

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA
UNIVERSITÀ DEGLI STUDI DI FERRARA
UNIVERSITÀ DEGLI STUDI DI PARMA

MOTORVEHICLE UNIVERSITY OF EMILIA-ROMAGNA

Master Thesis
in
Advanced Automotive Electronic Engineering

Design and realization of novel measurement
procedures and a measurement setup for the
characterization of LiDARs for automotive
applications

Candidate:
Di Loro Giorgio

Advisor:
Prof. Ing. Stefano Cattini

Co-advisors:
Ing. Davide Cassanelli

Academic Year 2019/2020
Session: III

Abstract

Advanced Driver Assistance Systems (ADAS) and Autonomous Vehicles are driving the innovation of cars and technology altogether, introducing challenges that require more and more information of the surrounding environment in order to take correct decisions in safety-critical scenarios. In this framework, car manufacturers are looking forward to adopt the best sensors available on the market and integrate them inside their vehicles. LiDAR (Light Detection and Ranging), and in particular 3D Time-of-Flight (ToF) scanning LiDAR, is thus object of an increasing interest because of its capability to scan the surrounding environment and to represent it in a 3D map. This sensor is of paramount importance for tackling complex problems such as decision making in autonomous driving, and this is why vehicle manufacturers are interested in LiDARs that can ensure the best performance or present the overall best characteristics. Because of the novelty of this technology, however, most LiDAR manufacturers highlight just some features of their devices and do not provide a full, standardized characterization that could allow to compare different products. The aim of this thesis work is to design, realize and ultimately provide both novel measurement procedures and a measurement setup that allow to characterize any LiDAR, assessing the relevant features that manufacturers do not always provide and allowing the comparison between different models. The proposed measurements performed exploiting the setup, in particular, allow to characterize warm-up time, stability, quantization and range error. The modular nature of the unique system developed, however, allows to further expand its capabilities and to easily perform additional characterizations without the need to build a new specific setup from scratch.

Keywords: ADAS, Autonomous vehicles, LiDAR, ToF, Automotive, Measurement, Characterization.

Contents

Introduction	ix
1 Theoretical Background	1
1.1 LiDAR	1
1.1.1 Measurement Principles	3
2 Methods	7
2.1 Measurement Procedures	7
2.1.1 Warm-Up and Stability	8
2.1.2 Range Error and Quantization	12
2.2 Measurement Setup	14
2.2.1 Warm-Up and Stability	16
2.2.2 Range Error and Quantization	18
2.3 Software and Firmware	20
3 Results	23
3.1 Warm-Up and Stability	23
3.1.1 7 m Distance	24
3.1.2 21 m Distance	31
3.2 Range Error and Quantization	34
4 Conclusions	43
Acknowledgements	47
Appendix A - Software Guides	49
I Ubuntu 16.04 LTS (Xenial Xerus) guide	49
II ROS for Ubuntu 16.04 guide	50

III	MATLAB ROS guide	52
IV	MRS6000 guide	53
V	Interferometer (HP5527A) guide	54
Appendix B - MATLAB code		59
I	Scan example	59
II	Crop and Fit example	59
III	Warm-Up and Stability, 7m acquisition	60
IV	Warm-Up and Stability, 7m plot	61
V	Warm-Up and Stability, 7m elaboration	62
VI	Warm-Up and Stability, 21m acquisition	63
VII	Warm-Up and Stability, 21m plot	64
VIII	Warm-Up and Stability, 21m elaboration	65
IX	Angular crop and data elaboration	66
X	Range Error and Quantization acquisition	69
XI	Range Error and Quantization elaboration	71
Bibliography		73

List of Figures

1	SICK MRS6000 3D scanning LiDAR.	xi
2	Example of a pointcloud obtained from the MRS6000.	xi
1.1	High level block scheme of a ToF LiDAR.	3
1.2	Measurement of the time elapsed between transmitted and received pulse during normal operation.	4
2.1	Instrument under test (SICK MRS6000) mounted on its bracket.	15
2.2	Portion of the custom rail used to align the target to the optical axis of the system.	16
2.3	Planar reflective target mounted on a sliding carriage.	17
2.4	Fluke 51 II thermocouple thermometer.	17
2.5	High level schematic of the Measurement Setup used for Warm-Up and Stability characterization.	18
2.6	HP5527A interferometer used for accurate distance measurements. The main components (laser head, laser detector, beam splitter and transducer electronics) are highlighted.	19
2.7	Newport M-BBR 1-1I corner cube used as reflective target for the laser beam emitted by the interferometer.	19
2.8	High level schematic of the Measurement Setup used for Range Error and Quantization measurements.	20
3.1	Top view of the planar fit of the target at 7 m.	25
3.2	Mean distance measured by the LiDAR at around 7 m with echo filter set to "first echo".	25
3.3	Normalized plot of mean distance measured over time at around 7 m with echo filter "first" during warm-up.	26

