arrival times of photons collected by a PIN, and M^2F applies to photon-detectors with gains, like APD's. The variable M is the APD gain, and F is the excess noise factor of the avalanche process. The M used here is the ensemble average gain, and the excess noise factor is to take care of the gain variation. Since each electron-hole pair does not always experience the same multiplication during the avalanche process, the variation causes additional noise (Webb, 1974). The excess noise factor can be calculated by

$$F = KM + (1 - K)(2 - 1/M),$$

where K is the ionization ratio (Kilpela, 2004). Under the assumption that all the shot noises are independent to each other, the total shot noise in a receiver is equal to the summation of the shot noise from all the currents:

$$\sigma_S = \sqrt{\sigma_{ss}^2 + \sigma_{bs}^2 + \sigma_{ds}^2} \ .$$

Thermal noise

Thermal noise is also called Johnson Noise, which originates from free electrons agitated by heat in electronic circuits (Johnson, 1928). Different than shot noise, thermal noise follows a Gaussian distribution and is independent on signal amplitude. On the other hand, thermal noise is a function of temperature and system resistance:

$$\sigma_{th} = \sqrt{\frac{4k_BTB}{R}} = NSD\sqrt{B}, \tag{5.7}$$

where k_B is the Boltzmann's constant, and NSD is the noise density spectral density (equivalent to noise equivalent power) which is usually available for a TIA product. In a receiver, the resistance of the amplifier is much greater than that of the photon-detector. Thus, the bandwidth of the amplifier is used in Equation (5.7). Note that the thermal noise increases with amplifier bandwidth, but as discussed in

Section 5.1.1, a wide bandwidth is required to capture a fast rise time. Therefore, a trade-off needs to be made between the rise time and the thermal noise.

Signal to Noise ratio

Given the shot noise and thermal noise, the standard deviation of the total noise can be calculated from Equation (5.6) and Equation (5.7), assuming the noise are independent of each other:

$$\sigma_{total} = \sqrt{\sigma_s^2 + \sigma_{th}^2}. \tag{5.8}$$

Given the total noise and signal we now can calculate the signal-to-noise ratio (SNR), which is a key metric describing the ratio of the signal to the noise in a system. The SNR can be expressed by :

$$SNR = \frac{I_{sig}M}{\sigma_{total}} = \frac{I_{sig}M}{\sqrt{\sigma_{ss}^2 + \sigma_{ds}^2 + \sigma_{bs}^2 + \sigma_{th}^2}}. \tag{5.9}$$

Each term has been described in the above sections. Note that, here we use current to define the SNR. Alternatively, we can also use electric power ($\propto I^2$) to defined the SNR in units of dB, giving $SNR_{dB} = 10 \log 10(SNR)$.

5.1.4 System Noise Modeling

In the receiver module, the shot noise and thermal noise are modeled differently. For the shot noise, since the Poisson distribution expresses the probability of distribution of positive numbers, we first convert the mean current I to mean number of electrons $\overline{N}$ using

$$\overline{N} = IM^2F \cdot \Delta t/q = \frac{IM^2F}{2Bq},$$

where Δt is the integration time of an ADC, and q is the charge of electrons. Then, we generate the random number of electrons following the Poisson distribution with the mean of $\overline{N}$, and convert the number of electrons back to current. Mathematically,

$$\sigma_i^2 = \sigma_N^2 \cdot (\frac{q}{\Delta t})^2 = \overline{N} \cdot (\frac{q}{\Delta t})^2 = 2qBIM^2F. \tag{5.10}$$

Thus, the random shot-noise current with a STD of σ_i is generated. On the other hand, the thermal noise is directly generated using σ_{th} as the STD and zero as the mean of the Gaussian distribution (Equation (5.7)). The total noise can be calculated by Equation (5.8) and added to each sample of a return signal. Note that the calculated σ in the above equations is for each sample in a signal, which means the value of σ may vary between samples depending on the type of noise. In addition, no optical noise was included in the model, such as speckle noise or overlapping of multiple returns. The values of the variables used in the equations can be found in Table A.1, or are already described in the particular sections. An example of the return pulse from a target at 15 m is given in Figure 5.2.

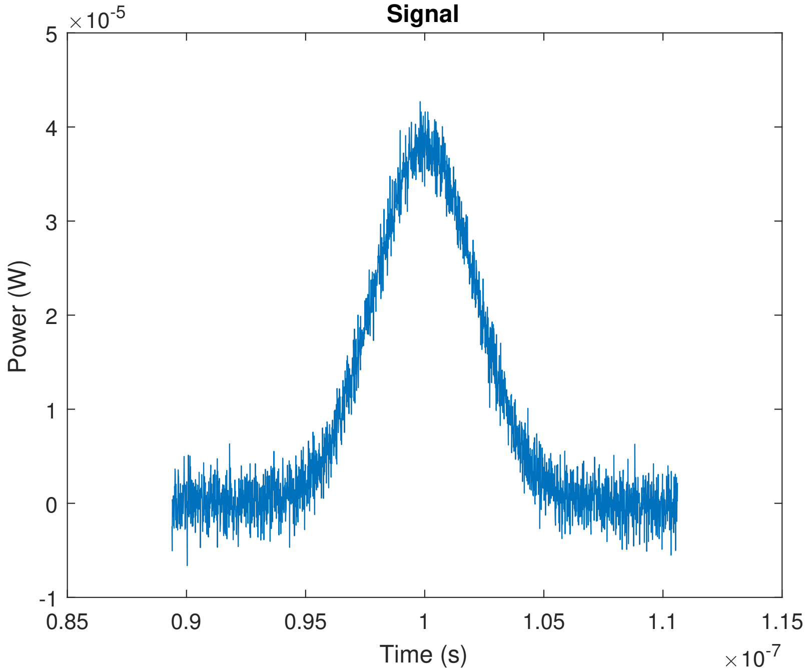


Figure 5.2: Return Signal with Modeled Noises.

5.1.5 White Noise and Structured Noise

In the above section, noise is classified based on their source. Alternatively, according their characteristics in time and frequency domain, noise can also be categorized into three classes: white noise, frequency-structured noise, and temporal-structured noise.

White noise

White noise is the noise that has a constant power spectral density through the whole frequency spectrum. Both shot noises and ideal thermal noise are white noise (Kittel, 2004; Blanter and Büttiker, 2000). Their power spectral densities can be expressed as

$$\begin{aligned} S_{shot}(f) &= 2q\overline{I} \\ S_{thermal}(f) &= 4k_BRT \end{aligned} \tag{5.11}$$

where S represents the power spectral density, $\overline{I}$ is the average current over the detector integration time, R is the system resistance, and T is the overall temperature of the system. From Equation (5.11), we can see the power spectral density of shot noise and ideal thermal noise is independent to frequency. In practice, thermal noise is only white through a finite bandwidth of an electrical system (Kasunic, 2016). Even though shot noise and thermal noise are all white noise, their magnitude distributions are different. Therefore, shot noise belongs to Poisson-white noise and thermal noise Gaussian-white noise.

Frequency-structured noise

Noises can also have certain structures in the frequency domain and periodical appearance if seen in the time domain. Such noise is referred to as frequency-structured noise

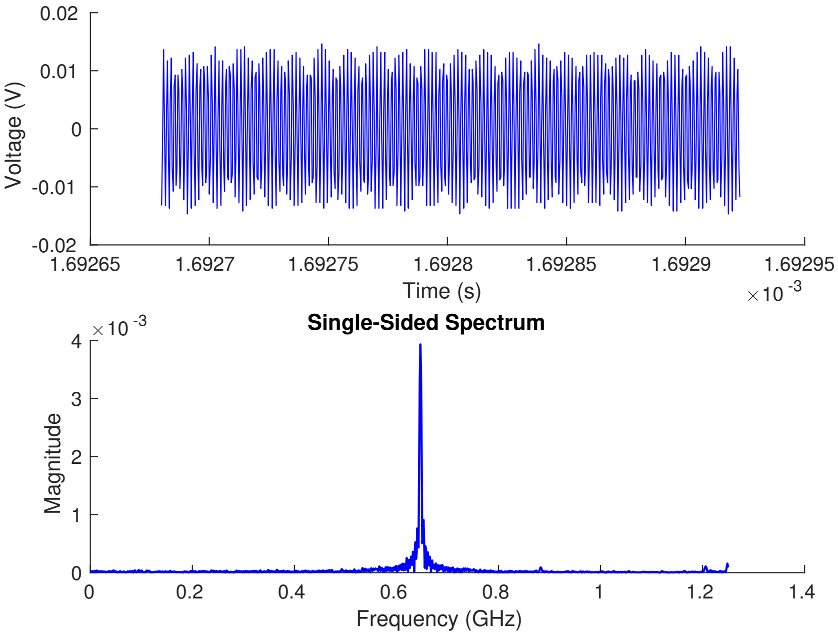


Figure 5.3: Frequency-Structured Noise and Its Power Spectrum. The noise oscillates at 645 MHz. The signal is from the prototype lidar.

in this study. Frequency-structured noises can be divided into in-band frequency-structured noise and out-of-band noise depending on whether the noise frequency is within the bandwidth of the signal of interests. Frequency-structured noise could result from unclean power supplies, impedance mismatch at connections between electrical components, or other periodic noise sources. An example of the frequency-structured noise is given in Figure 5.3, in which the noise oscillates with a frequency of 645 MHz.

Temporal-structured noise

Contrary to frequency-structured noise that has a periodic occurrence, noise can also occur randomly in the time domain, which is referred to as temporal-structured noise. Temporal-structured noises usually have certain shapes (structures) in the time

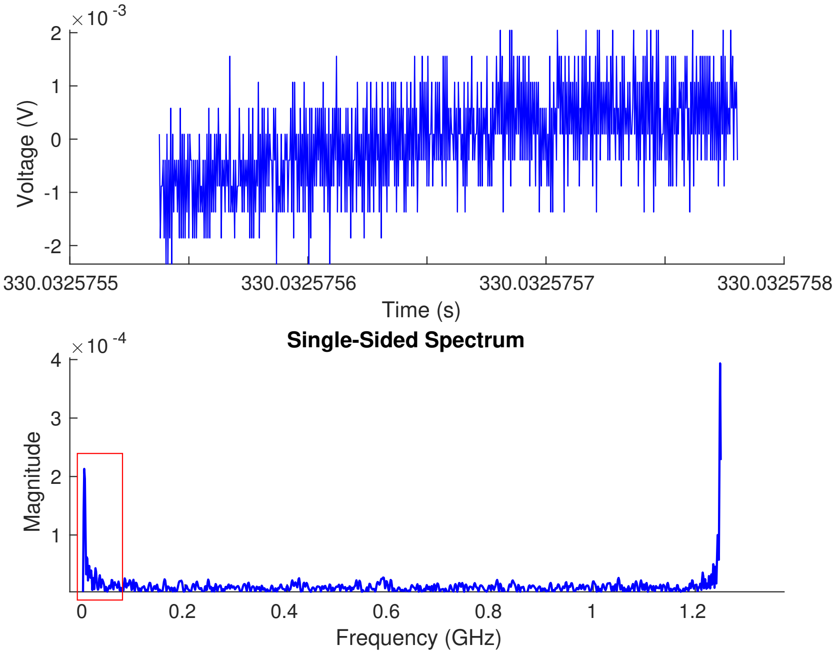


Figure 5.4: Temporal-Structured Noise. A frequency-structured noise at 1.22 GHz also exists on the signal. The temporal-structured noise causes the power spectrum in the red box. The signal is from the prototype lidar.

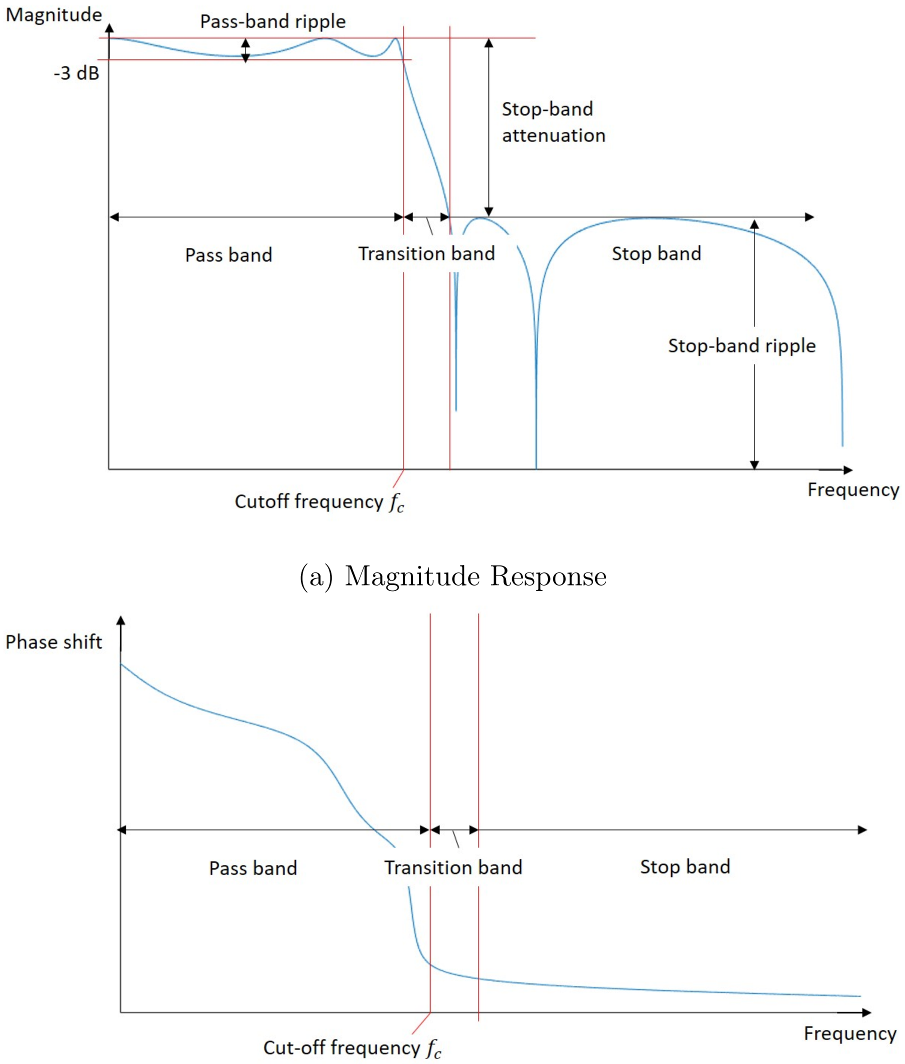
domain, but its appearance is unpredictable. Temporal-structured noises could result from ambient light, the reflection of a laser pulse from surrounding objects except the target, and loss connection of cables. An example of the temporal-structured noise is shown in Figure 5.4, in which the temporal-structured noise results in the power at frequencies inside the red box.

5.2 Analog Filters

In a TOF lidar, return signals can be contaminated by all the kinds of aforementioned noise. Distortion of signals could result in uncertainty of time measurement (Details will be described in Chapter 6, 8 and 9). Hence, the return pulses need to be separated from the noise and restored if their shape is distorted. The noise can be removed by placing an analog filter before a timing device, and such analog filters are also referred to as analog front ends (AFE's). Analog filters act on continuous-time (analog) signals and manipulate the spectra of a signal to reduce or remove the noise frequencies. The simplest analog filters are RLC filters which are composed of capacitors, inductors, and resistors, and other analog filters include crystal filters, microwave filters, and mechanical filters. This section will introduce the fundamental characteristics of analog filters, followed by the application of different analog filters to various types of noise.

5.2.1 Frequency-domain Characteristics

The characteristics of an analog filter can be described in the time and frequency domain. The frequency-domain characteristics will be introduced in this section, and time-domain ones will be discussed in the next section. The characteristics of an analog filter can be described by its frequency response which consists of the magnitude response and phase response. The magnitude and phase response are schematically shown in Figure 5.5, in which a low-pass filter is taken as an example. The magnitude response describes the magnitude change (gain) caused by a filter at different frequencies and the phase response presents the phase shift (or phase delay). The frequency response can be divided into three regions: *passband*, *transition band* and *stopband*. The passband is a range of frequencies passed by the filter, the stopband is the frequencies that are blocked, and the region in between is the transition band. According to the position of the passband over the whole spectra, filters could be classified as low-pass filters, high-pass filters, band-pass filters, band-stop filters, and all-pass filters. The passband is bounded by a *cut-off frequency* f_{cut}, defined to be where the frequency magnitude response is reduced to -3 dB. In the passband, the frequencies are desired to be unaltered (flat passband or unity gain), but the gain



(a) Magnitude Response

(b) Phase Response

Figure 5.5: Schematic of the Frequency Response of a Low-Pass Filter.

may vary in certain filter designs, and the fluctuation is called *passband ripple*. Since a large passband ripple alters the magnitude of in-band frequency components, the passband ripple should remain small in filter design. In the stopband, the blocked frequencies are attenuated by a *stopband attenuation* which is measured between the peak of the passband and the largest stopband lobe. To sufficiently attenuate the stopband frequencies, the stopband attenuation should be adequately large. Note that filters usually have different gains at the passband and stopband, but the all-pass filter, as indicated by the name, passes all the frequencies with a unity gain. Even though an all-pass filter has no change on the magnitude of a signal, it plays an important role in modifying the phase of the signal.

The *roll-off* of a filter defines the *steepness* or slope of the magnitude response in the transition band, and it also sets the corner frequency of the stopband. A fast roll-off is preferred for a filter to truncate undesired frequencies band with minimum impact on the rest frequency components. However, a fast roll-off could result in distortion (passband ripples) on the passband for certain filters. Another key characteristic of a filter is *order*. The order of a filter is defined as the highest exponent in the numerator or denominator of the transfer function. The concept of the transfer function will not be expanded in this work. In general, a filter with a higher order has more capacitors, inductors and resistors in the structure and it can achieve a better performance. However, a high order also increases the complexity of the filter structure.

5.2.2 Time-domain Characteristics

In addition to frequency-domain characteristics, the performance of an analog filter in the time domain is also important, and the step response of a filter is a useful tool to analyze a filter in the time domain. The time-domain parameters include *rise time* and *overshoot* of the step response. The rise time of a filter describes the transition period of the step response. It should be shorter than the duration of a signal event (*e.g.* changing from 'low' to 'high') so that the event can be distinguished from the step response. Second, overshoot changes the magnitude of a signal, so it should be avoided at a filter design. The phase response of a filter also plays a significant role in the time-domain characteristics of a filter. One of the key concepts is *group delay*, defined as

$$\tau_g = -\frac{d\Phi(\omega)}{d\omega}, \tag{5.12}$$

where $\Phi(\omega)$ is the phase response at angular frequency ω and $\omega = 2\pi f$. The group delay is equal to the slope of the phase response which depicts the linearity of the phase response. Here, the group delay should not be confused with the phase delay which is defined as

$$\tau_\Phi = -\frac{\Phi(\omega)}{\omega}. \tag{5.13}$$

The phase delay of a filter only describes the absolute phase shift at each frequency without any expression of the relationship between adjacent frequencies as the group delay does. The linearity of the phase response of a filter is a powerful indicator of the distortion enforced by the filter. Using the Fourier transform, a signal can be decomposed into several sinusoidal components with different frequencies. If a filter has a linear phase response, all the frequency components are delayed by the same phase. Thus, all the components can reconstruct the original signal, and only the time is shifted. In this case, the phase delay is equal to the group delay. However, if a filter has a nonlinear phase response, the frequency components of the signal are shifted by different amounts. Thus, they can no longer compose the original signal shape, *i.e.* the signal is distorted. If symmetric signals are used as in our case, the symmetry of the signal is impacted. Therefore, a linear-phase filter is preferred for applications that require the preservation of signal shapes. All-pass filters can also provide phase compensation. Since all-pass filters are designed to have different phase shifts among various frequencies, it can be used after other filters in a system to compensate the undesired phase shift.

5.2.3 Characteristics of Different Analog Filters

According to the characteristics of a filter in time and frequency domain, filters can be designed in various types. The most common filters are Butterworth filter, Chebyshev filter, Elliptic filter, and Bessel filter. The frequency response of the filters is shown in Figure 5.6. The magnitude response shows that both Butterworth and Bessel filter has no ripple at the passband, but the roll-off is slower than the other two. On the contrary, Chebyshev and Elliptic filters sacrifice the flatness of

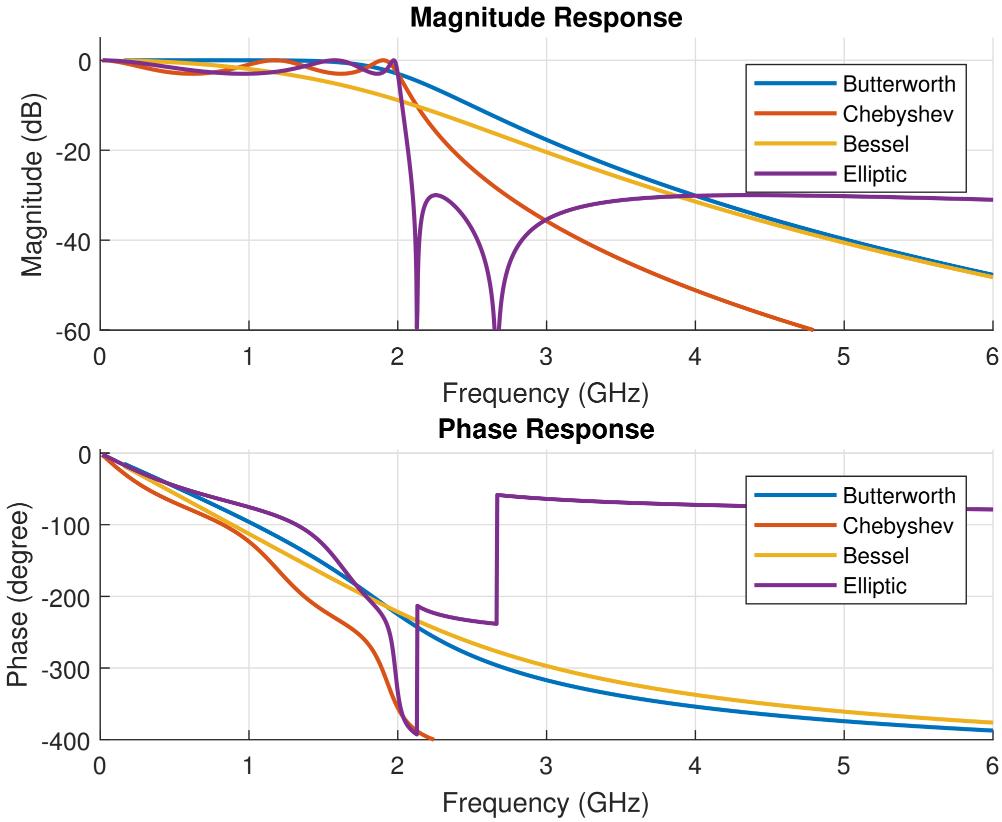


Figure 5.6: Frequency Response of Different Types of Filters. $f_{cut} = 2$ GHz, 5-th order, the passband ripple for the Chebyshev and the Elliptic filter are 3 dB, and the stopband ripple of the Elliptic filter is 30 dB.

the passband to achieve a faster roll-off, which is why they are commonly used for noise band removal. In the phase response, the Bessel filter has a maximally linear phase response, followed by the Butterworth filter, while the phase response of the Chebyshev and Elliptic filters are very nonlinear. Hence, the Bessel filter is the best option for applications where the shape of the filtered signal is crucial. The shape distortion of a signal needs to be considered when Chebyshev and Elliptic filters are used. The application of those filters to different types of noise will be discussed in the next section.

5.3 Effects of Analog Filter Characteristics on Signal Behaviors

In the section, the effects of different filter characteristics on the rise time, time delay, and magnitude attenuation of input signals are studied. The studied filters include a Butterworth filter, a Chebyshev filter, and an Elliptic filter. Since the Bessel filter is a pure analog filter which cannot be simulated by MATLAB, its application is not included in this work. The experimented filter characteristics are given in Table 5.1.

In the simulations, return pulses reflected from a target at 20 m (0.133 μs) were used as the signals feed into the filters. White noise was added to the return pulses,

Table 5.1: Characteristics of Analog Filters

Type	Butterworth	Chebyshev	Elliptic
Cut-off Frequency	2.5 GHz		
Order	3, 5, 7		
Passband ripple	-	3 dB, 5 dB, 10 dB	
Stopband ripple	-		30 dB

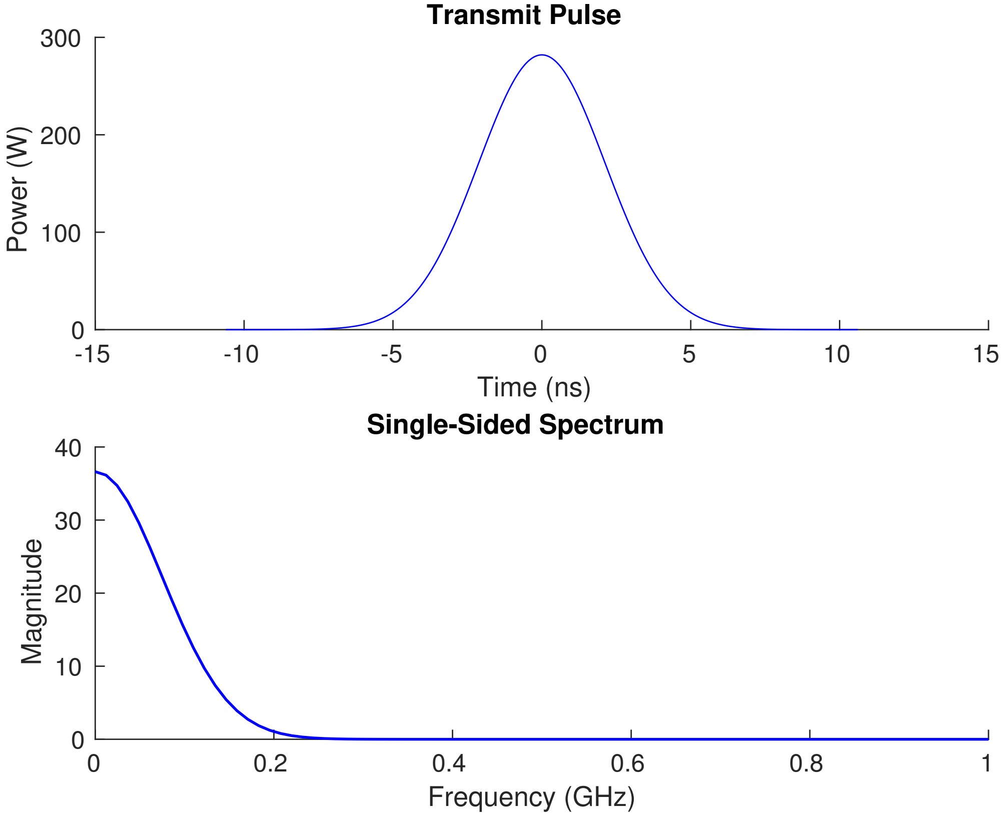


Figure 5.7: Transmit Pulse (Top) and the Single-Sided Power Spectrum (Bottom).

and the white noise was generated by the system noise model (Section 5.1.4). Here, we assume no other distortion or noises acted on the signal. A noise-free signal returned from the same distance without filtering was used as a reference. The transmit signal and its single-sided power spectrum is given in Figure 5.7 and the return signal contaminated by white noise and its power spectrum is shown in Figure 5.8. Figure 5.8 shows that the return signal has a noisy spectrum at high frequencies which causes the magnitude fluctuation in the time domain. By comparing with the noise-free signal, the following metrics are used for the evaluation of the influence of the filters on the signals:

- Rise-time difference: $\Delta t_r = t_{r,meas} - t_{r,ref}$, and $t_{r,ref} = 3.55\ ns$;

- Time shift: $\Delta t_{peak} = t_{peak,meas} - t_{peak,ref}$, and $t_{peak,ref} = 0.133\ \mu s$;

- Amplitude attenuation: $\Delta P = \frac{P_{means} - P_{ref}}{P_{ref}}$, and $P_{ref} = 20.17\ \mu W$.

Here, the subscript $meas$ represents the measured values of filtered signals, and the subscript ref represents the values of reference signals without filtering. t_{ref} is obtained from the laser source model (Section 3.2) given the pulse width. $t_{peak,ref}$ is the TOF of the reference signal corresponding to the target distance. P_{ref} is the power attenuation of the reference signal calculated in the propagation model (Section 4.2.1).

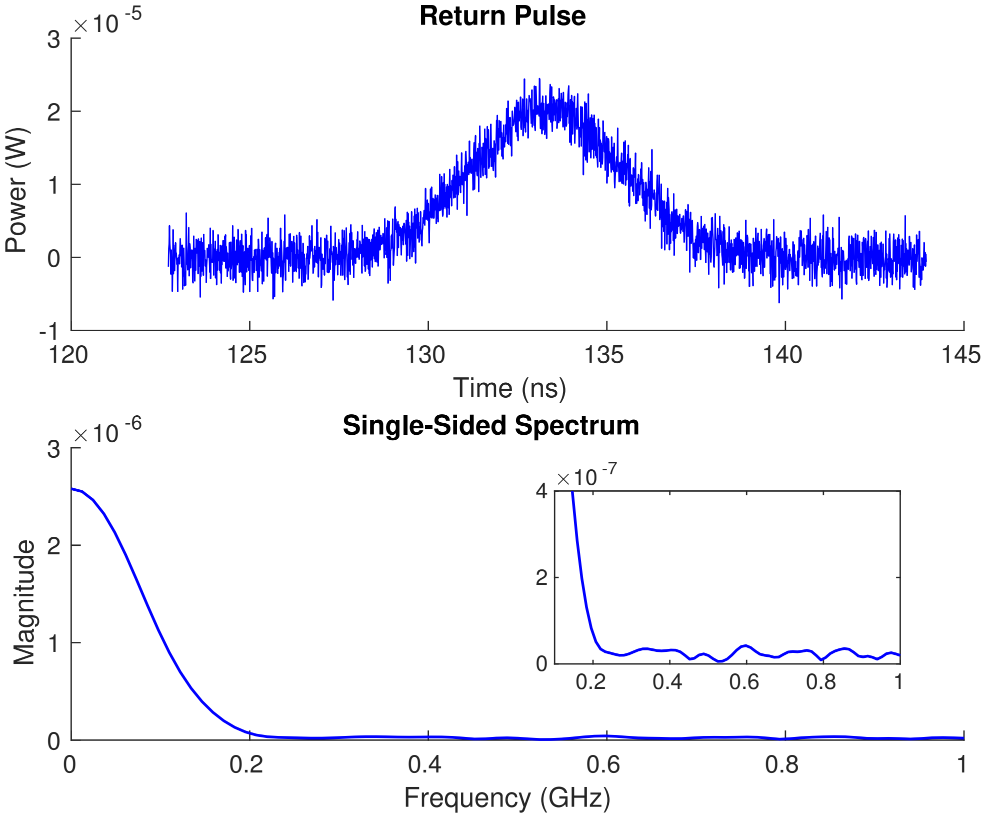


Figure 5.8: Return Pulse (Top) and the Single-Sided Power Spectrum (Bottom). The lower portion of the power spectrum is zoomed in and is shown in the sub-figure on the right.

5.3.1 Effect of Orders

In the study of the effect of filter order, the passband ripple of the Chebyshev and the Elliptic filters was set to 3 dB and the stopband of the Elliptic filter was set to 30 dB. The effects of the order on the signal characteristics are shown in Figure 5.9. Figure 5.9 shows that the magnitude attenuation by the filters increases with orders for the Chebyshev and the Elliptic filter. The Chebyshev filter has the largest attenuation, followed by the Elliptic filter at the same order, while the Butterworth filter has the least effect. The reason is that the oscillation of the passband ripple of the Chebyshev and the Elliptic filter increases with the order, as shown in the

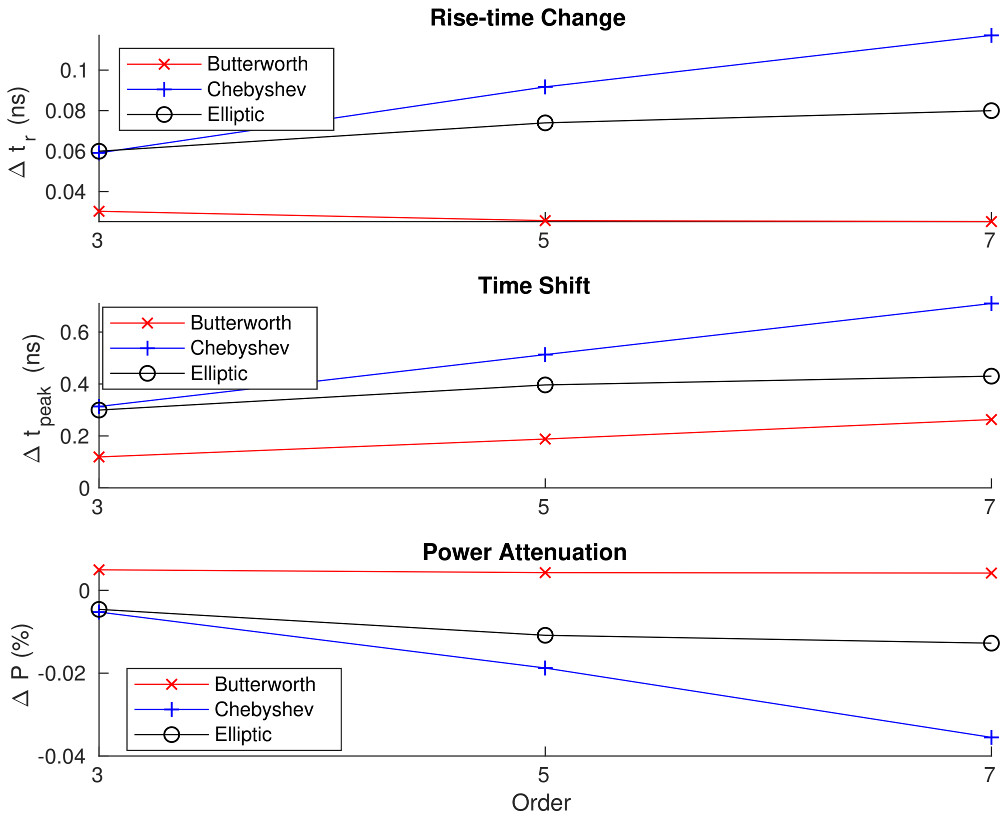


Figure 5.9: Effect of Order on Signal Characteristics.

magnitude response of the two types of filters (Figure 5.12b and Figure 5.12c). The increasing passband ripple results in a larger distortion of the signal magnitude. On the contrary, the Butterworth filter has a flat passband (Figure 5.12a), so the in-band frequency components are intact when the order changes.

The rise time and time shift of the signal also changes with the order. As shown in Figure 5.12a, Figure 5.12b and Figure 5.12c, the nonlinearity of the phase response increases with orders, and the Chebyshev filter has the worst linearity followed by the Elliptic filter. According to Equation (5.12), the non-constant slope of the phase response curve causes group delay distortion of the signal, which eventually alters the rise time. Since the time delay of the signal is equal to the slope of the response curve (group delay), a higher order also results in an increase of the time shift.

5.3.2 Effect of Passband Ripples

The effect of passband ripples on the signal characteristics is also studied. In the study, to control the number of variables, the order of all the filters was set to 3 and the stopband ripple of the Elliptic filter remains 30 dB. The signal characteristics after filtering with different passband ripples are presented in Figure 5.10. The results of the Butterworth filter are also provided for comparison. The frequency response of the Chebyshev and the Elliptic filter are shown in Figure 5.13a and Figure 5.13b. As discussed before, a high passband ripple causes a larger magnitude distortion, resulting in an increase of power attenuation. Furthermore, as the ripple value increases, the phase response becomes more nonlinear, which results in a larger change of rise time and the time shift.

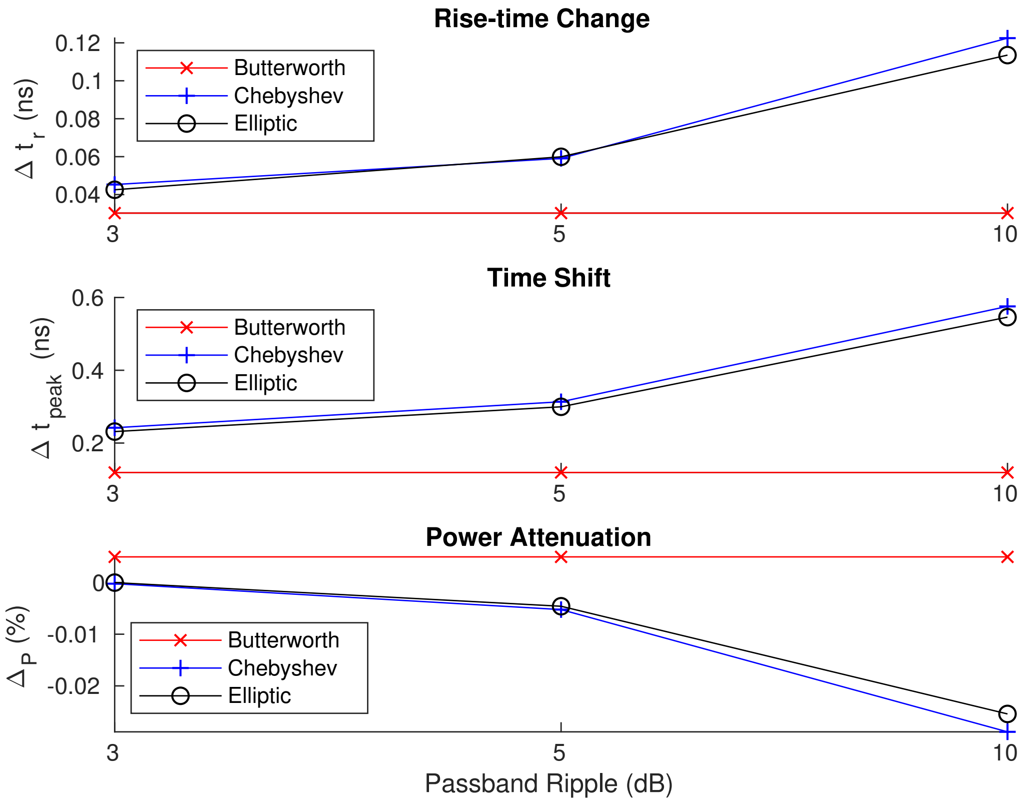


Figure 5.10: Effect of Passband Ripple on Signal Characteristics.

5.3.3 Zero-phase Filtering

Filters with nonlinear phase response could lead to an increase in the rise time and a time delay of the signal. Long rise time and time shift could bias distance measurement and increase measurement uncertainty. Therefore, a filter with a linear phase response is needed to reduce the impact, and the Bessel filter has a favorable characteristics. However, the trade-off is that the roll-off of the filter is slower than other filters like Chebyshev filters or Elliptic filters. One solution is to apply a filter to both START signal and STOP signal feed into the receiver. In this way, the time delay occurred at the two signals are canceled in the TOF calculation. Additionally,

Table 5.2: Results of Signal Characteristics after Phase-Shift Compensation

Filter Type	Butterworth		Chebyshev		Elliptic	
Zero-phase filtering	No	Yes	No	Yes	No	Yes
Rise-time difference (ns)	0.042	0.037	0.061	0.218	0.061	0.196
Time shift (ns)	0.632	-0.015	0.750	-0.030	0.726	-0.030
Power attenuation ratio (%)	0.010	-0.104	-2.045	-4.022	-1.984	-3.907

an all-pass filter with a customized phase response according to the time delay can also be added after the other filter to compensate the group delay.

In the signal processing, *zero-phase filtering* can compensate for the time shift caused by the filter and provide a zero-phase shift. Generally, a signal is only fed into the filter once, which is called 'forward filtering'. However, the zero-phase filtering has an additional step that feeds the signal backward after the forward filtering. Since the backward filtering causes a negative group delay, the positive and negative group delays are canceled, so a zero-phase signal is achieved. The zero-phase filtering can be performed by using any filters mentioned above. The effect of the zero-phase filtering on the signal characteristics is also studied in this work. In the study, the cutoff frequency of the filters was set to 1 GHz, the order of the filters set to 3, the passband ripple 3 dB and stopband ripple 30 dB. The filtered signals and the resultant characteristics are given in Figure 5.11 and Table 5.2. From the results, we can see that the time shift is theoretically eliminated. However, the power attenuation caused the Chebyshev and the Elliptic filter is doubled, and rise-time difference also becomes worse. The reason for those effects is that in fact, the zero-phase filter filters a signal twice, so the distortion of the signal is worse than the single forward filtering. Taking this experiment as an example, the signals were, in fact, filtered

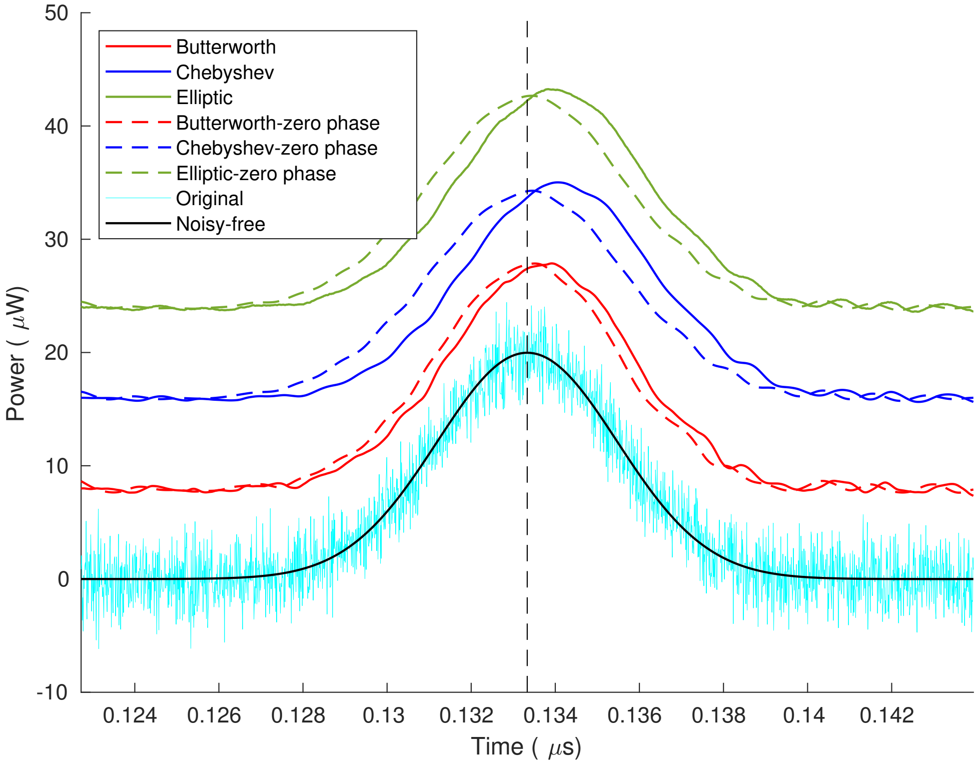
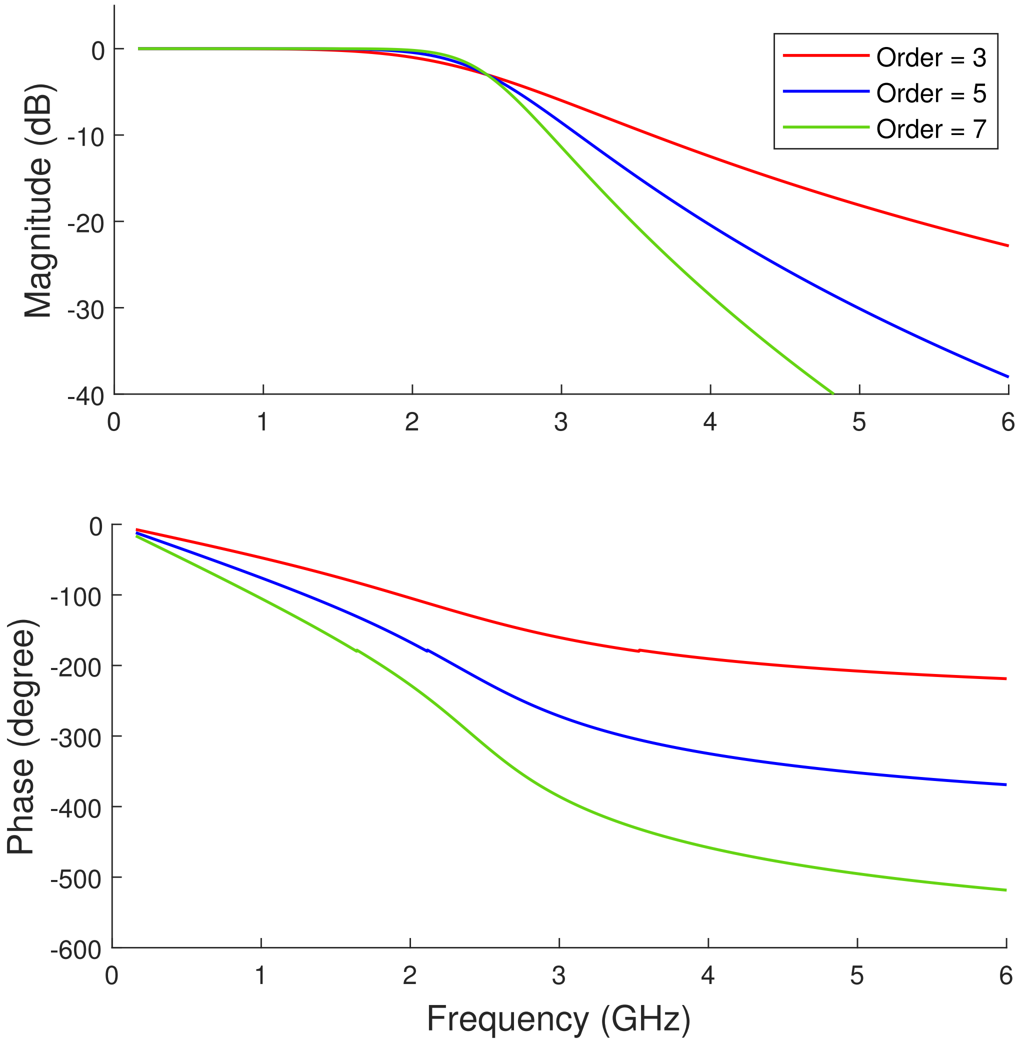
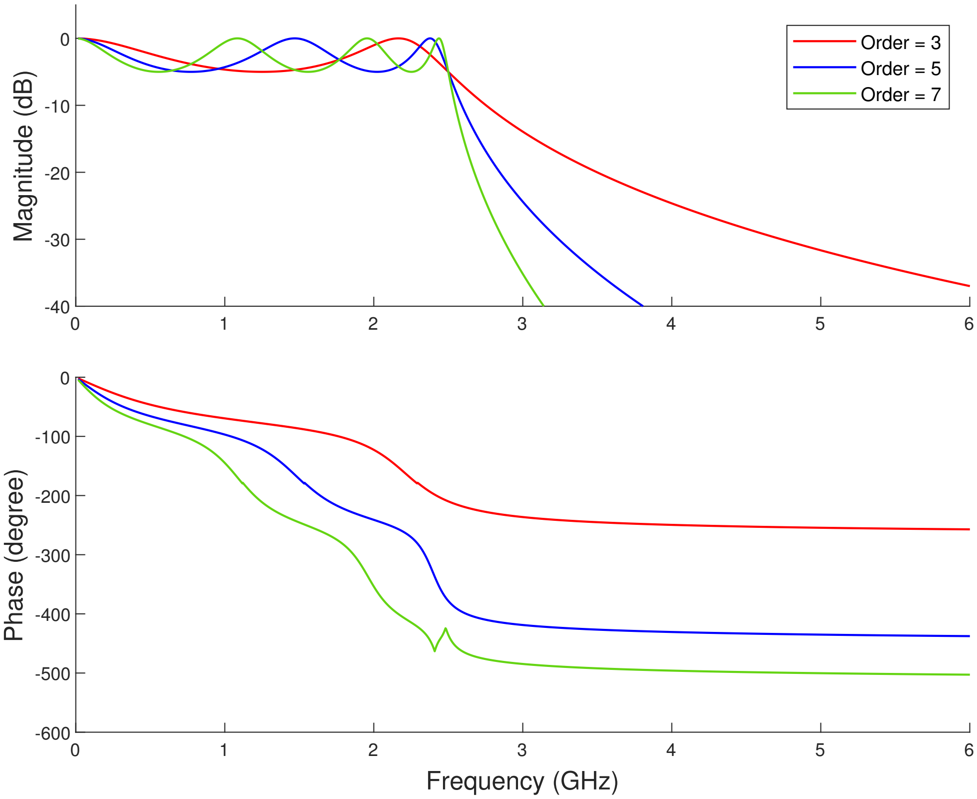


Figure 5.11: Signal before and after Zero-Phase Filtering. The dashed curves are signals after zero-phase filtering. The solid curves are signals after forward filtering. The black dashed line indicates the peak position of the reference signal (black curve).