3.4	Results of Warm-Up Time measurement at 7 m, with echo filter "first". . .	27
3.5	Normalized plot of mean distance measured over time at around 7 m with echo filter "all" during warm-up.	28
3.6	Results of Warm-Up Time measurement at 7 m, with echo filter "all". . . .	29
3.7	Normalized plot of mean distance measured over time at around 7 m with echo filter "last" during warm-up.	30
3.8	Results of Warm-Up Time measurement at 7 m, with echo filter "last". . .	30
3.9	Mean distance measured by the LiDAR at around 21 m with echo filter set to "first echo".	31
3.10	Results of Warm-Up Time measurement at 21 m, with echo filter "first". . .	32
3.11	Normalized plot of mean distance measured over time at around 21 m with echo filter "all" during warm-up.	33
3.12	Normalized plot of mean distance measured over time at around 21 m with echo filter "last" during warm-up.	34
3.13	Mapping of the planar target on concentric spheres with quantized radii. .	36
3.14	Schematic representation of how a single IUT channel scans its surroundings.	37
3.15	Quantization characterization at around 1.6 m, echo filter "first".	37
3.16	Quantization characterization at around 1.6 m, echo filter "last".	38
3.17	Quantization characterization at around 7 m, echo filter "first".	38
3.18	Quantization characterization at around 7 m, echo filter "last".	38
3.19	Distribution of the errors at around 1.6 m.	39
3.20	Distribution of the errors at around 7 m.	40
3.21	Distribution of the single-point analysis errors at around 1.6 m, echo filter "first".	40
3.22	Distribution of the single-point analysis errors at around 1.6 m, echo filter "last".	41
3.23	Distribution of the single-point analysis errors at around 7 m, echo filter "first".	41
3.24	Distribution of the single-point analysis errors at around 7 m, echo filter "last".	42
4.1	Sliding carriage motor unit prototype.	45

List of Tables

1.1	Technologies available for LiDARs based on the ToF measurement.	2
3.1	Number of Warm-Up and stability measurements performed for each different measurement condition, defined by target distance and LiDAR operating mode (selected echo filter).	23
3.2	Number of Range Error and Quantization measurements performed for each different measurement condition, defined by target distance ranges and LiDAR operating mode (selected echo filter).	35

Introduction

The race to develop Autonomous Vehicles (AV) and Advanced Driver Assistance Systems (ADAS) is on. Major vehicle producers and tier-1 suppliers, as well as disruptive newcomers, are accelerating AV and ADAS development efforts to get ahead in the race. Manufacturers are outfitting modern cars with a wide array of advanced control and sensing functions. Collision warning and avoidance systems, blind-spot monitors, lane-keeping assistance, lane departure warning, and adaptive cruise control are examples of established features that assist drivers and automate certain driving tasks, making driving a safer and easier experience. The increasing complexity of ADAS-related tasks and the challenges that are currently being tackled in the development of AV require vehicles to be able to sense the surrounding environment with a high level of fidelity in order to take safety critical decisions and perform the correct actions required by the situation detected. LiDARs are quickly becoming a fundamental technology to achieve this goal: together with cameras, RADARs and ultrasonic sensors, they are a crucial enabler of sensor fusion, which allows to obtain detailed and redundant spatial information by exploiting different sensors, based on different measurement principles and having different limitations. A high fidelity representation of the environment, however, requires reliable measurements, and the possibility of comparing LiDAR characteristics is absolutely necessary to design a system that is able to succeed in the tasks addressed by these cutting-edge technologies. Unfortunately, this comparison cannot be performed easily: LiDAR producers provide their own metrics to define the sensor's performance, making the evaluation of differences between sensors produced by different manufacturers (or even belonging to different model families) difficult to be obtained. As of today, no national/international standards or measurement procedures have been adopted for characterizing 3D LiDARs for automotive applications, mainly because each measurement application has its own peculiarities. Given the interest in these devices, many standards have been recently proposed or updated [1]-[7], but these have been developed for other areas and are, therefore, unable to fully analyze specific aspects

of potential interest for automotive applications. The majority of LiDAR manufacturers, on the other hand, do not give a comprehensive description of the performance of their systems, limiting the discussion to few device metrics like "measuring interval", "horizontal angular resolution", "vertical angular resolution" and "error", often without providing the metrological definition of these quantities. In this scenario, research efforts are producing a considerable number of studies that have been recently published on the characterization [15]-[27] of LiDAR systems and subsystems, confirming the high interest in these devices. The rapid growth of the technology is causing a gap of knowledge related to important aspects of the characterization of these instruments that are not yet covered by standards, provided by OEMs or analyzed in literature. The aim of this thesis work is to contribute to fill these gaps, first of all investigating which metrological characterizations could be of major interest for automotive applications. The next steps consist in designing, realizing and ultimately exploiting novel measurement procedures and a measurement setup that enable the characterization of the most important features of any type of LiDAR, including in particular LiDARs for automotive applications. The most relevant achievements of this work involve: the definition of the general measurement procedures that describe how to perform characterization measurements independently on the examined LiDAR and how to achieve repeatability by ensuring controlled operating conditions; the design and successive realization of the custom measurement setup that can be exploited for performing characterization measurements and further expanded to enable even more characterizations than those proposed in the following; the realization of the software used to interface the instruments with a PC in order to comfortably analyze measurement data and evaluate LiDAR features; the final test of the procedures, setup and software on a state of the art LiDAR sensor performed to validate the work and give an example of application. In particular, the LiDAR used to develop and validate the methods proposed in this work is the SICK MRS6000 3D scanning LiDAR shown in Fig. 1, one of the most popular LiDAR systems of its kind: this sensor is able to map the surrounding environment by means of 3D pointclouds, where each point is described by its Cartesian coordinates with respect to the sensor and by the intensity of the light reflected by it. An example of 3D pointcloud is provided in Fig. 2.

The characterization proposed in this work is aimed at investigating four relevant features of LiDAR sensors: warm-up time, stability, quantization, range error. Despite the importance of these four parameters, to the best of my knowledge, there are currently no



Figure 1: SICK MRS6000 3D scanning LiDAR.

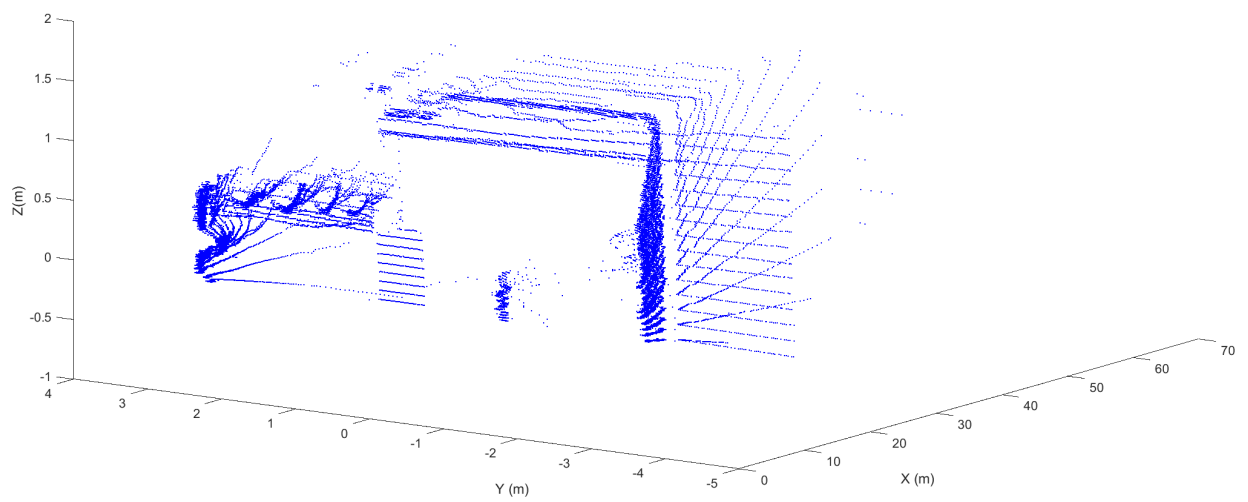


Figure 2: Example of a pointcloud obtained from the MRS6000.