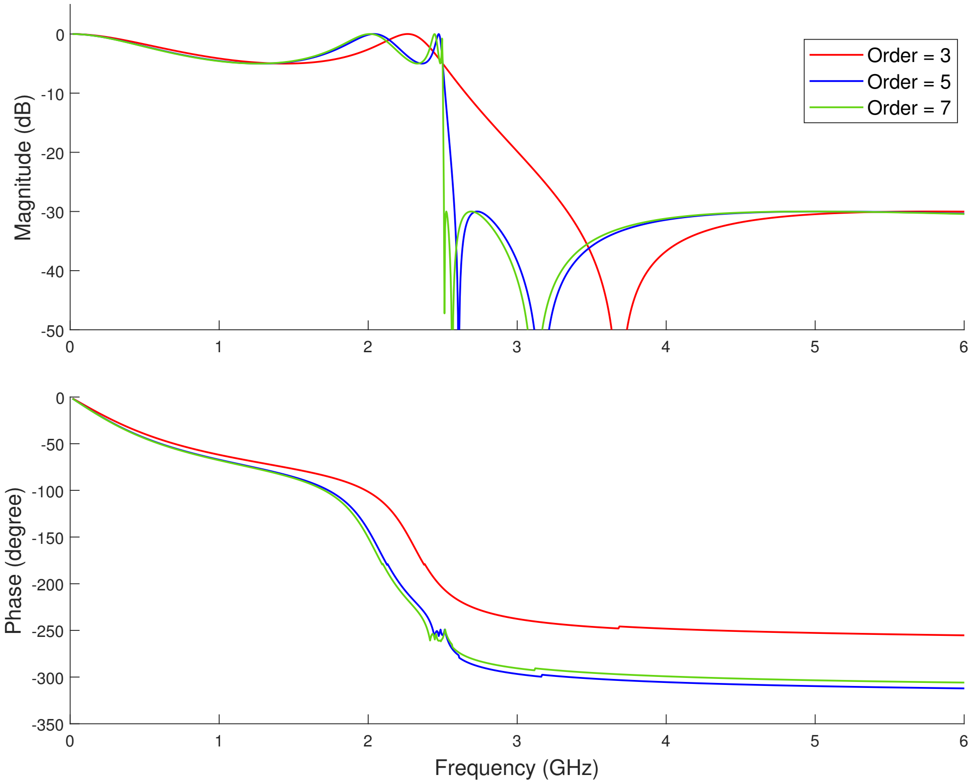
by filters with a passband ripple of 6 dB. In contrast, the Butterworth filter has the least distortion, because of its flat passband. A drawback of the zero-phase filtering is that the entire signal should be available for the zero-phase filtering, which means the zero-phase filtering cannot be realized in real-time by an analog circuit but can only be performed offline. Therefore, even though the zero-phase is powerful of removing the time-shift of the signal, it is only limited to offline post-processing signals that are stored beforehand.



(a) Magnitude Response (Top) and Phase Response (Bottom) of Butterworth Filters

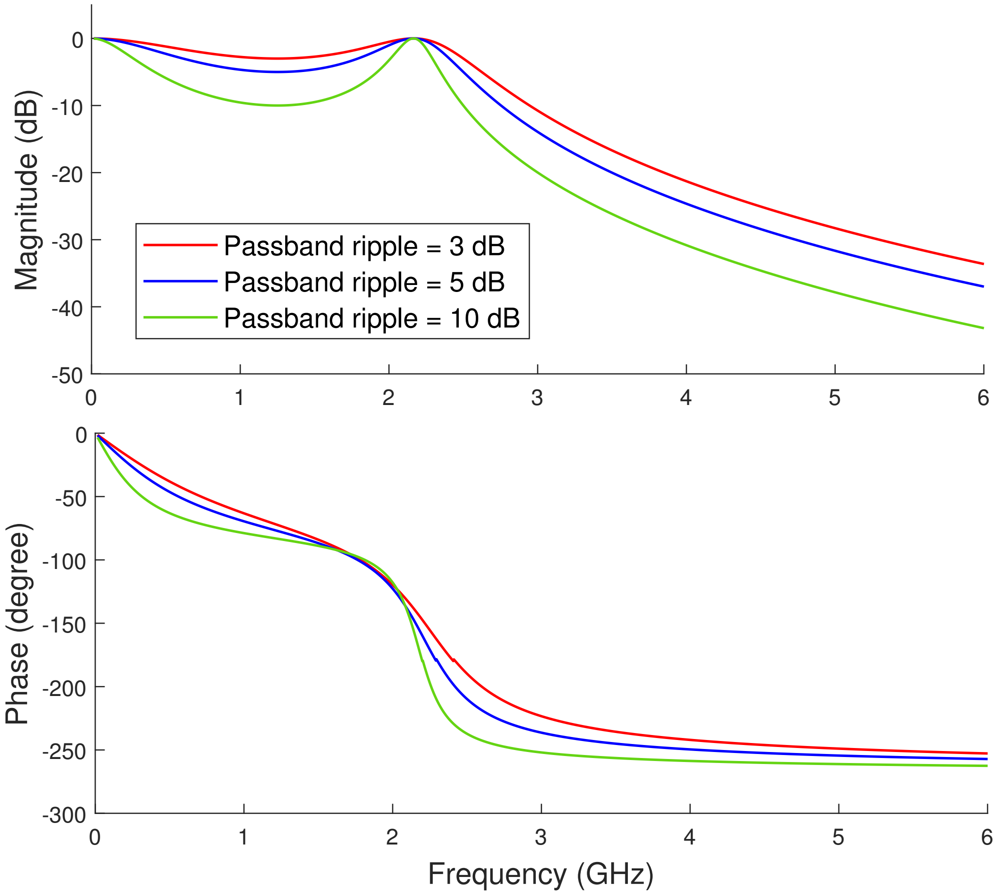


(b) Magnitude Response (Top) and Phase Response (Bottom) of Chebyshev Filters

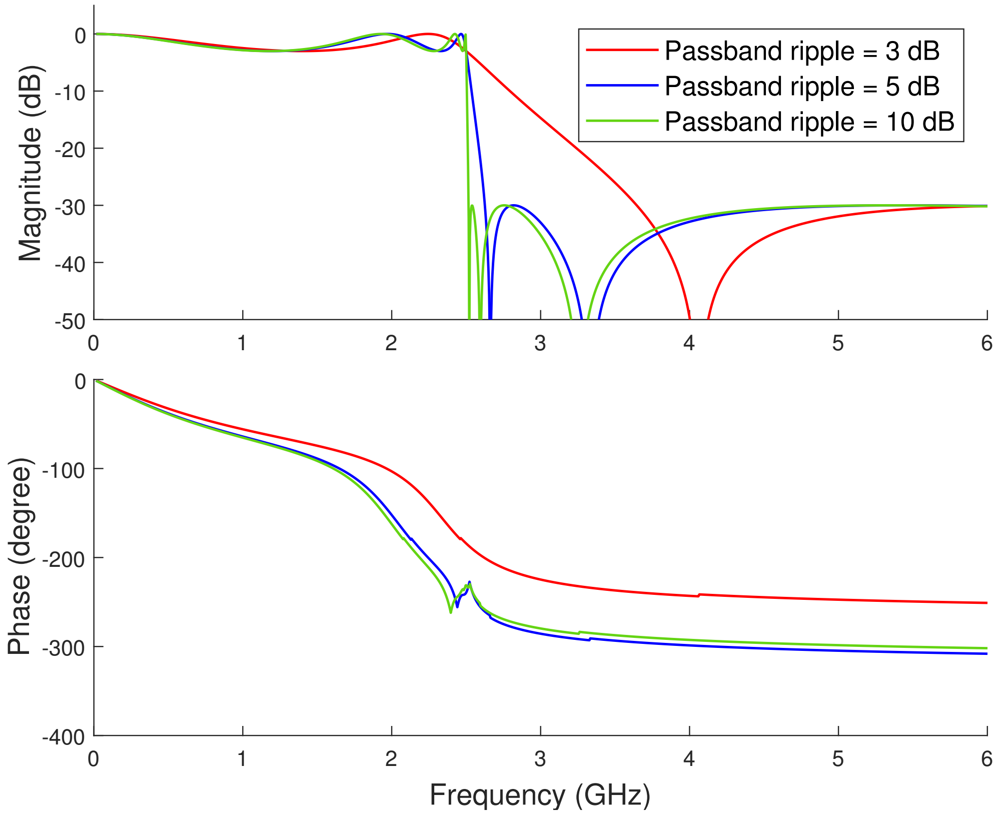


(c) Magnitude Response (Top) and Phase Response (Bottom) of Elliptic Filters

Figure 5.12: Frequency Response of Different Types of Filters with Various Orders.



(a) Magnitude Response (Top) and Phase Response (Bottom) of Chebyshev Filters



(b) Magnitude Response (Top) and Phase Response (Bottom) of Elliptic Filters

Figure 5.13: Frequency Response of Different Types of Filters with Various Passband Ripples.

5.4 Performance of Analog Filters on Different Types of Noise

As discussed in Section 5.1.5, noise can be classified as white noise, frequency-structured noise, and temporal-structured noise. In the previous section, different filters have been applied to white noise, and the Butterworth filter with an order of 3 demonstrates its best performance of smoothing the signal with least distortion compared to other filters at the same order. Also, we observe that the white noise cannot be entirely removed by a low-pass filter because the white noise spreads across all the spectrum. In this section, the performance of different filters on frequency-structured and temporal structured noise is studied, and the signal was collected by an ADC installed on a prototype lidar.

First, signals contaminated by frequency-structured noise with a frequency of 645 MHz (645MHz-Noise) are studied. The original signal and its power spectrum are shown in Figure 5.14. From the power spectrum, we observe the noise has strong power at 645 MHz. Three different types of filters were applied to the signal, and the orders of the three filters were all set to 3. The passband ripple of the Chebyshev and the Elliptic filter was set to 3 dB, and the stopband ripple of the Elliptic filter was set to 30 dB. For the purpose of illustration, the cut-off frequency was set at 0.5 GHz, which is close to the noise spectrum. The signal filtered by the three filters and their corresponding power spectra are given in Figure 5.15a, Figure 5.15b and Figure 5.15c, respectively.

From the power spectra, we notice that the noise after filtering by an Elliptic filter has the smallest power at 645 MHz, followed by the Chebyshev filter, while the Butterworth filter has the least truncation of the noise. The difference can be explained by the roll-off of the magnitude response of the three filters. The Elliptic filter has the fastest roll-off which means it acts a large attenuation of the frequency

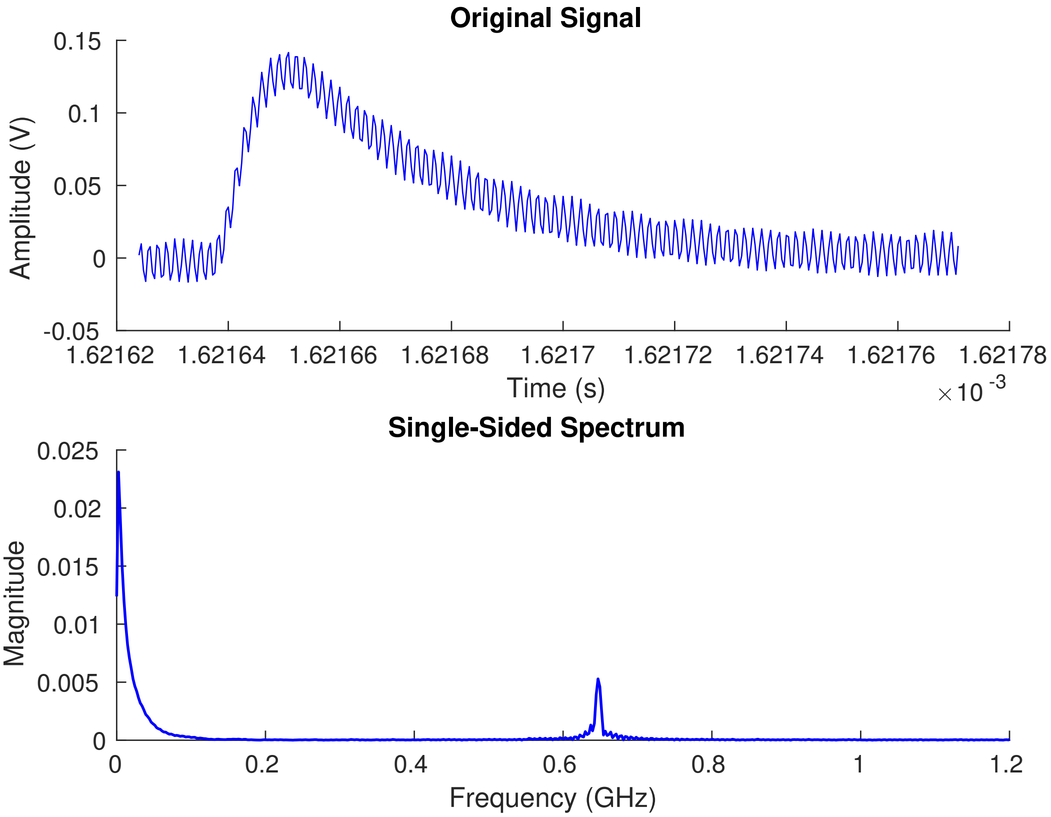
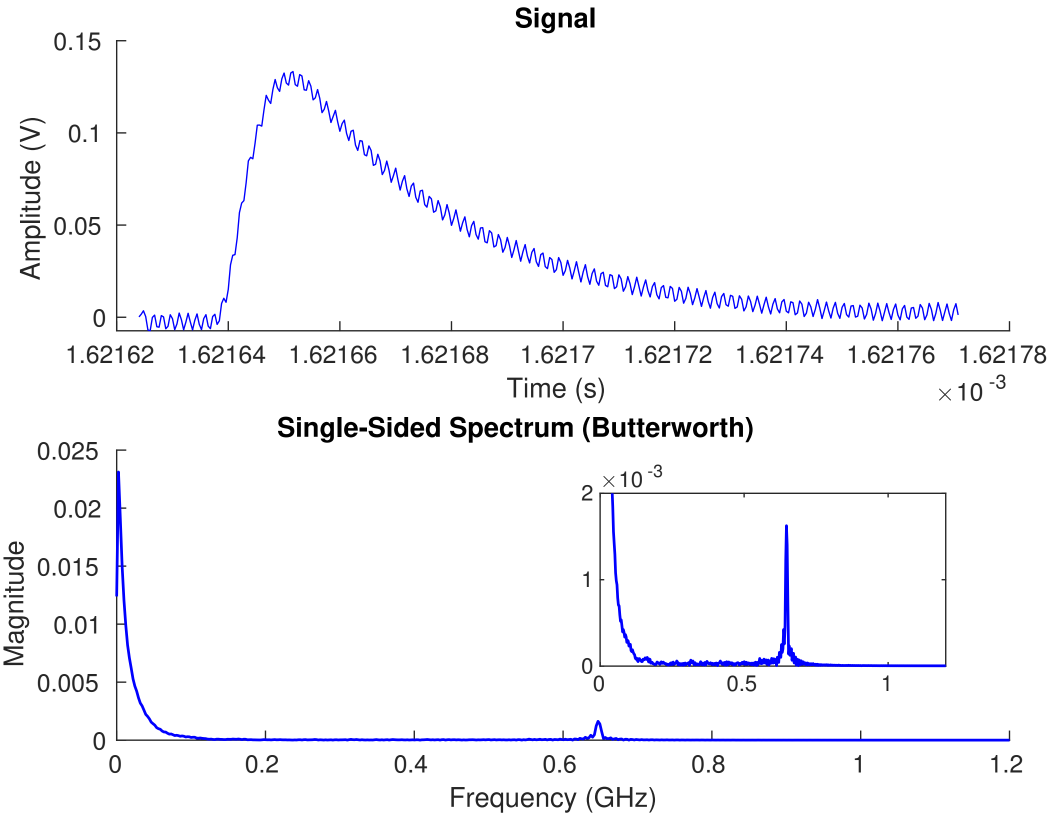


Figure 5.14: Original Signal Contaminated by Frequency-Structured Noise (625 MHz). The signal is from the prototype lidar.

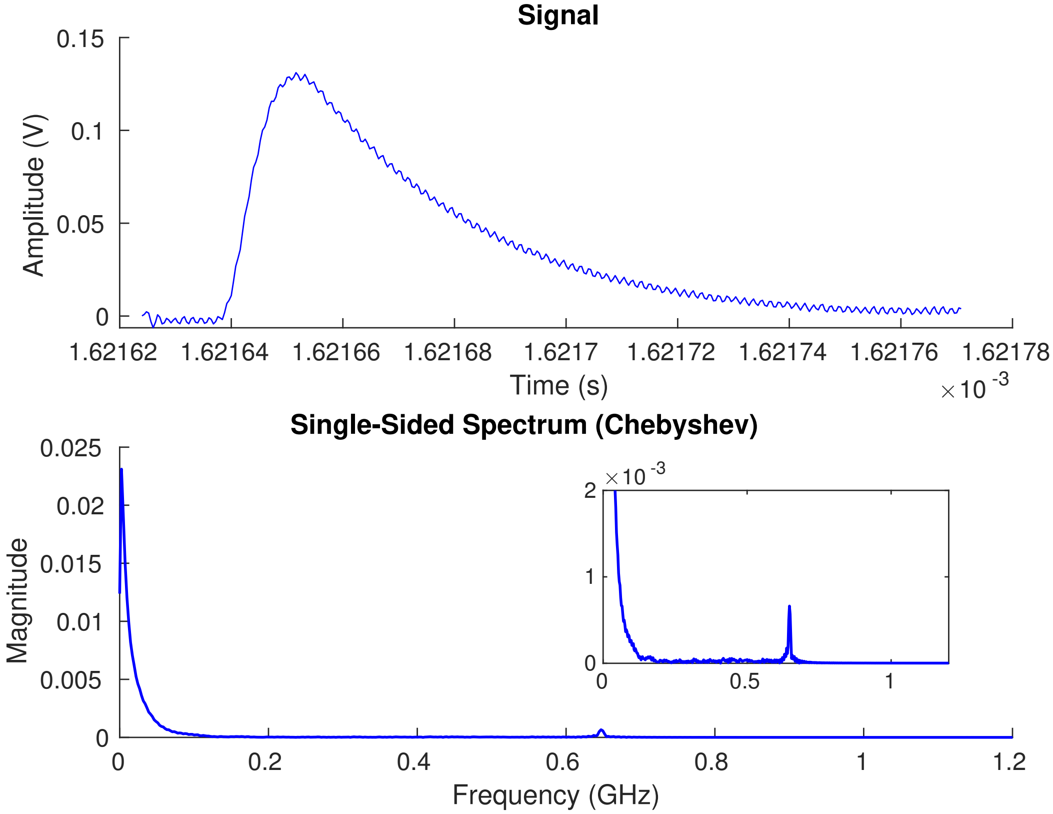
components in the transition band. Since the noise band is mostly inside the transition band of the filter, most of the noise is attenuated. On the contrary, the roll-off of the Butterworth filter is slower. Thus, the noise power inside the transition band is less attenuated, which results in a larger remaining noise. Therefore, a fast roll-off filter is necessary to remove noise in the transition band, especially in the case where the noise band of the signal spectrum is very close to the noise band. On the other hand, if the spectrum of a signal is far from the noise band, one can also set the cut-off frequency away from the noise band (*e.g.* 0.2 GHz in this case) to ensure the noise band falls inside the stopband of the filter rather than the transition band.

In this way, the noise band can be sufficiently truncated. The signal filtered by a Butterworth filter with $f_{cut} = 0.2$ GHz is given in Figure 5.16, from which we can see the noise is mostly removed by the Butterworth filter.

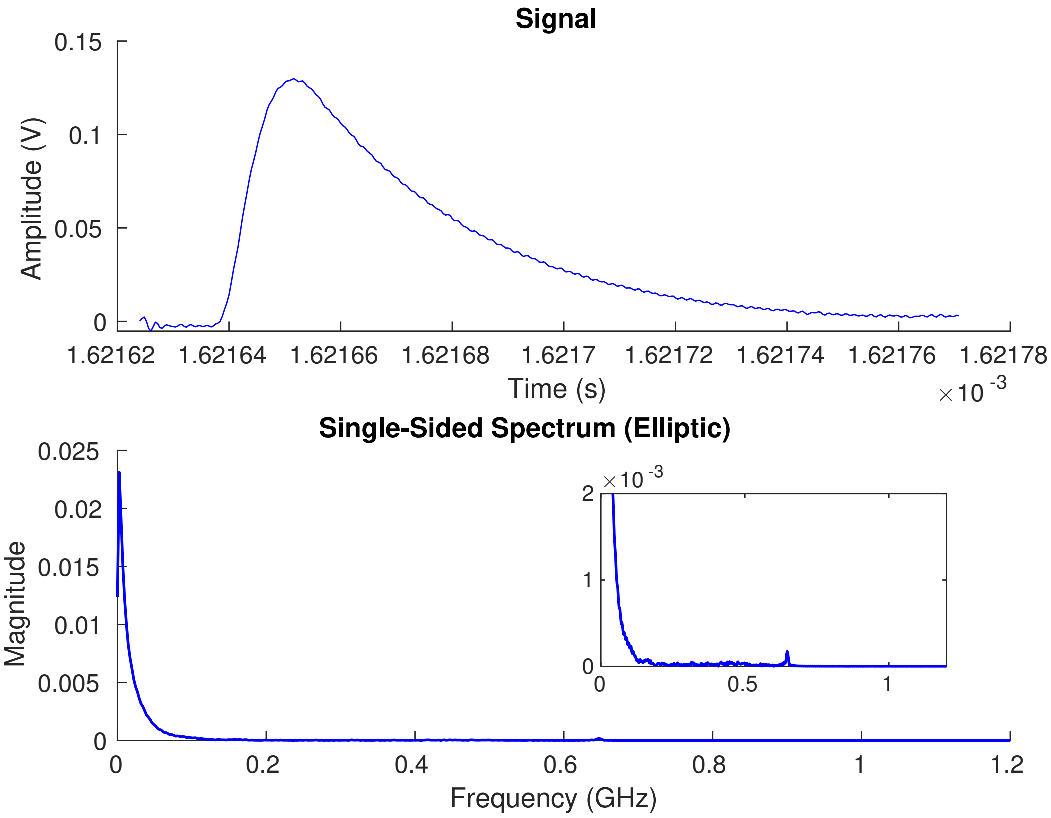
From the above discussion, we can see the low-pass filters efficiently remove out-of-band frequency-structured noise. However, they cannot reduce in-band frequency-structured noise without altering the spectrum of the original signal. Also, since the temporal-structured noise is also in-band noise (Figure 5.4), it is also difficult to be removed by the low-pass filters.



(a) Butterworth Filter (3rd Order, $f_{cut} = 0.5$ GHz)



(b) Chebyshev Filter (3rd Order, Passband Ripple = 3 dB, f_{cut} = 0.5 GHz)



(c) Elliptic Filter (3rd Order, Passband Ripple= 3 dB, Stopband Ripple=30 dB, $f_{cut} = 0.5$ GHz)

Figure 5.15: Signal Filtered by Different Filters (top) and Their Power Spectra (bottom). The signals are from the prototype lidar.

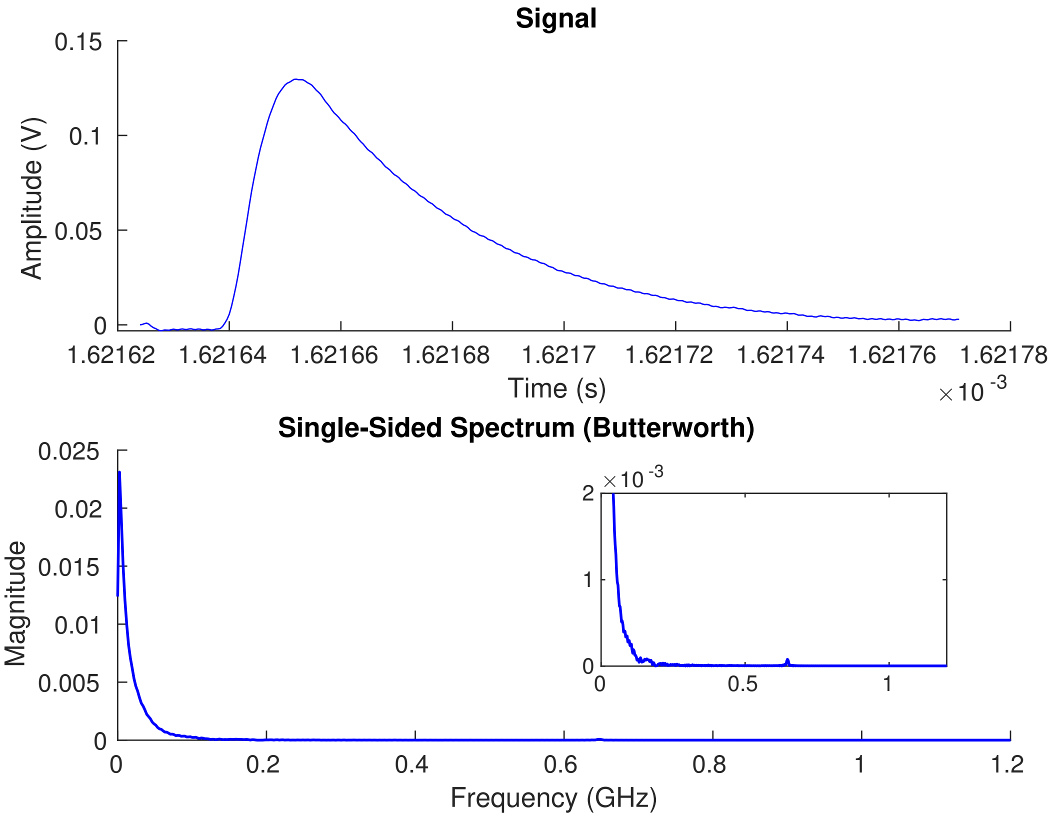


Figure 5.16: Signal Filtered by Butterworth Filter and Its Power Spectrum (3rd Order, f_{cut}=0.2 GHz). The signals are from the prototype lidar after filtering.

Chapter 6

TIME DISCRIMINATION: TIME-TO-DIGITAL CONVERTER

6.1 Time-to-Digital Converter

After the photon-detector converts the incident optical signals to analog electrical signals, a time discriminator measures the transmit time (start time) and arrival time (stop time) of the signals and calculates the TOF and corresponding distance. The detection algorithm attempts to determine whether signals are pulses or noises. The transmit time and the arrival time are usually measured by the same technique. This section will focus only on the time discrimination techniques for the arrival time.

Two commonly utilized timing devices are Time-to-Digital Converters (TDC's) and Analog-to-Digital Converters (ADC's). This chapter focuses on TDC's and TDC-based timing algorithms, and the ADC-based devices will be introduced in Chapter 7, Chapter 8 and Chapter 9. A TDC is presented schematically in Figure 6.1. Two key components of a TDC are a comparator and a timing circuit. The comparator generates logic signals by comparing analog signals with a predefined threshold. The timing circuit measures the timestamp when a logic signal changes its value from 'low' to 'high.' and it consists of a coarse counter and a finer counter. Taking the measurement of the arrival time for example, the coarse counter measures how many clocks have been passed since the start trigger event. The finer counter (usually made of ring oscillators) tracks the number of sub-divisions within a clock cycle from the last preceding clock recorded by the coarse counter. Using the recordings of the coarse and finer counter, the TDC can achieve a picosecond resolution. After measuring the start and stop timestamp, the timing circuit subtracts the start time from the stop

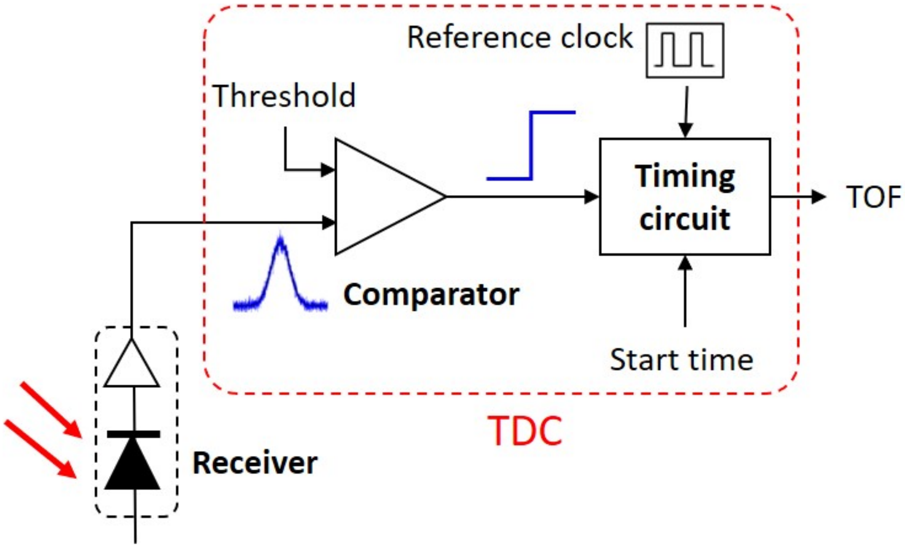


Figure 6.1: Schematic of a TDC.

time to calculate the TOF. Various time discrimination methods are available for timing using TDC's, and they generally fall into two main categories: leading-edge discrimination and constant fraction discrimination (CFD).

6.1.1 Leading-Edge Time Discrimination

In leading-edge time discrimination algorithms, a timestamp is measured when the leading edge of a signal crosses the threshold of the comparator. Since the measurement is carried out at the leading edge of a signal, the time detection method will still be functional even if the signal saturates a photon-detector. In other words, the leading-edge detection is not limited by the dynamic range of a photon-detector.

However, the major disadvantage of the leading-edge discrimination is walk error which is due to variation in the signal amplitude. Walk error is illustrated in Figure 6.2, where two signals are reflected by targets at the same distance, but the signals have different return powers. Keeping the comparator threshold and the rise-time of the two signals the same, the signal with a smaller amplitude triggers the threshold

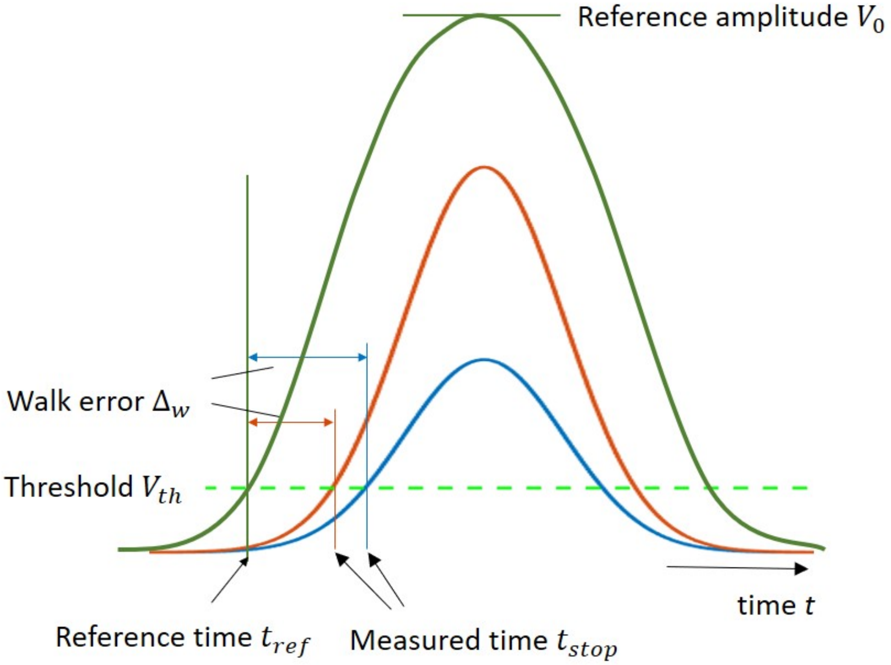


Figure 6.2: Illustration of Walk Error.

later than the larger one, which leads to a difference of arrival time between the signals. Such a time difference due to the amplitude change is called *walk* error and it is a major contributor to timing inaccuracy of leading-edge timing algorithms.

Gain-control compensation

Many methods have been developed to compensate for the walk error. One of them is to reduce the amplitude variation utilizing gain control. A variable-gain amplifier followed by an amplitude-insensitive time discriminator was proposed by Ruotsalainen *et al.* (2001). In the gain amplifier, the amplitude of a signal is first measured with a peak detector. The gain control circuit is adjusted to set the signal amplitude to a constant value. Ideally, the gain-control technique succeeds in presenting a constant amplitude signal to the comparator. In practice, the limited bandwidth of the amplifiers in the gain control circuit restrains the operational dynamic range to

$1 : 650$ and the delay variation of the amplifiers limits the timing precision to ±25 ps (Ruotsalainen *et al.*, 2001). The limited dynamic range makes the gain-control less practical for the autonomous vehicle application, since the dramatically varying atmospheric attenuation and target reflectivity can lead to a dynamic range of the signal amplitude of more than $1 : 100,\ 000$ (Ruotsalainen *et al.*, 2001).

Time-over-Threshold (TOT) compensation

A time-over-threshold (TOT) compensation was developed to allow a high operational dynamic range for input signals (Kurtti and Kostamovaara, 2009, 2011; Nissinen *et al.*, 2009; Williams, 2018). The schematic of the TOT compensation is shown in Figure 6.3. In the TOT compensation, both the trigger-crossing time of the leading edge and the falling edge of a signal are measured, and the time difference between the time stamps of the leading and the falling edges is referred to as time-over-threshold (TOT) (the TOT is the same as the 'pulse length' in Kurtti and Kostamovaara (2011)). As seen in Figure 6.3, the TOT increases monotonically with the pulse amplitude and the walk error Δ_w decreases with the amplitude. Thus, if the relationship among the amplitude, walk error and the TOT can be found, the walk error can be determined and compensated given a TOT.

In practice, a look-up table or a compensation curve containing the relationship of the walk error and the TOT can be made by calibrating a TDC (Kurtti and Kostamovaara, 2009). A compensation circuit was also proposed to realize the TOT compensation (Kurtti and Kostamovaara, 2009, 2011). Alternatively, in this dissertation, we provide an analytic description of the relationship among the TOT, pulse amplitude, and walk error.

$$\begin{aligned} V_r &= \frac{V_{th}}{exp(\frac{(TOT/2)^2}{2\sigma^2})} \\ \Delta_w &= |t_{ref}| - \sqrt{t_{ref}^2 - 2\sigma^2 \ln(\frac{V_{ref}}{V_r})} \end{aligned} \tag{6.1}$$

where V_{th} is a TDC threshold, and V_r is the signal amplitude. Variable σ is the standard deviation of the Gaussian distribution in the transmit pulse model whose value can be obtained given a pulse width (Equation (3.8)). Variable t_{ref} and V_{ref} represent convenient reference time and reference amplitude for the calculation of walk error Δ_w, respectively. Equation (6.1) is called the TOT compensation model (or TOT model) and its derivation is given in Appendix C.1. In the TOT model, the value of

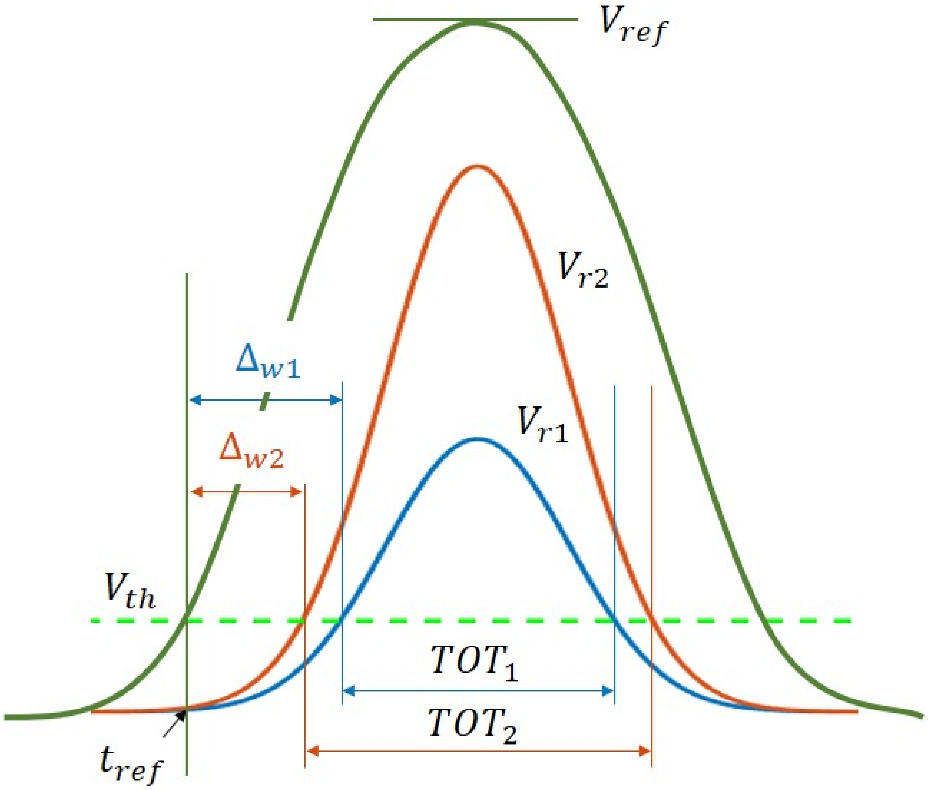


Figure 6.3: Schematic of TOT Compensation. The subscript 1 and 2 of V_r, TOT and Δ_w indicate the blue signal and the red signal, respectively.

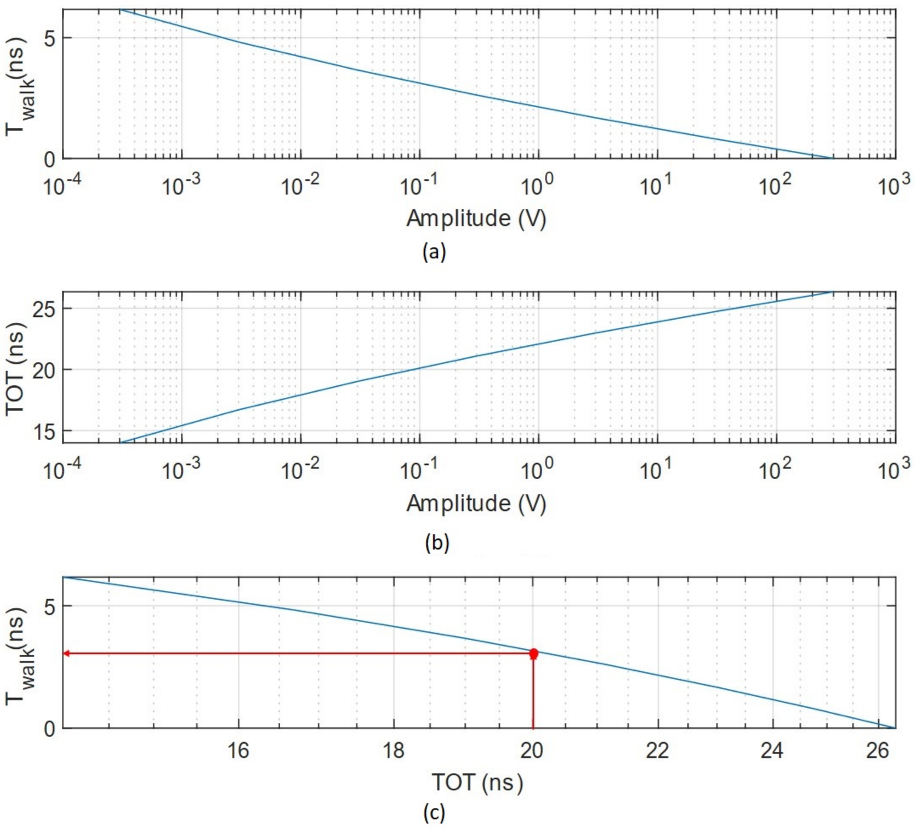


Figure 6.4: TOT Compensation Curve.

TOT should be measured from experiments, while V_r and V_{ref} are pre-defined variables and the rest variable can be obtained from pulse information. The relationships between the signal amplitude and TOT, amplitude and walk, and TOT and amplitude, are plotted in Figure 6.4. After a TOT is measured, using the TOT model or the TOT compensation curve, the walk error can be calculated analytically or found by using the conpensation curve. Then, the walk error on a time measurement can be compensated by subtracting T_{walk} from the original time measurement. The process of using the TOT compensation curve to find the walk error is shown in Figure 6.4c. The TOT method can achieve a time measurement precision of picosecond level and a larger dynamic range of a signal amplitude of $1 : 30,\ 000$ (Kurtti and Kostamovaara, 2009). However, we should notice that the TOT method is highly sensitive to the symmetricity of a signal. In other words, if the falling edge of a signal is heavily distorted due to, for example, multiple returns or slow response time of the system, the relation between the TOT and the walk error is impaired. In this case, the TOT compensation will not give accurate results.

Slew-rate compensation

An alternative way to compensate for walk error is to use the relation of the walk error and the slew rate of a pulse. This approach is called *slew-rate compensation.* The slew rate is the slope of a signal at the point of interest. For the slew-rate compensation, the TDC should be equipped with two comparators with two predefined thresholds. A schematic of the slew-rate compensation is shown in Figure 6.5. The lower threshold (V_{th1}) is set at a constant value as a normal TDC does, and the higher threshold (V_{th2}) is set C (C is an integer constant) times larger than V_{th1}, *i.e.* $V_{th2} = CV_{th1}$. The two

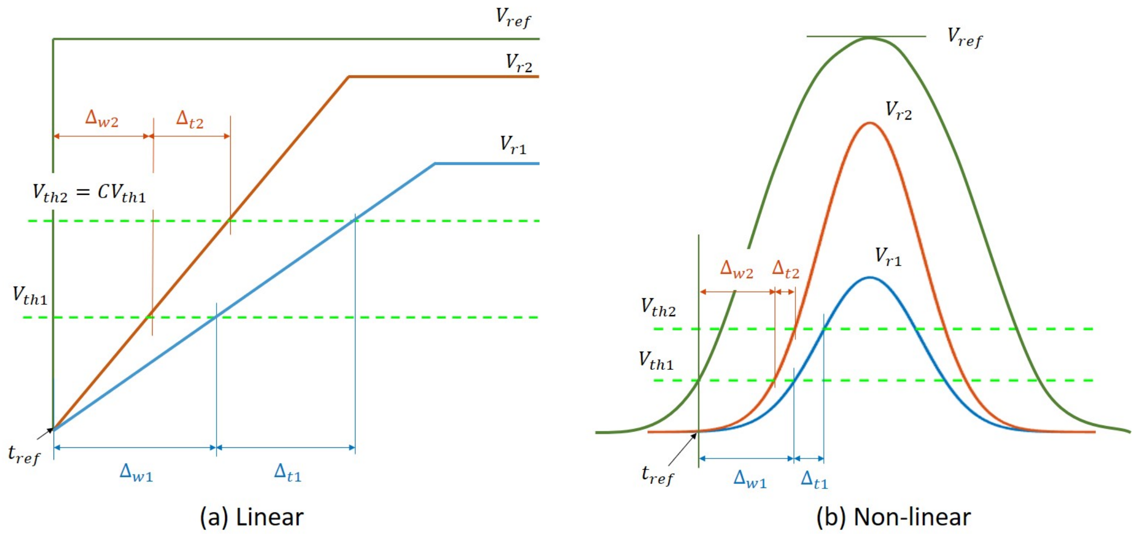


Figure 6.5: Slew-Rate Compensation: (a) Linear and (b) Nonlinear. The subscript 1 and 2 of Δ_w and Δ_t indicate the blue signal and the red signal, respectively.

time stamps, t_1 and t_2, are measured from the comparator for V_{th1} and V_{th2}, and the time difference Δ_t between t_1 and t_2 is calculated by $\Delta_t = t_2 - t_1$. The walk error Δ_w is the time difference between t_1 and a convenient time reference t_{ref}. Figure 6.5a shows an example of signals with linear slew-rates between the high and low threshold. In the example, since the slew rate can be calculated by $(V_{th2} - V_{th1})/\Delta_t$, the walk error can be related to Δ_t by $\Delta_w/\Delta_t = 1/(C-1)$ following the geometry of similar triangles. Thus, the walk error is a linear function of Δ_t (Nissinen *et al.*, 2009):

$$\Delta_w = \frac{\Delta_t}{C-1}. \tag{6.2}$$

A Gaussian signal usually has a nonlinear slew rate at the tails, which results in a nonlinear dependency of the walk error on Δ_t. Thus, in practice, the relation can be obtained by look-up tables or compensation curves by calibrating a TDC. In this work, we mathematically derived the nonlinear relation between the walk error and the time difference as

$$\begin{aligned} \Delta_t &= STOP_2 - STOP_1 \\ \Delta_w &= \frac{-2\sigma^2 \ln(C) - \Delta_t}{2\Delta_t} - t_{ref} \end{aligned} \tag{6.3}$$

where $STOP_1$ and $STOP_2$ are the time stamps measured by the high and low comparator, respectively. Equation (6.3) is called slew-rate compensation model (or slew-rate model), and the derivation is detailed in Appendix C.2. The relation between the walk error and the Δ_t is plotted in Figure 6.6. Using the slew-rate mode that takes the measured two time stamps, one can obtain the walk error, and subtracting the walk error from $STOP_1$, a walk-error-free TOF result can be achieved. A schematic of the compensation process is given in Figure 6.6b. The slew-rate compensation can also achieve picosecond-level precision with a large dynamic range (Nissinen *et al.*, 2009). In addition, its independence on the falling edge of a signal makes it more

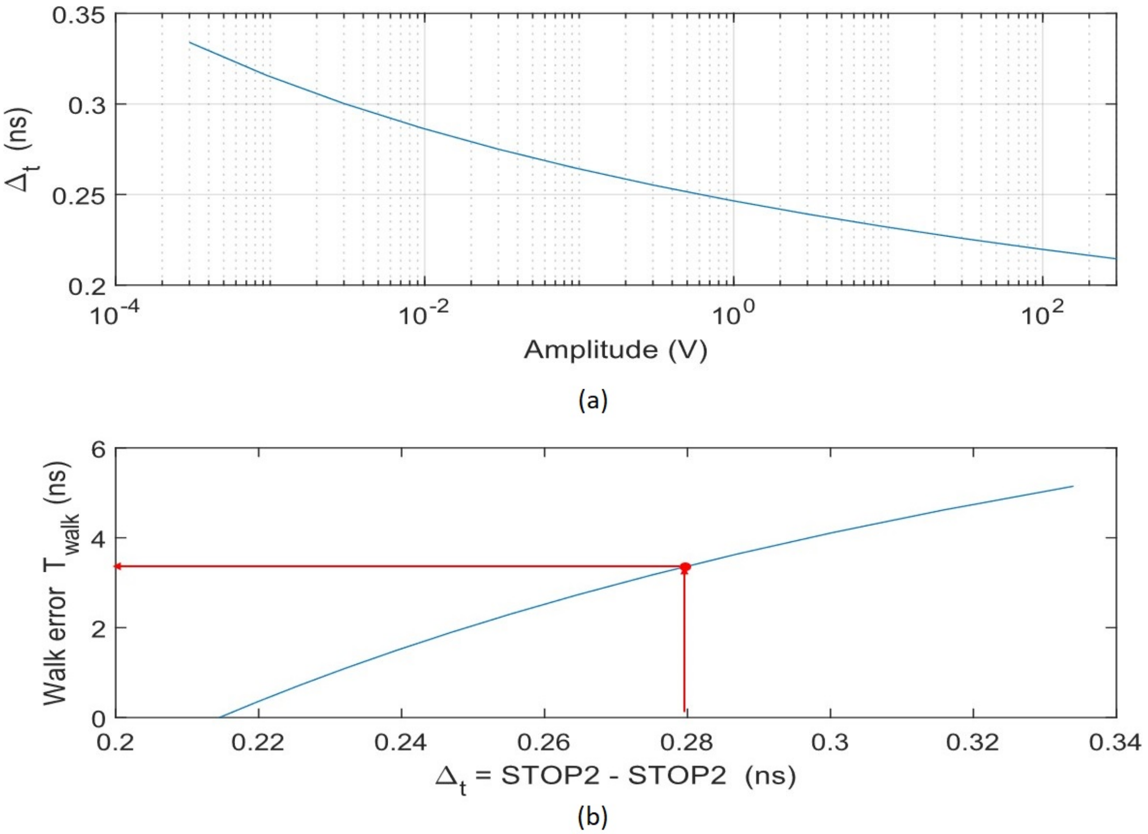


Figure 6.6: Slew-Rate Compensation Curve. (a) Time Difference-Amplitude Curve; (b) Slew-Rate Compensation Process. After measuring Δ_t, one can use the curve to find corresponding T_{walk}, and subtract the T_{walk} from $STOP_1$ to compensate the walk error.

applicable than the TOT compensation, but the additional comparator increases the complexity of the discriminator and has a limitation measuring signals weaker than the high threshold.