standards or studies aimed at analyzing these aspects. These characteristics are some of the most important parameters for describing the behavior of the sensor, and are thus very useful to compare different LiDARs. Warm-up time and stability characterize the behavior of LiDARs over time: the former describes the time needed by the instrument to reach steady state operation; the latter describes the ability of the sensor to maintain a stable measure during continuous operation. Quantization and range error, on the other hand, give information about how the LiDAR actually senses the surrounding environment and are able to assess its performance in terms of error with respect to more accurate sensors, as well as limitations due to the hardware-software components of the LiDAR itself. Regardless of their importance, as mentioned before, these features have not been currently investigated in depth by neither international standards nor OEMs, and this is why they were chosen as the object of the characterization discussed in this thesis.

The content is organized in four chapters. Chapter 1 contains a brief theoretical introduction required to understand LiDAR operation and Time-of-Flight measurement principles in general, as well as a small background of contemporary literature works that tackle the task of characterizing LiDARs.

Chapter 2 describes the materials and methods adopted to achieve the results proposed in this introduction. In particular, Section 2.1 gives an extensive perspective on the developed measurement procedures, whereas Section 2.2 contains the details on the measurement setup.

Chapter 3 describes the results of the measurements performed on the investigated LiDAR, validating the measurement procedures and setup presented previously and, thus, highlighting the qualities of both design and realization of the system. The instrument Finally, Chapter 4 contains the conclusions, a summary of all the work described throughout this thesis, and an insight on future developments.

Chapter 1

Theoretical Background

This chapter provides the basic information and the theoretical knowledge required to fully understand the methods and results proposed in Chapter 2 and Chapter 3.

1.1 LiDAR

Light Detection And Ranging (LiDAR) is a sensing method that detects objects and maps their distances by illuminating them with an optical beam generated by a laser source and measuring the characteristics of the reflected signal captured by a photodiode [28] [29]. The information on target distance is extracted from light signals' parameters such as pulse power at the receiving end, round-trip time, phase shift or pulse width. Because of their measurement principle and of their overall features, LiDAR sensors are also called "LADARs" (Laser Detection and Ranging sensors), "laser scanners" or "laser radars". For what concerns the automotive field, 3D scanning LiDARs based on Time-of-Flight distance measurements are the most common solution for autonomous driving applications. 3D ToF scanning LiDARs steer one or more collimated laser beams simultaneously emitted and map the environment sequentially by knowing the current steering direction and measuring the relative time of flight [28] [29]. An overview of the main LiDAR technologies currently being investigated for automotive applications is presented in Table 1.1.

The interest in characterization, performance evaluation and comparison of different LiDAR systems is now experiencing a sudden growth due to the significant request of this type of sensors for industrial and automotive applications [29]. The increasing demand, mainly driven by ADAS and autonomous driving, made 3D LiDARs experience a real boom in popularity and, consequently, their diffusion and performances are undergoing

Technology	Type	Description
Spinning	scanning	The beam is steered exploiting one (or more) rotating part/s
MEMS	scanning	The beam is steered exploiting MEMS (micro-electromechanical system) micromirrors
OPA	scanning	The beam is steered exploiting an array of antennas (optical phase arrays — OPA)
Flash	flash	The beam is not steered to sequentially scan the surroundings but expanded to illuminate the entire scene. The 3D-image of the surrounding is thus obtained exploiting an array of detectors

Table 1.1: Technologies available for LiDARs based on the ToF measurement.

a fast process of significant growth, marked by sensible innovation and technological advances. The extreme rapidity of this technological innovation trend resulted in the current lack of national/international standards or common measurement methods aimed at characterizing 3D LiDARs for automotive applications. This is an important practical issue: ADAS designers and car manufacturers often face obstacles related to the incompleteness of the metrological characterizations that they would require to correctly develop their systems; moreover, they frequently are not able to compare different LiDARs available on the market because of the lack of a common ground that would allow to assess the suitability for the final application (benchmarking).