The TOT and the slew-rate compensation significantly reduce the walk error, but also introduce additional noise to timing results. One example is the quantization error due to the utilization of extrapolation on the compensation curves, and the number of points used in the extrapolation affects the significance of the quantization error. In addition, the TDC-based timing techniques also include their inherent noises, *e.g.* jitter noise. Jitter is the uncertainty in the time measurement when a signal crosses the comparator threshold. If the signal is noise-free, the time when the threshold is exceeded is constant. However, amplitude variation always exists due to various noises, in which case, the measured TOF contains uncertainty. This kind of noise is called jitter noise. An example of the jitter noise is shown in Figure 6.7. Mathematically, the jitter noise is defined as (Skolnik, 1962):

$$\sigma_{jitter}^2 = \frac{\sigma_{noise}^2}{(dV/dt)^2} = \frac{t_r^2}{SNR^2}. \tag{6.4}$$

From Equation (6.4), we can see jitter has an inverse relationship with the slew rate at the trigger position (dV/dt). In the above discrimination methods, thresholds are set at a constant a certain level above the noise floor. Since the amplitude of

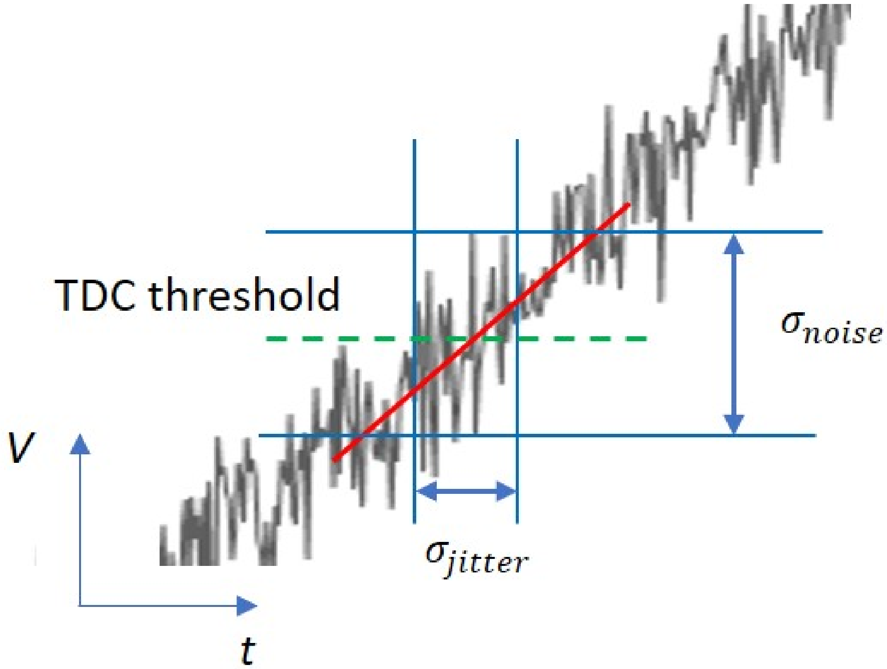


Figure 6.7: Schematic of Jitter Noise. The black curve is the signal with noise. (Figure is adapted from Williams (2017).)

incoming signals varies, the trigger point may not be at the optimal position that has the highest slew rate. Thus, additional noise is introduced. Ideally, the trigger position should be set at the linear portion ($30\% - 40\%$ of the amplitude of a pulse) of the leading or the falling edge of a pulse (Kilpelä *et al.*, 1998), where the signal has a sharp slew rate. In practice, keeping the threshold adjustable at a constant fraction of a pulse is a solution to minimize the jitter.

Time-variant threshold

A time-variant threshold method was proposed by Williams (2018) to optimize the trigger point for a leading-edge detection method, in which the threshold can be adaptively adjusted according to the time elapsed from the start time of a pulse. In this case, the threshold-crossing time can be kept around the optimal trigger position. The relation of the time-variant threshold and the time elapse is given by

$$V_{th}(t) = V_{th,lo} + (V_{th,hi} - V_{th,lo})e^{-\frac{t}{RC}}, \tag{6.5}$$

where $V_{th,hi}$ and $V_{th,lo}$ are predefined optimal thresholds for short and long-distance targets, respectively (Williams, 2018). One should note that the time t in Equation (6.5) is not the arrival time we are looking for, but the time value used to adjust the time-variant threshold. This time could be measured by a coarse counter in a TDC. After the time-variant threshold is adjusted, a more accurate time will be measured using the finer counter of the TDC and the above timing methods. The time constant RC is to approximate the $1/R^2$ power attenuation as a function of range, as shown in the Lidar Equation (Equation (4.2)), which is simplified to:

$$V_r = \frac{V_0 A}{R^2} e^{-2BR}, \tag{6.6}$$

where A and B are hyper-parameters, and V_0 and V_r are the peak voltage of the transmit and return signal, respectively. The usage of RC to model the power decay instead of modeling e^{-2BR} directly is because the time-variant threshold is achieved by an analog circuit. Comparing Equation (6.5) and Equation (6.6), we can see Equation (6.5) cannot exactly follow the change of power attenuation with range, which could lead to variation of the trigger position around the optimal position. The time constant RC also requires careful calibration to match the power attenuation with distance (time) in practice.

The time-variant threshold method provides an approach to obtain the optimal trigger position to reduce the time jitter, but it does not resolve the walk error issue. Therefore, the TOT compensation was combined with the time-variant threshold method (Williams, 2018). However, as aforementioned, the TOT compensation is limited to symmetrical signals.

6.1.2 Constant Fraction Discrimination

Traditional CFD

In addition to the time-variant threshold method, the Constant Fraction Discrimination(CFD) techniques, as illustrated in the name, can also trigger the time at a constant fraction of the pulse height. In the traditional CFD's, the input signal is copied into two identical signals, one of which is attenuated and inverted, and the other one is delayed. The schematic of the CFD principle is shown in Figure 6.8. The attenuation for the attenuated signal is selected as a fraction of the amplitude of the original amplitude. At that fraction the timing position is optimal, and the value usually ranges from 20% to 40%. Then, the time delay is tuned to make the fraction point on the leading edge of the delayed signal align with the peak of the attenuated signal. Subsequently, the two delayed and attenuated signals are added to generate a bipolar signal, which is then passed to a zero-crossing comparator to generate the

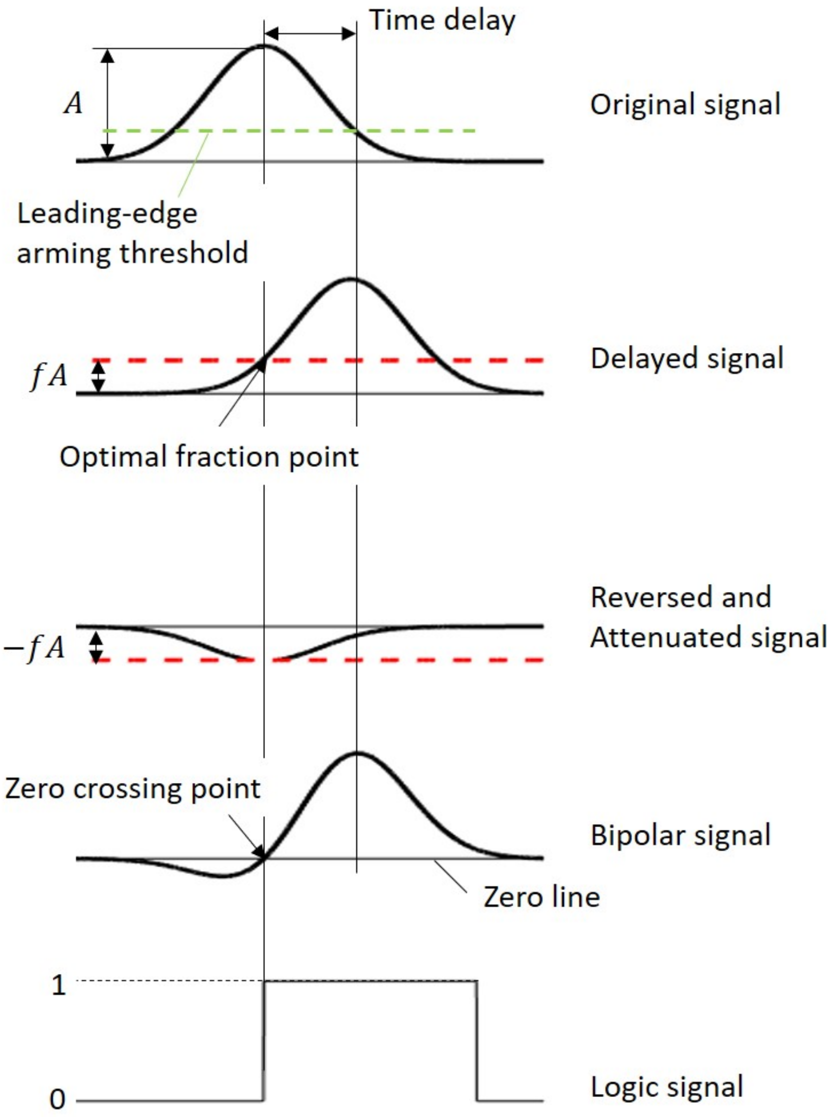


Figure 6.8: Schematic of the Principle of a Traditional CFD.

logic signal for time discrimination. The zero-crossing point corresponds to the time stamp at the optimal fraction point on the delayed signal, which is also the time at the peak of the original signal. In addition, a leading-edge arming discriminator triggered above the noise floor of the signal is applied to prevent the zero-crossing comparator from being mistakenly triggered on noise floor preceding the zero-crossing point.

The major advantage of the CFD is that it theoretically eliminates the walk error due to the independence of the zero-crossing point on signal amplitude (Figure 6.9). However, in practice, a finite amount of charge is required to move the comparator output from "0" to "1", which results in additional walk error on the timing results (Nakhostin, 2017). Additionally, if an incoming signal has strong amplitude beyond the dynamic range of a photon-detector that saturates the detector, or multiple returns come back in a minuscule time difference, the top of the pulse is flattened. The flattop could result in large walk error on TOF measurements, which is why the traditional CFD is normally used in a limited range of 1:100 or less for the input signal (Kurtti and Kostamovaara, 2011).

In addition to reducing walk errors, CFD's also leads to less time jitter. Even though in the summation process of the CFD, the noises of the attenuated and the delayed signals are added as well, in fact, the jitter is mitigated, because the statistical variation of the noise on the attenuated and the delayed signals cancel out with each other. Since only the leading edge and the peak of the signal are involved in the

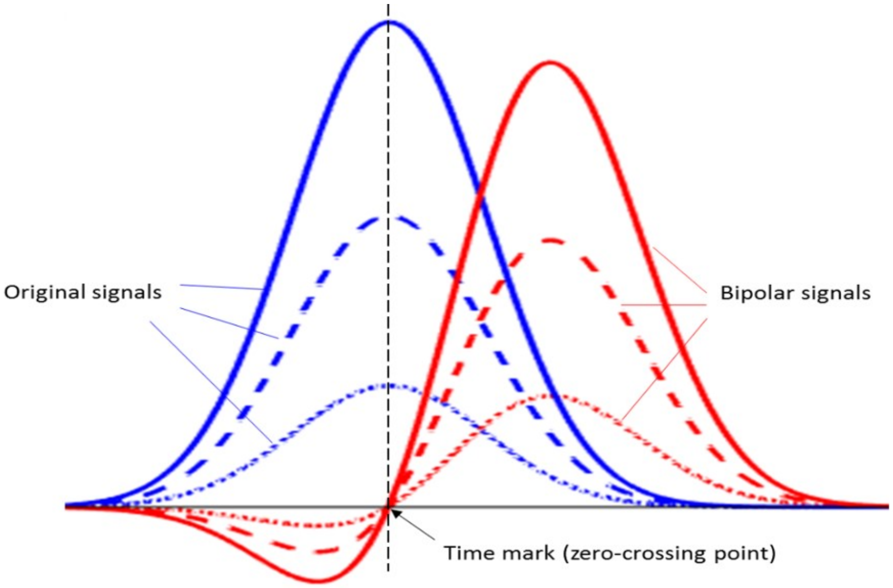


Figure 6.9: Independence of the Zero-Crossing Point on Signal Amplitude.

summation, the CFD is not affected by the pulse symmetricity.

Leading-falling edge CFD

In the traditional CFD, the delayed and attenuated signal is added to the top of the original signal, which could fail when the signal has a flat top (*e.g.* the photon-detector is saturated). In this case, a variation of the traditional CFD, leading-falling edge CFD was developed (Kilpela, 2004). The principle of a leading-falling edge CFD is similar to the traditional CFD, and it is shown in Figure 6.10. The leading-falling edge CFD copies the original signal into two identical signals, and one of the signals is delayed by a certain amount, such that the leading edge of the delayed pulse and the falling edge of the non-delayed pulse cross at an optimal constant fraction of the signal height. In this case, the two signals are intersected at a fraction point which has the largest slew rate rather than the peak. The leading-falling edge CFD can prevent the detection from suffering from flattops, but since the technique involves the falling edge of a signal, it is only practical for symmetric pulses.

6.1.3 Summary

From the comparison between the leading-edge and CFD detection, we can see that leading-edge detection utilizes a simpler circuit design than the CFD techniques which need delay and summation circuits, but the walk error is a major drawback for the leading-edge discrimination techniques. To reduce the walk error, additional compensation techniques that are operative in a large dynamic range are required to achieve high timing accuracy. On the other hand, the CFD techniques address the walk error and jitter to a large extent, but potential time delay could result from extra electrical components. The limited dynamic range of the CFD techniques could also restrain their application in the automotive industry.

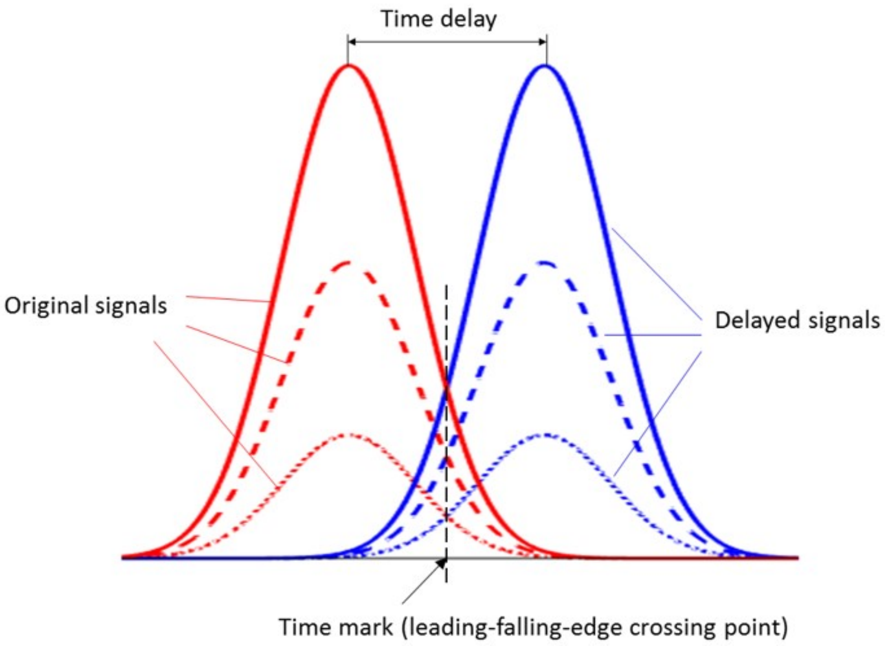


Figure 6.10: Principle of a Leading-Falling Edge CFD.

6.2 TDC Modeling

In this section, the TDC modeling will be introduced first followed by the application of different timing algorithms to simulated return signals from different target distances. The slew-rate compensation is not included due to its reduced applicability to weak signals. In addition, the time-variant threshold is not included either since the time constant RC of a photon-detector are not available in this study, which makes it difficult to calibrate the parameter A and B in Equation (6.5), and therefore, modeling the time-variant threshold is impractical.

6.2.1 Propagation Delay

The comparator is one of the key components of the TDC. In practice, a TDC circuit requires a small amount of time to respond and propagate signals for voltage comparison, so the comparators have a propagation delay, which limits the fastest frequency of a signal the TDC can process (Smat, 2012). However, on the other hand, even though the propagation delay restrains the frequency of the signal that can be processed, slower fluctuations still exist in the comparator, caused by the unstable transient response of the electrical device to high-frequency oscillation. To model the response of the TDC to the input analog signal, we have to simulate the transient signal, but the transient response of an electrical device to a rapid oscillating signal is usually ill-defined. Thus, it is very difficult to model the exact transient signals in a real device (Wahab *et al.*, 2016). In this study, inspired by Wahab *et al.* (2016), we approximated the transient signal by sub-sampling the original 'continuous' signal with a sampling rate of 2 GHz and smoothing the sub-sampled signal by a spline-line fitting, assuming no additional high-frequency noises were added to the transient signals before comparison. If a signal was sampled at a frequency lower than 2 GHz, no sub-sampling was performed before the spline-line fitting because the bandwidth of the signal is already smaller than the one of the comparator. The sub-sampling rate (2 GHz) is equal to the bandwidth of a high-speed comparator and the bandwidth can be obtained from the propagation delay: $bandwidth = 1/2\tau_{delay}$ and $\tau_{delay} = 0.25\ ns$ (Analog Devices, 2003). An example of the signal after subsampling is shown in Figure 6.11.

6.2.2 Linear Interpolation

If the sampling rate of an input signal is smaller than the TDC sampling rate, we resolve the TDC resolution by linearly interpolating the transient signals to the given resolution. Whenever a point on the grid exceeds the TDC threshold, the corresponding time was measured as the arrival time (take leading-edge detection for example). The linear interpolation is applied to all the TDC-based timing algorithms.

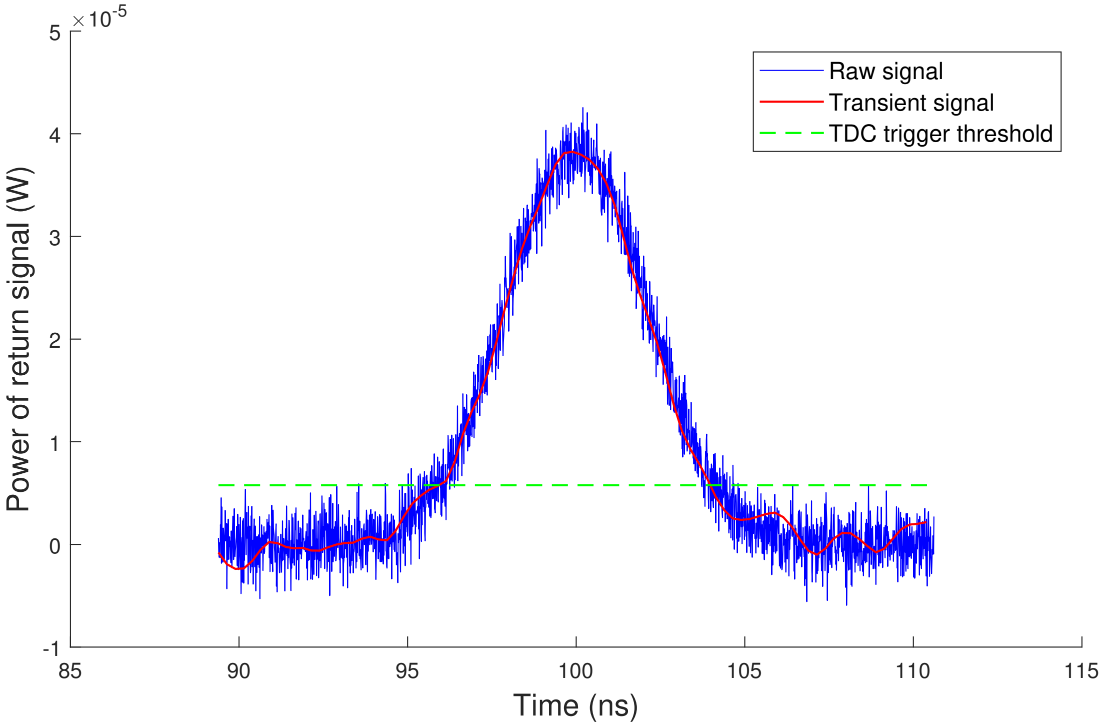


Figure 6.11: The Original 'Continuous' Return Signal (Blue) and Transient Signal (Red) after Subsampling. Dashed green line: TDC trigger threshold.

An example of the interpolation for the traditional leading-edge algorithm is shown in Figure 6.12. Note that in this example, the TDC resolution was set to 1 ps and the sampling rate of the signal was set to 100 GHz or 10 ps for illustration purpose. In the following simulation experiments, the signal sampling rate should be set according to the conclusion of Section 3.3.

6.2.3 TDC Trigger Threshold

Setting the trigger threshold of a TDC is important because a threshold that is too low or too high will increase the false triggering by the noise (probability of false alarm or PFA) or decrease the sensitivity of the algorithm to weak return signals (probability of detection or PD). In this section, the PFA is defined as the ratio of

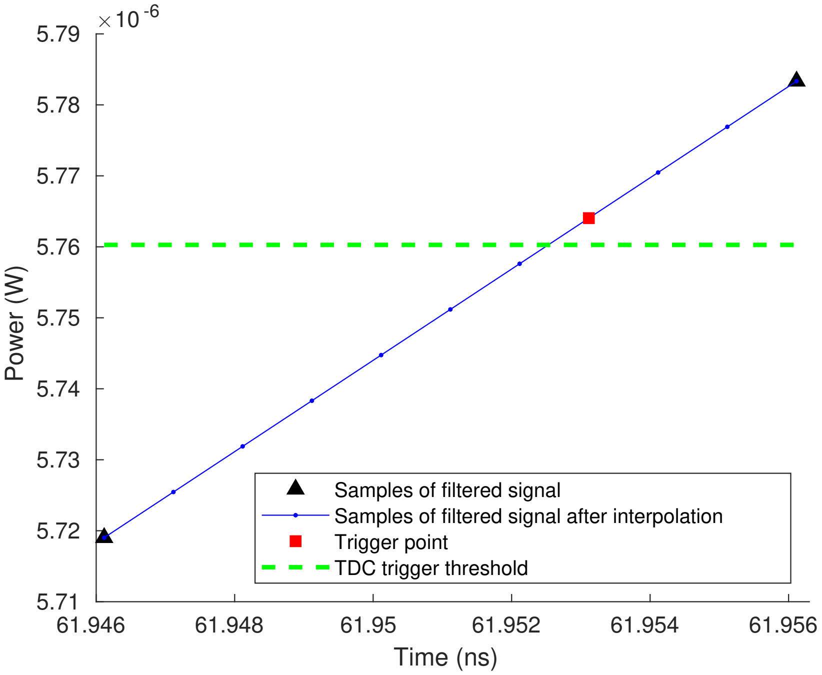


Figure 6.12: Illustration of Linear Interpolation (Sampling Rate: 100 GHz).

the number of samples of a noise signal above the TDC threshold to the total length of the signal. To balance the trade-off between the PFA and PD, we follows the Neyman-Pearson theorem to determine the threshold using a predefined PFA. More details of the Neyman-Pearson theorem is discussed in Chapter 9. We assume the noise of the electrical system is Gaussian-white noise. Theoretically, the threshold can be determined by:

$$V_{th} = \sigma_0 Q^{-1}\left(P_{fa}\right) + \mu_0, \tag{6.7}$$

where μ_0 is the mean of the noise which is equal to zero, σ_0 is the STD of noise signal (laser is turned off), and Q is the complementary cumulative distribution function (CCDF) of the Gaussian white noise, defined by (Kay, 1998)

$$Q\left(x\right) = \frac{1}{2} efc\left(\frac{x}{\sqrt{2}}\right). \tag{6.8}$$

Table 6.1: Parameters in the Lidar Model

Parameter	**Value**
Peak power	$300W$
Sampling rate	100 GHz
Target reflectivity	10%
TDC propagation delay	0.25 ns

6.3 Simulations

6.3.1 Simulation Design

A Monte-Carlo experiment was conducted using 500 observations of transmit and return signals. The target distance ranges from 5 to 60 m with an increment of 2.5 m. The return signals were generated by the pulse model, the propagation model, and the receiver model. The key parameters used in those models are given in Table 6.1, and other parameters remain unchanged.

6.3.2 TDC Trigger Threshold

The PFA is preset to 0.001 in this study, and it was evaluated on 100 observations of noise signals with the laser off. The noise signals were generated by the receiver model, each of which has 2123 data points, that is, there are total 212, 300 points in the 100 observations. The measured PFA is calculated by:

$$P_{fa,meas} = \frac{\sum_m k_m}{MN}, \tag{6.9}$$

where k_m is the number of data points in the m-th observation that is larger than the TDC threshold, and M and N are the number of observation and the number of points in each observation, respectively. The measured $P_{FA} = 0.000989$, which is smaller than the predefined P_{FA}.

6.3.3 Timing Algorithms

The traditional leading-edge timing algorithm, the leading-edge algorithm with TOT compensation, and the CFD algorithm were applied to return signals. The traditional leading-edge algorithm was applied to measure the start time when the leading-edge and the TOT algorithm were used. On the other hand, the start time of the CFD algorithm was measured by a simple peak detection. Since the transmit pulse is noise-free, the usage of the peak detection will not affect the TOF measurement. The constant fraction of the CFD is set to 30% of the peak power, and the arming threshold is set the same as the TDC threshold. The mean error and RMS error of the distance measurement are used for the evaluation of the performance of the timing algorithms.

6.4 Results and Discussion

6.4.1 Original Results

The results of the different timing algorithms are given in Figure 6.13. From the mean error, we can see that the leading-edge algorithm has an obvious positive bias from the ground truth, and it increases with distance. The walk error causes a positive bias. As the distance increases, the amplitude difference between the transmit and return signals increases, which results in a larger walk error. Compared to the traditional leading-edge algorithm, the bias from the results of the TOT compensation and the CFD algorithm is much less. The results of those two algorithms are given in Figure 6.13b to illustrate the details. The CFD has no bias at most the distances, while the TOT compensation has an increasing negative mean error as distance increases. There are two factors that contribute to the mean error. Note that the TOT compensation calculates the TOF by

$$TOF = t_{ret,lead} - \Delta_w - t_{start}, \tag{6.10}$$

where $t_{ret,lead}$ is the timestamp of the trigger event at the leading edge of the return signal, Δ_w is the walk error obtained from TOT curve, and t_{start} is the time when the pulse is emitted. The major factor of the negative bias is the inaccurate measurement of $t_{ret,lead}$. As known, the noise on the leading edge of the signal becomes comparable

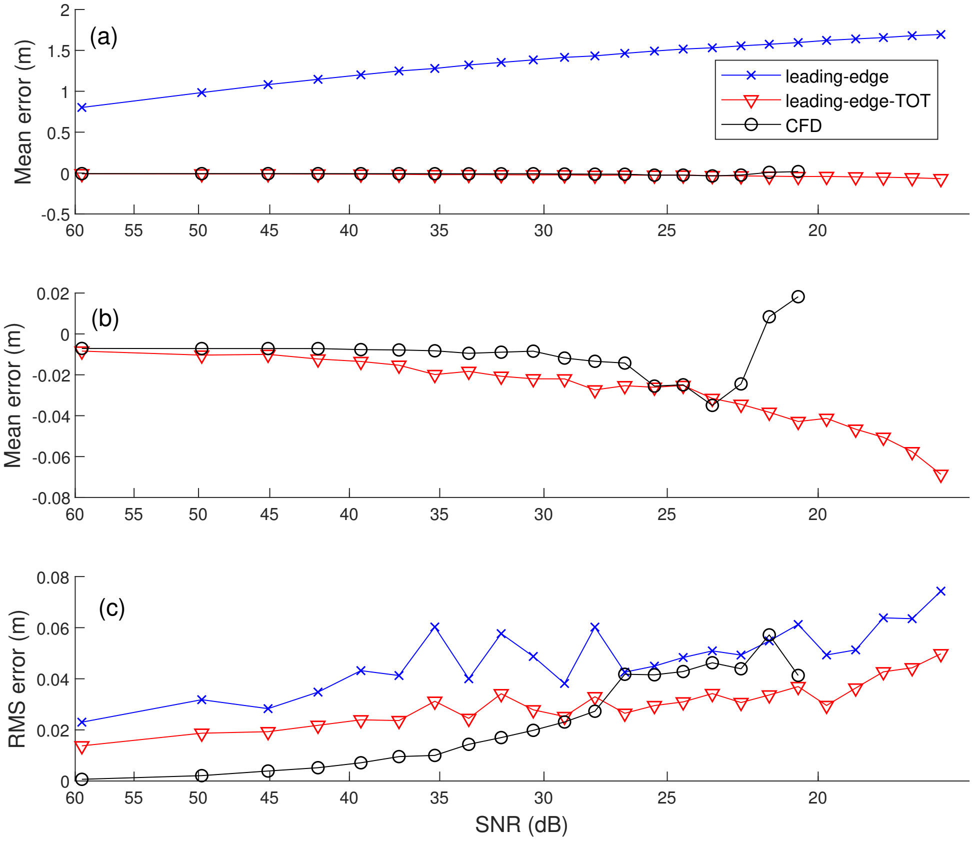


Figure 6.13: Mean Error and RMS Error of Different Timing Algorithms (without Low-Pass Filtering).

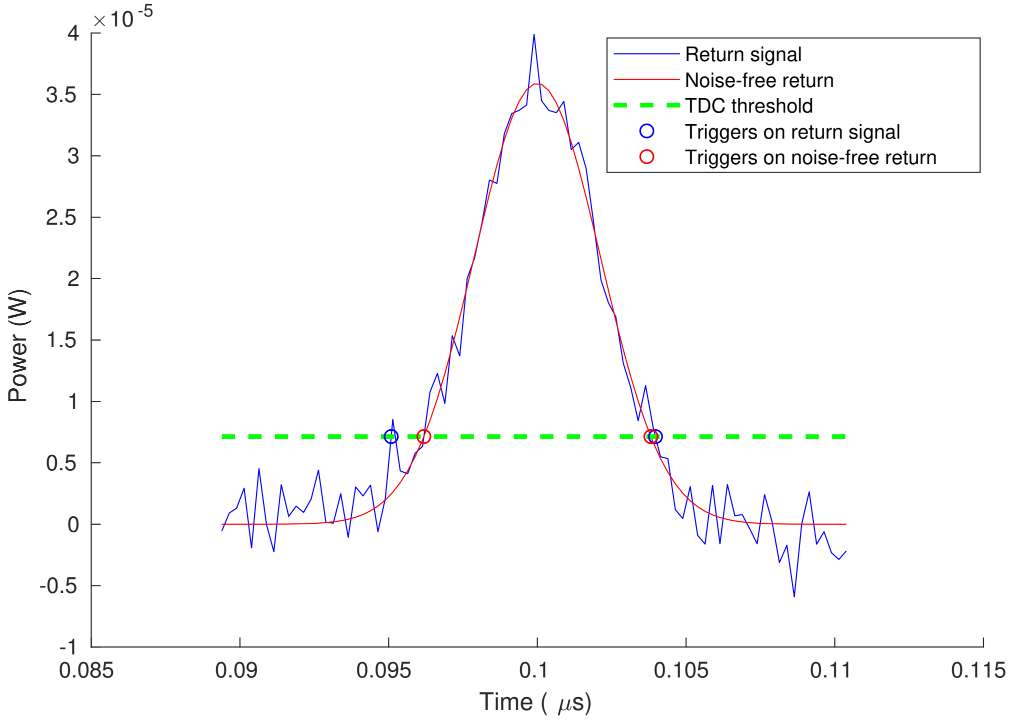


Figure 6.14: Early Triggering Caused by the Noise on the Leading Edge.

to the pulse at a further distance, which causes early triggering of the TDC. An example is given in Figure 6.14, in which the TDC is triggered earlier than it is supposed to be. In that case the $t_{ret,lead}$ is smaller than the ground-truth $t_{ret,lead}$, resulting in a smaller TOF estimation according to Equation (6.10). The statistical measurement of the 500 observations at distance 15m also provides evidence that the measured $t_{ret,lead}$ is 0.15 ns (or distance offset of 0.0233 m) less than the ground-truth value measured from the noise-free signals. Another factor for the mean error is the inaccurate measurement of the TOT when the SNR is low. Taking the 15 m distance as an example, the measured TOT from noisy return signals in the experiment is larger than the TOT of the corresponding noise-free signals, which results in the derived walk error being 0.002 m smaller than the ground-truth walk error. However, the inaccurate TOT measurement only contributes slightly to the mean error compared to the early triggering.

From the results of RMS error, we can see all the algorithms are subject to the noise on the signal when the SNR decreases. The traditional leading-edge algorithm is susceptible to the noise on the signal, while the effect of the noise is compensated to some extent in the TOT algorithm because both the leading edge and falling edge of the signal are taken into account. The CFD algorithm has the least RMS error because of the optimal triggering point. The fact that the random noise is canceled when the negative attenuated signal is added to the delay signal also reduces the RMS error. However, the CFD method does not work for low SNR conditions (< 21.47 dB), because in those cases the amplitude of the attenuated signal is lower than the arming threshold, and the algorithm ignores the signal.

6.4.2 Results after Low-pass Filtering

A low-pass filter with $f_{cut} = 0.4$ GHz was applied to the return signals before the timing algorithms to reduce the effect of the noise on the TOF estimation, and the phase shift was corrected. A small order of the low-pass filter was selected to avoid distortion of the pulse shape. Otherwise, the results of the TOT compensation and the CFD will be affected.

The results after low-pass filtering are shown in Figure 6.15. From the mean error, we can see the positive bias still exists in the traditional leading-edge algorithm because the bias is caused by the walk error which cannot be resolved by the low-pass filter. On the contrary, the low-pass filter improves the mean error of the CFD and the TOT compensation. An example of the filtered signal is given in Figure 6.16, in which the noise on the leading edge of the signal is significantly reduced. Statistical results at distance 15 m also show the $t_{ret,lead}$ of the filtered noisy signal is only 0.04 ns (0.006 m distance offset) smaller than the one of noise-free signals, and the difference of walk error is changed to 0.0038 m converted to distance. The reduction of the $t_{ret,lead}$ difference also demonstrates the influence of the noise on the TOF measurements. On the other hand, the RMS error of all the algorithms are reduced after the filtering, and the CFD still has the least RMS error followed by the TOT compensation.

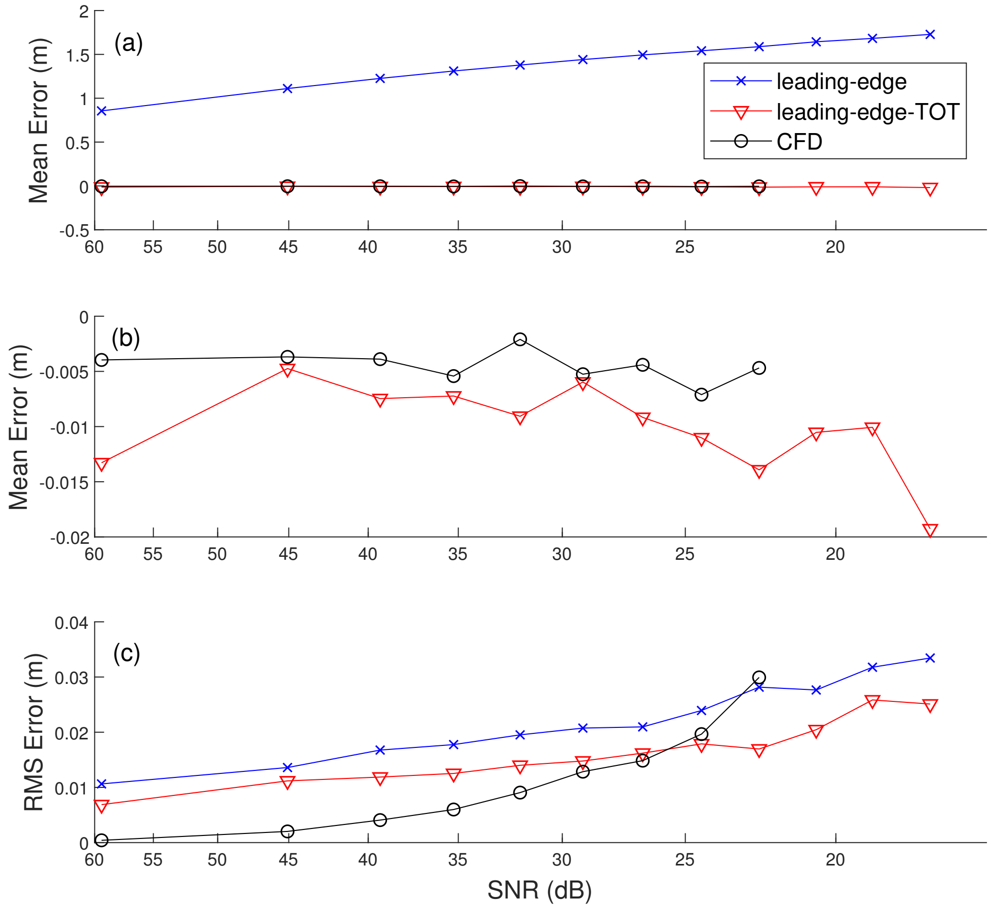


Figure 6.15: Mean Error and RMS Error after Filtering.

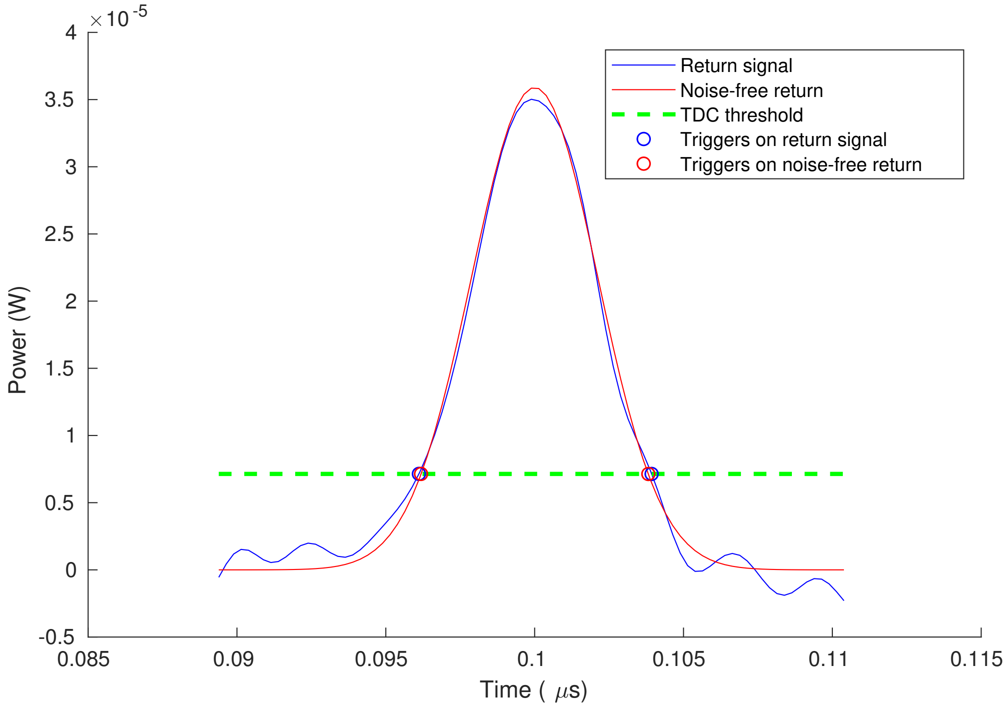


Figure 6.16: Signal after Filtering. Compared to Figure 6.14, the early triggering on the leading edge is significantly reduced by the low-pass filter.

Chapter 7

TIME DISCRIMINATION: ANALOG-TO-DIGITAL CONVERTER

7.1 Fundamentals of Analog-to-Digital Converter

7.1.1 Digital Processing

ADC-based time discrimination uses an ADC to convert filtered analog signals output from a photon-detector to digital signals, determine if the signal is a pulse or noise, and calculate the TOF if it is a pulse. The structure of the ADC-based discriminator is shown in Figure 7.1.

The digitization process of an ADC consists of two steps: sampling and quantization. The first step is to sample an analog signal in time by a sample-and-hold device that takes and holds a value from the continuous signal every T_s seconds. The time interval T_s is called sampling interval, and the sampling rate is denoted as $f_s = 1/T_s$. The resulting signal has a discrete time sequence but still has continuous values of amplitude. In the quantization step, the continuous values of the amplitude are converted to discrete values by a quantizer. Therefore, a signal with discrete values in both time and amplitude is generated, and such signals are referred to as 'digital'.

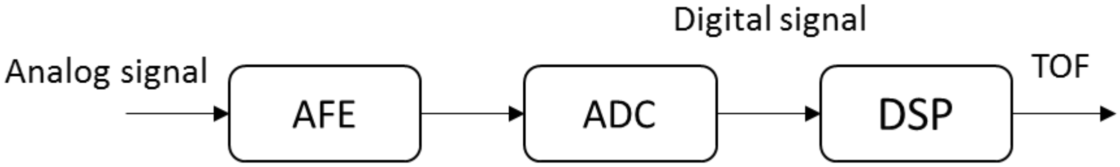


Figure 7.1: Schematic of ADC-Based Time Discrimination.

7.1.2 Quantization

In the quantization process, an important parameter of the quantizer is the resolution, which indicates the number of discrete levels it can produce over the dynamic range of the signal amplitude. The resolution is usually expressed in bits. For example, for a 6 bit ADC, there are 2^6 discrete levels to store the signal amplitude. The difference between two successive quantization levels is called quantization steps (Nakhostin, 2017)

$$\Delta = V_{k+1} - V_k \ . \tag{7.1}$$

Thus, the quantization steps Δ and the ADC resolution B, are related to the dynamic range V_{dy} of the input signal by

$$V_{dy} = 2^B \Delta. \tag{7.2}$$

The nature of the quantization process is to approximate an analog signal by discrete values. The finite number of quantization levels introduces error between the digital signal and the analog signal. The error is denoted as the quantization error, and its variance σ^2 is a function of the quantization step:

$$\sigma^2 = \frac{\Delta^2}{12}. \tag{7.3}$$

The signal-to-quantization noise ratio (SQNR) is defined as the ratio of the signal power to the noise power, and it can be written in dB as (Nakhostin, 2017)

$$SQNR_{dB} = 6.02B + 1.76 \ . \tag{7.4}$$

It should be noticed that an ADC with high resolution would have less quantization error and a large signal SQNR.

7.1.3 Sampling and Aliasing

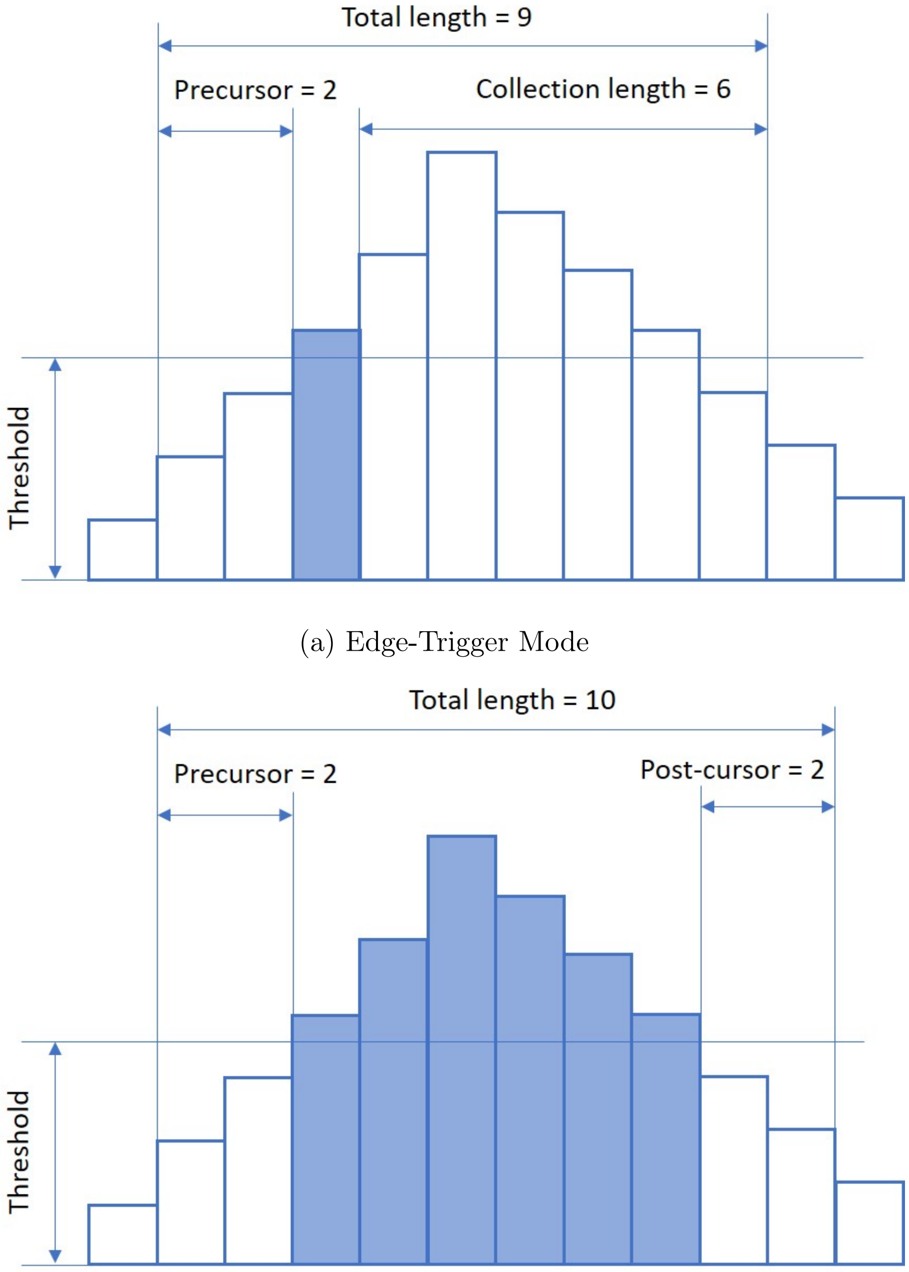
The sampling rate of an ADC affects the reconstruction of an analog signal. As known from the Nyquist's theorem, the sampling rate should be equal to or larger than twice the largest frequency (Nyquist frequency) of the signal to correctly represent the signal, *i.e.* $f_s \geqslant 2f_{max}$. Otherwise, aliasing occurs. However, the bandwidth of the input signal of an ADC may not always be known precisely, so the Nyquist criterion may not be met in practice. Therefore, to prevent aliasing, a low-pass filter is usually placed before an ADC, to truncate the frequency components larger than the Nyquist frequency. Such a filter is called anti-aliasing filter. In many cases, the anti-aliasing filter cannot completely remove all the high-frequency components due to the transition band of the analog filter. Therefore, to minimize the aliasing, an anti-aliasing filter could be designed to attenuate the frequency higher than the Nyquist frequency to a degree less than the quantization error (Mitra and Kuo, 2006). The anti-aliasing filtering could be achieved by selecting a filter with a sharp transition band or sacrificing some in-band frequencies by moving the transition band of the filter inside the signal bandwidth. However, either sharp transition band or cutting off in-band frequencies could distort the original signal. Alternatively, oversampling of the signal to a rate much higher than the Nyquist frequency is also often used to simplify the anti-aliasing filter and reduce the aliasing error.