In this framework, several national and international standards for possible applications of LiDAR systems have been recently proposed or updated. ISO standards ISO 17123-1:2014 [3], ISO 17123-9:2018 [4], ISO 17123-4:2012 [5], ISO 17123-6:2012 [6] for example, focus on evaluating the performance and characteristics of LiDARs for civil engineering and surveying measurements. Another international standard, ISO/TS 19159-2:2016 [6] is focused on the characterization of airborne LiDARs for remote sensing of geographic information. ASTM E2938-15 [1] investigates standard test methods for evaluating relative range measurement performance of medium-range 3D imaging systems whereas ASTM E3125-17 [2] focuses on point-to-point distance measurement errors for the same class of systems. These standards are of undoubted value, because they allow to define an initial common ground for metrological characterization of some types of ranging systems, they are still dependent on the specific application and do not provide a general methodology for characterizing any type of LiDAR. Moreover, as of today, specific characterization standards for automotive applications have not yet been defined.

Because of this, LiDAR OEMs frequently provide only general metrics for describing the performance of their products, typically limiting the characterization to:

- measuring interval, i. e. the minimum and maximum distances that the instrument is able to measure in the specified operating conditions, with specifications that may vary between different manufacturers;
- vertical and horizontal angular resolution, also in this case the specific definition is given by each manufacturer;
- measurement "error", that in most cases is just expressed quantitatively, without a complete metrological definition.

Because of the reasons stated above, literature presents some recent studies that try to focus on LiDAR design and characterization [8]-[27]. These studies, however, do not treat the characterization of LiDARs for automotive application extensively and, in particular, assess different metrics and features with respect to those characterized in this work, as will be further described in Chapter 2.

1.1.1 Measurement Principles

A high-level block scheme of a Time-of-Flight LiDAR sensor is shown in Fig. 1.1. The "Laser Transmitter" block drives the laser and issues the start signal to the "Time interval Measurement" block when the light pulse is emitted; the receiver photodiode collects the reflection of the beam hitting the target and the "Receiver" block issues the stop signal to the "Time interval Measurement" block.

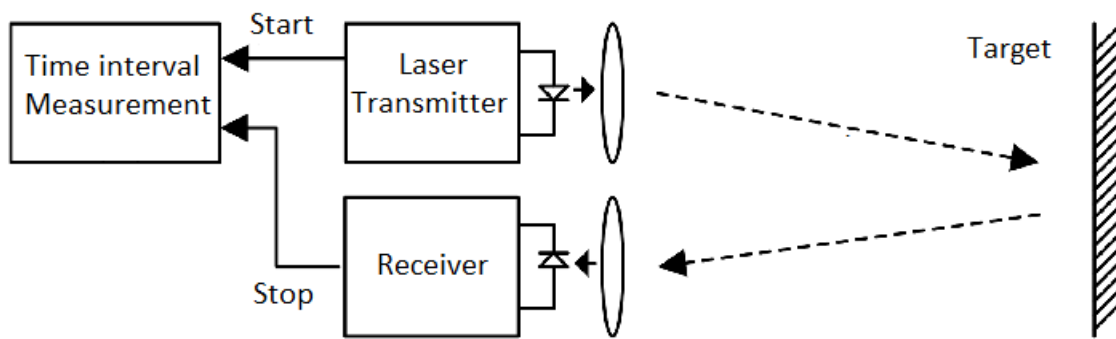


Figure 1.1: High level block scheme of a ToF LiDAR.

The information collected by the sensor in the first place, hence, is the round-trip time of the light pulse. This information is related to the distance of the target [28] by eq. 1.1:

$$t_{RT} = \frac{2 \cdot d_T}{c} \quad (1.1)$$

where t_{RT} is the round-trip time, d_T is the target distance, c is the speed of light in the medium and the factor 2 takes into account the fact that light must travel twice the distance that separates instrument and target. Fig. 1.2 depicts the time measurement principle during normal operation in a possible automotive application. Since the laser transmitter and the receiver photodiode are embedded in the same housing, they are usually placed close to each other.

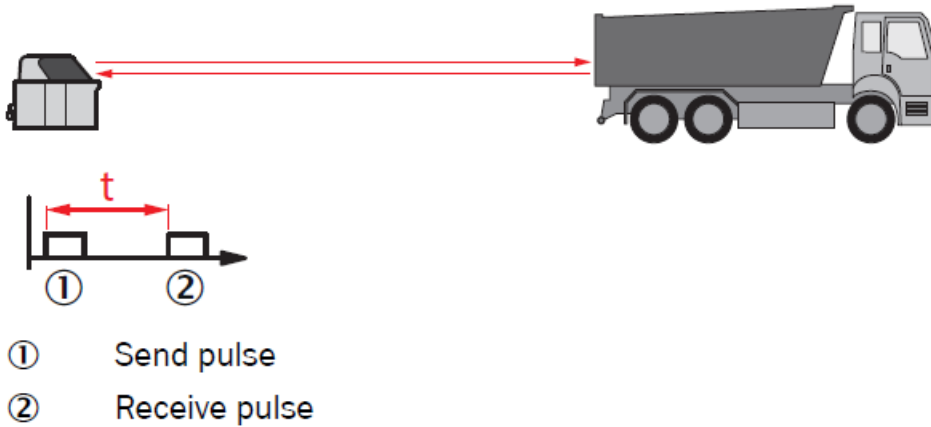


Figure 1.2: Measurement of the time elapsed between transmitted and received pulse during normal operation.

The acquired information about the round-trip time is exploited by the sensor to calculate target distance, according to eq. 1.2.

$$d_t = \frac{t_{RT} \cdot c}{2} \quad (1.2)$$

It is worth noting that the speed of light in a medium c depends on the medium's refractive index n and is always lower or equal to the speed of light in vacuum c_0 , as shown in eq. 1.3.

$$c = \frac{c_0}{n} \quad (1.3)$$

Distance resolution mainly depends on the bandwidth of the LiDAR: the minimum distance that the sensor is able to resolve depends on the duration of the transmitted light pulse t_P , which is related to bandwidth B as in eq. 1.4. Eq. 1.5 shows the expression that links distance resolution and system bandwidth.

$$t_P = \frac{1}{B} \quad (1.4)$$

$$\Delta d = \frac{c \cdot t_P}{2} = \frac{c}{2B} \quad (1.5)$$

If a Time to Digital Converter (TDC) is used by the LiDAR to obtain t_P , then the quantization of the TDC is directly translated in a quantization of the axial distance measurements, as in eq. 1.5.

Chapter 2

Methods

This Chapter describes the measurement procedures developed in order to characterize LiDARs as well as the setups, hardware and software that have been designed and developed for performing the measurements. The content of the chapter is organized as follows:

- Section 2.1 contains the description of the measurement procedures designed and adopted for characterizing and comparing the performance of LiDAR sensors;
- Section 2.2 contains the description of the measurement setups designed, realized and exploited for characterizing LiDAR sensors;
- Section 2.3 contains the description used to interface the PC used for data acquisition and processing with the rest of the measurement setup.

2.1 Measurement Procedures

This Section describes the measurement procedures specifically designed to characterize some of the most relevant parameters of LiDAR sensors that are not always provided by manufacturers or investigated in literature, even though they provide important information on the behavior of the instrument. Each measurement procedure follows the same structure:

- measurement goal, identification of the measurand and its definition;
- main influence quantities and their effects on the measure;
- measurement conditions and setup (explored in depth in Section 2.2);

- data acquisition and analysis methods.

2.1.1 Warm-Up and Stability

The first quantity to be measured (measurand) is the Warm-Up Time of the LiDAR sensor. Warm-Up Time is defined by the manufacturer as "the amount of time it takes for the sensor to attain its highest accuracy or performance level once the voltage supply has been applied. The warm-up time helps to achieve a thermo-mechanical balance in the measuring system as well as an optimal operating temperature of the electronic components". However, a significant amount of manufacturers do not provide any information about the Warm-Up Time of their products. In other cases, producers do state the Warm-Up Time of the device, but they only provide generic information that doesn't allow to understand which deviation from nominal performances should be expected before the Warm-Up Time has elapsed. In order to scientifically assess this characteristic of the Instrument Under Test (IUT), a more specific definition of what "highest accuracy or performance level" means must be provided. Moreover, the measured quantities and accuracies of interest must be identified. Since the goal is to characterize the LiDAR mainly through distance measurements, the repeatability (related to measurement precision) of distance measurements has been chosen as the metric for estimating the Warm-Up Time. To give a technical definition of "highest accuracy or performance level" and at the same time to define a standard procedure that can be applied to other LiDARs, Warm-Up Time was defined as the time it takes to the measured value of a property to stay within a tolerance interval (TI) defined by the upper (TU) and lower (TL) tolerance limits, implementing decision rules similar to the guarded acceptance and guarded rejection used in conformity assessment as described in JCGM 106:2012 [30], that will be mentioned before presenting the results in Chapter 3. The second measurand is the Stability of the LiDAR sensor. This quantity is defined as the "property of a measuring instrument, whereby its metrological properties remain constant in time. Stability may be quantified in several ways: in terms of the duration of a time interval over which a metrological property changes by a stated amount; in terms of the change of a property over a stated time interval." (definition provided by JCGM 200:2012 – International Vocabulary of Metrology, VIM [31]). Since the characterization of the LiDAR is being performed mainly through distance measurements, the definition adopted for Stability in this thesis work is the latter, "the change of a property over a stated time interval", quantified by means of the experimental standard deviation of the distance (the mean distance of the target) over a fixed amount of time starting from when the Warm-Up