7.1.4 Functionality

The important functions of an ADC include modes, zero-suppression, and trigger modes. An ADC usually has multiple channels taking analog inputs, and the channels could share one or multiple clocks. The ADC mode is the combination of channels that can be used. Taking a 4-channel ADC (Channel A, B, C, D) with a clock (5 GHz) as an example, the ADC can be operated in 1-channel mode (Channel A, B, C, D), 2-channel mode (*e.g.* AC, BD) and 4-channel mode (*e.g.* AAAA, ABCD). If all the channels share one clock, the 4-channel mode has a sampling frequency of 1.25 GHz signal, one-fourth of the speed of 1-channel mode (5 GHz). In the application of the TOF estimation, the 2-channel mode is commonly used which takes an analog signal from a laser source (START signal) and a return signal from a photon-detector (STOP signal). Another feature of an ADC is the analog offset. Using the analog offset function, users can remove the DC component of the signal to keep the signal remain within the ADC range, which maximizes the dynamic range.

The zero-suppression function is important for an ADC to reduce the load of data transfer. The zero suppression means that an ADC only starts to acquire data when the predefined ADC threshold is exceeded, and the stored waveform data is referred to as *packet*. A packet contains the sampled data and the absolute time of the last sample of the packet. The absolute time is measured since the data acquisition starts. The time stamp of each sample of a signal can be calculated using the time interval between the successive samples and the number of samples. An ADC acquires data in units of bins, which is a group of 8 points for the 2-channel mode.

The zero suppression function has two trigger modes: edge-trigger mode and level trigger mode, which collects data in two different ways. The edge and level trigger modes are illustrated in Figure 7.2. The edge-trigger mode specifies the length of the precursor in bins and the number of bins after the trigger point (collection length). It means all the data collected using the edge-trigger mode has the same length, but not all the samples are guaranteed to be above the threshold. On the contrary, the level-trigger mode saves all the samples above the threshold instead of saving a fixed number of samples. It means the length of the stored data is dependent on the signal amplitude. Both modes can also save some samples before the ADC trigger



(a) Edge-Trigger Mode

(b) Level-Trigger Mode

Figure 7.2: ADC Trigger Modes

point, which is called *precursor*. In addition, the level-trigger mode can also save samples after the sample becomes lower than the threshold, and such samples are called *post-cursor*.

7.2 ADC modeling

An ADC with 4 channels with a total sampling frequency of 5 GHz and a dynamic range of 10 bit are used in the prototype lidar (Cronologic, 2018). In this dissertation we uses two channels of the ADC and they share the same clock. Thus, each channel has a sampling frequency of 2.5 GHz, or the time interval between data samples is 0.4 ns. The ADC has a upper limit and a lower limit of the input signal amplitude, -0.25 V and 0.25V , respectively. The specification of the ADC is given in Table A.1. The ADC model takes return signals generated by the receiver module as inputs, and subsamples the signals with a rate of 2.5 GHz to model the ADC sampling process. The ADC model also digitizes the signal amplitude to 10 bit within the lower and upper limits. The ADC trigger threshold is set at 3x noise floor in the model, and the length of the precursor, collection length (edge-triggering) and post-cursor (level-triggering) are adjustable.

7.3 Time Discrimination Algorithms

The time discrimination using an ADC can be achieved by applying digital versions of the TDC-based timing algorithms to digital signals, or by taking advantages of the shape information contained in the digital signals.

7.3.1 Digital Version of TDC-based Timing Techniques

The digital versions of TDC-based timing algorithms include all the algorithms mentioned above like leading-edge detection techniques and CFD's. The difference from the TDC-based algorithm is that which algorithms take digital signals. The details of those methods are described in Chapter 6. A drawback of the digital algorithms is that since the digital signals only has discrete values, the TDC trigger threshold may not always coincide with the sample points of a digital signal. In other words, the exact trigger point may fall in between two successive samples. In this case, interpolation between neighborhood samples is needed to find the exact time mark corresponding to the threshold value. Linear interpolation is a common selection, but it could also affect the timing accuracy, depending on the local linearity of the signal (Nakhostin, 2017).

7.3.2 Detector and Estimator

The digital version of TDC-based timing methods utilizes the ADC as a TDC. This approach limits the potential of an ADC because the shape information contained in the digital signal is valuable, but ignored in TDC methods. Alternatively, ADC-based time discrimination algorithms can take advantage of the signal characteristics. ADC-based algorithms are generally composed of a detector and an estimator. The detector takes digital signals from an ADC as an input and determines the presence or absence of a transmit pulse from the signal contaminated by noise. Generally, we define a hypothesis $\mathcal{H}_1$ if the signal of interest (a pulse) is detected, and hypothesis $\mathcal{H}_0$ if the input signal is noise. If a pulse is detected, an estimator is applied to estimate the arrival time of the pulse.

The probability of detection (PD) and the probability of false alarm (PFA) are normally used to evaluate the performance of a detector. The probability of detection is defined as the ratio of the number of signals detected as $\mathcal{H}_1$ to the total number of $\mathcal{H}_1$ signals, while the probability of false alarm is defined as the percentage of false-positive signals to the total number of noise signals. In the next chapters, we will introduce two detection and estimation algorithms: centroid-based detection and estimation algorithm (benchmark method) and Neyman-Pearson (NP) detector.

Chapter 8

ANALOG-TO-DIGITAL CONVERTER: BENCHMARK ALGORITHM

8.1 Principle of Benchmark Detection and Estimation

8.1.1 Metrics for Signal Detection

In the benchmark detection, two metrics are utilized for the hypothesis determination of a return signal. The metrics are the number of points of a signal above the ADC threshold and the pulse width of the signal. In this section, the metrics and the corresponding criterion are defined first, followed by an introduction of the algorithm used for the pulse width calculation.

Metric 1: Number of points above threshold

Strong noises could false-trigger an ADC, but it is less likely that a large number of consecutive data points of white noise could exceed the threshold. In this case, we use the number of data points of a signal above the ADC threshold (*abbr.* Number above Threshold or NaT) as the first metric to prevent false triggering induced by random noise. Usually, the ADC threshold is set at a constant value (*i.e.* 3x noise floor), so when the amplitude of a signal changes, the NaT varies in a range according to the target distance, and stronger signals have a larger value of NaT. To make the metric applicable to signals from different distances, a range of the NaT is defined, and the upper and lower bound of the range is determined by measuring the NaT of noise-free return signals from the largest and smallest SNR's studied in this work, specifically, 60 dB and 8 dB, The corresponding measured NaT's are 52 points and 10 points, respectively. Therefore, when the ADC receives a signal, the NaT of a return signal is counted first, and if the NaT exceeds the range of 10 and 52, the signal is considered to be $\mathcal{H}_0$.

Metric 2: Pulse width

Usually, the pulse width of a signal remains constant through the whole signal propagation, assuming negligible distortion of the pulse shape due to reflection or by electrical systems. In this case, we can determine the hypothesis of a return signal by comparing its pulse width with that of the transmit pulse. The expression of the pulse width of a signal can be derived from Equation (3.4), such that

$$\Delta t_w = \frac{E}{P_{r,peak}}, \tag{8.1}$$

Here, we assume the approximation for the pulse energy mentioned in Section 3.1.1 also holds for digital signals. Theoretically, the energy of a signal can be obtained by integrating the signal over an infinitely long time duration, while in practice the signal is truncated due to a finite number of points that can be collected by an ADC. Similar to Metric 1, the criteria for Metric 2 is bounded by the pulse width of noise-free signals from the shortest and furthest distance. The pulse widths were calculated using the method that will be introduced in Section 8.1.2. The measured pulse width ranges from 4.91 ns (SNR = 8 dB) to 5.42 ns (SNR = 60 dB), and the variation of the pulse width from the mean value is 0.51 ns or 10% of the mean. Therefore, a tolerance of the measured pulse width is defined, such that if the difference between the measured pulse width $\Delta t_{w,meas}$ and the reference pulse width $\Delta t_{w,ref}$ is smaller than 10% of the reference, the signal passes Metric 2. In this study, the reference pulse width is 5.17 ns. Mathematically, the signal passes Metric 2 if

$$|\Delta t_{w,meas} - \Delta t_{w,ref}| < 10\% \Delta t_{w,ref}. \tag{8.2}$$

Signal detection

After defining the metric and the corresponding criterion, we can follow the flowchart shown in Figure 8.1 to determine if the signal is a pulse. In the detection process, the counting of NaT is straightforward, but the pulse width needs to be calculated from the signal. This will be illustrated next.

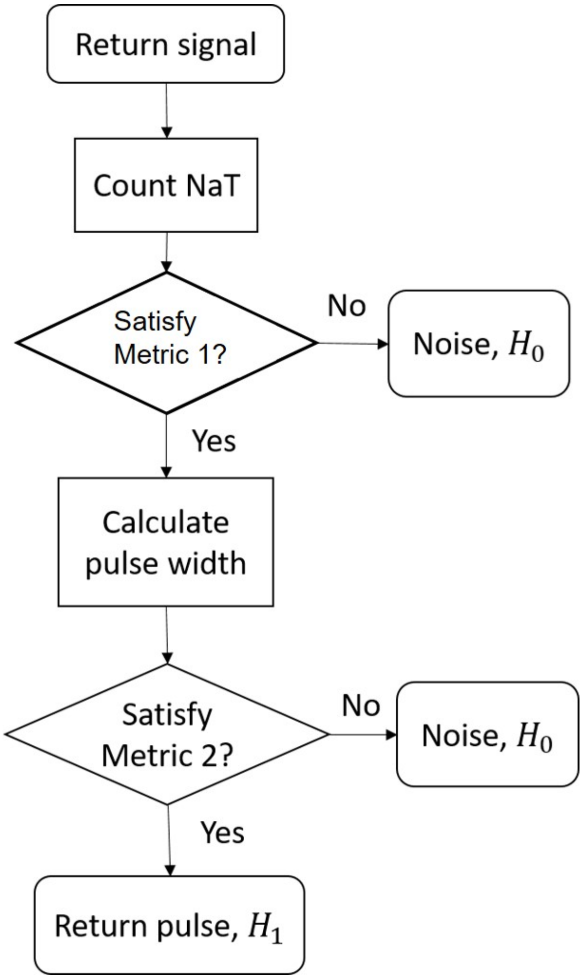


Figure 8.1: Flow Chart of the Benchmark Detection Algorithm.

8.1.2 Pulse Width Calculation

According to Equation (8.1), we need to obtain the energy and the peak power of a signal before calculating the pulse width. We first define a polygon with vertices as the data points of a signal. An example of the polygon is shown in Figure 8.2. The energy of the signal was approximated to be equal to the area of the polygon. A schematic of the numerical calculation of the area of a simple polygon is given in

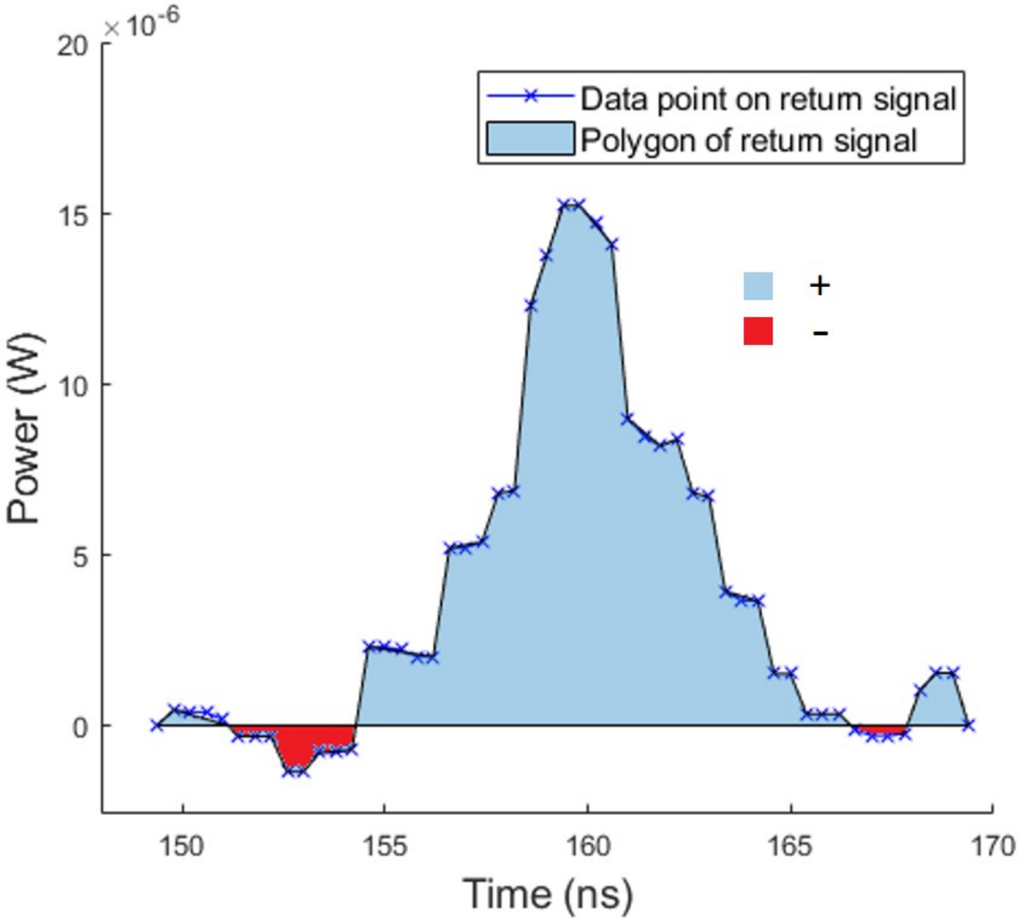


Figure 8.2: Example of a Polygon Defined by Data Points of a Return Signal. Regions with ’+’ signs (in blue) have positive area and regions with ’−’ sign (in red) have negative area.

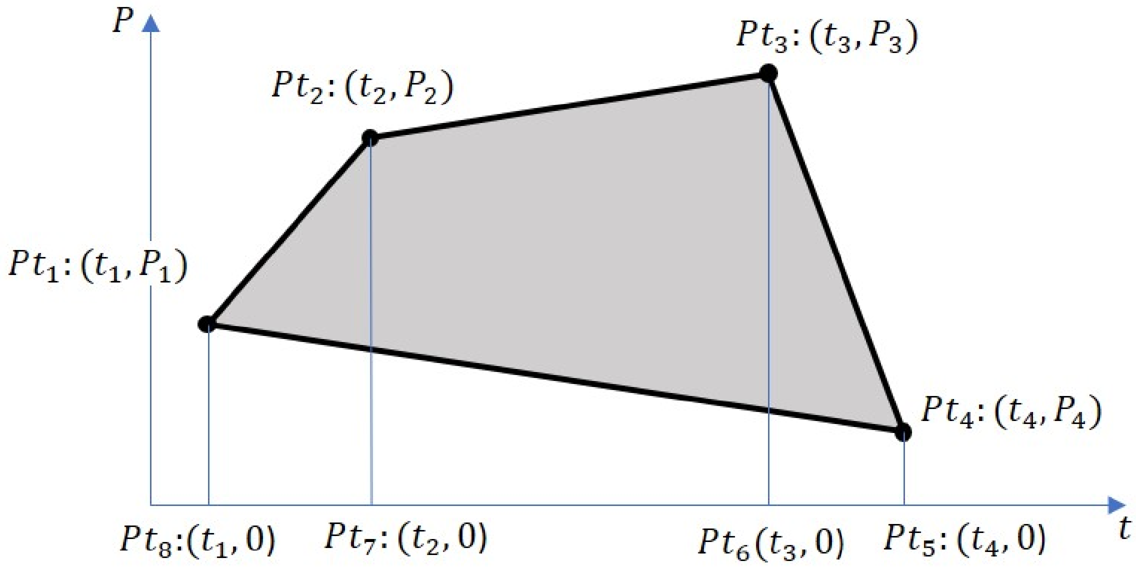


Figure 8.3: Schematic of the Numerical Calculation of the Area of a Simple Polygon (the Gray Region Bounded by Black Edges). Pt_5, Pt_6, Pt_7, Pt_8 are the data points projected on the t-axis.

Figure 8.3. In the figure, the polygon is defined by the sample points (t_n, P_n) and $n = 1, 2, 3, 4$. The area of the polygon A can be calculated by first dividing the polygon into several triangles (if the sample point is on the t-axis) and trapezoids, and then adding the areas of all the regions together (Bourke, 1988). In this case,

$$A = A_{(Pt_1, Pt_2, Pt_7, Pt_8)} + A_{(Pt_2, Pt_3, Pt_6, Pt_7)} + A_{(Pt_3, Pt_4, Pt_5, Pt_6)} - A_{(Pt_4, Pt_5, Pt_8, Pt_1)}$$

and the area of the trapezoid (Pt_1, Pt_2, Pt_7, Pt_8):

$$A_{(Pt_1, Pt_2, Pt_7, Pt_8)} = \frac{1}{2}(t_2 - t_1)(P_1 + P_2).$$

The same calculation can be applied to the other trapezoids. After reorganization, the polygon area can be expressed by

$$A = \frac{1}{2}\sum_{n=1}^{N}(t_n P_{r,n+1} - t_{n+1} P_{r,n}). \tag{8.3}$$

where the n-th vertex of the polygon is denoted by $(t_n, P_{r,n})$, and t_n and $P_{r,n}$ are the timestamp and the power value of the point, respectively. N is the total number of data points in a signal. Since the ADC removes the power offset of the digital signal in advance, the power can have negative values, which results in a self-crossing polygon, meaning the sides of the polygon intersect with itself. The negative values and the resultant negative areas of the self-intersecting polygon result in cancellation between the positive and negatives areas. The cancellation is consistent with our expectation that if a signal has powers with the same amplitude but opposite signs, the total energy of the signal is zero.

In this section, the x-direction indicates time and y-direction indicates power. The peak power can be approximated by the y-centroid C_y of the polygon using

$$P_{r,peak} = F_c \cdot C_y, \tag{8.4}$$

where F_c is a constant factor, approximately equal to 3. The value of 3 is derived from combining Equation (3.11) and the fact that the height of a Gaussian pulse $h_{pulse} \approx 2.817C_y$. The factor F_c indicates that the peak power of a Gaussian pulse is always F_c times the y-centroid regardless of the signal amplitude. The constant relationship provides a more stable way of estimating the peak power than directly measuring the peak value of a pulse which is subject to large fluctuation due to the noise at the peak.

Applying the same approach in Equation (8.3), the centroid of the polygon can be obtained using the area A by (Bourke, 1988):

$$\begin{aligned} C_x =& \frac{1}{6A}\sum_{n=1}^{N}(t_n + t_{n+1})(t_n P_{r,n+1} - t_{n+1} P_{r,n}) \\ C_y =& \frac{1}{6A}\sum_{n=1}^{N}(P_{r,n} + P_{r,n+1})(t_n P_{r,n+1} - t_{n+1} P_{r,n}) \end{aligned} \tag{8.5}$$

To simplify the algorithm, only the points above zero are used for the y-centroid calculation. Also, to close a polygon the value of $P_{r,N+1}$ at $n = N$ is set equal to the value of $P_{r,1}$ in Equation (8.3) and Equation (8.5), and to ensure the area cancellation, the value of $P_{r,N}$ and $P_{r,1}$ is set to zero, *i.e.* $P_{r,1} = P_{r,N} = 0$.

Now, given the energy E and the centroid of the polygon (C_x, C_y), the pulse width of a signal can be obtained by

$$\Delta t_w = \frac{E}{F_c \cdot C_y}. \tag{8.6}$$

8.1.3 Estimation of Arrival Time

After the determination of the hypothesis of a signal, the arrival time of return pulses should be estimated. A simple way for the estimation is to use the timestamp of the peak of a pulse. However, the noise on the peak of a signal could impact the measurement accuracy, and the result is also limited by the ADC resolution. Alternatively, we can approximate the lateral position of the peak by the x-centroid of the signal, and use the timestamp of C_x as the arrival time:

$$t_{STOP} = C_x. \tag{8.7}$$

In this case, the resultant TOF and distance measurement is much less sensitive to the noise near the peak of the signal, and an arrival time smaller than the ADC resolution can be resolved. To further improve the measurement accuracy, one can also use the data above the ADC threshold instead of the data above zero for Equation (8.7). The rational is that the data points above the threshold contain less noise than the data covering leading and falling edge regions. For a detailed discussion of using different data sets, please refer Section 8.2.2. It should be noticed that one should not use only the data above the threshold to calculate the y-centroid of a signal, since the constant factor F_c only holds for the data points that covers the entire Gaussian curve.

8.2 Application to Simulated Signals

8.2.1 Simulation

The performance of the benchmark detection and estimation algorithm was studied. The arrival time of the return signal was measured by the benchmark algorithm and the arrival time of the transmit signal (start time) was measured by a peak detection algorithm. Since the transmit signal is noise-free, the usage of the peak detection will not affect the TOF measurement. A Monte-Carlo experiment was performed on 500 observations of return signals generated by the propagation and receiver model, and 500 observations of pure white noise. The parameters used in the above models are given in Table 4.1 and Table A.1. The tested SNR ranges from 60 dB to 8 dB. Both edge and level trigger modes were tested. The settings of the trigger modes are given in Table 8.1, in which s is the number of bins that a signal contains above the ADC threshold and the value is depended on the signal amplitude. Also, the number of bins is set to be sufficiently large to cover the $1/e^2$ width of the Gaussian pulse.

Sparse random outliers can occur on the signals. Therefore, the signals are filtered

Table 8.1: Settings of ADC Trigger Modes

Trigger mode	Edge		Level		
Length of precursor (*bins*)	1	1	1	2	3
Collection-length (*bins*)	4	5	s[1]		
Length of post-cursor (*bins*)	-	-	1	2	3
Total length (*bins*)	6	7	$2+s$	$4+s$	$8+s$
Physical length (*ns*)	19.2	22.4	$6.4+3.2s$	$12.8+3.2s$	$25.6+3.2s$

[1] Variable s is signal dependent.

by a median filter with a kernel with a size of 3 to reduce the effect of the noise on the detection algorithm. The small size of kernels was used to avoid distortion of the signal by the filter.

8.2.2 Results and Discussion

Pulse width measurement

The pulse width of return signals with different SNR's was calculated, and the results are given in Figure 8.4a. In the figure, the number after the trigger mode indicates the collection-length in *bins* for edge-triggering and the number of bins of precursor and post-cursor for level-triggering. The results show that for a particular trigger

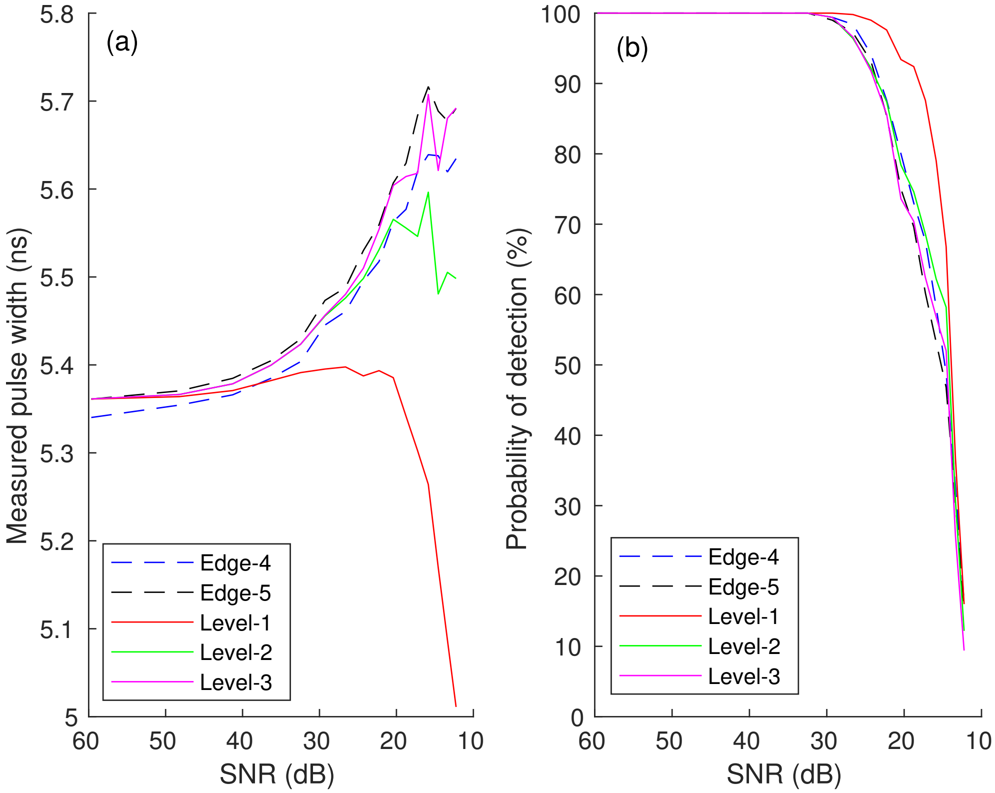


Figure 8.4: (a) Pulse Width Measurement and (b) Probability of Detection.

setting the pulse width increases/decreases with the SNR, and at the same distance (SNR) the pulse width increases with the length (number of bins) of the signal. These trends apply to both edge-triggering and level-triggering. The reason for the trends is that the amplitude of the signal decreases with distance, and the edge-triggering samples a constant length of a signal. Therefore, a larger portion of the leading and falling edge of the signal was captured for a signal from a longer distance, which lowers the y-centroid of the signal. Additionally, as the distance increases, the leading and falling edge of the signal become noisier. Also, the noise contributes to a large area at the lower portion of the signal which reduces the y-centroid. On the other hand, the measured area under the Gaussian curve remains almost constant as the positive and negative areas induced by the noise were canceled. Therefore, the pulse width increases according to Equation (8.1), and more data points captured by the ADC give a longer pulse width. The same reason applies to the edge-triggering as well. Level-3 triggering that captures more data points gives a longer pulse width than the other two. As the SNR decreases, the edges of a signal become noisier, which increases the pulse width for level-2 and level-3, respectively. An exception is level-1, which captures the smallest portion of the edges. In this case, the effect of the noise on the pulse width is the minimum. However, on the other hand, the small size of the precursor and the post-cursor truncates the Gaussian pulse, which results in a rise of the y-centroid of the signal and therefore, a smaller pulse width.

Probability of detection and probability of false alarm

The PD's were calculated using the same return signals for the pulse width calculation, and the PFA's at different trigger modes and ADC-trigger thresholds were computed using 500 noise signals generated by the receiver model with laser pulse turned off. The results of PD is shown in Figure 8.4b. From the results we observe that the probability of detection remains greater than 99% at an SNR of 30 dB. Moreover, all the trigger modes give a probability of false alarm of zero for the ADC trigger at 3x the noise floor and 2% for the trigger at 1x the noise floor. The reason is that the first metric (NaT) is the dominant factor of the PFA. Since the noise is white, the randomness of the white noise makes it unlikely for the NaT of noise to exceed toleration range. The median filter also removes the outliers that could cause a false detection. Based on the probability of detection and false alarm, we conclude that the detection algorithm that uses the number above the threshold and the pulse width of a signal as the metrics demonstrates a high performance on the signal detection for SNR larger than 30 dB. For lower-SNR (long distance) signals, the increased pulse width reduces the probability of detection. However, if the trigger mode and the length of the precursor and post-cursor is selected properly (*e.g.* level-1), the SNR can be extended to 23 dB with $P_d = 99\%$ and 19 dB with $P_d = 92\%$.

Accuracy and RMS error of distance measurement

The accuracy (mean error) and the RMS error of the distance measurement using the benchmark estimation algorithm are also evaluated on the 500 observations of the return signal at different distances using both edge and level triggering. The mean error $\overline{\Delta d}$ and the RMS error σ_d are defined as

$$\begin{aligned}\overline{\Delta d} &= \frac{1}{M}\sum_{m=1}^{M}(d_{meas,m} - d_{GT}) \\ \sigma_d &= \sqrt{\frac{1}{M}\sum_{m=1}^{M}(d_{meas,m} - \overline{d_{meas}})^2}\end{aligned} \tag{8.8}$$

where $d_{meas,m}$ is the distance measurement of the m-th observation of the total M observations, after converting the measured TOF to distance. $\overline{d_{meas}}$ is the average distance measurement over the M observations for a certain distance and d_{GT} is the corresponding ground-truth distance. The results of the errors are shown in Figure 8.5 and Figure 8.6.

From the results, we notice that the mean distance error for the edge triggering using data above zero presents a negative or positive bias. The reason is that at a low SNR the noise on the signal could cause an early triggering on the signal and since the edge-triggering samples a constant length of a signal, a longer leading edge and a shorter falling edge of the signal are captured. The asymmetric signal with respect to the vertical center line of the Gaussian pulse causes the left-shift of the x-centroid and

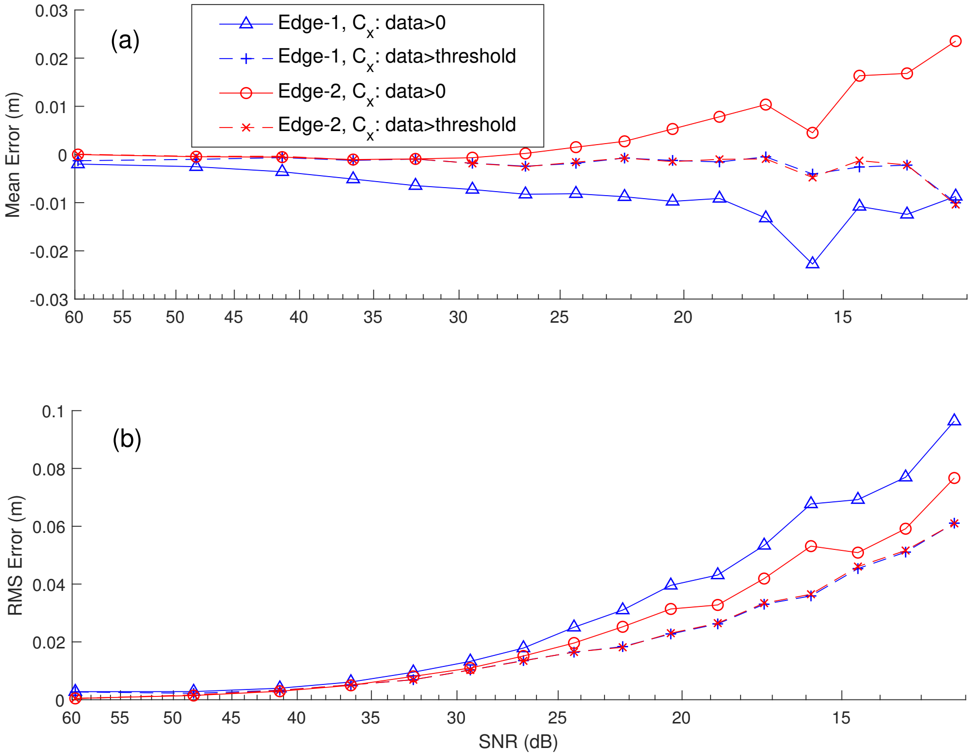


Figure 8.5: Results of Edge-Trigger Mode (Edge-1 and Edge-2): (a) Mean Error and (b) RMS Error. Distance calculated using data greater than zero is shown in the solid lines and using data greater than the ADC threshold is shown in the dashed lines.

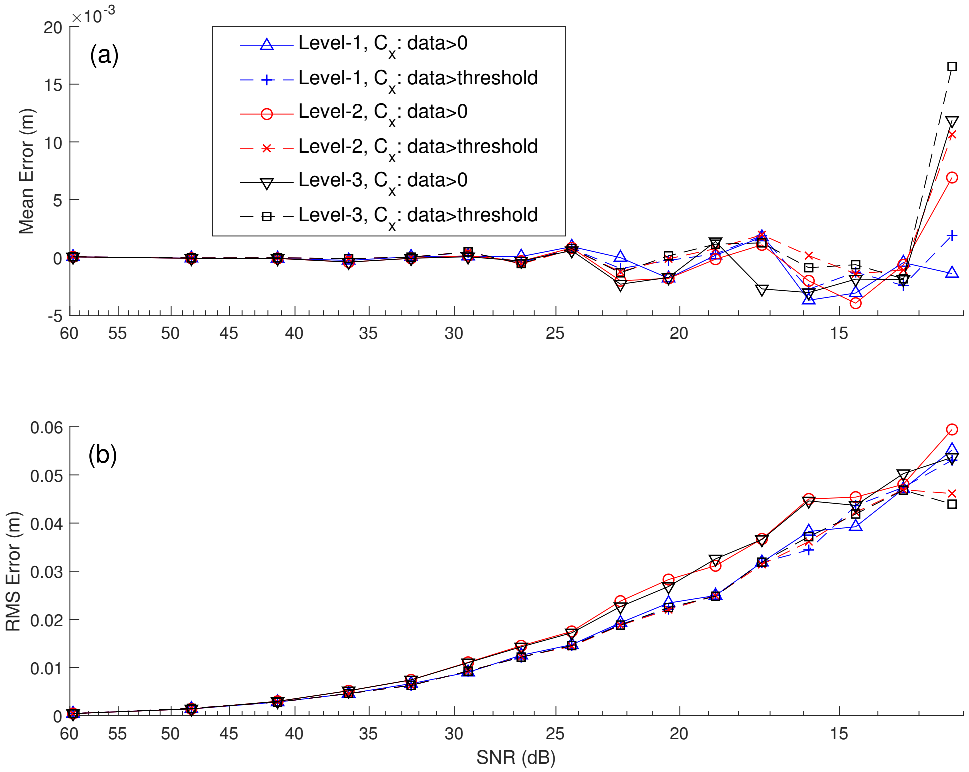
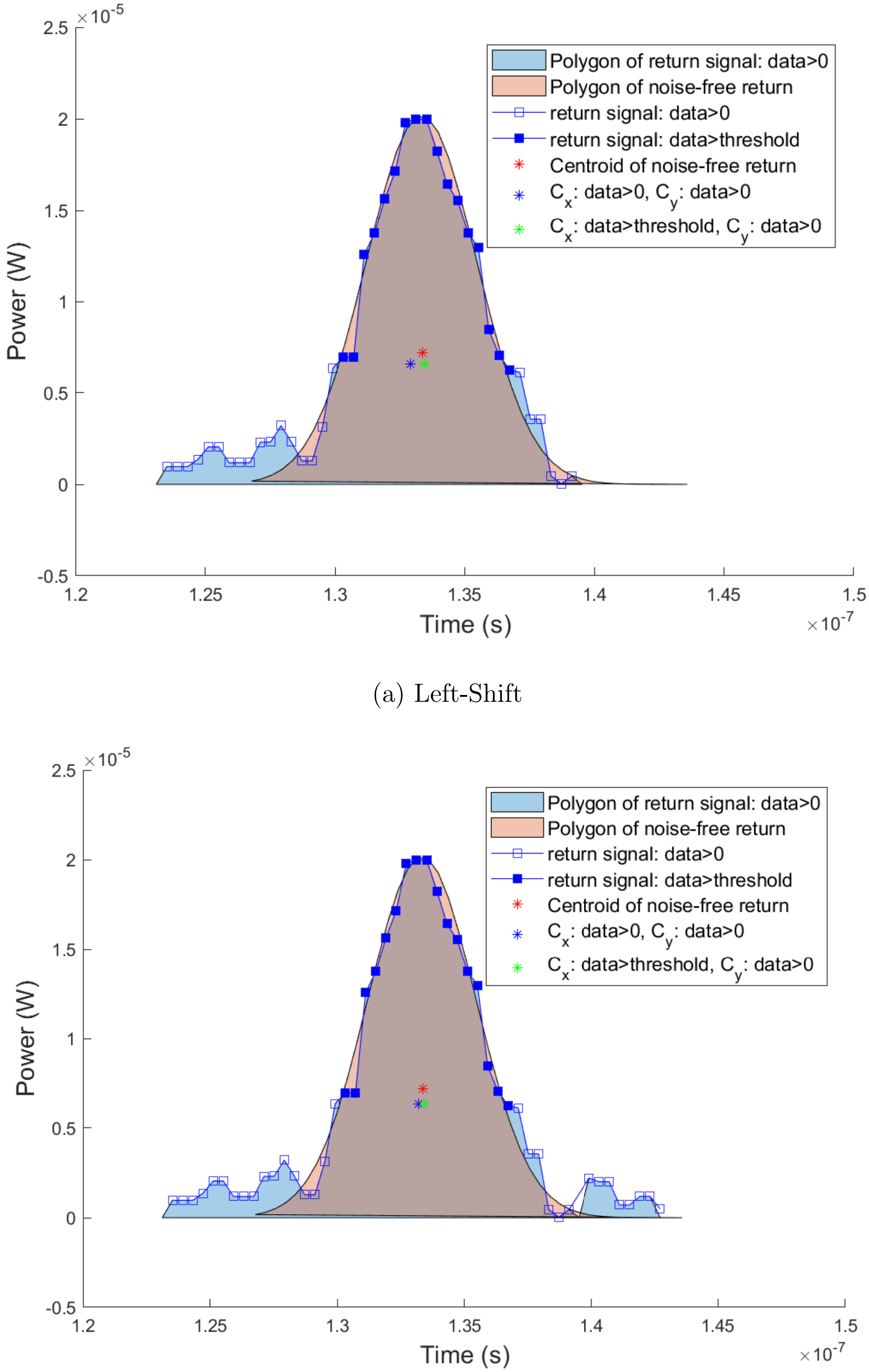


Figure 8.6: Results of Level-Trigger Mode (Level-1, Level-2, and Level-3): (a) Mean Error and (b) RMS Error. Distance calculated using data greater than zero is shown in the solid lines and using data greater than the ADC threshold is shown in the dashed lines.

consequently, earlier arrival time and shorter target distance. Figure 8.7a illustrates the left-shift of the x-centroid of the signal, in which the blue star stands for the x-centroid using data greater than zero and the red star represents the x-centroid of a corresponding noise-free signal. The blue star is shifted to the left of the red star because of the larger portion of leading-edge included in the signal. Setting a longer length of the precursor and post-cursor to cover a longer falling edge could partially reduce the left-shift. An example is shown in Figure 8.7b, in which the same signal is sampled by twice the length of the precursor and post-cursor used in Figure 8.7a. In this case, the blue star moves closer to the red star. However, since the trigger position on the signal is random and difficult to predict, a longer falling edge could also cause a right-shift of the x-centroid and a further target distance (solid red line in Figure 8.5). An example of the right shift of the x-centroid is provided in Figure 8.7b, in which the blue star shifts to the right of the red star because of the longer falling-edge captured by the ADC.

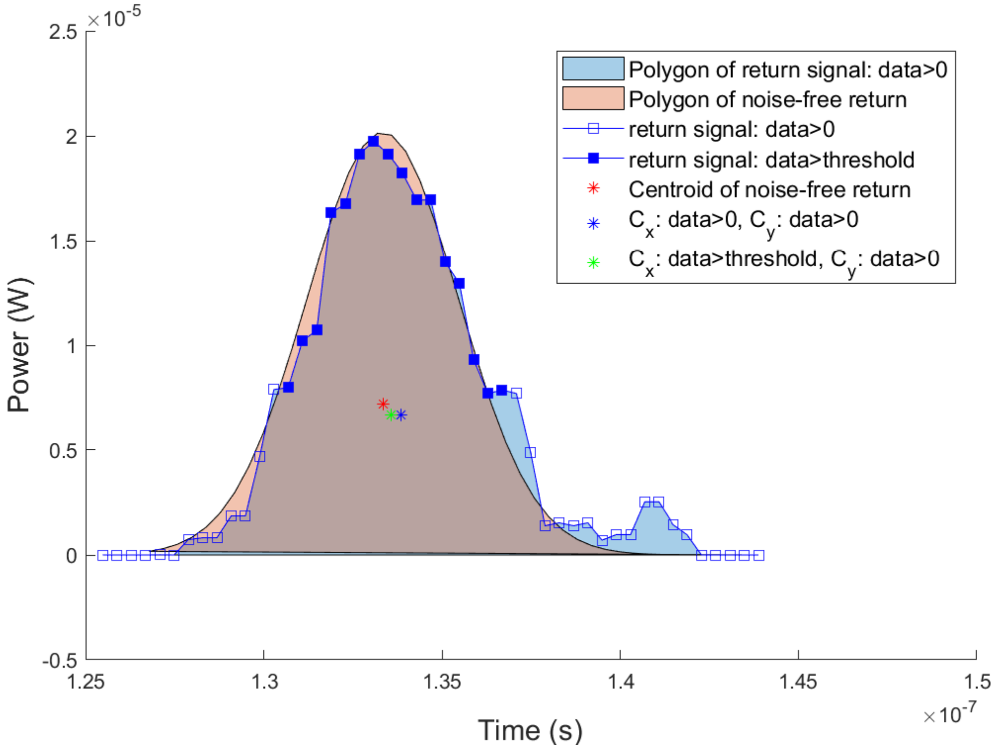
Adjusting the length of the precursor and the post-cursor of the edge-trigger is not an optimal solution to the negative/positive bias of the distance measurement due to the randomness of noise. An alternative solution is to use the data above the ADC threshold rather than the data greater than zero. The benefit of using those data points is that most of the data points above the threshold belong to the Gaussian pulse instead of the noise, and they have symmetrical distribution around the vertical center line of the pulse. In this case, the x-centroid is calculated from symmetric signals. The results of using data above the threshold are shown in the dashed lines in Figure 8.5, which presents reduced bias compared to the results using the data points above zero. Contrary to the edge-triggering, the positive/negative bias does not appear to affect the results of level-triggering (Figure 8.6). The reason is that the nature ofcxcxcxc the level-triggering is to capture the data points above the threshold and symmetric signals are captured in the x-centroid calculation.

From the comparison of the RMS error of edge-1 and edge-2 (Figure 8.5b), we observe that the edge-2, that covers a larger portion of the pulse yielded a smaller RMS error. The reason could be that the more samples at the falling edge of signals compensate for the random early-triggering at the leading edge, so the fluctuation of distance measurement is reduced. However, the length should be chosen with caution to avoid positive or negative bias on the distance measurement. Level-triggering is less affected by the early-triggering at the edges, but it is sensitive to the noise at the edges. Therefore, signals with fewer samples included at the edges have better precision (smaller RMS error), which is evidenced by Figure 8.6b. Also, notice that the level-triggering generally gives better precision than edge-triggering. The RMS error due to using data greater than zero and data above the ADC threshold are also compared. The results of both edge-triggering and level triggering show the former gives a larger RMS error, due to noise which is included in the signal and causes a larger fluctuation on the x-centroid measurement.



(a) Left-Shift

(b) No-Shift



(c) Right-Shift

Figure 8.7: Examples of Signals with Different x-centroid Shifts (Edge-Triggering). Case (a) and (b) are from the same signal with different trigger settings, and Case (c) is from a different signal. SNR=20.41 dB in all the cases. The trigger settings are: (a) edge-1, (b) edge-2, and (c) edge-2.

Chapter 9

ANALOG-TO -DIGITAL CONVERTER: NEYMAN-PEARSON DETECTOR

The Neyman-Pearson(NP) detector was first introduced by Jerzy Neyman and Egon Pearson, and has shown to be the favorible detector for radar/ laser signal detection (Neyman and Pearson, 1933; Kay, 1998). The NP detector has been widely utilized in many applications like range sensing, medicine, nuclear energy, and gravitational-wave astronomy (Hoover *et al.*, 2000; Bousselham and Bohm, 2007; Grönwall, 2006; Grönwall *et al.*, 2007; Gu and Tjahjadi, 2002; Jordan, 2009; Roman *et al.*, 2000; Ofek and Zackay, 2018; Vio and Andreani, 2018). In general, the NP detector conducts a likelihood ratio test (LRT) to determine the existence of a known deterministic signal (*i.e.* a signal that has no unknown parameters), and a generalized likelihood ratio test (GLRT) for detection of signals with unknown parameters. In our case, since the arrival time and the amplitude of the return signal are unknown, a GLRT is utilized. According to maximum likelihood estimation, the arrival time can also be derived during the detection process (Kay, 1993). Therefore, the detection and estimation problem are combined by using the NP detector. The problem is defined in the next section.

9.1 Problem Definition

Hypothesis $\mathcal{H}_1$ states the signal $\vec{x} = \{x[0], x[1], \ldots, x[n]\}$ collected by an ADC is Gaussian distributed white noise. Hypothesis $\mathcal{H}_1$ states that the signal is the summation of the noise-free return pulse with unknown amplitude and white noise following another Gaussian distribution. Symbolically,

$$\begin{aligned} &\mathcal{H}_0 : x[n] = w_0[n] && n = 0, 1, \ldots, N-1 \\ &\mathcal{H}_1 : x[n] = As[n - n_0] + w_1[n] && n = 0, 1, \ldots, N-1 \end{aligned} \tag{9.1}$$

where $\vec{w}_0 = \{w_0[0], w_0[1], \ldots, w_0[n]\}$ and $\vec{w}_1 = \{w_1[0], w_1[1], \ldots, w_1[n]\}$ are the white noise under $\mathcal{H}_0$ and $\mathcal{H}_1$. Each sample of $\vec{w}_0$ and $\vec{w}_1$, $w_0[n]$ and $w_1[n]$, has a variance W_0 and W_1, respectively. The signal $\vec{s}$ is the template of the transmit signal normalized to unit amplitude which is also called the *kernel*. The kernel has a size of M and is scaled by a unknown amplitude A representing the attenuation of the signal during the propagation and reflection. The index n represents the timestamp of the n-th sample of the signal with a size of N, and n_0 stands for the unknown arrival time. The time period $[0, N-1]$ should cover the signal for all the possible arrival time, *i.e.* $n_0 \in [0, N-M]$.

Note in Equation (9.1), the noise on each sample is approximated to be Gaussian distributed, even though the shot noise follows a Poisson distribution. This approximation is explained in Wall (1979) that, despite the fact that the incident number of photons follows Poisson statistics, the distribution is reduced to Gaussian if a photon-detector integrates sufficient number of photons (> 10) per integration period. Since the collected number of photons is far beyond 10 in our case, the Gaussian approximation is valid. In addition, the white noise $\vec{w}_0$ and $\vec{w}_1$ usually have different variances because the shot noise induced by the laser pulse also contributes to the variance of W_1. However, the shot noise is amplitude dependent and the amplitude of incident signals is unknown, so the W_1 is usually unavailable. Therefore, the $\vec{w}_0$ and $\vec{w}_1$ are assumed to be identical and denoted by $\vec{w}$ in this study. The samples in the white noise $\vec{w}$ has a variance W. In next few sections, we will see that this approximation does not affect the probability of false alarm and has negligible influence on the probability of detection.

9.1.1 Neyman-Pearson Theorem

In this section, we introduce the Neyman-Pearson theorem which allows us to decide the hypothesis of a signal. The Neyman-Pearson theorem states that, to maximize the probability of detection (PD) P_d of a signal for a given probability of false alarm (PFA) $P_{fa} = \alpha$, decide $\mathcal{H}_1$ if the generalized likelihood ratio $L_G(\vec{x})$ is greater than a threshold γ; otherwise, decide $\mathcal{H}_0$:

$$L_G(x) = \frac{p(\vec{x};\hat{n}_0, A, \mathcal{H}_1)}{p(\vec{x};\mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma. \tag{9.2}$$

The threshold γ can be derived from the definition of the PFA

$$P_{fa} = \int_{\{\vec{x}:L_G(\vec{x})>\gamma\}} p(\vec{x};\mathcal{H}_0)d\vec{x} = \alpha, \tag{9.3}$$

and the PD can be obtained by:

$$P_d = \int_{\{\vec{x}:L_G(\vec{x})>\gamma\}} p(\vec{x};\hat{n}_0, A, \mathcal{H}_1)d\vec{x}. \tag{9.4}$$

The probability density distribution (PDF) of signals with unknown parameters $\hat{n}_0$ and A under hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ are denoted by $p(\vec{x};\hat{n}_0, A, \mathcal{H}_1)$ and $p(\vec{x};\mathcal{H}_0)$, respectively. $\hat{n}_0$ is the estimation of the arrival time n_0 to be determined. Equation (9.2) is called the generalized likelihood ratio test (GLRT), which evaluates if the likelihood of a signal being $\mathcal{H}_1$ versus being $\mathcal{H}_0$ satisfies the required PFA. After mathematical reorganization, the left-hand side of Equation (9.2) is reduced to (The derivation is provided in Appendix D):

$$T_{max} = T(x[\hat{n}_0]) = \sum_{n=\hat{n}_0}^{\hat{n}_0+M-1} x[n]s[n-\hat{n}_0], \tag{9.5}$$

and the right-hand side is reformed to

$$\gamma' = \frac{A}{2}\sum_{n=0}^{M-1} s^2[n] + \frac{W \ln\gamma}{A}. \tag{9.6}$$

Equation (9.3) can be expressed as

$$P_{fa} = Pr(T(\vec{x}) > \gamma'; \mathcal{H}_0). \tag{9.7}$$

In the above equations, $T(\vec{x})$ is the test statistic which needs to be calculated. Equation (9.5) indicates the GLRT first calculates the cross-correlation $T(x[n_0])$ of the signal $\vec{x}$ with the kernel $\vec{s}$ for all possible n_0. Then, the GLRT finds the maximum value of $T(x[n_0])$ at $n_0 = \hat{n}_0$, denoted by T_{max}. By comparing the T_{max} with the threshold γ', the presence of the pulse can be determined. Mathematically, the T_{max} can be expressed by

$$T_{max} = \max_{n_0 \in [0, N-M]} T(x[n_0]) = \max_{n_0 \in [0, N-M]} \sum_{n=n_0}^{n_0+M-1} x[n]s[n-n_0], \tag{9.8}$$

and the GLRT (Equation (9.2)) becomes

$$T_{max} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma'. \tag{9.9}$$

If the pulse is present, the arrival time $\hat{n}_0$ can be estimated as the timestamp of T_{max}, such as

$$\hat{n}_0 = \underset{n_0 \in [0, N-M]}{\operatorname{argmax}} \; T(\vec{x}). \tag{9.10}$$

Equations (9.8) to (9.10) represent the core equations for signal detection using Neyman-Pearson Theorem, and such a detector is called NP detector, or *matched filter* (MF). In this work, the NP detector and matched filter are used interchangeably.

In practice, $T(\vec{x})$ can be obtained from the correlation between the signal and the kernel, or convolution of the signal with the conjugated time-reversed kernel $s'[n]$. In other words, at each timestamp n, the convolution $T(x[n])$ can be obtained by

$$\begin{aligned} T(x[n]) &= \sum_{m=0}^{n} x[m]s'[n-m] \\ s'[n] &= s[N-1-n] \end{aligned} \tag{9.11}$$

The correlation and the convolution approach are proved mathematically equivalent to each other (Kay, 1998). The major difference is the way of implementation. The correlation approach utilizes multipliers followed by an integrator, while the convolution approach (MF) is an finite impulse response filter. Both approaches will give us the same final results. For simplicity, the convolution approach is applied in this dissertation.

9.1.2 Fast Fourier Transform Approach

The convolution approach of the NP detector can be achieved by sliding the kernel over a signal in the time domain (sliding-window approach, or SW approach). However, the time complexity is equal to $\mathcal{O}(MN)$ (M and N are the respective number of samples in the kernel and the signal), which means the NP detector could be computationally intensive if the signal contains a large number of sample points. Therefore, an alternative approach is necessary for such situations or real-time computation is a demand. Since the convolution of two signals in the time domain is equivalent to a dot product of the Fourier transform of the two signals in the frequency domain, Equation (9.11) can be implemented in the frequency domain by using a fast Fourier transform (FFT). This approach is called the FFT approach. Mathematically,

$$\begin{aligned} X &= \mathcal{F}[\vec{x}] \\ S &= \mathcal{F}[\vec{s}] \\ Y &= X \cdot S \\ T(\vec{x}) &= \mathcal{F}^{-1}[Y] \end{aligned} \tag{9.12}$$

where the variable X, S, and Y are the Fourier transform of signal $\vec{x}$, $\vec{s}$ and $T(\vec{x})$ in the frequency domain. The symbols $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier transform and inverse-Fourier transform operators. Since the time complexity of an FFT operation is $\mathcal{O}(N \log N)$, the computational cost is dramatically reduced. Using FFT with zero-paddings can also resolve arrival time finer than the ADC sampling interval.

The zero-padding technique is to add zeros to signals in time or frequency domain, so that the sampling rate of signals in frequency or time domain increases after the Fourier transform or the inverse-Fourier transform. In an FFT operation without zero-padding, if the number of samples of a signal in frequency domain is equal to the number used in the FFT in the time domain, the resolution of the signal is unchanged, *i.e.* there is no improvement in the resolution. However, for a signal having N samples (sampling rate of $1/N$) in the time domain, if N zeros are padded at the end of the Fourier transformed signal in the frequency domain (assuming the frequency is centered around zero), the inverse-Fourier transformed signal (time domain) will have $2N$ samples. Since the time range is unchanged, the resolution of the signal is doubled. Furthermore, the FFT is most efficient when the number of samples in a signal is an integer power of two, which means zero padding in the time domain is also necessary for signals that have varying lengths. Note that the time-domain zero-padding only extends the length of the signal rather than increases the signal resolution.

The application of zero-padding to an FFT is explained next with a demonstration given in Figure 9.1. In this demonstration, the original signal $\vec{x}$ has $N = 21$ samples, and the kernel has $M = 5$ samples. First, we zero-pad the signal $\vec{s}$ and $\vec{x}$ to the total length of the two signals ($M + N - 1 = 25$) in the time domain. If the total length ($M+N-1$) is not a power of two, additional zeros are padded to achieve the smallest power of two larger than $M+N-1$. The length is denoted by $L_{p2} = 32$ (Figure 9.1a). The second time-domain zero-padding is to ensure the maximum efficiency of the FFT. After an FFT and centering the spectrum around zero, KL_{p2} zeros are added to the dot-product Y ($KL_{p2}/2$ zeros at the tail on each side), in which K is an

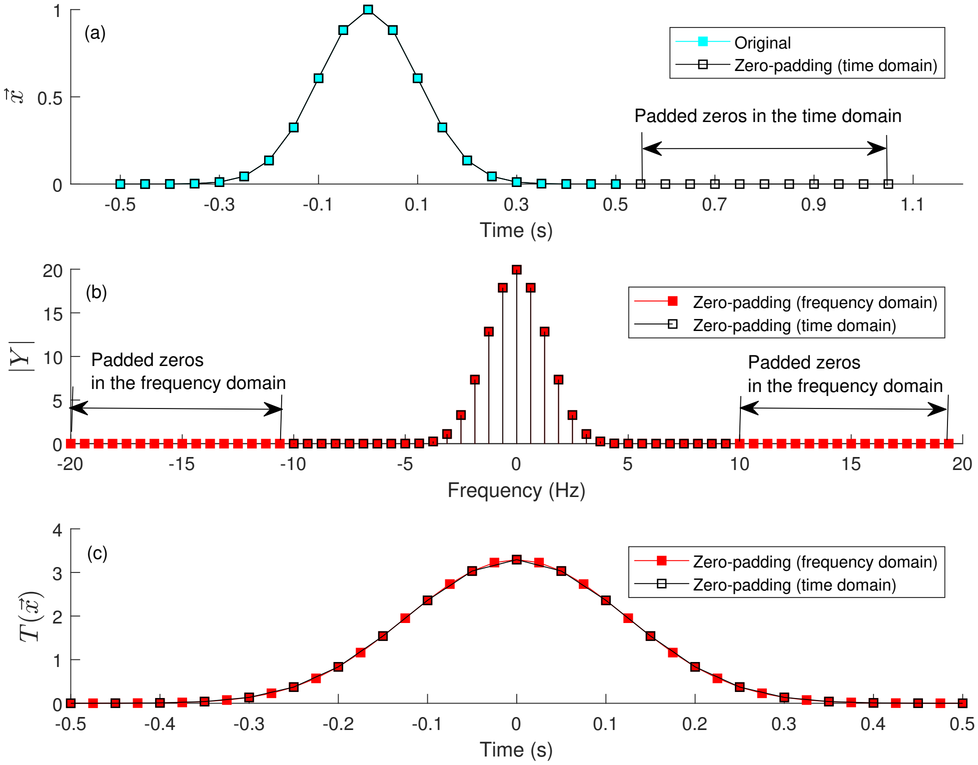


Figure 9.1: Demonstration of FFT with Zero-Padding in the Time Domain and Frequency Domain. Only zero-padding on signal $\vec{x}$ is shown in (a). In (c) the convolution result $T(\vec{x})$ has been truncated to $(K+1)N = 42$ length of the original signal $\vec{x}$.

non-negative integer and $K = 1$ in Figure 9.1. After the zero-padding in frequency domain, the length of the resultant signal is extended to $(K+1)L_{p2} = 2L_{p2} = 64$ (Figure 9.1b). Then, the resultant signal is converted back to the time domain using an inverse-FFT, and the length of the signal remains the same. The convolution result $T(\vec{x})$ has additional zeros due to the time-domain zero-padding at the first step, so the additional zeros are removed from the convolution result. After the zero-removal, the length of the result $T(\vec{x})$ becomes $(K+1)(M+N-1) = 42$ (Figure 9.1c), *i.e.* the resolution increases $K+1 = 2$ times than the SW approach. Accordingly, the initial time interval Δt is reduced to $\Delta t/(K+1) = \Delta t/2$. Note that the length of the convolution result is $(K+1)(M+N-1)$, larger than length of the new time series $((K+1)N)$ at this step, so only the center portion of the convolution result is kept, so that both the length of the convolution result and the time series are the same $((K+1)N)$. Additionally, since the amplitude of the convolution result is reduced due to the frequency-domain zero padding, the amplitude of $T(\vec{x})$ has to be amplified $(K+1)$ times to remain the original amplitude. After the zero-padding in the time domain and the frequency domain, we can see in Figure 9.1c, the resolution of the convolution result is doubled.

9.1.3 PDF's of $T(\vec{x})$

The threshold γ' needs to be determined in a NP detector, and it can be achieved using Equation (9.7) given a PFA $P_{fa} = \alpha$. Therefore, the PDF's of $T(\vec{x})$ under both hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ are required. Since $T(\vec{x})$ is a linear combination of Gaussian distributed variables, the PDF's of $T(\vec{x})$ should also follow a Gaussian distribution. The estimation of the mean $\hat{\mu}$ and the variance $\hat{\sigma}^2$ for hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ are given below (derivation is given in Appendix D):

$$\begin{aligned}
\hat{\mu_0} &= 0 \\
\hat{\sigma_0^2} &= W \sum_n s[n]^2 \\
\hat{\mu_1} &= A \sum_n s^2[n] \\
\hat{\sigma_1^2} &= W \sum_n s^2[n],
\end{aligned} \tag{9.13}$$

and the PDF's of $T(\vec{x})$ follow

$$\begin{aligned}
\mathcal{H}_0 :& T(\vec{x}) \sim N(\hat{\mu_0}, \hat{\sigma}_0^2) \\
\mathcal{H}_1 :& T(\vec{x}) \sim N(\hat{\mu_1}, \hat{\sigma}_1^2).
\end{aligned} \tag{9.14}$$

After normalization by the mean and standard deviation, the normalized test statistic $T'(\vec{x}) = \frac{T(\vec{x})-\hat{\mu}}{\hat{\sigma}^2}$ follows the standard normal distribution $\mathcal{N}(0,1)$.

9.1.4 Estimation of Threshold γ'

Having the PDF of $T(\vec{x})$, Equation (9.7) can be expressed by

$$\begin{aligned} P_{fa} &= Pr(T(\vec{x}) > \gamma'; \mathcal{H}_0) \\ &= Pr(T'(\vec{x}) > \frac{\gamma' - \hat{\mu}_0}{\hat{\sigma}_0}; \mathcal{H}_0) \\ &= Q(\frac{\gamma' - \hat{\mu}_0}{\hat{\sigma}_0}), \end{aligned} \tag{9.15}$$

where

$$\begin{aligned} Q(x) &= \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2) dt \\ &= \frac{1}{2}[1 - erf(\frac{x}{\sqrt{2}})] \\ &= \frac{1}{2} erfc(\frac{x}{\sqrt{2}}) \\ &= 1 - \Phi(x), \end{aligned} \tag{9.16}$$

and $\Phi(x)$ is the cumulative distribution function (CDF) for a standard normal distributed variable, and $Q(x)$ is the complementary cumulative distribution function (CCDF). Then, we can derive the expression of γ'

$$\gamma' = \hat{\sigma}_0 Q^{-1}(P_{fa}) + \hat{\mu}_0. \tag{9.17}$$

Here, we can see the PFA is not a function of the amplitude A or the variance $\vec{w}_1$, which means PFA will not be affected by the signal amplitude or the variance under $\mathcal{H}_1$.

The theoretical PD can be obtained from Equation (9.4)

$$\begin{aligned} P_d &= Pr(T(\vec{x}) > \gamma'; \hat{n}_0, A, \mathcal{H}_1) \\ &= Pr(T'(\vec{x}) > \frac{\gamma' - \hat{\mu}_1}{\hat{\sigma}_1}; \hat{n}_0, A, \mathcal{H}_1) \\ &= Q(\frac{\gamma' - \hat{\mu}_1}{\hat{\sigma}_1}) \\ &= \frac{1}{2} erfc(\frac{\gamma - \hat{\mu}_1}{\sqrt{2}\hat{\sigma}_1}). \end{aligned} \tag{9.18}$$

Plugging Equation (9.17) in Equation (9.18), we have

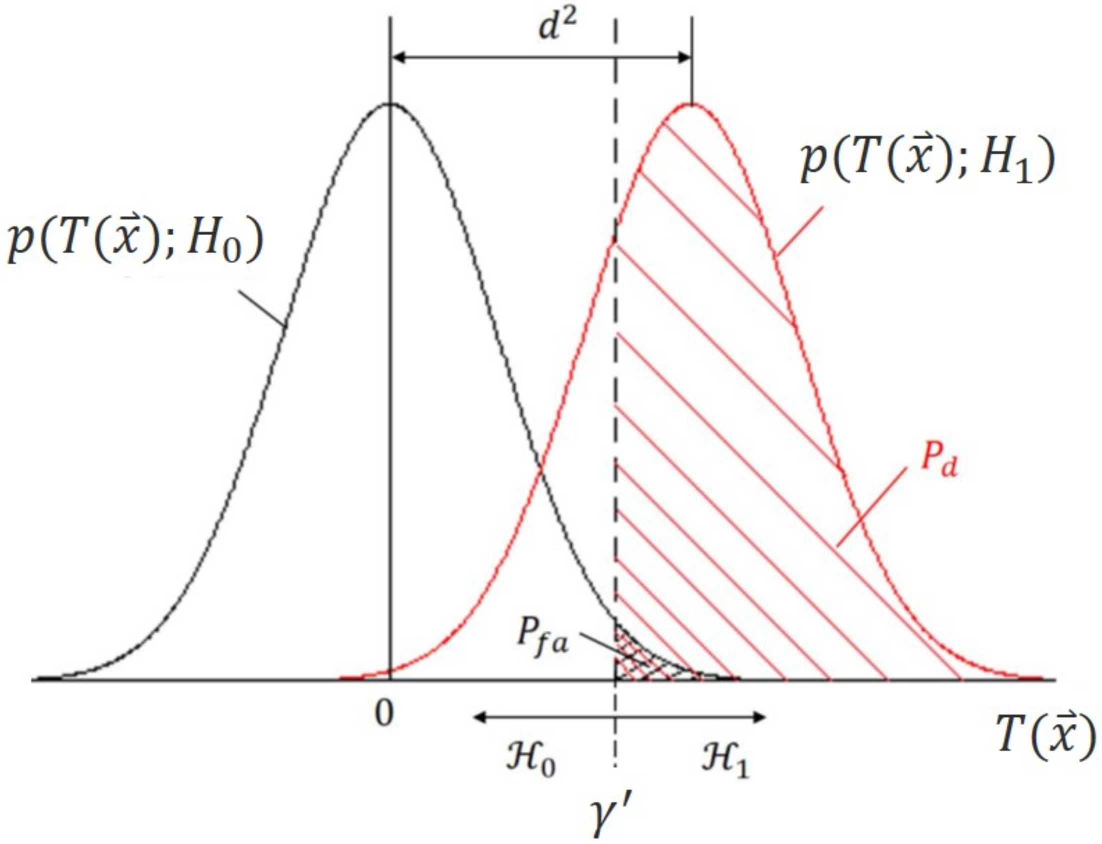


Figure 9.2: Schematic of the PDF's of $T(\vec{x})$ under $\mathcal{H}_0$ and $\mathcal{H}_1$ and the Deflection Coefficient d^2. (Figure is adapted from Kay (1998)).

$$P_d = Q(Q^{-1}(P_{fa}) - \sqrt{d^2}) \tag{9.19}$$

$$d^2 = \frac{A^2 s^2[n]}{W}, \tag{9.20}$$

where d^2 is the deflection coefficient, which can be interpreted as the separation between the PDF of $T(\vec{x})$ under $\mathcal{H}_1$ and that under $\mathcal{H}_0$. The deflection coefficient is illustrated in Figure 9.2. From the figure, we can see if the deflection coefficient is large, the position of the threshold has less impact on the PD, and the detector is approaching a perfect detector ($P_d = 1$). However, the deflection coefficient is a function of the amplitude A and variance $W$of a signal, which means the smaller amplitude and larger variance could move the two PDF's closer which causes a decrease of PD.

9.1.5 Procedure of the NP Detector

Now, we summarize the procedure of using a NP detector for the signal detection and estimation:

1. find the kernel $s[n]$;
2. find the variance of noise $x[n]$;
3. calculate the PDF's of the test statistic $T(\vec{x})$ (Equation (9.13));
4. calculate the threshold γ' for $T(\vec{x})$, given a predefined PFA (Equation (9.17));
5. calculate $T(\vec{x})$ over all possible n_0 using SW approach or FFT approach (Equation (9.8)). Find T_{max} and the corresponding $\hat{n}_0$ (Equation (9.10));
6. compare T_{max} with γ', and decide the hypothesis of the signal (Equation (9.9));
7. if $\mathcal{H}_1$, estimate the arrival time $\hat{n}_0$ (Equation (9.10)).

9.2 Application to Simulated Signals

In this section, the application of the NP detector to noise and pulse signals was studied. The experiment design will be introduced first followed by the results and discussion.

9.2.1 Study Cases

A case with pure noise signals and one with noisy return pulses were studied in this work. For both cases, the ADC threshold was set 3x noise floor above zero, and the edge-trigger model with a precursor length of 1 and the collection length of 5 was used. More details of the two cases are explained below.

Case 1: noise (laser off)

For the case of noise, 1000 observations of pure noise signals (with the laser turned off) were generated by the receiver model with a sampling rate of 100 GHz as suggested in Section 3.3.

Case 2: pulse

Return signals reflected from targets at different distances were tested for the case with the laser on. The distance ranges from 5 m to 60 m with an increment of 5 m and the corresponding SNR ranges from 44.4 dB to 1.27 dB. For each distance, 1000 return signals were generated by the pulse model, the propagation model, and the receiver model. The process of moving averaging and subsampling is the same as in Case 1.

9.2.2 NP Detector

The NP detector was used for signal detection and TOF calculation. The arrival time of the return signal was measured by the NP detector and the arrival time of the transmit signal (start time) was measured by a peak detection algorithm. Since the transmit signal is noise-free, the usage of the peak detection will not affect the TOF measurement. The kernel of the NP detector was obtained from the pulse model with the amplitude normalized to one. The PFA is set to 0.001 which is a typical value for a laser range finder (Burns *et al.*, 1991). In Case 1, the variance W is calculated dynamically from each observation, meaning the threshold is dynamic for the NP detector. In Case 2, as mentioned in Section 9.1, the variance W_1 is assumed to be equal to W_0, *i.e.* $W_1 = W_0 = W$. The PDF's of the $T(\vec{x})$ and the threshold can be obtained from Equation (9.13).

9.2.3 Results and Discussion

Threshold

The threshold for the NP detector is determined by the STD of the $\mathcal{H}_0$ signals σ_0^2, and different ways of calculating the STD results in different PFA's. The STD of the noise signal can be obtained from the datasheet of an electronic device or measurement of an oscilloscope or signals collected by an ADC. A comparison of the STD obtained from signals before ADC sampling (analogous to oscilloscope measurements) and after sampling (analogous to signals collected by an ADC) is shown in Figure 9.3. The figure shows the before-sampling signals have a smaller STD compared to after-sampling signals, which results in a lower threshold, and consequently a high PFA. The reason for the difference of the STD is that the ADC only samples a limited number of data points from analog signals, and the variance of the subset of a signal

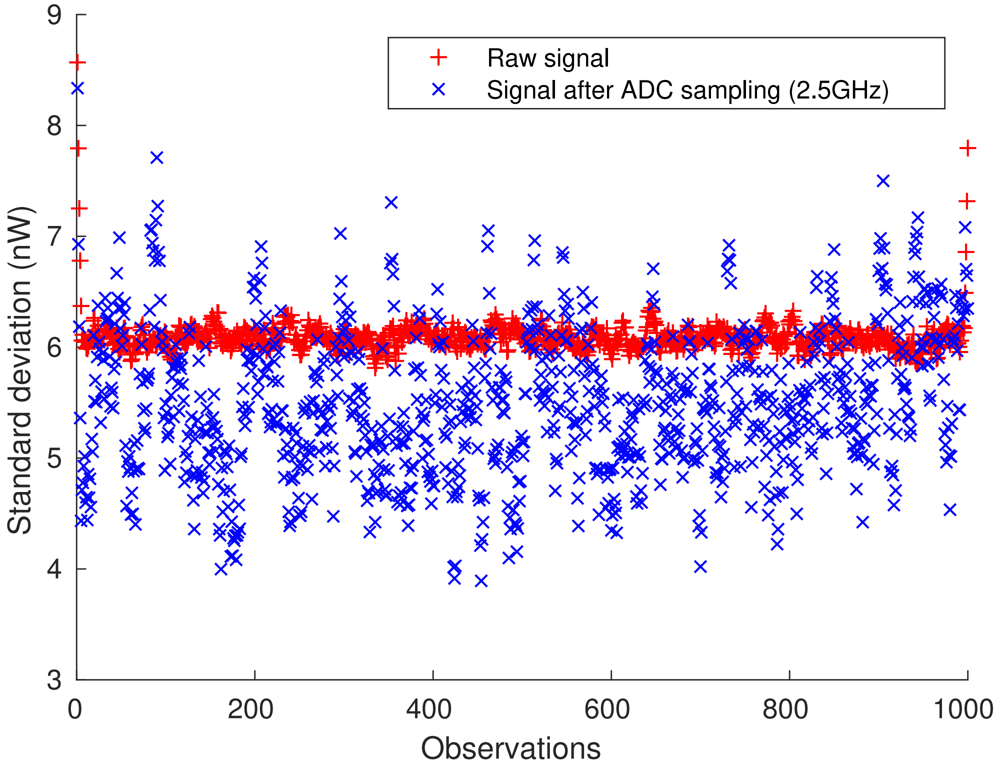


Figure 9.3: The STD of 1000 Noise Signals before ADC Sampling (Red) and after ADC Sampling (Blue).

is usually higher than the entire analog signal. Consequently, the threshold for a given PFA is underestimated by using the STD of before-sampling signals. Therefore, to achieve the given PFA, the STD of the $\mathcal{H}_0$ signals should be calculated from the signals collected by an ADC, rather than before the ADC sampling, from oscilloscope measurements or the datasheet of an electronic device.

Probability of false alarm

A Monte-Carlo simulation was performed on 100 tests, each of which contains 1000 observations of noise (Case 1). The PFA of each test was measured, and the average PFA was obtained by averaging over the PFA's of the 100 tests. The resultant $P_{fa,aveg}$

is equal to 0.08 with an STD of 0.025. The measured PFA is higher than the predefined value (0.001). The reason is described in the following. $T(\vec{x})$ is the convolution of the kernel and the signal at a single position n_0 ($n_0 \in [0, N-M]$) (Equation (9.8)). $T(\vec{x})$ is referred to as 'single pass' in this work and the 'pass' means how many convolutions are made in the sliding window process (taking the SW approach for example). On the other hand, T_{max} is the maximum of $T(\vec{x})$, meaning it results from p passes of $T(\vec{x})$, and p depends on the length of the single and the kernel. The calculated threshold γ' using the predefined PFA is for an LRT on signals with known arrival time $n_0 = \hat{n}_0$. In other words, the threshold is used to compare with a single pass $T(\vec{x})$ rather than T_{max}. However, the arrival time is unknown in our study, so convolutions have to be conducted over the entire signal (p passes). In that case, the random noise on the signal is more likely to exceed the threshold and be detected as a pulse. Kay (1998) gives an estimation of the measured PFA in this situation that, the measured PFA is approximately to be p times the predefined PFA. In our case, the kernel and the signal have the same length of 54 samples, resulting in $p = 54 \times 2 - 1 = 107$. Therefore, we should expect the measured PFA is approximately 0.1 ($p \times 0.001 \approx 0.1$) which is close to our simulation result (0.08).

Probability of detection

All the signals in Case 2 give $P_d = 1$. The reason is that even though the PD is theoretically dependent on the amplitude of the signal Equation (9.19), the deflection coefficient is large, even for low-SNR signals. A signal observation with an SNR of 1.31 dB is shown in Figure 9.4 along with the convolution results and the threshold. In the example, the square root of the deflection coefficient ($d = 15.7$) is much larger than the normalized threshold ($\gamma'/\sigma_0 = 3.09$), which means the PDF of hypothesis $\mathcal{H}_1$ is separated from that of the hypothesis $\mathcal{H}_0$. One can also use the maximum

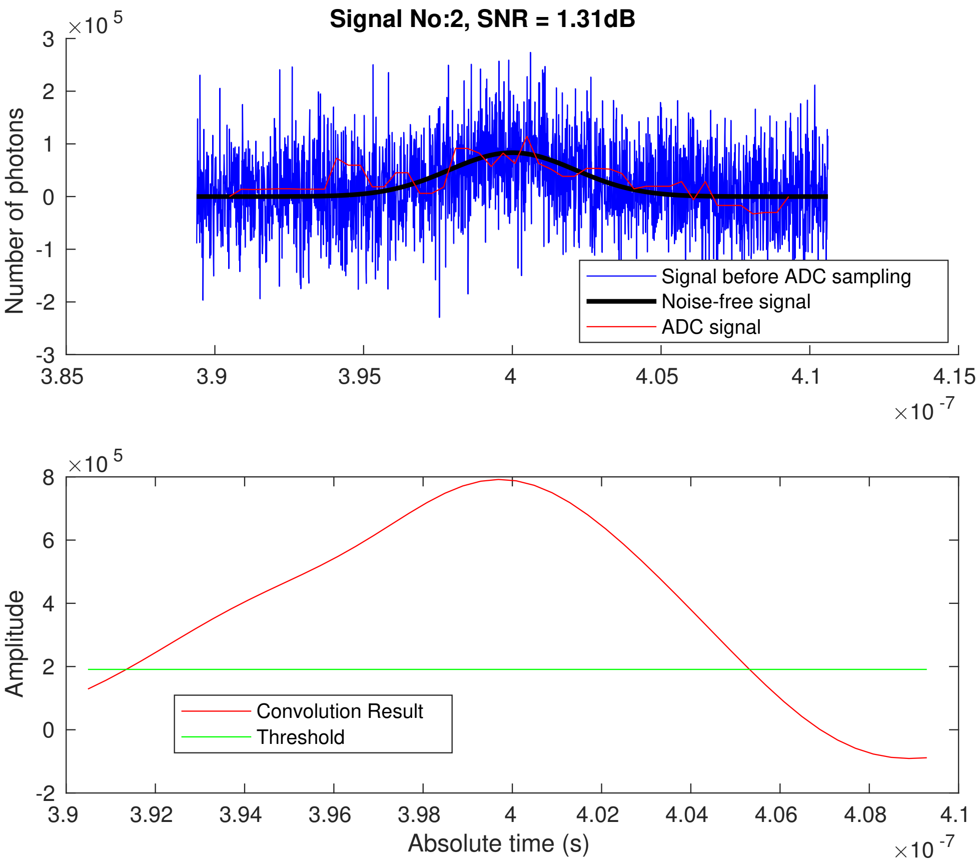


Figure 9.4: Simulated Signal with an SNR of 1.31 dB with the $T(\vec{x})$ and the Threshold. The maximum value of the convolution result is still higher than the threshold.

likelihood estimation (MLE) to estimate the amplitude of the signal to reduce the effect of the amplitude on the PD (Kay, 1998):

$$A = \frac{\sum_{n=0}^{N-1} x[n]s[n]}{\sum_{n=0}^{N-1} s^2[n]} \tag{9.21}$$

However, since the P_d is already equal to 1 in our case, the MLE estimation is skipped, but it could be an option for readers who have situations that have a smaller amplitude, and a high PD is needed.

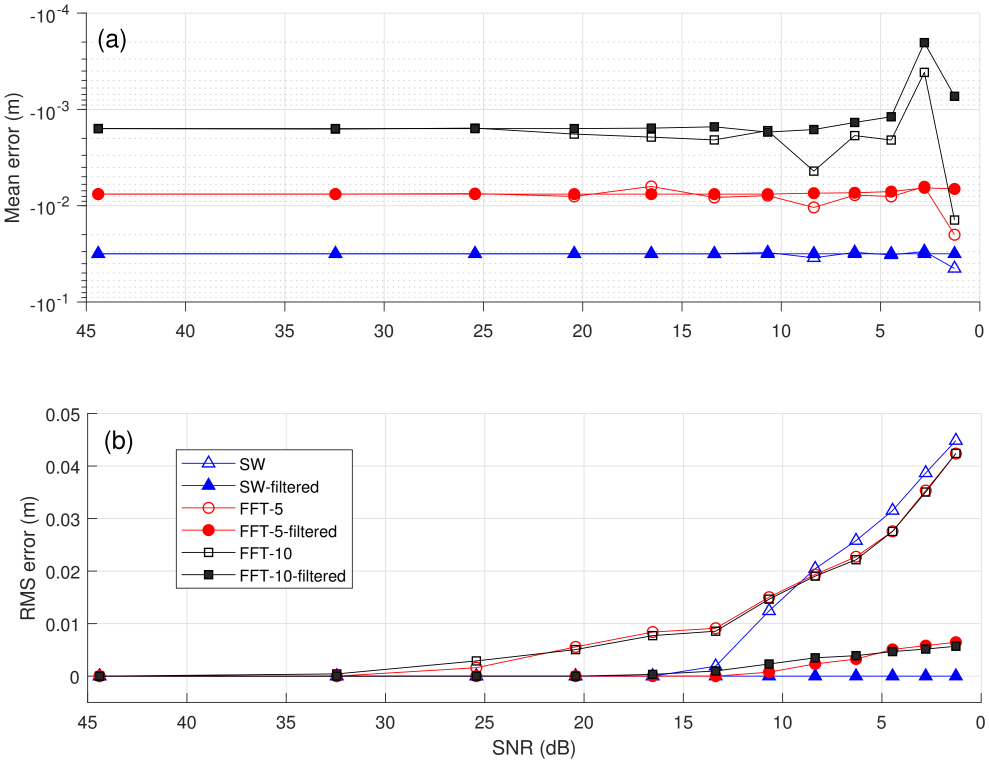


Figure 9.5: Mean and RMS Error of the NP Detector using the SW and FFT Approaches. The number after FFT indicates how many times the resolution is refined. The hollow markesr indicate the results without filtering, and the solid markers denote the results after filtering.

Accuracy and precision

The mean error and the RMS error are calculated to evaluate the performance of the NP detector. Both the sliding-window and FFT approaches were applied. Also, to reduce the effect on the noise on the measurement, a 3rd order Butterworth low-pass filter with a cutoff frequency of 0.4 GHz and a sampling rate of 2.5 GHz was applied on both START and STOP signals, which reduces the phase shift caused by the filter. The results are shown in Figure 9.5. From the result of the mean error, we can see the FFT approaches have a smaller mean error than the SW approach, and a larger number of zero-padding gives even better accuracy. The reason is that the mean error is affected by the resolution of the signal. The sliding-window approach has the coarsest resolution, so any errors caused by the noise is at least 0.4 ns (0.06 m). On the contrary, the FFT-approach refines the resolution 5 and 10 times than the sliding-window approach, which results in a less mean error and the FFT-10 provides the least mean error. Additionally, the low-pass filtering slightly reduces the mean error for long distance. The comparison of the effect of the resolution on the measurement accuracy between different approaches is given in Figure 9.6 and Figure 9.7. From the figure we can see because of the finer resolution, the peak position of FFT-10 is closer to the ground-truth position than FFT-5, and the peak position of the sliding-window approach has the largest separation from the ground-truth.

From the RMS error, we notice that the sliding-window has better precision than

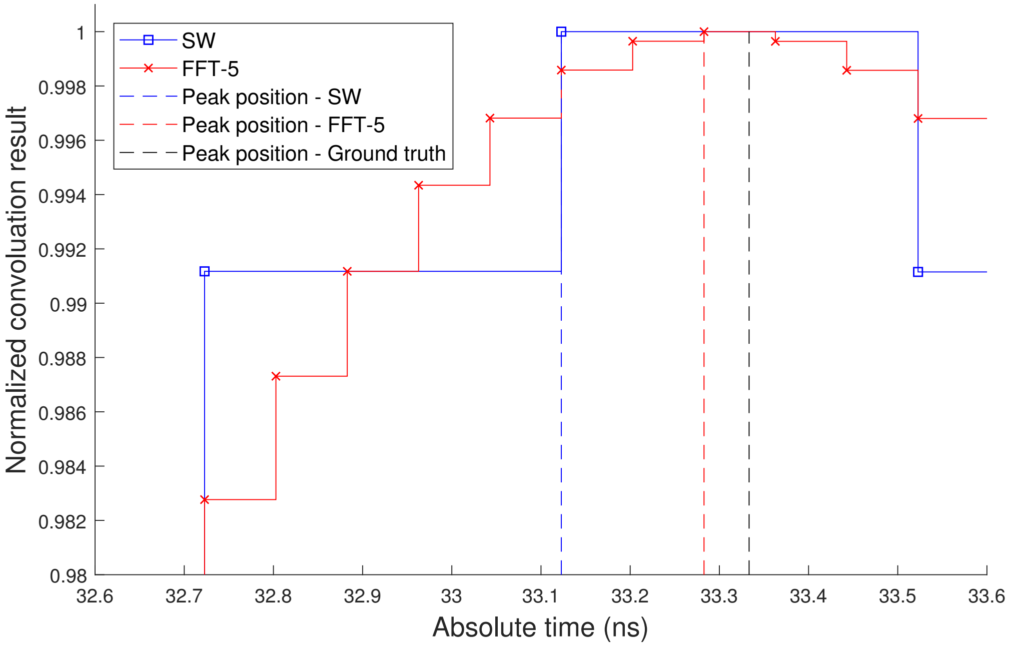


Figure 9.6: Convolution Result of FFT-5 (Zoomed in on the Top).

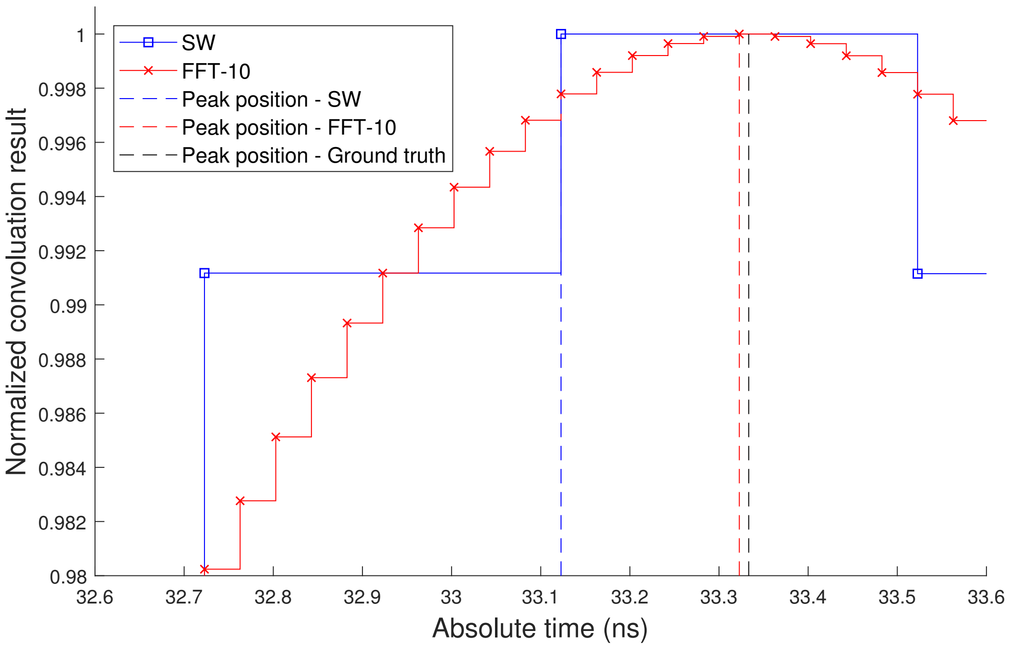


Figure 9.7: Convolution Result of FFT-10 (Zoomed in on the Top).

the FFT-approaches for a shorter distance, but the precision becomes worse as the distance increases. The difference also results from the resolution of the result. At a short distance, the signal is less, and the samples of a signal are more separated for the sliding-window approach than the FFT's, so the peak of the convolution result is less challenged by the adjacent points, which results in smaller variation of the time(distance) calculation. On the contrary, the samples at the top portion of the signal are likely to have similar values for the FFT approaches, due to the smaller separation between the sample points, which results in a larger variation of the peak position, and consequently, larger RMS error. On the other hand, as the distance increases, the signal becomes noisy, so the values at the top portion of the result from the SW-approach also become competitive, which increases the variation of the peak position. Therefore, the larger time interval between the samples of the SW approach than the FFT approaches results in a larger RMS error of the former one. The low-pass filter improve the precision of both approaches (the RMS errors reduce to less than 1 cm). Also, note that the RMS error of the SW approach becomes smaller than the one of the FFT approaches at a longer distance. It is because the low-pass filter reduces the noise of the signal, which makes the adjacent points at the top portion of the result of the SW approach less competitive (similar to the situation for short distance). Therefore, the variation of the peak position of the SW approach decrease.

9.3 Application to Experimental Signals

The NP detector was also applied to real signals collected by the ADC installed on the prototype lidar. The experiment design including the experimental setup will be introduced first in this section, followed by the results and discussion.

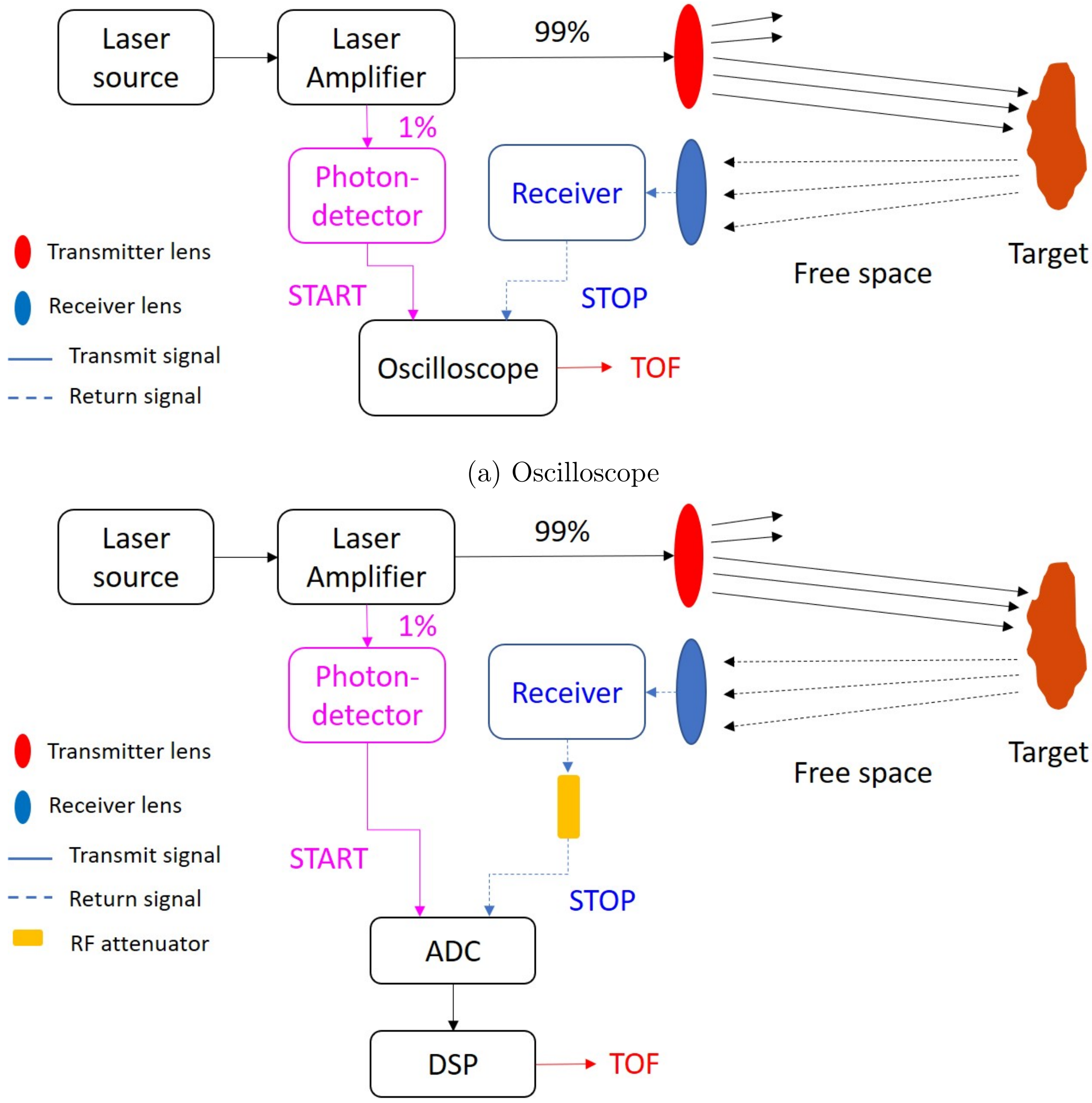
9.3.1 Experimental Setup

The setups for an ADC and oscilloscope experiment are given in Figure 9.8. The RF attenuator for the ADC experiment is to reduce the return voltage to less than the upper limit of the ADC input. In this study, off-line MATLAB scripts serve as the digital signal processor (DSP) in Figure 9.8b. In a real-time mode, the MATLAB scripts would be replaced wit a DSP or onboard ADC programs.

9.3.2 Study Cases

Case 1: noise (laser off)

For Case 1, the ADC threshold was set to 3x the noise floor, and the edge-triggering with precursor length of 5 and the collection length of 40 were used. The settings guarantee sufficient data points are collected in each observation. During the experiment, there are 67, 370 noise signals with laser off that were collected from the START channel of the ADC and 73, 133 noise signals collected from the STOP channel. The



(a) Oscilloscope

(b) ADC

Figure 9.8: Experimental Setups of TOF Measurements using an (a) Oscilloscope (b) ADC.

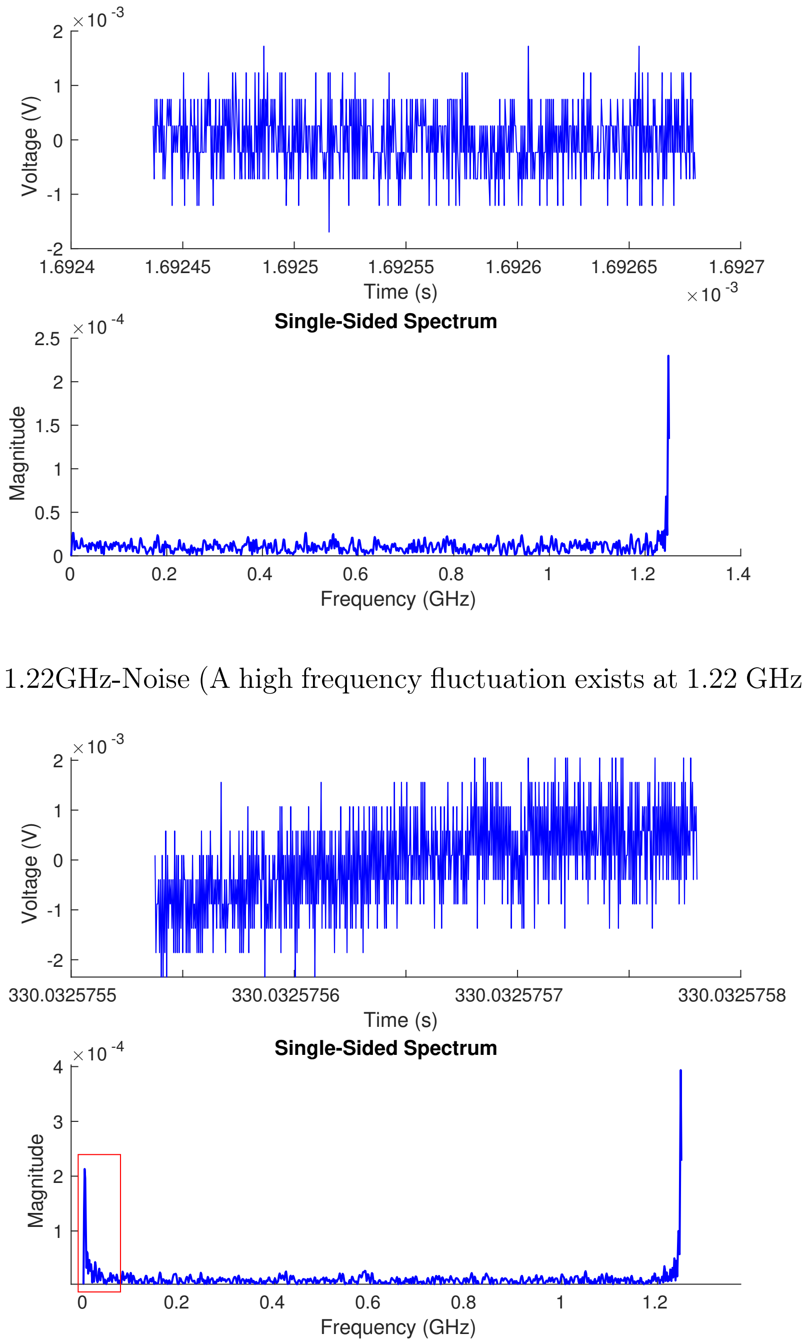
START and STOP channels contain multiple types of noise including white noise, temporal-structured noise, and frequency-structured noise. Examples of the noise in the START channel are given in Figure 9.9 with the corresponding spectrum for which the DC component of the time-domain signal was removed beforehand. Due to the random appearance, the Ramp-Noise I is a temporal-structured noise. The 1.22GHz-Noise is almost white, but it has a high-frequency oscillation at 1.22 GHz.