Time has elapsed, thus when the LiDAR reaches its rated operating condition. This is the reason why stability measurements are performed right after Warm-Up Time measurements, therefore the two procedures share the same setup and methodologies, as well as the same influence quantities. Moreover, since Warm-Up Time is not known beforehand, the duration of the measurement must be large enough to contain both the Warm-Up Time and the maximum time span of interest for the evaluation of the Stability (ideally, the amount of time in which the instrument would be employed in normal operating conditions).

Influence quantities that can affect Warm-Up Time and Stability measurements are:

- **Target distance** - the SNR (signal to noise ratio) is affected by the distance at which the target is placed, thus a greater target distance (lower SNR) could have an impact on the Warm-Up Time and Stability measurement. Because of this, it is necessary to define different distance ranges (short range and long range, eventually medium range) at which the target must be placed and to use the same distances for all the different LiDARs whose Warm-Up Time and Stability must be characterized, in order to be able to compare the measurement results obtained by different sensors;
- **Ambient temperature** - the temperature of the environment in which the IUT performs its operation influences the Warm-Up Time and Stability measures. This happens both because the sensor's initial temperature is related to the temperature of the environment where it operates and because temperature may influence the refractive index of air. Since temperature variations can be significant throughout the long time required by Warm-Up and Stability measurements, environmental temperature must be monitored in order to fully characterize measurement conditions. Moreover, different sensors should be characterized when operating in the same temperature conditions or reducing the discrepancies between different operating conditions as much as possible, in order to obtain comparable results;
- **Ambient light** - differences in light levels may influence LiDAR measurements, thus the system must be characterized in an environment with controlled lighting levels. Moreover, different sensors should be characterized when operating in the same lighting conditions or reducing the discrepancies between different operating conditions as much as possible, in order to obtain comparable results;
- **Target reflectance** - differences in targets' reflectance may affect the SNR, thus a single, fully characterized target must be selected for performing the Warm-Up

Time and Stability measurement of all the different LiDARs under test. Performing measurements on targets with different reflective properties might be an interesting way to further investigate sensors' features, as long as the target is fully characterized;

- **Angle of reflection** - the angle at which the laser beam is reflected can affect the SNR, thus the alignment between target and sensor must be ensured throughout the measurements [32];
- **Air composition, humidity** - the medium through which beams travel is characterized by a refraction index (dependent on the medium's physical composition and conditions) that determines the speed of LiDAR beams, thus the atmosphere in which the measurement is performed should be completely characterized to distinguish between measurement errors and drifts due to the speed of the beams being lower than the speed of light in vacuum, that would make measured distance an overestimation;
- **Shape and size of the target** - LiDARs are not able to detect objects that are significantly smaller than the laser spot that illuminates them, thus this effect becomes relevant when the target is at a very large distance from the IUT or when the adopted target is small with respect to the resolution of the sensor.

Since target distance might affect both Warm-Up Time and Stability, two main distances have been identified for investigating the behavior of the IUT: a target distance equal to 7 meters is exploited for evaluating the measurands at close range, whereas a distance of 21 meters is used for evaluating the long range behavior. Further measurement conditions and operating modes are related to the specific IUT and to its functionalities, thus they will be discussed in Section 2.2 when talking about the LiDAR analyzed in this thesis work. An initial pointcloud acquisition and a planar fit of the detected target points must be performed for each different measurement condition and every time the target is moved at a different distance, in order to determine and eliminate any systematic error (possible tilting of the target with respect to the optical axis of the system). Moreover, for each measurement condition, the complete measurement procedure must be performed at least 3 times, in order to obtain an estimate, however rough, of the type A measurement uncertainty of Warm-Up Time and Stability.