The STOP channel contains two frequency-structured noises with a frequency of 200 MHz and 645 MHz in addition to the 1.22GHz-Noise. The two frequency-structured noises are named 200MHz-Noise and 645MHz-Noise, respectively. In addition, the STOP channel also has a ramp-like Noise, called Ramp-Noise II. Examples of the 200MHz-Noise, 645MHz-Noise, and Ramp-Noise II are given in Figure 9.10. From the figures, we observe that the 645MHz-Noise has a constant appearance, while the 200MHz-Noise and the Ramp-Noise II occurs occasionally. It means the last two noises overlap with the 645MHz-Noise noise and appear on top of the 1.22GHz-Noise. The average STD's of the noise signals were calculated by averaging over the STD of each observation. The STD's of the START and STOP signals are 0.000921 V and 0.0094 V respectively.

Case 2: pulse

In Case 2, different trigger modes with different lengths were used to collect signals from different target distances. The ADC settings are given in Table 9.1. The ADC thresholds for the START and STOP channels were set differently, because the amplitude of the transmit and return signal is different. The target distances are 1.87 m, 2.11 m, 2.58 m, 3.19 m, 3.66 m, 4.80 m and 5.08 m. For each distance, 20 files each of which contains more than 1000 observations were collected.

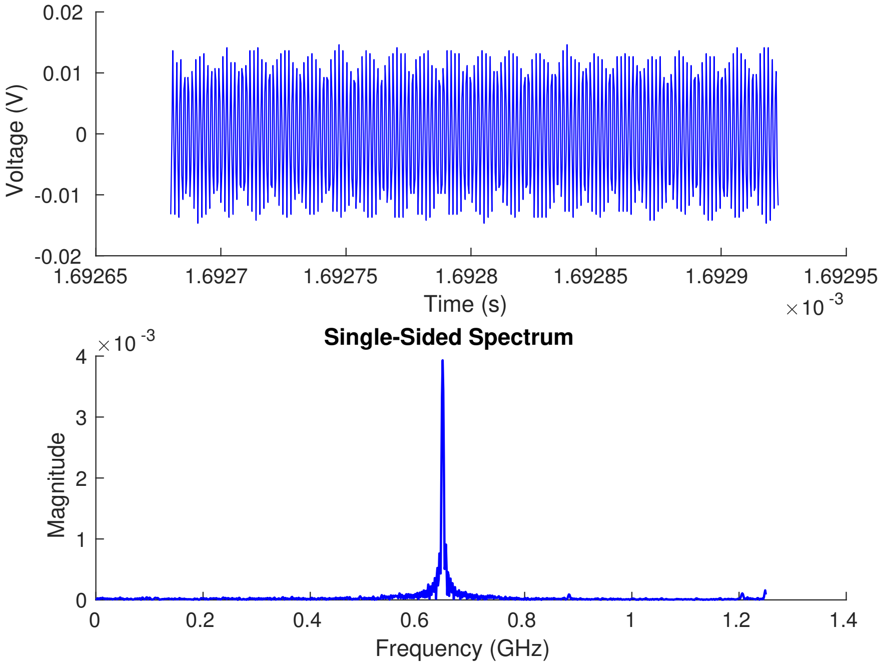
The signals collected at the START and STOP channels were contaminated by the



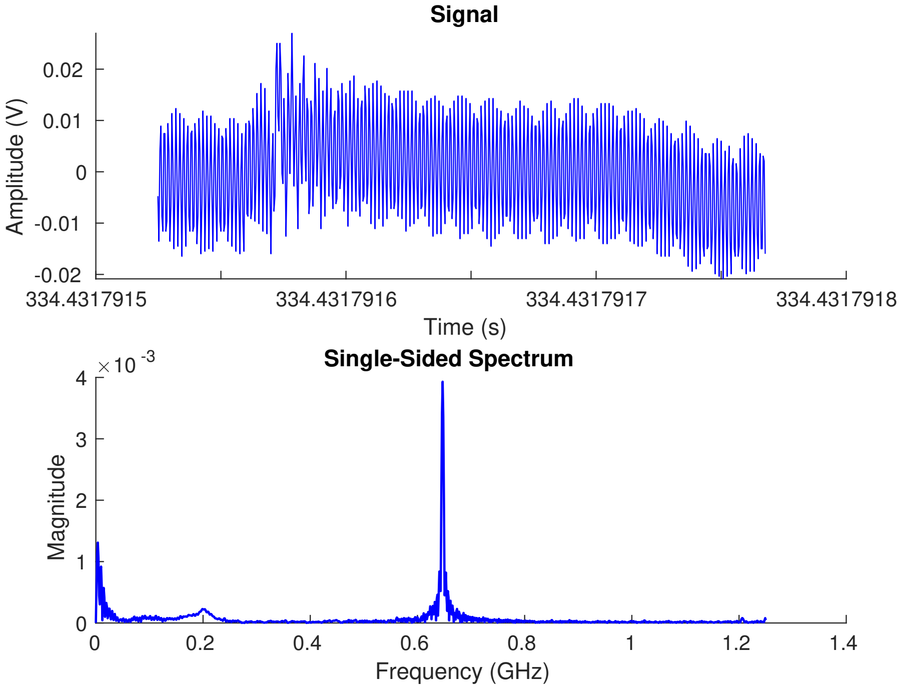
(a) 1.22GHz-Noise (A high frequency fluctuation exists at 1.22 GHz.)

(b) Ramp-Noise I + 1.22GHz-Noise

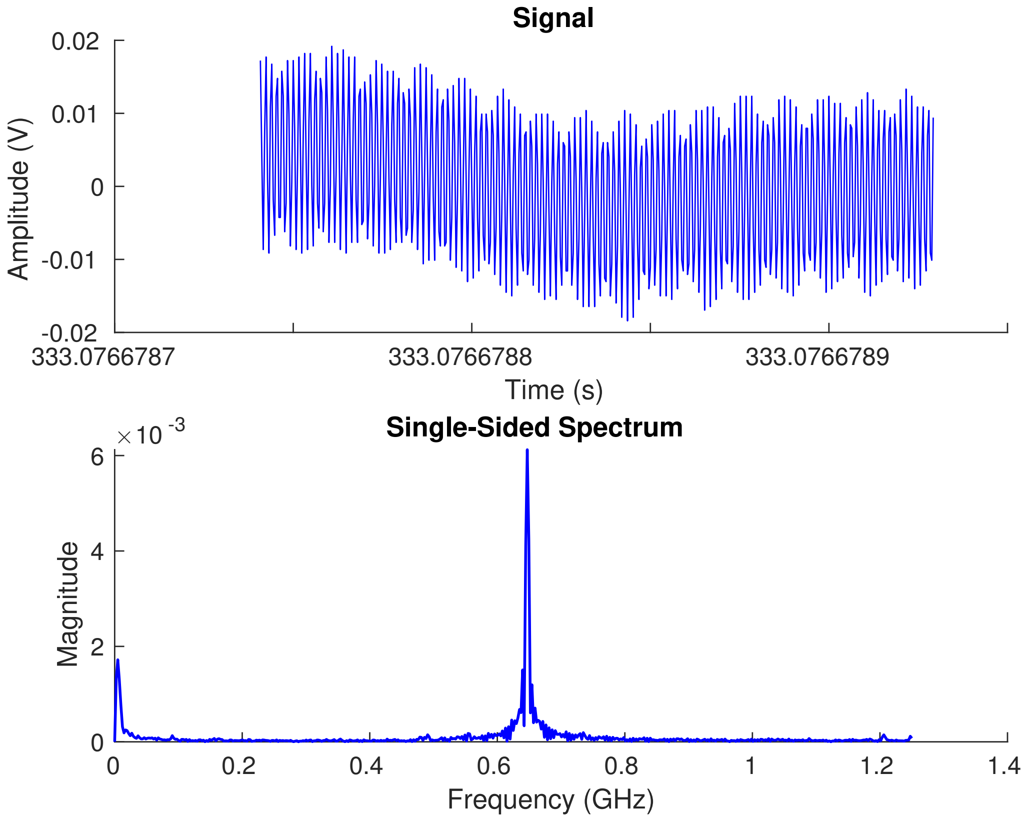
Figure 9.9: Noises in the START Channel. Top: signal; bottom: single-sided spectrum.



(a) 645MHz-Noise + 1.22GHz-Noise



(b) 645MHz-Noise + 200MHz-Noise + 1.22GHz-Noise



(c) 645MHz-Noise + Ramp-Noise II + 1.22GHz-Noise

Figure 9.10: Noises on the STOP Channel. Top: signal; bottom: single-sided spectrum.

noises mentioned in Case 1. Therefore, similar in Section 9.2.3, the same low-pass filter was applied to the START and STOP signals, and the cut-off frequency is determined according to the spectrum of the frequency-structured noises (Figure 9.10).

9.3.3 Detection and Estimation Algorithm

The NP detector was applied for the signal detection and TOF estimation, as well as a peak estimation algorithm for comparison. The centroid detection and estimation algorithm were not applied in this experiment because the non-Gaussian shape of

Table 9.1: ADC Settings

Trigger mode	Edge		Level
ADC threshold (START channel)	530 (8.789 mV)		
ADC threshold (STOP channel)	600 (0.043 V)		
Precursor (bins)	5	25	5
Collection-length (bins)	40	50	-
Post-cursor (bins)	-	-	40
Total length (bins)	46	76	46+s[1]
Physical length (ns)	147.2	243.2	147.2+ 3.2s

the signal (Figure 9.18) makes the algorithm less practical. The peak estimation algorithm utilizes the timestamp of the peak of the transmit and return signals as the arrival time, and the TOF is equal to the difference between the times. The TOF of the signals collected by the oscilloscope was also calculated using the peak estimation method for comparison.

For the NP detector, the STD of the noise in Case 1 was measured in the same way as for the simulated signals, meaning the threshold was adjusted dynamically, and the σ_1^2 was assumed to be equal to σ_0^2 for Case 2. The PFA for Case 1 and 2 were both set to 0.001. For the real signal experiment, the kernel for the real signals was not obtained from the pulse model as did in the simulated signal experiment, since a longer falling edge of the collected transmit and return signal are observed (Figure 9.18), which makes the signal shape non-Gaussian. One should note that the pulse transmitted from the laser source is still Gaussian, and the long falling edge is due to the slow RC response of the detector. Also, the values of the resistance (R) and

[1] Variable s is signal dependent.

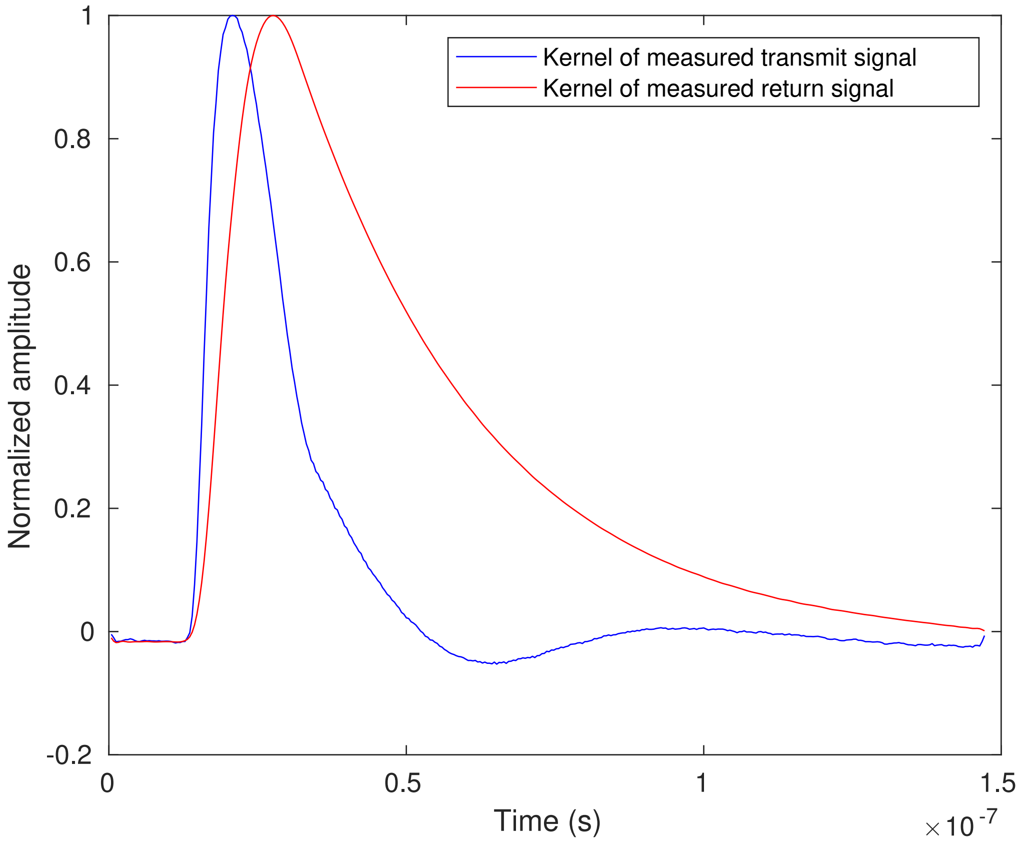


Figure 9.11: Kernels of the Transmit and the Return Signals. Their amplitudes are normalized to one.

the capacitance (C) of the detector are not available, which means the kernel cannot be modeled by a Gaussian pulse model or an RC response. Therefore, we obtained the kernel from the measurements. We first filtered the collected transmit and return signals by the low-pass filter to remove the 645MHz-Noise and then averaged the signals over 1000 observations of the transmit and return signals, respectively, to reduce the random noise. We also checked the signals to ensure the 200MHz-Noise contaminated no signals. Then, we used the averaged filtered signals as the kernels of the respective transmit and return signals, which are shown in Figure 9.11.

9.3.4 Results and Discussion

Trigger modes

For the trigger modes, both the level-triggering and edge-triggering with different collection-lengths give the same results using different detection algorithms. The reason is that all the trigger modes were able to capture the top of the signal where the TOF is measured from, and the only difference is the length of the leading and falling edge, which does not influence the TOF measurement. Therefore, only the results of signals collected by the edge-triggering with a precursor of 25 and a collection-length of 50 is discussed in this section.

Probability of false alarm

In Case 1, both the signals before and after filtering were used for the PFA measurement. The results are shown in Table 9.2. The results show that the NP detector has poor performance ($P_{fa} \approx 20\%$) on the START signal and applying the low-pass filter does not make an improvement. The high PFA and large STD are due to Ramp-Noise I. The Ramp-Noise I occurs randomly, which makes it difficult to include its fluctuation in the STD measurement and consequently, the threshold was underes-

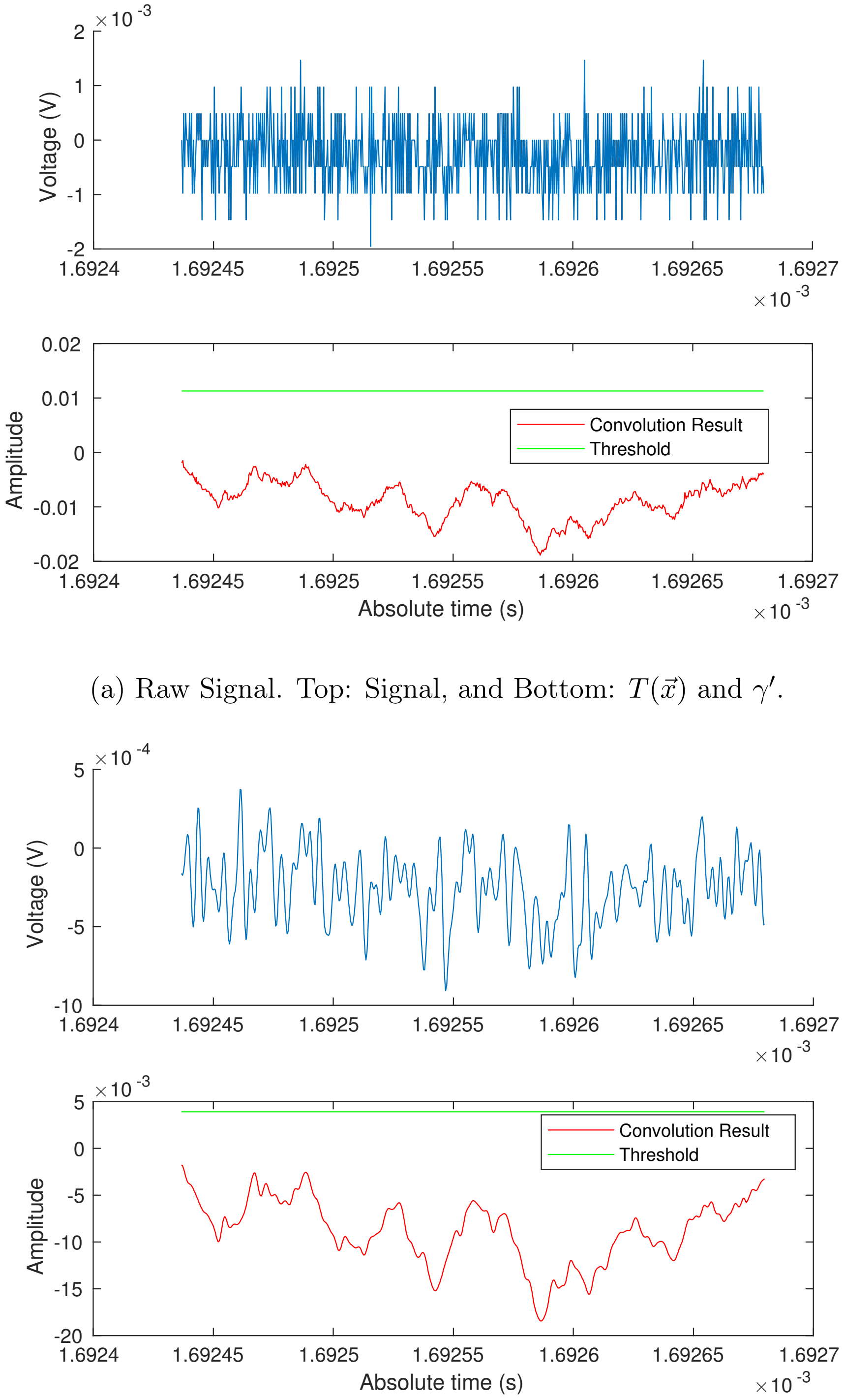
Table 9.2: PFA of Signals with and without Low-Pass Filtering

Channel	**Noise**	**Filtering**	**Measured P_{fa}**
START	1.22GHz-Noise, Ramp-Noise I	No	0.207 ± 0.208
		Yes	0.201 ± 0.192
STOP	1.22GHz-Noise, Ramp-Noise II	No	0.0052 ± 0.0057
	645MHz-Noise, 200MHz-Noise	Yes	0.2306 ± 0.16

timated. In that case, when the Ramp-Noise I occurs, the convolution results can easily exceed the threshold and the Ramp-Noise I was identified as a pulse. Moreover, since the frequency of the Ramp-Noise I is inside the bandwidth of the pulse, it is not possible to filter out using a low-pass filter without distorting the pulse. Examples of the START signals having the 1.22GHz-Noise and the Ramp-Noise I are given in Figure 9.12 and Figure 9.13 along with the convolution result $T(\vec{x})$.

On the other hand, the NP detector was also challenged on the STOP signals by the Ramp-Noise II, 645MHz-Noise and 200MHz-Noise. In the STOP signal, the 645MHz-Noise and the 200MHz-Noise have strong amplitude while the Ramp-Noise II is buried inside the two noises. Therefore, before the low-pass filtering, the PFA is affected majorly by the first two noises. The NP detector behaved well on the pure 645MHz-Noise without other noises overlapped. The convolution results of 1112 signals and the corresponding thresholds are shown in Figure 9.14a. In the figure we can see that the convolution results of most of the signals contaminated by the pure 645MHz-Noise are below the threshold which is because of the consistent appearance of the 645MHz-Noise. The dynamic STD calculation was able to measure the fluctuation, so that the NP detector can adjust the threshold accordingly and distinguish the noise successfully. Examples of the signals contaminated by the pure 645MHz-Noise and the corresponding convolution results and thresholds are given in Figure 9.14b.

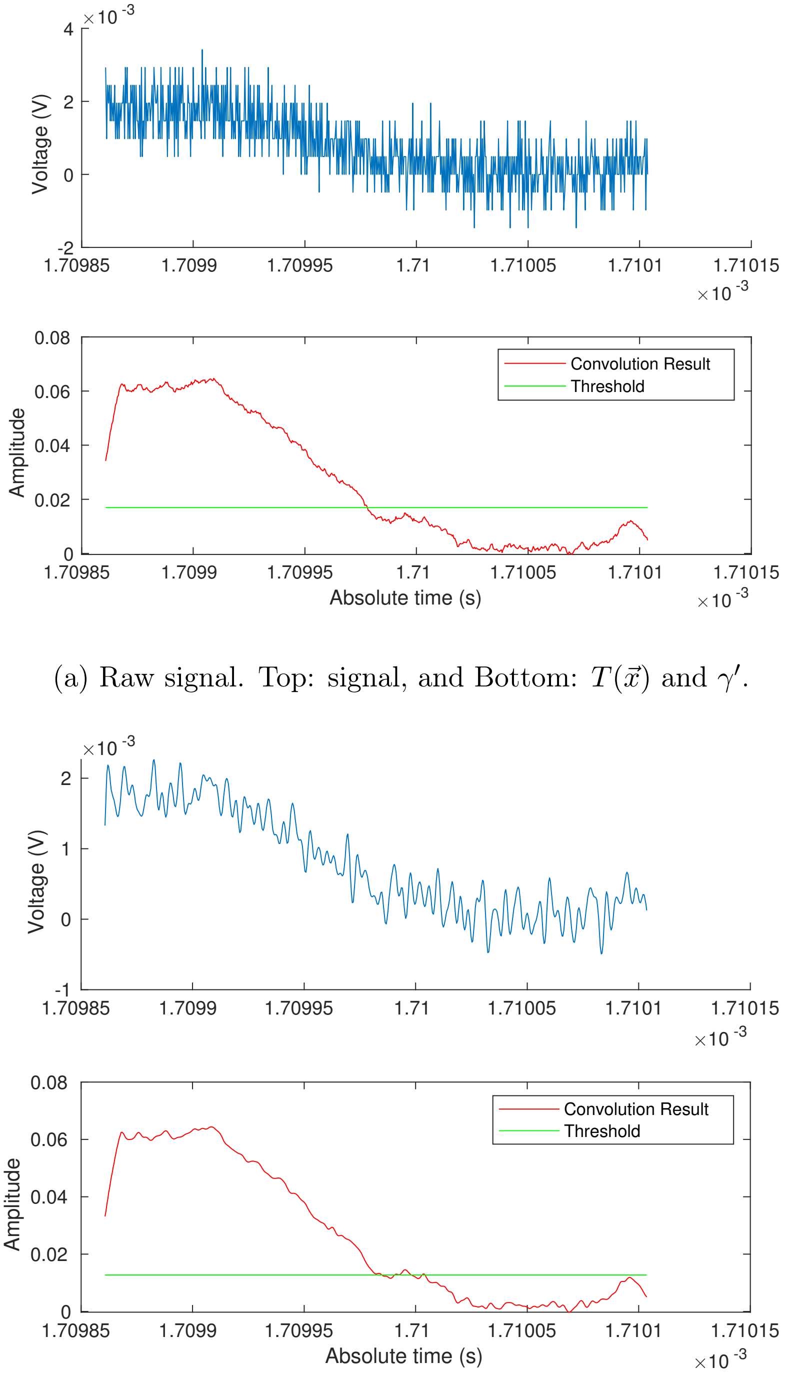
However, the NP detector fails on the 200MHz-Noise. An example is shown in Figure 9.15 in which the No. 664 observation was contaminated by both the 200MHz-Noise and the 645MHz-Noise. The peak of the convolution result exceeds the threshold due to the high amplitude of the 200MHz-Noise and its random appearance, which makes it difficult to include the large amplitude in the STD measurements. Also, the failure of the NP detector on the 200MHz-Noise results in the dramatic variation of PFA among different files.



(a) Raw Signal. Top: Signal, and Bottom: $T(\vec{x})$ and γ'.

(b) Filtered Signal. Top: Signal, and Bottom: $T(\vec{x})$ and γ'.

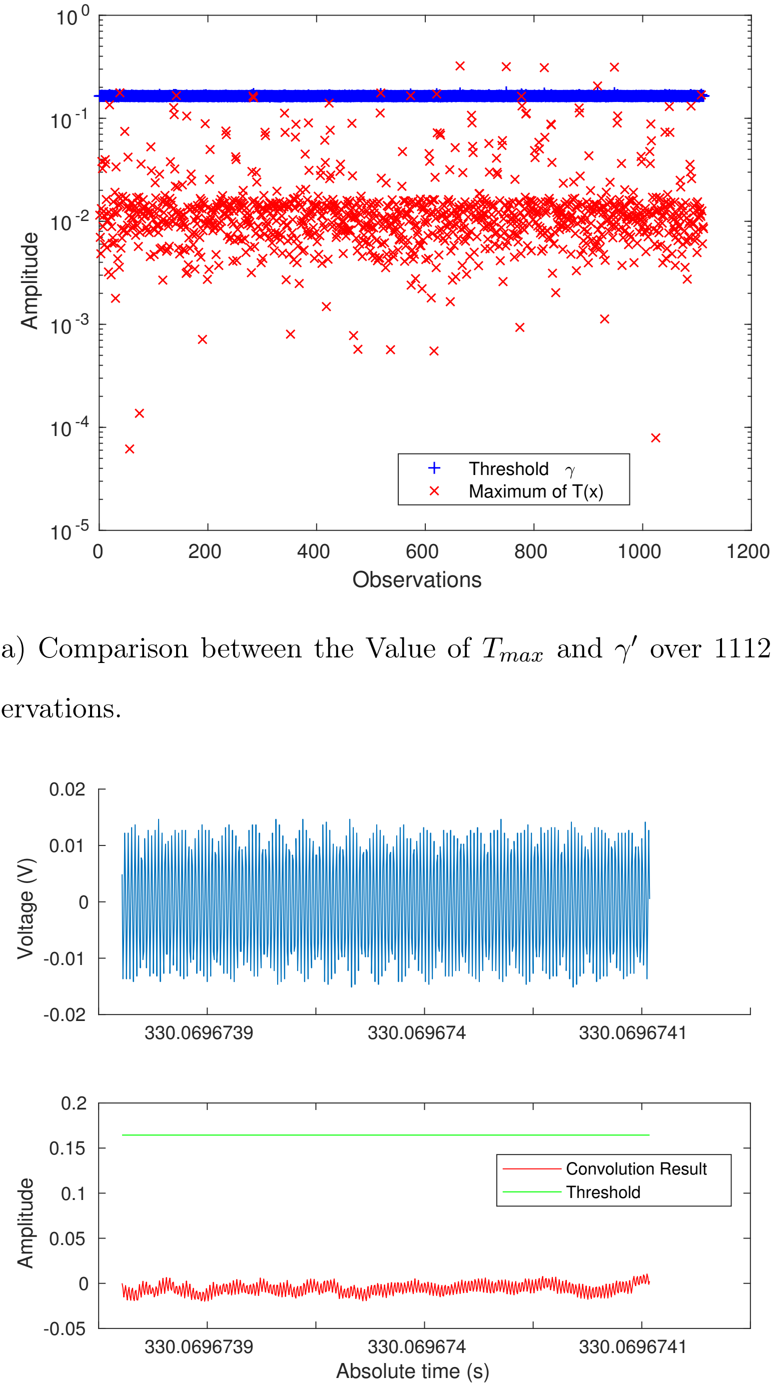
Figure 9.12: 1.22GHz-Noise in the START Channel in Case 1 (Observation No.1).



(a) Raw signal. Top: signal, and Bottom: $T(\vec{x})$ and γ'.

(b) Filtered signal. Top: signal, and bottom: $T(\vec{x})$ and γ'.

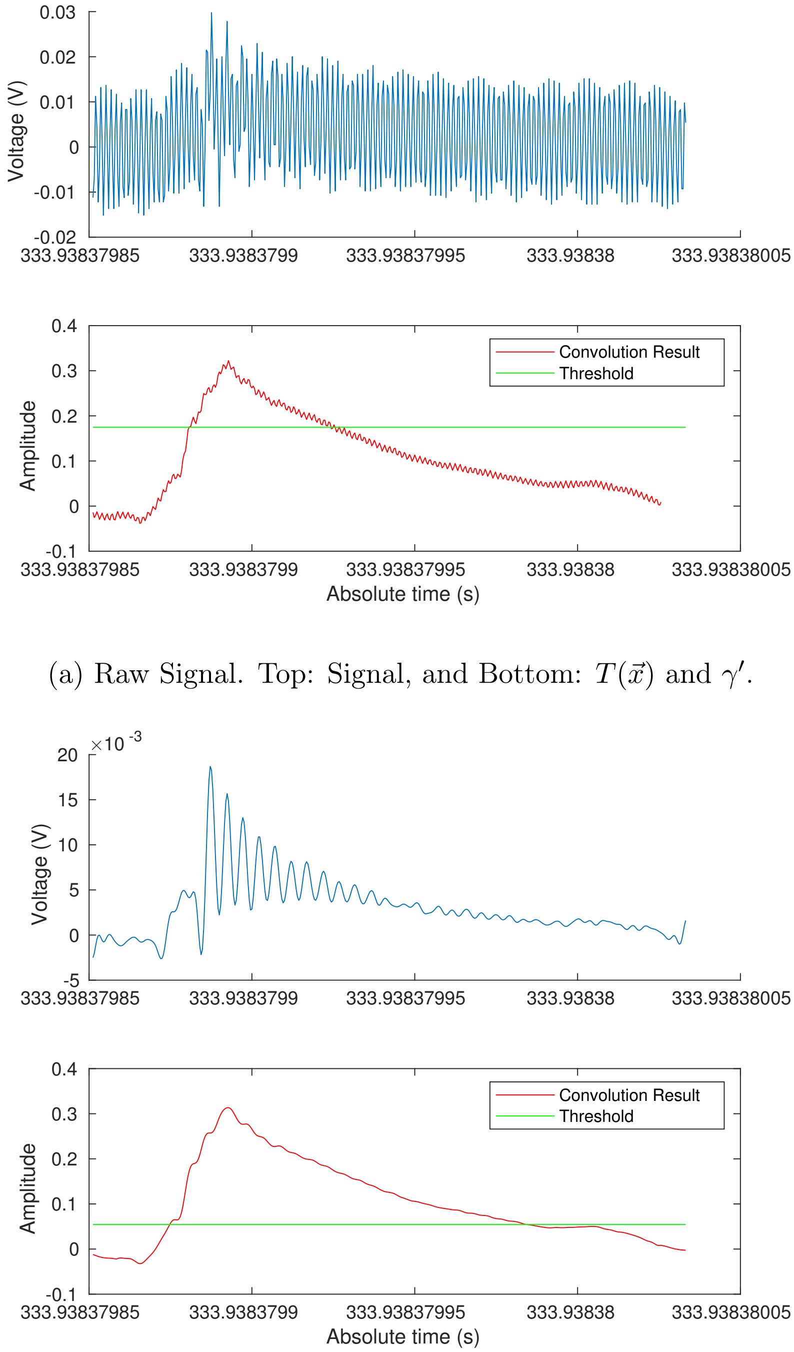
Figure 9.13: Ramp-Noise I + 1.22GHz-Noise in the START Channel in Case 1 (Observation No.70).



(a) Comparison between the Value of T_{max} and γ' over 1112 Observations.

(b) Top: Raw Signal (Observation No.1); Bottom: $T(\vec{x})$ and γ'.

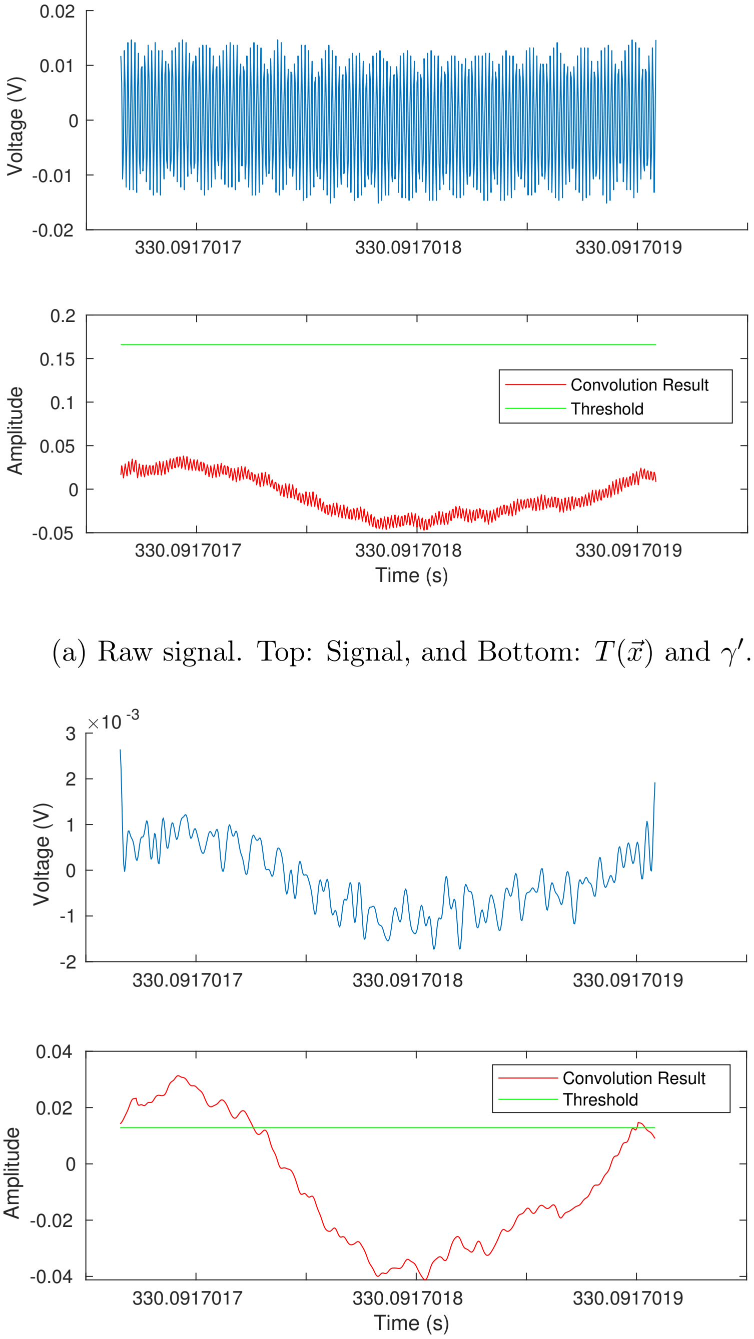
Figure 9.14: 645MHz-Noise + 1.22GHz-Noise in the STOP Channel in Case 1.



(a) Raw Signal. Top: Signal, and Bottom: $T(\vec{x})$ and γ'.

(b) Filtered signal. Top: Signal, and Bottom: $T(\vec{x})$ and γ'.

Figure 9.15: 200MHz-Noise + 645MHz-Noise + 1.22GHz-Noise in the STOP Channel in Case 1 (Observation No.664).



(a) Raw signal. Top: Signal, and Bottom: $T(\vec{x})$ and γ'.

(b) Filtered Signal. Top: Signal, and Bottom: $T(\vec{x})$ and γ'.

Figure 9.16: Ramp-Noise II + 645MHz-Noise + 1.22GHz-Noise in the STOP Channel in Case 1 (Observation No.5).

The Ramp-Noise II slightly affects the PFA without filtering, since the Ramp-Noise II is buried inside the 645MHz-Noise. However, when the 645MHz-Noise is removed by the low-pass filter, the Ramp-Noise II starts to influence the NP detector, which causes an increase of PFA. An example of the Ramp-Noise II buried inside the 645MHz-Noise and the noise after the low-pass filter is given in Figure 9.16. From the figures, we can see that it is difficult to observe the Ramp-Noise II before filtering, but it becomes obvious after the low-pass filter and difficult to be distinguished from a laser pulse by the NP detector.

Probability of detection

Signals from different distances in Case 2 were tested by the NP detector and $P_d = 1$ is achieved for all the signals. The same reason for the PD on simulated signals can be applied here, so it is skipped in this section.

Accuracy and precision

The mean error and the RMS error are calculated by using the peak estimation and the SW and FFT approaches for the NP detector on the ADC measurements. The distance errors from the oscilloscope measurement were also calculated. The signals from different trigger modes were examined for the TOF measurements as well.

Mean error The results of the mean error from different estimation algorithms are shown in Figure 9.17. All the estimation methods and oscilloscope measurements give a positive bias of the mean distance from the ground truth distance (*i.e.* longer distance than the ground-truth distance), which could be due to the inaccurate measurement of the ground truth. More evidence is given by the constant discrepancy between the distance measurements of the oscilloscope and the ground-truth distance.

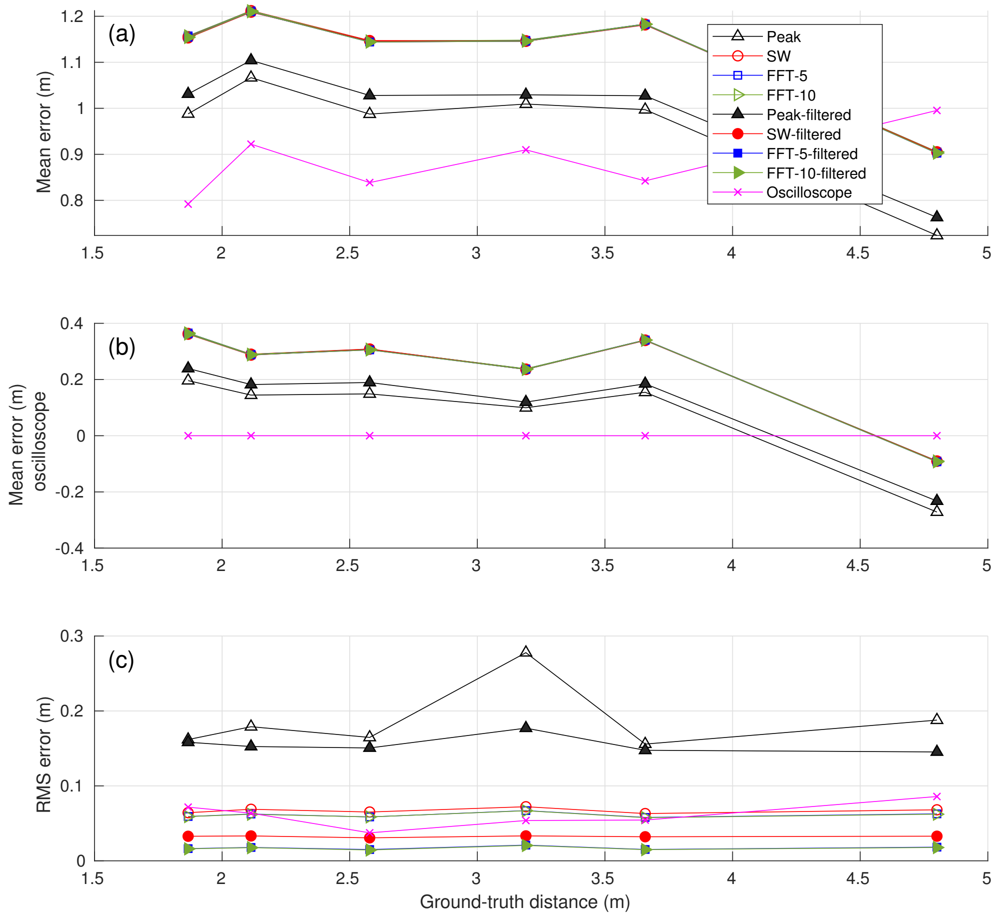


Figure 9.17: (a) Mean Error and (b) RMS Error of the NP Detector using the SW and FFT Approaches. The number after FFT (*e.g.* FFT-5) indicates the times the resolution is refined. The hollow markers indicate the results without filtering, and the solid markers denote the results after filtering. (b) is obtained by using the distance of signals collected by the oscilloscope as the reference for the mean error calculation.

Therefore, we used the distance measured by the oscilloscope as the ground-truth in this case, and the resultant mean error is shown in Figure 9.17b.

However, the positive bias still exists even though it is reduced after changing the reference, and the peak estimation gives less mean error than the NP detectors. The positive bias could be due to the noise on the peak and the long falling edge of the measured signals. The noise could make the peak of the ground-truth signal less detectable, and the long falling edge plateaus the top of the signal, which tends to move peak towards the falling edge. The different slopes of the falling edge of the transmit and return signal could also cause the shift of the peak. Specifically, for the NP detectors, in addition to the flat top of the convolution results, the convolution with asymmetric kernels and the mismatch of the shape of the kernel with each signal could be the reasons for the shift. Consequently, the right-shift of the peak will result in a larger TOF and longer distance measurement.

The peak estimation algorithm has less mean error than the NP detectors, which could be due to that the peak estimation algorithm is only sensitive to the noise on the peak. On the other hand, the NP detectors are dependent on the noise on the whole signal, and the noise on the falling edge of the signal could cause a larger right shift of the convolution result.

The low-pass makes the mean error worse than the one without filtering. The reason could be as the noise decreases, the effect of the long falling edge on the algorithm is larger than the noise at the peak which results in a larger right shift of the peak. On the other hand, different NP detectors with and without low-pass filtering give similar mean errors (the difference is less than 1 cm), which demonstrates effects of the noise on the peak shift of the results of the NP detector. It also shows that the mean error is dominated by the noise on the signal rather than the resolution of the results even after filtering.

RMS error The RMS errors from different TOF estimation algorithms are shown in Figure 9.17c. The reduction of the RMS error after low-pass filtering demonstrates that the capability of the low-pass filter on the removal of the out-of-band frequency-structured noise (645MHz-Noise) and improvement of the measurement precision. The RMS error of the peak estimation is worse than the NP detector since the peak estimation is highly sensitive to the noise at the peak of a signal. On the other hand, the NP detectors weight the noise at the peak by a kernel, which reduces the effect of noise. Among different approaches of the NP detector, the FFT-approach has less RMS error than the SW-approach. The reason could be that the signal is relatively noisy even after low-pass filtering (similar to the long-distance situation in the simulated signal (Section 9.2.3). Therefore, even though the variation of the peak position of the convolution result exists in both SW and FFT approaches, the FFT approaches with a smaller sample separation provide less RMS error.

In the figure, we also observe the RMS error of the oscilloscope measurements is better than the peak estimation method. The reason could be that the oscilloscope has a bandwidth of 50 MHz, which works like a low-pass filter and cuts off noise with frequencies higher than 50 MHz, while the cut-off frequency of the applied low-pass filter is 0.4 GHz. Therefore, the signal output from the oscilloscope is less noisy than the one after the low-pass filter, which results in a lower RMS error.

From the previous discussion, we can see the noise at the peak of a signal needs to be treated carefully for the peak estimation algorithm and the NP detector, and a low-pass filter is a powerful way of removing any out-of-band frequency-structured noise and improve the measurement precision. Moreover, an ADC with a high sampling rate is required to achieve higher precision. On the other hand, to achieve a high measurement accuracy, asymmetric signals or kernel should be avoided which requires a faster detector that has a fast RC response and allows symmetric output signals.

9.4 Comparison between Benchmark Algorithm and NP Detector

After the description of the benchmark and NP signal detection and estimation algorithm, their advantages and shortcomings are discussed in this section. The algorithms were compared from the aspects of measurement accuracy and precision, the probability of detection and probability of false alarm and their application limitations.

From the comparison of the mean error of the benchmark algorithm and the NP detector, the FFT approach of the NP detector shows the most accurate TOF measurements even at very low SNR. The benchmark detection with level-triggering also provides decent accuracy at high SNR, but the accuracy decreases at low SNR. On the other hand, the SW approach of the NP detector can only resolve the arrival time larger than the signal time interval. In terms of precision, the benchmark algorithm provides the worst precision due to its sensitivity to noise, while the SW and FFT approaches in the NP detector give a better precision especially at low SNR. The application of a low-pass filter also improves the precision.

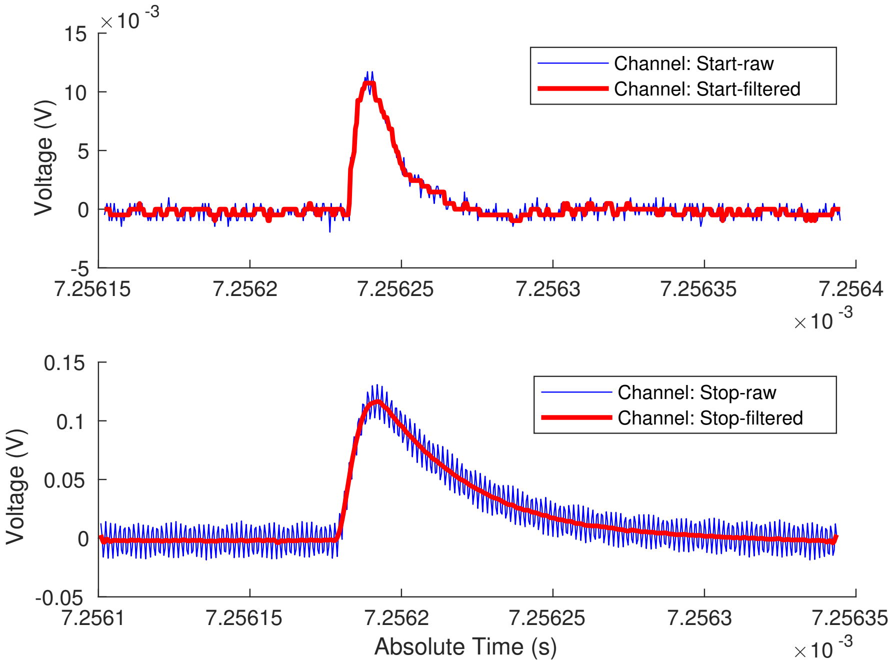
In terms of signal detection, the benchmark algorithm provides a lower PFA than the NP detector on random noise. The reason is that the benchmark algorithm is almost immune to white noise that has either Gaussian distribution or random strong spike. However, the NP detection is highly dependent on the measurement of the noise floor. If the noise is not well characterized, the NP detector fails. Moreover, both the benchmark algorithm and the NP detectors fail on poor characterized structured noise. The detection of $\mathcal{H}_1$ signals is opposite. The NP detector has perfect detection of weak signals of interest. However, the pulse width calculation of the benchmark algorithm is highly sensitive to noise, so the calculated pulse width is likely not to stay within the criteria at low SNR, which degrades the capability of detection of weak signals.

The benchmark and the the NP detector have different requirements to achieve a desired performance. The NP detector has a limitation to identify poorly characterized noise. Thus, an accurate measurement of the noise floor is crucial which requires a large amount of data and a long time to cover all the possible situations. The benchmark detection is limited to symmetric signals, and the criteria range of two metrics need to be carefully calibrated in advance.

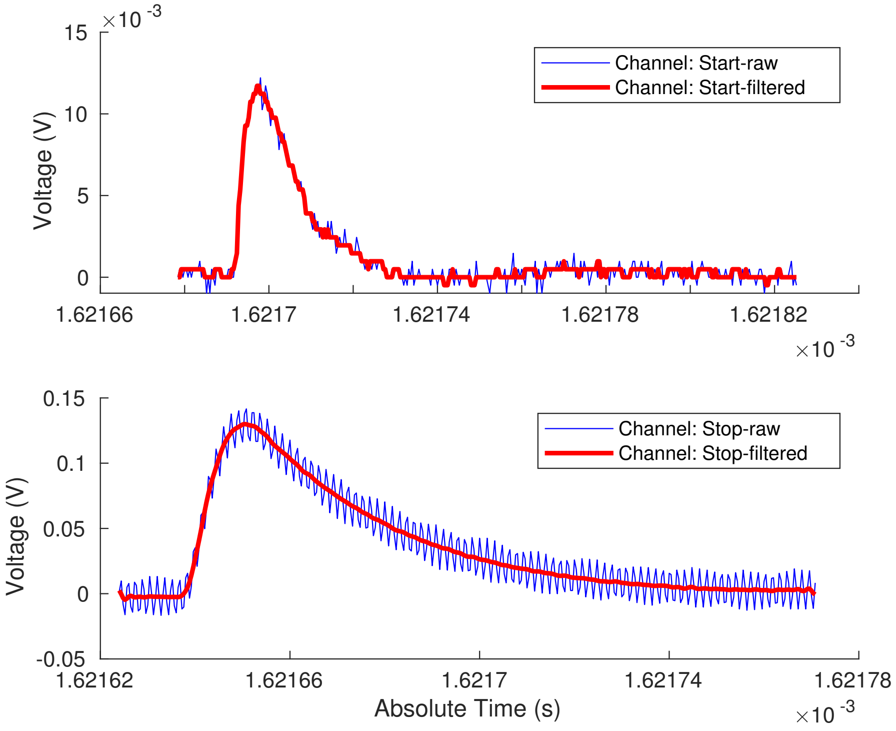
9.5 Comparison between TDC-based and ADC-based Timing Methods

Since both TDC-based and ADC-based timing algorithms are studied, it is beneficial to compare their performance. TDC-based algorithms provides the best performance of filtering out well-characterized white noise between the TDC and ADC timing algorithms, but they are unable to distinguish noise from true signals when the noise amplitude exceeds the threshold. The TDC-based algorithms also have difficulty of detecting true signals when their amplitude are smaller than the trigger threshold. In terms of measurement accuracy, analog filters play an important role of improving the accuracy. After the filtering, the TDC-based methods with TOT compensation and the CFD method achieve a comparable accuracy with FFT-10 and the benchmark algorithm (the best two algorithms among the ADC-based algorithms). The TDC-based algorithms provide precision between the NP detector and the benchmark algorithm. The degraded performance could result from the fluctuation of the trigger point and TOT measurement due to noise.

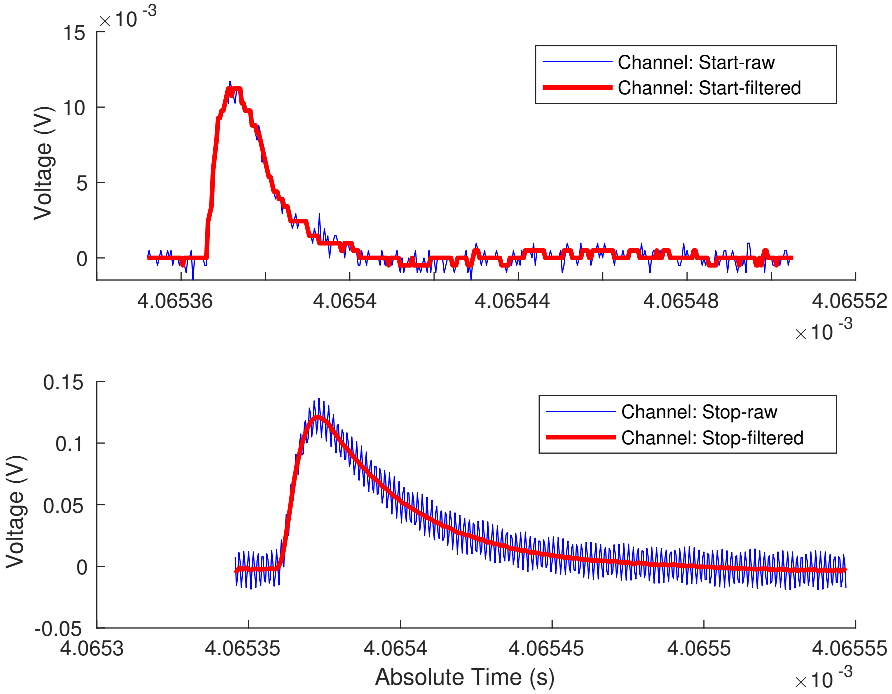
The TDC-based algorithms also have their advantages over the ADC-based ones. One major advantage is the simplicity of the system and the fast processing speed. Without storing a large number of samples and signal processing using DSP circuits, TDC-based approaches can output TOF measurements much faster. Because of the simplicity, a TDC also costs much less than an ADC. Additionally, TDC-based algorithms are less affected by the saturation of a photon-detector, and fast timing circuits also allow a much higher time resolution than what an ADC can achieve. Therefore, both TDC-based and ADC-based algorithms have their strength and weakness, and careful consideration should be taken when a timing device is selected for a lidar system.



(a) Edge Triggering: Precursor: 25, Collection Length: 50

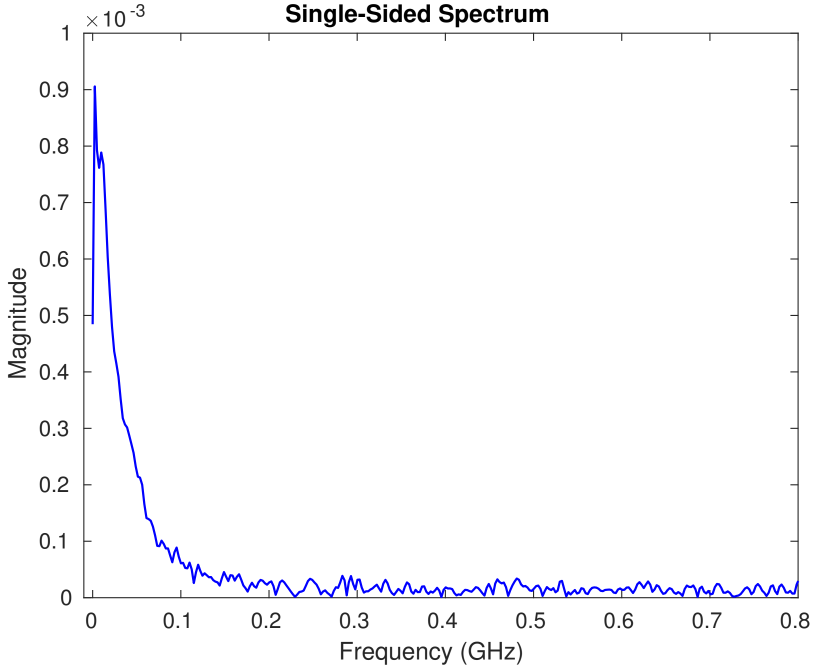


(b) Edge Triggering: Precursor: 5, Collection Length: 40

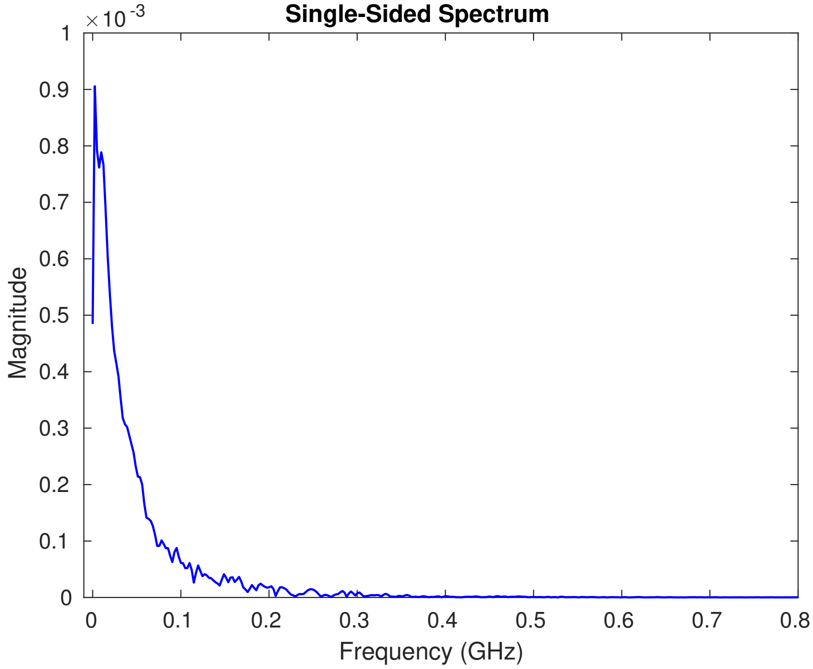


(c) Level Triggering: Precursor: 5, Post-Cursor:40

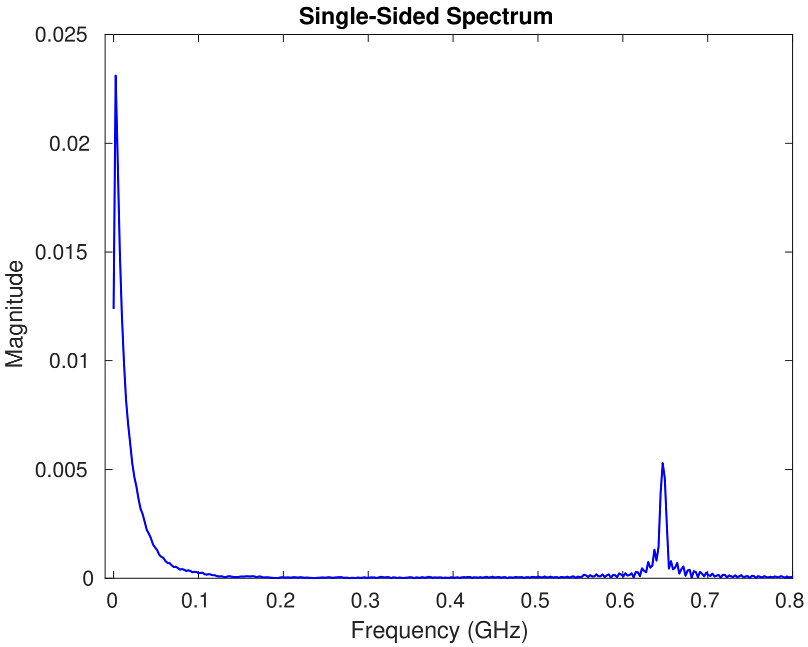
Figure 9.18: Examples of Raw (Blue) and Filtered (Red) Signals Collected using Edge and Level Triggering. Top: START signals; bottom: STOP signals.



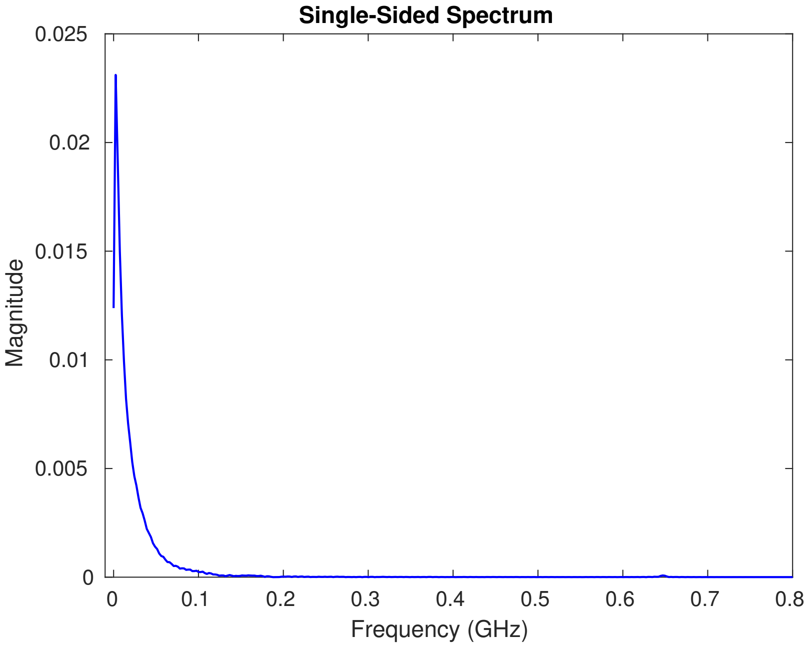
(a) Raw START Signal



(b) Filtered START Signal



(c) Raw STOP Signal.



(d) Filtered STOP Signal

Figure 9.19: Spectrum of Raw and Filtered Signals. Edge-triggering: precursor of 5 and collection length of 40.

Chapter 10

CONCLUSION

10.1 Summary

In the first part of this dissertation, a novel detecting and tracking algorithm for the study of large-scale wind gusts using a long-range Doppler lidar was proposed. A prediction model of the impact of downstream wind turbines, using the Gaussian plume model and random walk theory, was introduced and corrected by the size dependency. The prediction model was validated in a limited set of conditions, and applied to a virtual wind turbine array to estimate the impact time and locations of the gusts on the turbines. A comprehensive conclusion was given in Section 2.3.

In the second part of this dissertation, a modulated full-system TOF lidar simulation was proposed. A pulse model was introduced in the laser source module, and the sampling rate that is used to model continuous analog signals using digital signals was discussed. We observe that a sampling rate equal to the bandwidth of a timing device is sufficient for MATLAB to model analog signals using discrete signals, given the resolution of the timing device is achieved externally.

The propagation module takes into account multiple factors in the laser propagation, including beam divergence, target characteristics, atmospheric attenuation, and optics. An attenuation model that uses visibility and wavelength to represent atmospheric conditions was included in the simulation. The propagation model was compared with measurements from a prototype lidar, and the same trends of signal amplitude against distance were presented. However, due to the limitation of the device and the dataset, only results from short distance were compared.

The receiver module models various electric noise in a lidar receiver. Several analog filters are also included in the receiver module. The effect of the analog filters on signal behaviors was investigated. Butterworth filters demonstrate the least signal distortion, but also the least attenuation of the noise spectrum in the transition band. On the other hand, Chebyshev and Elliptic filters show their efficiency of noise removal, even though their distortion on signals is obvious. Zero-phase filtering provn to be useful to remove the phase shift, but it is limited to post-processing applications.

TDC-based methods and ADC-based time-discrimination methods were included in the timing module. In the TDC module, a TDC model was first introduced. The TOT and slew-rate compensation for the leading-edge timing method and the CFD method are modeled. The determination of the TDC threshold was discussed. The results show good noise recognition if the threshold is set properly. The TDC-based algorithms were tested on simulated signals with different SNRs. The TOT compensation and CFD demonstrated their efficiency on walk-error removal. The sensitivity of these algorithms on noise was also observed.

A benchmark signal detection and estimation algorithm was proposed in the ADC module. The benchmark algorithm was tested on simulated pulses and noise under different ADC configurations. Perfect detection of whiten noise was observed, but the detection of pulses degrades at low SNR. The level-trigger mode with a short length of post-cursor presents the most accurate and precise distance measurement. Using data above the ADC threshold for centroid calculation improves the distance measurements accuracy.

A frequency-domain approach using FFT for the Neyman-Pearson (NP) detector was proposed. Both the time-domain approach (SW approach) and the FFT approach were evaluated on simulated pulses and noise, and measurements from a prototype lidar. The measurements were contaminated by white noise and different frequency-structured and time-structured noises. The FFT approach using frequency-domain zero-paddings efficiently improves the measurement resolution. The NP detector presented a perfect probability of pulse detection. However, the probability of false alarm is higher than expected due to the poor characterization of noise. The SW and FFT approaches show biased mean distance measurements, but they present a good measurement precision. The FFT approach provides a better performence than SW approach.

Finally, the TDC-based and ADC-based time discrimination algorithms were compared, and their advantages and limitations were discussed.

10.2 Future Work

Future research for the wind gust characterization could include the validation of the detecting and tracking algorithms on other datasets with a shorter time interval between frames and a higher spatial resolution. Using Kalman filters or other multiple object tracking algorithms, the detecting and tracking algorithm might be improved. Deep learning approaches might be expected to recognize patches quickly with acceptable accuracy. For the prediction model since the size dependency was derived empirically from the current dataset, the parameters need fine calibration for other datasets, and the dependency of gusts with scales larger than $5.5D$ is not presently clear. In addition, the quality of the prediction model needs to be validated, which could be addressed in a dataset with longer time and in more varied atmospheric conditions. Finally, using real-time measurements, like lidar or meteorological tower, the Lagrangian time scale estimation can be improved.

The next step for the full-system TOF lidar simulation could focus on the integration of more sophisticated models of pulse transmission, signal-target optical interaction, and the effect of atmosphere particles. The proposed receiver model does not consider effect of the response time of a photon-detector and a TIA on the signal shape, which could be taken into account in further development. Implementation of different digital filters in the receiver module could be another direction for future work. Additionally, the analytic models of the TOT and slew-rate compensation needs further validation with a real TDC device, and compensation curves might be generated and integrated into the receiver module. For the ADC benchmark algorithm, a more robust way of the centroid and pulse width calculation could be investigated. A dynamic criteria of NaT and pulse width, that can be adaptively adjusted according to distance, trigger models, and pulse width in use, could be researched to improve the probability of detection. An integration of the metrics in the benchmark algorithm to the NP detector could be explored to reduce the sensitivity to white noise. A more accurate way of noise characterization in experiments could also be investigated to decrease the probability of false alarm. A thorough validation is needed for the models in each module and the entire simulator. For better validation, measurements that are collected with longer data acquisition times under various environmental and system conditions could be beneficial.

REFERENCES

Adams, M., “Lidar design, use, and calibration concepts for correct environmental detection”, IEEE Transactions on Robotics and Automation **16**, 6, 753–761 (2000).

Adams, M. and P. Probert, “The interpretation of phase and intensity data from AMCW light detection sensors for reliable ranging”, The International journal of robotics research **15**, 5, 441–458 (1996).

Aitken, M., R. Banta, Y. Pichugina and J. Lundquist, “Quantifying wind turbine wake characteristics from scanning remote sensor data”, Journal of Atmospheric and Oceanic Technology **31**, 4, 765–787 (2014).

Al-Temeemy, A., “The Development of a 3D LADAR Simulator Based on a Fast Target Impulse Response Generation Approach”, 3D Research **8**, 3, 31 (2017).

Albota, M., B. Aull, D. Fouche, R. Heinrichs, D. Kocher, R. Marino, J. Mooney, N. Newbury, M. O’Brien, B. Player *et al.*, “Three-dimensional imaging laser radars with Geiger-mode avalanche photodiode arrays”, Lincoln Laboratory Journal **13**, 2, 351–370 (2002).

Amidror, I., “Scattered data interpolation methods for electronic imaging systems: a survey”, Journal of electronic imaging **11**, 2, 157–177 (2002).

AMS, “TDC-GPX2 4-Channel Time-to-Digital Converter”, (2017).

Analog Devices, “ADCMP567 Dual Ultrafast Voltage Comparator”, (2003).

Andrews, L., R. Phillips and C. Hopen, *Laser beam scintillation with applications*, vol. 99 (SPIE press, 2001).

ASTMG159-98, *Standard tables for reference solar spectral irradiances: Direct normal and hemispherical on 37° tilted surface(Withdrawn 2005)* (ASTM international, 1998), west Conshohocken, PA.

Aull, B., “Geiger-mode avalanche photodiode arrays integrated to all-digital CMOS circuits”, Sensors **16**, 4, 495 (2016).

Banks, P., “Systems and methods of high resolution three-dimensional imaging”, US Patent 8,471,895 (2013).

Banta, R., Y. Pichugina, W. Brewer, J. Lundquist, N. Kelley, S. Sandberg, R. Alvarez II, R. Hardesty and A. Weickmann, “3D volumetric analysis of wind turbine wake properties in the atmosphere using high-resolution Doppler lidar”, Journal of Atmospheric and Oceanic Technology **32**, 5, 904–914 (2015).

Batchelor, G., “Diffusion in a field of homogeneous turbulence: II. the relative motion of particles”, in “Mathematical Proceedings of the Cambridge Philosophical Society”, vol. 48, pp. 345–362 (Cambridge University Press, 1952).

Becker, W., *Advanced time-correlated single photon counting applications*, vol. 111 (Springer, 2015).

Beer, M., O. Schrey, J. Haase, J. Ruskowski, W. Brockherde, B. Hosticka and R. Kokozinski, "SPAD-based flash LiDAR sensor with high ambient light rejection for automotive applications", in "Quantum Sensing and Nano Electronics and Photonics XV", vol. 10540, p. 105402G (International Society for Optics and Photonics, 2018).

Beer, M., O. Schrey, B. Hosticka and R. Kokozinski, "Coincidence in SPAD-based time-of-flight sensors", in "Ph.D. Research in Microelectronics and Electronics (PRIME), 2017 13th Conference on", pp. 381–384 (IEEE, 2017).

Beljaars, A., *The measurement of gustiness at routine wind stations: a review* (Weteschappelijke rapport (Koninklijk Nederlands Meteorologisch Instituut), WR87-11., 1987).

Bhaskaran, S., *Direct Detection Time of Flight Lidar Sensor System Design and A Vortex Tracking Algorithm for a Doppler Lidar*, Ph.D. thesis, Arizona State University, Tempe, USA (2018).

Blanter, Y. and M. Büttiker, "Shot noise in mesoscopic conductors", Physics reports **336**, 1-2, 1–166 (2000).

Bourke, P., "Calculating the area and centroid of a polygon", Swinburne Univ. of Technology (1988).

Bousselham, A. and C. Bohm, "Sampling pulses for optimal timing", IEEE Transactions on Nuclear Science **54**, 2, 320–326 (2007).

Branlard, E., "Wind energy: On the statistics of gusts and their propagation through a wind farm", ECN-Wind-Memo-09 **5** (2009).

Brazda, V., V. Schejbal and O. Fiser, "Rain impact on FSO link attenuation based on theory and measurement", in "Antennas and Propagation (EUCAP), 2012 6th European Conference on", pp. 1239–1243 (IEEE, 2012).

Budge, S., B. Leishman and R. Pack, "Simulation and modeling of return waveforms from a ladar beam footprint in USU LadarSIM", in "Laser Radar Technology and Applications XI", vol. 6214, p. 62140N (International Society for Optics and Photonics, 2006).

Burns, H., C. Christodoulou and G. Boreman, "System design of a pulsed laser rangefinder", Optical Engineering **30**, 3, 323–330 (1991).

Chamorro, L., S. Lee, D. Olsen, C. Milliren, J. Marr, R. Arndt and F. Sotiropoulos, "Turbulence effects on a full-scale 2.5 MW horizontal-axis wind turbine under neutrally stratified conditions", Wind Energy **18**, 2, 339–349 (2015).

Cherukuru, N., R. Calhoun, R. Krishnamurthy, S. Benny, J. Reuder and M. Flügge, "2D-VAR single Doppler lidar vector retrieval and its application in offshore wind energy", Energy Procedia **137**, 497–504 (2017).

Chevalier, T. and O. Steinvall, "Laser radar modeling for simulation and performance evaluation", in "Electro-Optical Remote Sensing, Photonic Technologies, and Applications III", vol. 7482, p. 748206 (International Society for Optics and Photonics, 2009).

Cronologic, "Ndigo5G-8, Ndigo5G-10 User Guide", (2018).

Der, S., B. Redman and R. Chellappa, "Simulation of error in optical radar range measurements", Applied Optics **36**, 27, 6869–6874 (1997).

Dosio, A., J. Guerau de Arellano, A. Holtslag and P. Builtjes, "Relating Eulerian and Lagrangian statistics for the turbulent dispersion in the atmospheric convective boundary layer", Journal of the atmospheric sciences **62**, 4, 1175–1191 (2005).

Dyson, N., "Peak distortion, data sampling errors and the integrator in the measurement of very narrow chromatographic peaks", Journal of Chromatography A **842**, 1-2, 321–340 (1999).

Ebben, T., D. Begley and R. Marshalek, "Phase-noise-limited accuracy of distance measurements in a frequency-modulated continuous-wave LIDAR with a tunable twin-guide laser diode", Optical Engineering **34**, 3, 896–904 (1995).

Fersch, T., R. Weigel and A. Koelpin, "Challenges in miniaturized automotive long-range lidar system design", in "Three-Dimensional Imaging, Visualization, and Display 2017", vol. 10219, p. 102190T (International Society for Optics and Photonics, 2017).

Fried, D., "Optical resolution through a randomly inhomogeneous medium for very long and very short exposures", JOSA **56**, 10, 1372–1379 (1966).

Godbaz, J., M. Cree and A. Dorrington, "Understanding and ameliorating non-linear phase and amplitude responses in AMCW lidar", Remote Sensing **4**, 1, 21–42 (2011).

Grönwall, C., *Ground object recognition using laser radar data: geometric fitting, performance analysis, and applications*, Ph.D. thesis, Linköping University, Linköping, Sweden (2006).

Grönwall, C., O. Steinvall, F. Gustafsson and T. Chevalier, "Influence of laser radar sensor parameters on range-measurement and shape-fitting uncertainties", Optical Engineering **46**, 10, 106201 (2007).

Gu, I. Y. H. and T. Tjahjadi, "Detecting and locating landmine fields from vehicle-and air-borne measured ir images", Pattern Recognition **35**, 12, 3001–3014 (2002).

Guo, J., H. Zhang and X. J. Zhang, "Propagating characteristics of pulsed laser in rain", International Journal of Antennas and Propagation **2015** (2015).

Hanna, S., G. Briggs and R. Hosker, “Handbook on atmospheric diffusion”, Tech. rep., National Oceanic and Atmospheric Administration, Oak Ridge, TN (USA) (1982).

Heck, M., “Highly integrated optical phased arrays: photonic integrated circuits for optical beam shaping and beam steering”, Nanophotonics **6**, 1, 93 (2017).

Hinshaw, J., “Handling fast peaks”, LC-GC North America **21**, 3, 268–272 (2003).

Hoover, A., V. Kouznetsova and M. Goldbaum, “Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response”, IEEE Transactions on Medical imaging **19**, 3, 203–210 (2000).

Hussmann, S., T. Ringbeck and B. Hagebeuker, “A performance review of 3D TOF vision systems in comparison to stereo vision systems”, in “Stereo Vision”, (InTech, 2008).

International Electrotechnical Commission *et al.*, “IEC 61400-1: Wind turbines part 1: Design requirements”, International Electrotechnical Commission (2005).

Jo, S., H. Kong, H. Bang, J. Kim, J. Kim and S. Choi, “High resolution three-dimensional flash LIDAR system using a polarization modulating Pockels cell and a micro-polarizer CCD camera”, Optics express **24**, 26, A1580–A1585 (2016).

Johnson, J., “Thermal agitation of electricity in conductors”, Physical review **32**, 1, 97 (1928).

Jordan, S., “Range estimation algorithms comparison in simulated 3-D flash LADAR data”, in “Aerospace conference, 2009 IEEE”, pp. 1–7 (IEEE, 2009).

Kadambi, A., A. Bhandari and R. Raskar, “3D depth cameras in vision: Benefits and limitations of the hardware”, in “Computer Vision and Machine Learning with RGB-D Sensors”, pp. 3–26 (Springer, 2014).

Kasunic, K., *Laser Systems Engineering* (SPIE Press monograph, 2016).

Kay, S., “Fundamentals of statistical signal processing, Volume I: Estimation theory (v. 1)”, PTR Prentice-Hall, Englewood Cliffs (1993).

Kay, S., *Fundamentals of Statistical Signal Processing: Detection theory*, Prentice Hall Signal Processing Series (Prentice-Hall PTR, 1998).

Kelley, N., M. Shirazi, D. Jager, S. Wilde, J. Adams, M. Buhl, P. Sullivan and E. Patton, “Lamar low-level jet project interim report”, National Renewable Energy Laboratory, National Wind Technology Center, Golden, CO, Technical Paper No. NREL/TP-500-34593 (2004).

Kelly, P. and G. Horlick, “Practical considerations for digitizing analog signals”, Analytical Chemistry **45**, 3, 518–527 (1973).

Kilpela, A., *Pulsed time-of-flight laser range finder techniques for fast, high precision measurement applications*, Ph.D. thesis, University of Oulu, Oulu, Finland (2004).

Kilpelä, A., J. Ylitalo, K. Määttä and J. Kostamovaara, “Timing discriminator for pulsed time-of-flight laser rangefinding measurements”, Review of Scientific Instruments **69**, 5, 1978–1984 (1998).

Kim, I., B. McArthur and E. Korevaar, “Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications”, in “Optical Wireless Communications III”, vol. 4214, pp. 26–38 (International Society for Optics and Photonics, 2001).

Kittel, C., *Elementary statistical physics* (Courier Corporation, 2004).

Kristalny, M. and D. Madjidian, “Decentralized feedforward control of wind farms: prospects and open problems”, in “2011 50th IEEE Conference on Decision and Control and European Control Conference”, pp. 3464–3469 (IEEE, 2011).

Kromidas, S. and H. Kuss, *Quantification in LC and GC: a practical guide to good chromatographic data* (John Wiley & Sons, 2009).

Kruse, P., L. McGlauchlin and R. McQuistan, “Elements of infrared technology: Generation, transmission and detection”, New York: Wiley (1962).

Kumar, A., E. Bossanyi, A. Scholbrock, P. Fleming, M. Boquet and R. Krishnamurthy, *Field Testing of LIDAR-Assisted Feedforward Control Algorithms for Improved Speed Control and Fatigue Load Reduction on a 600-kW Wind Turbine* (United States. Department of Energy. Office of Energy Efficiency and Renewable Energy, 2015).

Kumar, S. and J. Deen, *Fiber optic communications: fundamentals and applications* (John Wiley & Sons, 2014).

Kumer, V., J. Reuder, B. Svardal, C. Sætre and P. Eecen, “Characterisation of single wind turbine wakes with static and scanning WINTWEX-W LiDAR data”, Energy Procedia **80**, 245–254 (2015).

Kundu, P. and I. Cohen, *Fluid Mechanics* (Elsevier Science, 2010).

Kurtti, S. and J. Kostamovaara, “Pulse width time walk compensation method for a pulsed time-of-flight laser rangefinder”, in “Instrumentation and Measurement Technology Conference, 2009. I2MTC’09. IEEE”, pp. 1059–1062 (IEEE, 2009).

Kurtti, S. and J. Kostamovaara, “An integrated laser radar receiver channel utilizing a time-domain walk error compensation scheme”, IEEE Transactions on instrumentation and measurement **60**, 1, 146–157 (2011).

Lane, R., A. Glindemann, J. Dainty *et al.*, “Simulation of a Kolmogorov phase screen”, Waves in random media **2**, 3, 209–224 (1992).

Leone Jr, J., J. Nasstrom, D. Maddix, D. Larson, G. Sugiyama and D. Ermak, “Lagrangian Operational Dispersion Integrator (LODI) User’s Guide”, Tech. rep., Report UCRL-AM-212798, Lawrence Livermore National Laboratory, Livermore, CA (2005).

Li, D., G. Katul and S. Zilitinkevich, “Revisiting the turbulent Prandtl number in an idealized atmospheric surface layer”, Journal of the Atmospheric Sciences **72**, 6, 2394–2410 (2015).

Mayor, S., P. Dérian, C. Mauzey and M. Hamada, “Two-component wind fields from scanning aerosol lidar and motion estimation algorithms”, in “Lidar Remote Sensing for Environmental Monitoring XIV”, vol. 8872, p. 887208 (International Society for Optics and Photonics, 2013).

Mayor, S., J. Lowe and C. Mauzey, “Two-component horizontal aerosol motion vectors in the atmospheric surface layer from a cross-correlation algorithm applied to scanning elastic backscatter lidar data”, Journal of Atmospheric and Oceanic Technology **29**, 11, 1585–1602 (2012).

McManamon, P., P. Banks, J. Beck, D. Fried, A. Huntington and E. Watson, “Comparison of flash lidar detector options”, Optical Engineering **56**, 3, 031223 (2017).

Middleton, W. E. K., *Vision through the atmosphere* (University of Toronto Press, 1952).

Mitra, S. and Y. Kuo, *Digital signal processing: a computer-based approach*, vol. 2 (McGraw-Hill New York, 2006).

Nakhostin, M., *Signal Processing for Radiation Detectors* (John Wiley & Sons, 2017).

Neyman, J. and E. Pearson, “On the problem of the most efficient tests of statistical inference”, Biometrika A **20**, 175–240 (1933).

Niclass, C., *Single-photon image sensors in CMOS: picosecond resolution for three-dimensional imaging* (Hartung-Gorre, 2008).

Niclass, C., M. Soga and E. Charbon, “3D Imaging based on single photon detectors”, in “2nd Symposium on Range Imaging (RIM'07)”, (2007), invited paper.

Niclass, C., M. Soga, H. Matsubara, S. Kato and M. Kagami, “A 100-m range 10-frame/s 340×96-pixel time-of-flight depth sensor in 0.18-μm cmos”, IEEE Journal of Solid-State Circuits **48**, 2, 559–572 (2013).

Nissinen, J., I. Nissinen and J. Kostamovaara, “Integrated receiver including both receiver channel and TDC for a pulsed time-of-flight laser rangefinder with cm-level accuracy”, IEEE journal of solid-state circuits **44**, 5, 1486–1497 (2009).

Ofek, E. and B. Zackay, “Optimal matched filter in the low-number count Poisson noise regime and implications for X-ray source detection”, The Astronomical Journal **155**, 4, 169 (2018).

Oren, M. and S. Nayar, “Generalization of Lambert's reflectance model”, in “Proceedings of the 21st annual conference on Computer graphics and interactive techniques”, pp. 239–246 (ACM, 1994).

Petersen, W., J. Catalano, T. Chico and T. Yuen, "INPUFF-A single source gaussian puff dispersion algorithm", User's Guide. US EPA, Research Triangle Park, NC (1984).

Pierrottet, D., F. Amzajerdian, L. Petway, B. Barnes, G. Lockard and M. Rubio, "Linear FMCW laser radar for precision range and vector velocity measurements", MRS Online Proceedings Library Archive **1076** (2008).

Poulton, C., A. Yaacobi, D. Cole, M. Byrd, M. Raval, D. Vermeulen and M. Watts, "Coherent solid-state LIDAR with silicon photonic optical phased arrays", Optics letters **42**, 20, 4091–4094 (2017).

Price, J., "Lagrangian and Eulerian representations of fluid flow: Part I, kinematics and the equations of Motion", Clark Laboratory Woods Hole Oceanographic Institution: Woods Hole, MA, USA (2005).

Rahman, A., M. Anuar, S. Aljunid and M. Junita, "Study of rain attenuation consequence in free space optic transmission", in "Telecommunication Technologies 2008 and 2008 2nd Malaysia Conference on Photonics. NCTT-MCP 2008. 6th National Conference on", pp. 64–70 (IEEE, 2008).

Richmond, R. and S. Cain, *Direct-detection LADAR systems*, Tutorial Text Series (SPIE Press, 2010).

Richmond, R., R. Stettner and H. Bailey, "Laser radar focal plane array for three-dimensional imaging", in "Laser Radar Technology and Applications", vol. 2748, pp. 61–68 (International Society for Optics and Photonics, 1996).

Roman, J., M. Rangaswamy, D. Davis, Q. Zhang, B. Himed and J. Michels, "Parametric adaptive matched filter for airborne radar applications", IEEE Transactions on Aerospace and Electronic Systems **36**, 2, 677–692 (2000).

Ruotsalainen, T., P. Palojarvi and J. Kostamovaara, "A wide dynamic range receiver channel for a pulsed time-of-flight laser radar", IEEE Journal of Solid-State Circuits **36**, 8, 1228–1238 (2001).

Scheeren, P., Z. Klous, H. Smit and D. Doornbos, "A software package for the orthogonal polynomial approximation of analytical signals, including a simulation program for chromatograms and spectra", Analytica Chimica Acta **171**, 45–60 (1985).

Schlipf, D., L. Pao and P. W. Cheng, "Comparison of feedforward and model predictive control of wind turbines using LIDAR", in "2012 IEEE 51st IEEE Conference on Decision and Control (CDC)", pp. 3050–3055 (IEEE, 2012).

Schmidt, B., S. Tuvey and P. Banks, "3D sensor development to support EDL (entry, descent, and landing) for autonomous missions to mars", in "Nanophotonics and Macrophotonics for Space Environments VI", vol. 8519, p. 851905 (International Society for Optics and Photonics, 2012).

Scholbrock, A., P. Fleming, L. Fingersh, A. Wright, D. Schlipf, F. Haizmann and F. Belen, "Field testing LIDAR-based feed-forward controls on the NREL controls advanced research turbine", in "51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition", p. 818 (2013).

Skolnik, M., "Introduction to radar", Radar handbook **2** (1962).

Smalikho, I., V. Banakh, Y. Pichugina, W. Brewer, R. Banta, J. Lundquist and N. Kelley, "Lidar investigation of atmosphere effect on a wind turbine wake", Journal of Atmospheric and Oceanic Technology **30**, 11, 2554–2570 (2013).

Smat, R., "Introduction to comparators, their parameters and basic applications", STMicroelectronics, App. Note **4071** (2012).

Steinvall, O., "Effects of target shape and reflection on laser radar cross sections", Applied Optics **39**, 24, 4381–4391 (2000).

Steinvall, O. and T. Carlsson, "Three-dimensional laser radar modeling", in "Laser Radar Technology and Applications VI", vol. 4377, pp. 23–35 (International Society for Optics and Photonics, 2001).

Steinvall, O., L. Sjöqvist, M. Henriksson and P. Jonsson, "High resolution ladar using time-correlated single-photon counting", in "Laser Radar Technology and Applications XIII", vol. 6950, p. 695002 (International Society for Optics and Photonics, 2008).

Stoppa, D. and A. Simoni, "Single-photon detectors for time-of-flight range imaging", in "Single-Photon Imaging", pp. 275–300 (Springer, 2011).

Süss, A., V. Rochus, M. Rosmeulen and X. Rottenberg, "Benchmarking time-of-flight based depth measurement techniques", in "Smart Photonic and Optoelectronic Integrated Circuits XVIII", vol. 9751, p. 975118 (International Society for Optics and Photonics, 2016).

Sykes, R., S. Parker, D. Henn, C. Cerasoli and L. Santos, "PC-SCIPUFF version 1.2 PD technical documentation", ARAP Rep **718**, 180, 08543–2229 (1998).

Thykier-Nielsen, S., S. Deme and T. Mikkelsen, "Description of the atmospheric dispersion module RIMPUFF", Riso National Laboratory, PO Box **49** (1999).

Vandeginste, B., D. Massart, S. De Jong, D. Massaart and L. Buydens, *Handbook of chemometrics and qualimetrics: Part B* (Elsevier, 1998).

Vio, R. and P. Andreani, "Matched filter in the low-number count Poisson noise regime: an efficient and effective implementation", arXiv preprint arXiv:1801.02859 (2018).

Wahab, F., P. Dasgupta, A. Kadjo and D. Armstrong, "Sampling frequency, response times and embedded signal filtration in fast, high efficiency liquid chromatography: A tutorial", Analytica chimica acta **907**, 31–44 (2016).

Wall, J., "Practical statistics for astronomers. I-definitions, the normal distribution, detection of signal", Quarterly Journal of the Royal Astronomical Society **20**, 138–152 (1979).

Wang, H. and R. Barthelmie, "Wind turbine wake detection with a single Doppler wind lidar", in "Journal of Physics: Conference Series", vol. 625, p. 012017 (IOP Publishing, 2015).

Webb, P., "Properties of avalanche photodiodes", RCA review **35**, 234 (1974).

Weisstein, E., "Moore neighborhood", URL `http://mathworld.wolfram.com/MooreNeighborhood.html`, accessed on 3/2/2019 (2019).

Williams, G., "Optimization of eyesafe avalanche photodiode lidar for automobile safety and autonomous navigation systems", Optical Engineering **56**, 3, 031224 (2017).

Williams, G., "Range-Walk Correction Using Time Over Threshold", Tech. rep., Voxtel Inc (2018).

Wojtanowski, J., M. Zygmunt, M. Kaszczuk, Z. Mierczyk and M. Muzal, "Comparison of 905 nm and 1550 nm semiconductor laser rangefinders' performance deterioration due to adverse environmental conditions", Opto-Electronics Review **22**, 3, 183–190 (2014).

Zhang, P., X. Du, J. Zhao, Y. Song and H. Chen, "High resolution flash three-dimensional LIDAR systems based on polarization modulation", Applied optics **56**, 13, 3889–3894 (2017).

Zhou, K. and R. Calhoun, "Lidar Measurements of Wind Flow through a Rotor-swept Area in Tehachapi Pass", in "EPJ Web of Conferences", vol. 119, p. 14005 (EDP Sciences, 2016).

Zhou, K., N. Cherukuru, X. Y. Sun and R. Calhoun, "Wind gust detection and impact prediction for wind turbines", Remote Sensing **10**, 4, 514 (2018).

Ziemer, R. and W. Tranter, *Principles of communications: system modulation and noise* (John Wiley & Sons, 2006).

APPENDIX A

SPECIFICATIONS OF PROTOTYPE LIDAR

Table A.1: Specifications of Prototype Lidar

Laser Source

Parameter	**Value**
Type	Fiber laser
Wavelength	1550 nm
Peak power	300 W
Pulse width	5 ns
PRF	1 MHz

Optics and Receiver

Parameter	**Value**
Optical path	bi-static
FOV overlap	1 [1]
Material	InGaAs
Responsivity	0.9 A/W
APD gain	10
APD bandwidth	10 GHz
TIA gain	1200 Ω
Receiver diameter	2.54 cm
Bandwidth of TIA	750 MHz
Standard deviation of extra noise	0.0288 V [2]

TDC

Parameter	**Value**
Resolution	1 ps

ADC (Cronologic, 2018)

Parameter	**Value**
Number of channel	4
Sampling rate	5 Gsps (one channel-mode) [3]
Dynamic range	10 bit
Input signal amplitude limits	±0.25 V

[1] The full-overlapping zone begins at 1.78 m with a separation of the transmitter and receiver of 7 cm (Bhaskaran, 2018).

[2] Measured from the prototype lidar.

[3] Gsps: gigasample per second.

APPENDIX B

DERIVATION OF RELATIONSHIPS IN THE PULSE MODEL

The transmit pulse model is given in Equation (3.7):

$$P_t(t) = \frac{E}{\sigma\sqrt{2\pi}} e^{-\frac{-t^2}{2\sigma^2}}. \tag{B.1}$$

Equation (3.9) and Equation (3.8) are derived next. The rise time is defined as the time taken by a pulse to rise from 10% to 90% of its peak power. At the 90% of the peak power, the instantaneous power is

$$0.9P_0 = P_0 e^{-\frac{-t_{90\%}^2}{2\sigma^2}}, \tag{B.2}$$

from which we obtain

$$t_{90\%} = \sqrt{-2ln(0.9)} \approx 0.459\sigma. \tag{B.3}$$

Similarly, we can obtain $t_{10\%} \approx 2.146\sigma$ from

$$0.1P_0 = P_0 e^{-\frac{-t_{10\%}^2}{2\sigma^2}}. \tag{B.4}$$

Thus, the rise time t_r can be expressed by:

$$t_r = t_{10\%} - t_{90\%} \approx 1.687\sigma, \tag{B.5}$$

or

$$\sigma \approx \frac{t_r}{1.687}. \tag{B.6}$$

The pulse width of a laser pulse, which is usually defined as the Full Width Half Maximum (FWHM) of the pulse, can also be derived from Equation (3.7):

$$0.5P_0 = P_0 e^{-\frac{-t_{FWHM}^2}{2\sigma^2}}. \tag{B.7}$$

Thus,

$$FWHM = 2\sqrt{2ln2\sigma} \approx 2.355\sigma. \tag{B.8}$$

APPENDIX C

DERIVATION OF WALK-ERROR COMPENSATION ALGORITHMS

C.1 TOT Compensation Derivation

In the TOT compensation, the TOT is measured from the two stop channels of a TDC. As we know $TOT = 2(|t_{ref}| - \Delta_w)$, plugging the data points (t_{ref}, V_{th}) and ($TOT/2$, V_{th}) into the pulse model (here we express the pulse model in voltage):

$$V(t) = V_r e^{-\frac{t^2}{2\sigma^2}}, \tag{C.1}$$

we have:

$$\begin{cases} V_{th} = V_{ref} e^{-t_{ref}^2/2\sigma^2} & \text{(C.2)} \\ V_{th} = V_r e^{-(|t_{ref}|-\Delta_w)^2/2\sigma^2}. & \text{(C.3)} \end{cases}$$

Combining Equation (C.2) and (C.3), we have:

$$\frac{V_{ref}}{V_r} = e^{-[(|t_{ref}|-\Delta_w)^2 - t_{ref}^2]/2\sigma^2} \tag{C.4}$$

$$\Delta_w^2 - 2\Delta_w|t_{ref}| + 2\sigma^2 \ln(\frac{V_{ref}}{V_r}) = 0. \tag{C.5}$$

Solving (C.5) we obtain

$$\Delta_w = |t_{ref}| \pm \sqrt{t_{ref}^2 - 2\sigma^2 \ln(\frac{V_{ref}}{V_r})}. \tag{C.6}$$

Variable V_r can be derived from Equation (C.1) by setting $t = TOT/2$. As we know Δ_w is usually smaller than t_{ref}, we ignore the positive sign in Equation (C.6). Thus, we can derive the walk error Δ_w as a function of TOT and the comparator threshold V_{th}:

$$\Delta_w = f(TOT, V_{th}, t_{ref}, V_{ref}) = |t_{ref}| - \sqrt{t_{ref}^2 - 2\sigma^2 \ln(\frac{V_{ref}}{V_r})} \tag{C.7}$$

$$V_r = \frac{V_{th}}{e^{-(\frac{TOT}{2})^2/2\sigma^2}}. \tag{C.8}$$

C.2 Slew-rate Compensation Derivation

In the slew-rate compensation, the TDC provides two time stamps at the stop channel $STOP_1$ and $STOP_2$ at thresholds V_{th1} and $V_{th2} = CV_{th1}$, respectively. C is a preset constant integer. The time difference between the two time stamps $\Delta t = STOP_2 - STOP_1$.

Plugging the measured time stamps $STOP_1$ and $STOP_2$ and amplitudes V_{th1} and V_{th2} into the pulse model (here we express the pulse model in voltage):

$$V(t) = V_r e^{-\frac{t^2}{2\sigma^2}}, \tag{C.9}$$

we have:

$$\begin{cases} V_{th1} = \dfrac{V_r}{e^{-STOP_1^2/2\sigma^2}} & \text{(C.10)} \\ V_{th2} = \dfrac{V_r}{e^{-STOP_2^2/2\sigma^2}}. & \text{(C.11)} \end{cases}$$

Solving Equation (C.10) and (C.11), we have

$$\begin{aligned} STOP_2 &= -\sqrt{-2\sigma^2 \ln C + STOP_1^2} \\ \Delta t &= STOP_2 - STOP_1 \\ &= -\sqrt{-2\sigma^2 \ln C + STOP_1^2} - STOP_1. \end{aligned} \tag{C.12}$$

Thus,

$$STOP_1 = \frac{-2\sigma^2 \ln C - \Delta t^2}{2\Delta t}. \tag{C.13}$$

Since we define that $\Delta_w = STOP_1 - t_{ref}$, we can get the relation of Δ_w with the threshold constant C and time difference Δt as:

$$\Delta_w = f(C, \Delta t, t_{ref}) = \frac{-2\sigma^2 \ln C - \Delta t^2}{2\Delta t} - t_{ref}. \tag{C.14}$$

APPENDIX D

DERIVATION OF NEYMAN-PEARSON DETECTOR

D.1 GLRT

Signals under hypothesis $\mathcal{H}_1$ and $\mathcal{H}_0$ are given in Equation (9.1) and the noise is assumed to be white following a Gaussian distribution with a variance of W. The PDF of a Gaussian distributed variable x can be expressed by

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp[-\frac{1}{2\sigma^2}(x-\mu)^2] \tag{D.1}$$

Thus, the PDF of signals under $\mathcal{H}_1$ is given by

$$\begin{aligned} p(\vec{x}; n_0, A, \mathcal{H}_1) &= \prod_{n=0}^{N-1} p(x[n]; n_0, A, \mathcal{H}_1) \\ &= \prod_{n=0}^{n_0-1} \frac{1}{\sqrt{2\pi\sigma^2}} exp\big(-\frac{1}{2\sigma^2}x^2[n]\big) \\ &\times \prod_{n=n_0}^{n_0+M-1} \frac{1}{\sqrt{2\pi\sigma^2}} exp\big(-\frac{1}{2\sigma^2}(x[n]-As[n-n_0])^2\big) \\ &\times \prod_{n=n_0+M}^{N-1} \frac{1}{\sqrt{2\pi\sigma^2}} exp\big(-\frac{1}{2\sigma^2}x^2[n]\big) \\ &= \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\sigma^2}} exp\big(-\frac{1}{2\sigma^2}x^2[n]\big) \\ &\times \prod_{n=n_0}^{n_0+M-1} \frac{1}{\sqrt{2\pi\sigma^2}} exp\big(-\frac{1}{2\sigma^2}(-2Ax[n]s[n-n_0]+A^2s^2[n-n_0])\big) \end{aligned} \tag{D.2}$$

and the PDF of signals under $\mathcal{H}_0$:

$$\begin{aligned} p(\vec{x}; \mathcal{H}_0) &= \prod_{n=0}^{N-1} p(x[n]; \mathcal{H}_0) \\ &= \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\sigma^2}} exp\big(-\frac{1}{2\sigma^2}x^2[n]\big) \end{aligned} \tag{D.3}$$

Thus, the left-hand side of GLRT (Equation (9.2)) can be expressed by:

$$\begin{aligned} L_G(\vec{x}) &= \frac{p(\vec{x}; n_0, A, \mathcal{H}_1)}{p(\vec{x}; \mathcal{H}_0)} \\ &= \prod_{n=n_0}^{n_0+M-1} exp\big(-\frac{1}{2\sigma^2}(-2Ax[n]s[n-n_0]+A^2s^2[n-n_0])\big) \end{aligned} \tag{D.4}$$

Taking the logarithm on both sides:

$$-\frac{1}{2\sigma^2}\sum_{n=n_0}^{n_0+M-1} -2Ax[n]s[n-n_0] + A^2s^2[n-n_0] > \ln\gamma \tag{D.5}$$

Then, we have

$$T_{max} = \max_{n_0\in[0,N-M]} T(x[n_0]) = \max_{n_0\in[0,N-M]} \sum_{n=n_0}^{n_0+M-1} x[n]s[n-n_0] > \gamma' \tag{D.6}$$

$$\gamma' = \frac{A}{2}\sum_{n=0}^{M-1} s^2[n] + \frac{W\ln\gamma}{A} \tag{D.7}$$

D.2 PDF's of $T(\vec{x})$

Since the PDF's of $T(\vec{x})$ should also follow a Gaussian distribution, the mean and variance under hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ can be derived as:

$$\begin{aligned}
\hat{\mu_0} &= E[T(\vec{x})] = E\big[\sum_n w[n]s[n]\big] = \sum_n E\big[w[n]\big]s[n] = 0 \\
\hat{\sigma_0^2} &= Var\big[T(\vec{x})\big] = Var\big[\sum_n w[n]s[n]\big] \\
&= \sum_n Var\big[w[n]\big]s^2[n] = W\sum_n s^2[n] \\
\hat{\mu_1} &= E[T(\vec{x})] = E\big[\sum_n (As[n]+w[n])s[n]\big] \\
&= \sum_n E\big[As^2[n]+w[n]s[n]\big] = \sum_n As^2[n] \\
\hat{\sigma_1^2} &= Var[T(\vec{x})] = Var\big[\sum_n (As[n]+w[n])s[n]\big] \\
&= \sum_n Var\big[As^2[n]+w[n]s[n]\big] = 0 + \sum_n Var\big[w[n]s[n]\big] \\
&= W\sum_n s^2[n]
\end{aligned} \tag{D.8}$$

In short,

$$\begin{aligned}
\hat{\mu_0} &= 0 \\
\hat{\sigma_0^2} &= W\sum_n s^2[n] \\
\hat{\mu_1} &= A\sum_n s^2[n] \\
\hat{\sigma_1^2} &= W\sum_n s^2[n],
\end{aligned} \tag{D.9}$$

Thus, $T(\vec{x})$ under hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ follows Gaussian distributions:

$$\begin{aligned}
\mathcal{H}_0 :& T(\vec{x}) \sim N(\hat{\mu_0}, \hat{\sigma}_0^2) \\
\mathcal{H}_1 :& T(\vec{x}) \sim N(\hat{\mu_1}, \hat{\sigma}_1^2)
\end{aligned} \tag{D.10}$$

APPENDIX E

MATLAB SCRIPTS FOR DEMONSTRATION

MATLAB scripts to measure the TOF using the lidar simulator with an ADC or TDC are given in this appendix. Users should keep all the files under the same directory, and run the main functions (Section E.1 for ADC and Section E.6 for TDC) in MATLAB for demonstration.

The main functions call the functions of different modules and output TOF and distance measurements in the variable *Results*. Examples of transmit pulses and return pulses can also be plotted (refer to the comments in the scripts for details). Different analog filters and TDC/ADC detection algorithms can also be selected in the main functions.

Laser parameters can be set in the main functions and in Section E.2. Target distance can be set in the main functions. Parameters used in the radiometry calculation, target reflectivity, and the receiver configurations can be set in Section E.3.

Characteristics of analog filters including the filter type (*e.g.* low-pass or high-pass filter), order, cut-off frequency, passband ripple, and stopband ripple can be set in Section E.4.1.

The ADC sampling rate, bit resolution, and trigger threshold can be set in Section E.5. Section E.5 also contains the functions of edge-trigger, level-trigger mode, and the amplitude digitization. The precursor, collection length, and post-cursor can be set in the function *funADCSamplingEdge* and the function *funADCSamplingLevel* (refer to the comments in the functions for details).

The criteria of NaT and pulse width, and tolerance percentage for the benchmark algorithm can be set in Section E.5.1. Examples of polygons can also be plotted. The PFA, kernel, and number of zeros in the zero-padding for the NP detector can be set in Section E.5.2.

The TDC resolution can be set in Section E.7.1 and Section E.7.2. The TDC threshold can be set in Section E.7.1 and the value is also used in Section E.7.2. The constant fraction for the CFD algorithm can be set in Section E.7.2. The TOT compensation and the CFD algorithm are achieved in Section E.7.3 and Section E.7.4. The function of linear interpolation is given in Section E.7.5.

E.1 Main Function (ADC)

```
clear
close all

%% Constants
global c sampleRate Nobs P0 tStart TargetDistance
Nobs = 50; % number of observations in the Monte-Carlo experiment
c = 3E8; % speed of light [m/s]

%% Lidar settings
% laser source
P0 = 300; % Peak power of transmit pulse [W]
riseTime = 3.58E-9; % rise time [s]
sampleRate = 1e11; % sampling rate to model continuous 'analog' ...
    signal [Hz]

% Target characteristics

TargetDistance = 5; % Target distance [m]
timeTrue = 2*TargetDistance/c; % ground-truth TOF

%% Tranmit Pulse Module
% OUTPUTS: time series, transmit pulse and other laser parameters

[tTran, yTran, Energy, PowerAveg] = fun_PulseModel(P0,riseTime);
%
% figure
% hold on
% plot(tTran*1e9, yTran,'b')
% ylabel('Power (W)')
% xlabel('Time (ns)')
% title('Transmit Pulse')

%% Propagation and Receiver Module
% OUTPUTS: return time series, return pulse with noise
tRet = tTran + timeTrue; % return time series
[SNRSquare, yRetClean, yRetNoisy] = ...
    fun_receiver(TargetDistance,yTran); % return pulse with noise

% figure
% hold on
% plot(tRet*1e9, yRetClean,'k')
% plot(tRet*1e9, yRetNoisy(1,:),'b')
% ylabel('Power (W)')
% xlabel('Time (ns)')
% title('Return Pulse')
% legend('Noise-free', 'Signal with noise')
%% Analog Filter Module
% INPUTS:
% -1st term: filter on or off: 'on' | 'off',
% -2nd term: forward filter with time compensation or zero-phase ...
    filtering: 'analog' | 'filtfilt'
% -3st term: type of filter: butterworth, chebyshev or elliptic: ...
    'butter'
```

```
% | 'cheby' | 'ellip'

% OUTPUTS: filtered return signals
for k = 1: Nobs
    yAFE(k,:) = fun_AFE('on','filtfilt', 'butter', ...
        yRetNoisy(k,:), sampleRate);
end
% apply the same filter to START signal to compensate time delay
yAFE_Start = fun_AFE('on','filtfilt', 'butter', yTran, sampleRate);
% time shift of START time (will be subtracted from the STOP time ...
    measured in the timing module)
tStart = tTran(yAFE_Start == max(yAFE_Start));

% figure
% hold on
% plot(tRet*1e9, yRetClean,'k')
% plot(tRet*1e9, yRetNoisy(1,:),'b')
% plot(tRet*1e9, yAFE(1,:),'r')
% ylabel('Power (W)')
% xlabel('Time (ns)')
% title('Return Pulse')
% legend('Noise-free', 'Signal with noise', 'Filtered signal')
%% Timing Module
% INPUTS: types of detection algorithm: benchmark or NP detector: ...
    'bm' | 'np'
% OUTPUTS: measured TOF and distance

Flag_detector = 'bm'; % detection algorithm (bm: benchmark ...
    algorithm; np: NP detector
Results= fun_ADC(tRet, yRetClean, yAFE, Flag_detector);

% Results
Probability = Results.prob; % PFA or PD
DistanceMean = Results.dMean; % mean distance measurement
DistanceSTD = Results.dStd; % std of distance measurement
```

E.2 Function of Pulse Module

```
function [t, signal, Energy, PowerAveg] = ...
    fun_PulseModel(PowerPeak, riseTime)
%% Gaussian Pulse Module
% Inputs:
%   -PeakPower: peak power of transmit pulse [W]
%   -riseTime: rise time of the pulse [s]
%   -sampleRate: sampling rate to model 'analog' signals [Hz]
%   -PRF: pulse repetition rate [Hz]
% Outputs:
%   -t: transmit time series
%   -signal: transmit pulse
%   -pulseWidth: pulse width (FWHM) = 5 ns
%   -Energy: energy of laser pulse {J}
%   -PowerAveg: laser average power [W]

% Inputs
global pulseWidth sampleRate RMS
PRF  = 1e6; % PRF of laser [Hz]

% Outputs
RMS = riseTime / 1.687; % standard deviation of the Gaussian pulse
pulseWidth = RMS * 2*sqrt(2*log(2));
Energy = PowerPeak * pulseWidth;
PowerAveg = Energy * PRF;

tStart = -5*RMS;
tEnd = 5*RMS;

t = tStart : 1/sampleRate : tEnd; % time series of transmit pulse
signal = 0.94* PowerPeak * exp(-0.5*(t/RMS).^2); % transmit pulse
end
```

E.3 Function of Propagation and Receiver Module

```
function [SNRSquare, powerRet, powerRetNoisy] = ...
    fun_receiver(TargetDistance, yTran)
%% Propagation and Receiver Module
% Inputs:
%   -TargetDistance: target distance [m]
%   -PowerTran: transmitted pulse

% Outputs:
%   -SNRSquare: SNR of power [dB]
%   -powerRet: noise-free return pulse
%   -powerRetNoisy: return pulse with noise

%% Constants
global e integrationTime R0 Gain F Nobs NShotBackMean ...
    NShotDarkMean powerNoiseFloor
rng(0,'twister'); % random machine seed
e = 1.602E-19; % charge of electrons
k = 1.38E-23; % Boltzmann constant [J/K]
Srr = 0.26226; % solar spectral irradiance W/m^2/nm

%% Variables
% Laser source
wavelength = 1550; % laser wavelength [nm]

% Atmospheric attenuation
visibility =10; % visibility [km]

if visibility > 50  % for high visibility
    q = 1.6;
elseif visibility > 6 && visibility < 50  % for average visibility
    q = 1.3;
elseif visibility > 1 && visibility < 6   % for haze visibility
    q = 0.16 * visibility + 0.34;
elseif visibility > 0.5 && visibility < 1 % for mist visibility
    q = visibility - 0.5;
else                                      % for fog visibility
    q = 0;
end
alpha = 3.91/visibility *(wavelength/550)^(-q); % [/km] specific ...
    attenuation
coefAtten = exp(-2 * TargetDistance/1000 * alpha); % attenuation ...
    coefficient, round trip

% Target characteristics
beta = .1; % reflectivity

% Photon-detector
R0 = 0.9; % responsivity [A/W]
Gain = 10; % APD gain
K = 0.45; % ionization ratio of InGaAs
F = K*Gain + (1-K) * (2 - 1/Gain); % excess noise factor
BW = 10E9; % system bandwidth [Hz]
integrationTime = 1/2/BW; % integration time [s]
```

```
NF = 1.6; % noise factor of InGaAs
T = 300; % temperature [K]
Teff = NF * T; % effective temperature [K]
RTIA = 1200; % TIA resistance [ohm]
extraNoise = 0.0288; % STD of extra noise from prototype lidar[V]

% Receiver lens
d_lambda = 2;  % spectral filter bandwidth [nm]
FOV = 0.002; % receiver FOV [rad]
angle = 0; % angle between the target surface normal and ...
    receiver's look-angle (angle = 0: receiver faces the target.)
Ax = pi * (FOV/2 * TargetDistance)^2 / cos(angle); % area ...
    coverage of the receiver FOV at the target distance [m^2]
Dr = 25.4E-3; % receiver diameter [m]
Ar = pi * Dr^2 / 4; % area of the receiver aperture [m^2]

% Optical coefficients
coefOpticTrans0 = 0.9; % transmission through transmitter optical ...
    path
coefT = 1 ;  % collection efficiency of transmitter lens
coefFilterTranRet = 0.7; % transmission of receiver spectral filter
coefOpticTranRet = 0.9; % transmission through receiver lens
FOVoverlap = 1; % FOV overlap between transmitter and receiver

%% Propagation Module
% OUTPUTS: return pulse and power of ambient light
powerRet = (yTran * coefOpticTrans0 * coefT * FOVoverlap) * Ar / ...
    TargetDistance^2 * coefAtten * beta * coefFilterTranRet * ...
    coefOpticTranRet;
Pamb = (Srr * d_lambda) * Ax * Ar / TargetDistance^2 * exp ...
    (-TargetDistance/1000 * alpha) * beta * coefOpticTranRet ...
    *coefFilterTranRet;


%% Receiver Module
% SHOT NOISE
% mean current
iBckgrd = Pamb * R0; % background current [A]
iDark = 150E-9; % dark current [A]

% mean number of electrons during integration time (STD)
NShotNoiseSigMean = powerRet * R0  * Gain^2 * F * integrationTime ...
    / e;
NShotBackMean = iBckgrd * Gain^2 * F * integrationTime / e;
NShotDarkMean = iDark * Gain^2 * F * integrationTime / e;

% generate shot noise current at each sample
iShotSigNoise= (poissrnd(NShotNoiseSigMean .* ...
    ones(Nobs,length(powerRet)))- NShotNoiseSigMean) * e / ...
    integrationTime;
iShotAmbNoise = (poissrnd(NShotBackMean * ...
    ones(Nobs,length(powerRet))) - NShotBackMean) * e / ...
    integrationTime;
iShotDarkNoise = (poissrnd(NShotDarkMean * ...
    ones(Nobs,length(powerRet))) - NShotDarkMean) * e / ...
    integrationTime;

```

```
% total shot noise current
iShotNoise = iShotSigNoise + (iShotAmbNoise + iShotDarkNoise);

% THERMAL NOISE
% thermal noise current (STD)
sigmaThermalAmp = sqrt(4 * k * Teff * BW / RTIA); % TIA thermal noise

% generate thermal noise current at each sample
iThermal = (extraNoise / RTIA + sigmaThermalAmp) * randn(Nobs, ...
    length(powerRet));

% TOTAL NOISE CURRENT
iNoiseElec = iShotNoise + iThermal;

% RETURN PULSE (with NOISE)
% total return current (with noise)
iRetNoisy = powerRet * R0 * Gain + iNoiseElec;
% total return power (with noise)
powerRetNoisy = iRetNoisy / R0 / Gain;
powerNoise = (iNoiseElec - iShotSigNoise) / R0 / Gain; % Noise ...
    power (laser off)
powerNoiseFloor = std(powerNoise,1,2); % noise floor
SNRSquare = 20*log10(max(powerRet) * R0 * Gain ./ ...
    std(iNoiseElec,1,2)); % SNR of power in dB
end
```

E.4 Function of Analog Filter Module

```
function yAFE = fun_AFE(isON, methods, name, signal, samplingRate)
%% Analog Filter Module
% Inputs:
%   −isOn:AFE is on ('on') | off ('off')
%   −methods: type of filter
%       −'analog': forward analog filter
%       −'filtfilt': zero−phase filtering
%   −name: type of analog filters
%       −'cheby': Chebyshev filter
%       −'butter': Butterworth filter
%       −'ellip': Elliptic filter
%   −signal: return pulse with noise
%   −sampleRate: sampling rate to model 'analog' signals
% Outputs:
%   −yAFE: filtered return signals


% Check if analog filter option is on
if strcmp(isON, 'off')
    yAFE = signal;
    return;
end

% Call different analog filter functions
switch methods
    case 'filtfilt'
        [sos, g] = fun_AnalogFilters(name, samplingRate);
        yAFE = filtfilt(sos, g, signal); % zero−phasing filtering
    case 'analog'
        [sos, g] = fun_AnalogFilters(name, samplingRate);
        yAFE = sosfilt(sos,signal)*g;
end
```

E.4.1 Function of Different Analog Filters

```matlab
function [sos, g] = fun_AnalogFilters(name, samplingRate)
%% Analog Filter Module
% Inputs:
%   -sampleRate: sampling rate to model 'analog' signal
%   -name: type of analog filters
% name:
%   'cheby': Chebyshev filter
%       -need to set:
%           -order: order
%           -rpPass: passband ripple [dB]
%           -fcut: cutoff frequency [Hz]
%           -Flag_Type: 'low': low-pass filter
%   'butter': Butterworth filter
%       -need to set:
%           -order: order
%           -fcut: cutoff frequency [Hz]
%           -Flag_Type: 'low': low-pass filter
%   'ellip': Elliptic filter
%       -need to set:
%           -order: order
%           -rpPass: passband ripple [dB]
%           -rpStop: stopband ripple [dB]
%           -fcut: cutoff frequency [Hz]
%           -Flag_Type: 'low': low-pass filter
% Output:
%   -sos: second-order section of the filter
%   -g: gain of the filter

fNy = samplingRate / 2; % Nyquist rate
fcut = 0.2e9; % cutoff frequency
wp = fcut/fNy;
Flag_Type = 'low';
order = 3;
rpPass = 5;
rpStop = 30;

switch name
    case 'cheby' % Chebyshev filter
        [z, p ,k] = cheby1(order,rpPass,wp,Flag_Type); % ...
            zero-pole-gain expression of the filter
        [sos, g] = zp2sos(z, p, k);
    case 'butter' % Butterworth filter
        [z, p ,k] = butter(order,wp,Flag_Type);
        [sos, g] = zp2sos(z, p, k);
    case 'ellip' % Elliptic filter
        [z, p ,k] = ellip(order,rpPass,rpStop, wp,Flag_Type);
        [sos, g] = zp2sos(z, p, k);
end
```

E.5 Function of ADC Module

```
function TOF = fun_ADC(tRet0, yRetClean0, yRet0, flag_detector)
%% ADC Timing Module
% Inputs:
%   -tRet0: time series
%   -yRetClean0: noise-free return signal
%   -yRet0: return signal
%   -flag_detector: detection algorithm: benchmark algorithm or ...
    NP detector: 'bm' | 'np'
% Outputs:
%   -TOF: measured TOF structure

global powerNoiseFloor ADCTriggerLevel sampleRate skip ResolADC
ADCTriggerLevel = 3*powerNoiseFloor; % ADC trigger threshold

%% ADC settings
ResolADC = 4e-10; % ADC time interval
skip = ResolADC*sampleRate;
bits = 8;

%% ADC sampling
% edge-triggering
% [tRetADC, yRetADC, ¬] = fun_ADCSamplingEdge(tRet0, yRet0, ...
    bits); % time and return pulse after ADC sampling

% level-triggering
[tRetADC, yRetADC, ¬] = fun_ADCSamplingLevel(tRet0, yRet0, bits); ...
    % time and return pulse after ADC sampling
yRetCleanADC = yRetClean0(1:skip:end); % noise-free return pulse ...
    after sampling
tRetCleanADC = tRet0(1:skip:end); % time of noise-free return ...
    pulse after sampling

%% Timing algorithm
switch flag_detector
    case 'bm'
        TOF = fun_detector_BM(tRetCleanADC, tRetADC,yRetCleanADC, ...
            yRetADC);
    case 'np'
        TOF = fun_detector_NP(tRetADC, yRetADC);
end

%% function of ADC edge-trigger mode
function [tRetADC, yRetADCDigi, ind] = fun_ADCSamplingEdge(tRet0, ...
    yRet0, bits)
global ADCTriggerLevel skip
lenADC = 31; % collection length [samples]
yRetADC = zeros(size(yRet0,1), lenADC+1+8*2);
tRetADC = zeros(size(yRet0,1), lenADC+1+8*2);
for i = 1: size(yRet0,1)
    ind = find(yRet0(i,:) > ADCTriggerLevel(i),1); % find first ...
        sample higher than the threshold
    startInd = ind-8*skip;
    endInd = ind+(lenADC+8)*skip;
    if endInd > size(yRet0,2)
        endInd = size(yRet0,2);
    end
    if startInd < 1
        startInd = 1;
    end
    temp = yRet0(i,startInd : skip :  endInd);
    yRetADC(i,1:length(temp)) = temp;
    tRetADC(i,1:length(temp)) = tRet0(startInd : skip :  endInd);
end
yRetADCDigi = fun_digitizer(yRetADC, bits, max(yRetADC,[],2)); % ...
   digitize amplitude

%% function of ADC level-trigger mode
function [tRetADC, yRetADCDigi, ind] = ...
   fun_ADCSamplingLevel(tRet0, yRet0, bits)
global ADCTriggerLevel skip
precursor = 16; % precursor length [samples]
postcursor = precursor;  % postcursor length [samples]
lenADC = 40;
yRetADC = zeros(size(yRet0,1), lenADC+precursor+ postcursor);
tRetADC = zeros(size(yRet0,1), lenADC+precursor+ postcursor);
% sampling
yRet0 = yRet0(:,1:skip:end);
tRet0 = tRet0(:,1:skip:end);
for i = 1: size(yRet0,1)
    ind = find(yRet0(i,:) > ADCTriggerLevel(i),1);
    startInd = find(yRet0(i,:) > ADCTriggerLevel(i),1) - ...
       precursor; % find first sample higher than the threshold
    endInd = find(yRet0(i,:) > ...
       ADCTriggerLevel(i),1,'last')+postcursor; % find first ...
       sample lower than the threshold
    if endInd > size(yRet0,2)
        endInd = size(yRet0,2);
    end
    if startInd < 1
        startInd = 1;
    end
    temp = yRet0(i,startInd:endInd);
    yRetADC(i,1:length(temp)) = temp;
    tRetADC(i,1:length(temp)) = tRet0(startInd :  endInd);
end
yRetADCDigi = fun_digitizer(yRetADC, bits, max(yRetADC,[],2)); % ...
   digitize amplitude

%% function of digitizing the signal amplitude
function yDigiY = fun_digitizer(signal, bit, yMax)
% Inputs:
%   -signal: transmitted pulse
%   -bit: ADC resolution
%   -yMax: max value of the signal
% Output:
%   -y: digitized pulse
for i = 1: size(signal, 1)
    yDigiY(i,:) = double(uencode(signal(i,:), bit, ...
       yMax(i),'signed'))/(2^(bit -1)) .* yMax(i);
end
```

E.5.1 Function of Benchmark Detection Algorithm

```
function TOF = fun_detector_BM(tRetClean, t,yRetCleanDigi, yRetDigi)
%% Benchmark Algorithm Module
% Inputs:
%   −tRetClean: time series of noise−free return signal
%   −t: time series of return signal
%   −yRetCleanDigi: digitized noise−free return signal in power
%   −yRetDigi: digitized return signal with noise in power
% Outputs:
%   −TOF structure:
%       −−isPulse: hypothesis of a signal: −1 (H0) | 1(H1)
%       −−tof: raw TOF measurement
%       −−d: raw distance measurement
%       −−name: detector name
%       −−prob: probability
%       −−tofMean: mean of TOF
%       −−tofStd: STD of TOF
%       −−dMean: mean distance
%       −−dStd: STD of distance
global  Nobs c cri ADCTriggerLevel tStart

%% Detector
% CRITERIA SETTING
% Criteria 1: NaT
cri.NaT_up = 52; % upper limit of NaT
cri.NaT_low = 10; % lower limit of NaT

% Criteria 2: pulse width
cri.width_mean = 5.33e−9; % mean pusle width
cri.tol = 0.1; % tolerate percentage

% Calculate NaT
meas.NaT = sum(yRetDigi > ADCTriggerLevel,2);

% Calcualte pulse width
[meas.width, cen] = fun_pulseWidth(t, yRetDigi, tRetClean, ...
    yRetCleanDigi);
meas.width = meas.width';

% Determine the hypothesis of a signal
TOF.tof = ones(size(yRetDigi,1),1)*−1;
isPulse1 = (meas.NaT ≥ 10 & meas.NaT ≤ 52);
isPulse2 = (abs(meas.width − cri.width_mean) ≤ ...
    cri.tol*cri.width_mean);
TOF.isPulse = isPulse1 & isPulse2;

%% Estimator: calcualte TOF (with compensation of time shift)
TOF.tof(TOF.isPulse == 1) = cen.x_above(TOF.isPulse == 1) − tStart;

%% Save data
TOF.d(TOF.isPulse == 1) = TOF.tof(TOF.isPulse == 1)*c/2;
TOF.d(TOF.isPulse ≠ 1) = NaN;
TOF.name = 'bm';
TOF.prob = sum(TOF.isPulse)/Nobs;
TOF.tofMean = nanmean(TOF.tof(TOF.tof≠−1));
TOF.tofStd = nanstd(TOF.tof(TOF.tof≠−1));
if sum(TOF.tof≠−1) ≤ 0.1*Nobs % if less than 10% of the signals ...
   are pulses, skip the test
    TOF.tofStd = NaN;
    TOF.tofMean = NaN;
end
TOF.dMean = TOF.tofMean*c/2;
TOF.dStd = TOF.tofStd*c/2;
end

%% function to calculate pulse width and x− and y− centroid
function [pw, cen] = fun_pulseWidth(t, y0, tRetClean, yRetCleanDigi)
global  Nobs ADCTriggerLevel
Factor = 2.817; % constant factor
y_cenup = zeros(1,Nobs); % y−centroid using data>0
A_cancel = zeros(1,Nobs); % area with cancellation between ...
   positive and negative area
for i = 1:Nobs
        ind = find(t(i,:)≠0);
    if isempty(ind)
        cen.x_above(i) = NaN;
        cen.x_above(i) = NaN;
        A_cancel(i) = NaN;
        y_cenup(i) = 0;
        continue
    end
    %% find all the data not equal to 0
    x = t(i,ind);
    y = y0(i,ind);
    y(1) = 0; y(end) = 0; % bound two ends of a polygon with zeros

    %%  calcualte the area and x−centroid and y−centroid
    yup = y; yup(yup<0) = 0; % extract data > 0
    polyup(i) = polyshape(x,yup); % define polygon > 0
    Aup = area(polyup(i)); % area of polygon >0
    [¬, y_cenup(i)] = centroid(polyup(i)); % calculate y_cen ...
       using data>0

    ydown = y; ydown(ydown>0) = 0; % extract data < 0
    polydown(i) = polyshape(x,ydown); % define polygon < 0
    Adown = area(polydown(i)); % area of polygon <0

    A_cancel(i) = Aup − Adown; % area
end

%% Calcualte pulse width
pw = A_cancel./y_cenup/Factor; % pulse width using data>0 to ...
   calculate y−centroid

% %% noise free signal
% xfree = tRetClean(tRetClean≠0);
% yfree = yRetCleanDigi(tRetClean≠0);
% % check if noise−free signal is below the ADC trigger
% if isempty(xfree)
%     xfree = [0,0,0];
%     yfree = [0,0,0];
% end
% polyinfree = polyshape(xfree,yfree);
% [x_cenfree,y_cenfree] = centroid(polyinfree);

%% Calculate TOF using x−centroid (data > threshold)

for i = 1: Nobs
    indAbove = find(y0(i,:) > ADCTriggerLevel(i),1); % find index ...
        of samples > ADC threshold
    if isempty(indAbove)
        cen.x_above(i) = NaN;
        continue;
    end
    startInd = find(y0(i,:) > ADCTriggerLevel(i),1);
    endInd = find(y0(i,:) > ADCTriggerLevel(i),1,'last');
    xab= t(i,startInd:endInd);
    yab = y0(i,startInd:endInd);
    yab(1) = 0; yab(end) = 0;
    polyinAb = polyshape(xab,yab); % define a polygon with data > ...
        threshold
    [cen.x_above(i),¬] = centroid(polyinAb); % calculate the ...
        x−centroid
end

%% Plot polygons
% ind = Nobs;
% figure
% hold on
% plot(polyup(ind))
% plot(polyinfree)
% plot(x,yup,'bs−')
% plot(xab(2:end−1),yab(2:end−1),'b−s','MarkerFaceColor','b')
% plot(x_cenfree,y_cenfree,'r⋆')
% plot(cen.x_above(ind),y_cenup(ind),'g⋆')
%
% xlabel('Time (s)')
% ylabel('Power (W)')
% legend('Polygon of return signal: data>0', 'Polygon of ...
    noise−free return', ...
%     'return signal: data>0','return signal: data>threshold',...
%     'Centroid of noise−free return','C_x: data>threshold, C_y: ...
    data>0',...
%     'Location','northeast')
end
```

E.5.2 Function of NP Detector

```
function TOF= fun_detector_NP(t, yRetDigi)
%% NP Detector Module
% Inputs:
%   -t: time series
%   -yRetDigi: digitized return signal with noise in power
% Outputs:
%   -TOF structure:
%       --isPulse: hypothesis of a signal: -1 (H0) | 1(H1)
%       --tof: raw TOF measurement
%       --name: detector name
%       --prob: probability
%       --tofMean: mean of TOF
%       --tofStd: STD of TOF
%       --dMean: mean distance
%       --dStd: STD of distance

global e integrationTime R0 Gain Nobs c tStart
%% Median filter
yRetDigi = medfilt1(yRetDigi, 3,[],2); % return pulse after ...
    median fitler with a kernel size of 3

%% Convert return pulse in power to number of electrons

NRetDigi = (yRetDigi * R0 * Gain) * integration Time /e; % return ...
    pulse
noiseFloorADC_power = 5.38e-7; % precalculated noise floor when ...
    laser is off
% noiseFloorADC_power = fun_calNoisefloor(yRetDigi); % function ...
    to calculate noise floor when laser is off
W = (noiseFloorADC_power*R0 *Gain) *integrationTime  /e; % ...
    convert noise power to number of electrons

%% Kernel generation
amplitude = 1; % set kernel amplitude to 1
kernel  = fliplr(fun_genKernel(amplitude));

%% Threshold
pfa = 0.001; % predefined PFA
u_T0 = 0; % mean of Tx under H0
var_T0 = W.^2'.* kernel*kernel'; % variance of Tx under H0
Q = erfcinv(2*pfa) * sqrt(2);
thres = Q.* sqrt(var_T0) + u_T0; % threshold of NP detector
if length(thres) == 1
    thres = repmat(thres,size(yRetDigi,1),1);
end

%% NP detector
TOF.isPulse = zeros(size(yRetDigi,1),1);
TOF.tof = ones(size(yRetDigi,1),1)*-1;
kerneltemp = kernel;
ResConv = struct('time',[],'data',[]);

for i = 1:size(yRetDigi,1)
```

```
    ind = find(t(i, t(i,:)≠0));
    if isempty(ind)
        TOF.isPulse(i) = 0;
        TOF.tof(i) = NaN;
        continue;
    end
    ttemp = t(i,ind);
    Ntemp = NRetDigi(i,ind);
    [Tmax(i), timeMax(i), ResConv(i)] = fun_convFFT(Ntemp, ...
       kerneltemp, ttemp);
end
%% Determine hypothesis of a signal
indPUlse = find(Tmax> thres');
TOF.isPulse(indPUlse) = 1;
TOF.tof(indPUlse) = timeMax(indPUlse);

%% SW approach
% for i = 1:size(yRetDigi,1)
%     ind = find(t(i, t(i,:)≠0));
%     if isempty(ind)
%         TOF.isPulse(i) = 0;
%         TOF.tof(i) = NaN;
%         continue;
%     end
%     ttemp = t(i,ind);
%     Ntemp = NRetDigi(i,ind);
%     T_full = conv(Ntemp, kerneltemp); % calculate convolution ...
   result
%     T_same = conv(Ntemp, kerneltemp,'same');
%     if max(T_full) > thres(i)
%         TOF.isPulse(i) = 1;
%         TOF.tof(i) = nanmean(ttemp(T_same == max(T_same)));
%     else
%         TOF.isPulse(i) = 0;
%         TOF.tof(i) = −1;
%     end
% end

%% Save data
TOF.name = 'np';
TOF.prob = sum(TOF.isPulse)/Nobs; % calculate probability of ...
   signal detection
TOF.tofMean = nanmean(TOF.tof(TOF.tof≠−1) − tStart); % calculate ...
   mean TOF
TOF.tofStd = nanstd(TOF.tof(TOF.tof≠−1)); % calculate std of TOF
if sum(TOF.tof≠−1) ≤ 0.1⋆Nobs % if less than 10% of the signals ...
   are pulses, skip the test
    TOF.tofStd = NaN;
    TOF.tofMean = NaN;
end
% Convert TOF to distance
TOF.dMean = TOF.tofMean⋆c/2;
TOF.dStd = TOF.tofStd⋆c/2;
end

%% function to calculate noise floor
function noisefloorADC = fun_calNoisefloor(signal)
```

```matlab
noisefloorADC = nanstd(signal,1,2);
end

%% function to generate kernel
function kernel = fun_genKernel(A)
global RMS ResolADC
tStart = -5*RMS;
tEnd = 5*RMS;
t = tStart : ResolADC : tEnd; % time series of transmit pulse
kernel = A*exp(-0.5*(t/RMS).^2); % transmit pulse
end

%% function of FFT approach of NP detector
function [convMax, timeMax, ResConv] = fun_convFFT(signal, ...
    kernel, time)
zeroPad = 4; % number of zeropaddings

tNew_min = time(1);
tNew_max = time(end);
dt = (tNew_max - tNew_min)/(length(time)-1);
timeNew = tNew_min:dt/(zeroPad+1):tNew_max;

LA = 2^nextpow2(length(kernel)+length(signal)-1);
Zeros = zeros(1,LA/2*zeroPad);

%% FFT
temp0 = fft(signal, LA,2); % calculate FFT of signal with time ...
    domain zero-padding
kernelFFT0 = fft(kernel, LA,2); % calculate FFT of kernel with ...
    time domain zero-padding
convFFT = kernelFFT0.*temp0; % dot product
convFFT = fftshift(convFFT); % center the result around zero
Ypad = [Zeros, convFFT,Zeros]; % frequency-domain zero-padding
convY = abs(ifft(Ypad)); % convert convolution result to time domain

ind = ceil((length(kernel)*(zeroPad+1)-1-zeroPad)/2)...
+(1:length(signal)*(zeroPad+1));
Conv = convY(ind)*(zeroPad+1); % recover amplitude
Conv = Conv(1:end-zeroPad); % time-domain truncation

convMax = max(Conv);
indMax = find(Conv == convMax); % find the peak of the ...
    convolution result
timeMax = timeNew(indMax(1)); % find the corresponding time
ResConv.data = Conv;
ResConv.time = timeNew;
end
```

E.6 Main Function (TDC)

```
clear
close all

%% Constants
global c sampleRate Nobs P0 TargetDistance skip
Nobs = 100; % number of observations in the Monte-Carlo experiment
c = 3E8; % speed of light [m/s]

%% Lidar settings
% laser source
P0 = 300; % Peak power of transmit pulse [W]
riseTime = 3.58E-9; % rise time [s]
sampleRate = 1e11; % sampling rate to model continuous 'analog' ...
    signal [Hz]

% Target characteristics
TargetDistance = 5; % Target distance [m]
timeTrue = 2*TargetDistance/c; % ground-truth TOF

%% Tranmit Pulse Module
% OUTPUTS: time series, transmit pulse and other laser parameters
[tTran, yTran, Energy, PowerAveg] = fun_PulseModel(P0,riseTime);

%% Propagation and Receiver Module
% OUTPUTS: return time series, return pulse with noise
tRet = tTran + timeTrue; % return time series
[SNRSquare, yRetClean, yRetNoisy] = ...
    fun_receiver(TargetDistance,yTran); % return pulse with noise

%% Timing Module
% TDC smoothing
% OUTPUTS: return signal after TDC smoothing
bw_TDC = 2e9; % TDC bandwidth
intcomp = 1/bw_TDC/2; % TDC 'integration' time
skip = floor(sampleRate*intcomp);
tRetAveg = tRet(:,1:skip:end); % time series after 'smoothing'
yRetAveg = yRetNoisy(:,1:skip:end); % return pulse after 'smoothing'

% Timing algorithms
% INPUTS: types of timing algorithm: leading-edge, TOT ...
    compensation or CFD:
% 'lead' | 'tot' | 'cfd'
% OUTPUTS: measured TOF and distance

Flag_TDC = 'lead';
timeStart = fun_TDC_Start(tTran,yTran, Flag_TDC); % measure start ...
    time
Results = fun_TDC(tRetAveg,yRetAveg, timeStart, Flag_TDC); % ...
    measure TOF

% Results
DistanceMean = Results.dMean; % mean distance measurement
DistanceSTD = Results.dStd; % std of distance measurement
```

E.7 Function of TDC Module

E.7.1 Function of TDC for Transmit Signals

```
function timeMeas = fun_TDC_Start(tRet, signal, Flag_TDC)
%% function to measure the start time
% Inputs:
%   -tRet: return time series
%   -signal: return pulse
%   -Flag_TDC: timing algorithm
% Outputs:
%   -timeMeas: measured TOF [s]

global powerNoiseFloor RMS TDCTriggerLevelGlobe
pfa = 0.001; % predefined PFA
TDCTriggerLevelGlobe = fun_TDCTrigger(mean(powerNoiseFloor), ...
    pfa); % calculate TDC trigger threshold

%% Measured real pfa
% pfaMeas = sum(sum(signal > ...
    TDCTriggerLevelGlobe))/size(signal,1)/size(signal,2);

switch Flag_TDC
    % leading-edge and TOT uses simple leading-edge detection to ...
        measure
    % the start time
    case 'lead'
        Vth =TDCTriggerLevelGlobe;
        VthReal = Vth;
        Vth = 0.3 * max(signal);
        tSamp = tRet;
        sigSamp = signal;
        y01 = zeros(1,length(sigSamp));
        y01(sigSamp > Vth) = 1;
        ind = find(y01==1,1);
        if (ind == 1)
            timeMeas = tSamp(ind);
        else
            % linear interpolation
            timeMeas = interp1([sigSamp(ind - 1), ...
                sigSamp(ind)],[tSamp(ind - 1), tSamp(ind)], Vth);
        end
        timeMeas = -sqrt(timeMeas^2 + 2*RMS^2 * ...
            log(Vth/VthReal)); % time stamp measured at the ...
            leading edge

    case 'tot'
        Vth =TDCTriggerLevelGlobe;
        VthReal = Vth;
        Vth = 0.3 * max(signal);
        tSamp = tRet;
        sigSamp = signal;
        y01 = zeros(1,length(sigSamp));
        y01(sigSamp > Vth) = 1;
        ind = find(y01==1,1);
        if (ind == 1)
            timeMeas = tSamp(ind);
        else
            % linear interpolation
            timeMeas = interp1([sigSamp(ind - 1), ...
               sigSamp(ind)],[tSamp(ind - 1), tSamp(ind)], Vth);
        end
        timeMeas = -sqrt(timeMeas^2 + 2*RMS^2 * ...
           log(Vth/VthReal)); % time stamp measured at the ...
           leading edge

        % CFD uses peak detection to measure the start time
    case 'cfd'
        timeMeas = tRet(signal == max(signal)); % time stamp ...
           measured at the peak
end
%% function to calculate TDC threshold
function TDCTriggerLevel= fun_TDCTrigger(W, pfa)
u_T0 = 0;
var_T0 = W^2;
Q = erfcinv(2*pfa) * sqrt(2);
TDCTriggerLevel = Q.* sqrt(var_T0) + u_T0;
```

E.7.2 Function of TDC for Return Signals

```
function TOF= fun_TDC(tRet, signal, timeStart, method)
%% TDC Module
% Inputs:
%   -tRet: return time series
%   -signal: return pulse
%   -timeStart: start time
%   -method: TDC methods
% method:
%   Flag_detection = 'lead':
%       -leading-edge detection, measure time at the leading edge ...
    (without offset compensation)
%       -fixed threshold
%   Flag_detection = 'tot':
%       -time-over-threshold(TOT) compensation
%       -fixed threshold
%   Flag_detection = 'cfd':
%       -CFD detection
%       -constant fraction is set at 30% of the pulse height
% Outputs:
%   TOF measurements

global Nobs c grids TDCTriggerLevelGlobe
grids = 1e-12; % TDC resolution
TDCTriggerLevel  = TDCTriggerLevelGlobe; % calculated in ...
    fun_TDC_Start.m

switch method
    case 'lead'
        timeStop = zeros(1,size(signal,1));
        for i = 1 : size(signal,1)
             % error handling: if threshold is larger than the ...
                 signal height
            if (TDCTriggerLevel > max(signal(i,:)))
                timeStop(i) = NaN;
                disp('!!! Error: Return power < Threshold. No ...
                    object detected !!!')
                continue;
            end
            % linear interpolation
            timeStop(i) = fun_interp(TDCTriggerLevel,tRet, ...
                signal(i,:));

        end
        % calculate TOF
        timeMeas = (timeStop - timeStart);

    case 'tot'
        timeMeas = fun_TOT(TDCTriggerLevel,tRet, signal, ...
            timeStart); % call TOT function

    case 'cfd' % CFD
        TDCTriggerLevelPerc = 0.3; % constant fraction
        timeMeas = fun_CFD(tRet, signal, TDCTriggerLevelPerc); % ...
            call CFD function
end

% save data
TOF.name = method;
TOF.tof = timeMeas;
TOF.d = timeMeas⋆c/2;
TOF.tofMean = nanmean(timeMeas);
TOF.tofStd = nanstd(timeMeas);
if sum(¬isnan(timeMeas)) ≤ 0.1*Nobs
    TOF.tofStd = NaN;
    TOF.tofMean = NaN;
end
TOF.dMean = TOF.tofMean⋆c/2;
TOF.dStd = TOF.tofStd⋆c/2;
```

E.7.3 Function of TOT Compensation

```
function timeMeas = fun_TOT(TDCTriggerLevel,tRet, signal, time0)
%% function of TOT compensation
% Inputs:
%   -tRet: return time series
%   -signal: return pulse
%   -TDCTriggerLevel: TDC trigger threshold
%   -time0: start time
% Outputs:
%   -timeMeas: measured TOF [s]

global RMS P0 grids
timeMeas = zeros(1,size(signal,1));

for i = 1 : size(signal,1)
    % error handling: if threshold is larger than the signal height
    if (TDCTriggerLevel > max(signal(i,:)))
        timeMeas(i) = NaN;
        disp('!!! Error: Return power < Threshold. No object ...
            detected !!!')
        continue;
    end

    %% measure time stamp and the power at the leading edge
    [timeLead(i), yLead(i)] = fun_interp(TDCTriggerLevel,tRet, ...
        signal(i,:));

    %% measure time stamp at the falling edge
    ind1 = find(signal(i,:) < TDCTriggerLevel);
    if isempty(ind1)
        timeMeas(i) = NaN;
        continue;
    end
    indFall = ind1(find(ind1 > size(signal(i,:),2)/2,1)); % find ...
        the index of the sample at the falling edge lower than the ...
        threshold
    t1 = tRet(indFall-1); y1 = signal(i,indFall-1);
    t2 = tRet(indFall); y2 = signal(i,indFall);
    tq = t1:grids:t2;
    yq = interp1([t1, t2], [y1, y2], tq);
    timeFall(i) = tq(find(yq<TDCTriggerLevel,1));
    yFall(i)= yq(find(yq<TDCTriggerLevel,1)); % power at the time ...
        stamp at the falling edge

    %%  Calculate TOT
    tot(i) = timeFall(i) -timeLead(i);
    if tot<0
        timeMeas(i) = NaN;
        continue;
    end

    %% calculate corresponding peak power
    PrEst = TDCTriggerLevel / exp(-(tot(i)/2)^2 / 2/ RMS^2);

    %% Calculate walk error
    walk(i) = abs(time0) - sqrt(time0^2 - 2 * RMS^2 * log(P0/PrEst));
    timeMeas(i) = timeLead(i) - walk(i) - time0;
end
```

E.7.4 Function of CFD Algorithm

```
function timeMeas = fun_CFD(tRet, signal, VthPerc)
%% Constant Fraction Discriminators (CFD) Model
% Inputs:
%   −tRet: return time series
%   −signal: return pulse
%   −VthPerc: constant fraction of the trigger point of the pulse ...
    peak
% Outputs:
%   −timeMeas: measured TOF [s]

%% Generate bipolar signal
global RMS sampleRate grids Nobs skip TDCTriggerLevelGlobe

CFDArmTrigger = TDCTriggerLevelGlobe⋆ones(Nobs,1); % CFD arming ...
    threshold
offsetTime = sqrt(−log(VthPerc) ⋆ 2 ⋆ RMS^2);
if ¬isreal(offsetTime)
    error('!!! Error: Vth of TDC is larger than signal. No time ...
        detected. Check Vth !!!')
end

% find delay time
NumberOffset = floor(offsetTime / (1/sampleRate)/skip);

% generate delayed signal
try
    yDelay = [zeros(size(signal,1),NumberOffset), signal(:,1:end ...
        − NumberOffset)];
catch ME
    if (strcmp(ME.identifier, 'MATLAB:Error using zeros: Size ...
        vector must be a row vector with real elements.'))
        msg = 'Threshold of TDC is larger than signal. No signal ...
            detected !!!';
        causeException = MException('MATLAB:myCode:dimensions',msg);
        ME = addCause(ME,causeException);
    end
    rethrow(ME)
end

% generate reversed and attenuated signal
yNeg = −VthPerc ⋆ signal;

% generate bipolar signal: delay + attenuated
yBipolar = yNeg + yDelay;

% generate logic signal of of bipolar signal

yBi01 = zeros(size(yBipolar,1),size(yBipolar,2));
yBi01(yBipolar > 0) = 1;

%% Measure TOF
% apply arming threshold
yArm01 = zeros(size(signal,1),size(signal,2));
```

```
yArm01(yNeg <- CFDArmTrigger) = 1;

% generate logic CFD signal
yCFD01 = (yBi01 & yArm01);
timeMeas = zeros(1,size(signal,1));

for i = 1 : size(signal,1)
    ind =find(yCFD01(i,:) ==1,1);
    if (isempty(ind) || ind == 1 || CFDArmTrigger(i) > ...
       1*max(signal(i,:)))
        timeMeas(i) = NaN;
        continue;
    end
    t1 = tRet(ind-1); y1 = yBipolar(i,ind-1);
    t2 = tRet(ind); y2 = yBipolar(i,ind);
    tq = t1:grids:t2;
    yq = interp1([t1, t2], [y1, y2], tq);
    if isempty(find(yq ≥0,1))
        timeMeas(i) = t2;
    else
        timeMeas(i) = tq(find(yq ≥0,1));
    end
end
```

E.7.5 Function of Linear Interpolation

```
function [time, P]= fun_interp(TDCTriggerLevel,tRet, signal)
%% function of linear interpolation
% Inputs:
%   -tRet: return time series
%   -signal: return pulse
%   -TDCTriggerLevel: TDC threshold
% Outputs:
%   -time: trigger time stamp
%   -P: power at the trigger time

ind =find(signal > TDCTriggerLevel,1);
if (isempty(ind) || ind == 1)
    time = NaN;
    P = NaN;
    return;
end

global grids % TDC resolution
t1 = tRet(ind-1); y1 = signal(ind-1); % find the data point ...
    before the trigger point
t2 = tRet(ind); y2 = signal(ind); % find the data point after the ...
    trigger point
tq = t1:grids:t2;
yq = interp1([t1, t2], [y1, y2], tq); % linear interpolation
if isempty(find(yq > TDCTriggerLevel,1))
    time = t2;
    P = y2;
else
    time = tq(find(yq > TDCTriggerLevel,1));
    P = yq((find(yq > TDCTriggerLevel,1)));
end
```