Considering a typical operating condition to last several hours, even 8 hours or more in some cases, in order to guarantee that Warm-Up Time plus the 8 hours considered are reasonably contained inside the temporal window analyzed in this characterization, a total duration

of 15 hours has been selected for carrying out Warm-Up and Stability measurements. The whole data acquisition process has, thus, a total duration of 15 hours (900 minutes) with each acquisition being performed every minute, in order to have a large amount of data available for evaluating the metrics related to the characterization of the sensor. Every minute, a pointcloud (in which each point is defined by its Cartesian coordinates x , y , z , and by the intensity of the reflected light) must be acquired and spatially filtered in order to extract the information on target distance. Spatial filtering is performed for two reasons: first of all because of memory optimization (it allows to discard most of the points that don't belong to the target before storing them in memory), but most importantly because it reduces the complexity of the data set by reducing the amount of information that will be processed for characterizing the IUT. The quantity of interest is the distance of points that belong to the target, associated to its respective time of acquisition, which starts when the IUT is powered on.

At the end of the 15 hour long acquisition process, 900 spatially filtered pointclouds (one per minute) are available for processing. The information belonging to one single channel of the IUT that hits the target is extracted to allow the comparison of ideally any LiDAR, since generally each sensor has a different amount of channels. Measured target distance $\bar{z}(t)$ is then evaluated according to eq. 2.1, as the mean of the distances (z coordinates in the reference frame adopted) of the N points of the target available at each time stamp.

$$\bar{z}(t) = \frac{1}{N} \cdot \sum_{i=1}^N z_i(t) \quad (2.1)$$

Concurrently, the relative experimental standard deviation of the mean is evaluated according to eq.2.2.

$$s_{\bar{z}}(t) = \sqrt{\frac{1}{N(N-1)} \sum_{i=1}^N [z_i(t) - \bar{z}(t)]^2} \quad (2.2)$$

Eq 2.2 clearly shows that increasing the number of samples N , $s_{\bar{z}}$ decreases. Thus, in order to be able to compare measurements performed at different distance ranges with the same IUT, as well as measurements performed with different IUTs, it is important to select a specific number of points N and to adopt the same value for all the measurements and characterizations that must be performed. Further elaboration of the data obtained from eq. 2.1 and eq. 2.2, as well as the results of Warm-Up Time and Stability measurements are presented in Chapter 3.

2.1.2 Range Error and Quantization

The first measurand is the range error, defined as the difference between the target distance measured by the IUT and the one measured by an instrument that provides a higher degree of accuracy, i.e. the interferometer that will be described in Section 2.2.

The second feature of the IUT investigated in this subsection is the quantization of distance measurements performed internally by the LiDAR, assessed by evaluating the distribution of measurements in the different quantization bins defined by the instrument.

Influence quantities that can affect Range Error and Quantization measurements are:

- **Target distance** - differences in target distance may vary the repeatability of distance measurements and the distribution of points in the different quantization bins, thus multiple ranges must be considered for characterizing the instrument;
- **Ambient light** - differences in light levels may influence LiDAR measurements, thus the system must be characterized in an environment with controlled lighting levels. Moreover, different sensors should be characterized when operating in the same lighting conditions or reducing the discrepancies between different operating conditions as much as possible, in order to obtain comparable results;
- **Target reflectance** - differences in targets' reflectance may affect the distance measurement thus a single, fully characterized target must be selected for performing the Range Error and Quantization measurement of all the different LiDARs under test;
- **Angle of reflection** - the angle at which the laser beam is reflected can impact the distance measurement, thus the alignment between target and sensor must be ensured throughout the measurements. Quantization is heavily affected by the possible non-orthogonality of the target with respect to the optical axis, which must be avoided;
- **Shape and size of the target** - LiDARs are not able to detect objects that are significantly smaller than the laser spot that illuminates them, thus this effect becomes relevant when the target is at a very large distance from the IUT or when the adopted target is small with respect to the resolution of the sensor.

Since target distance might affect both Range Error and pointcloud arrangement, two main subsets of distances have been identified for investigating the behavior of the IUT: a set of target distances around 1.6 meters is exploited for evaluating the measurands at very close range, whereas a set of distances around 7 meters is used for evaluating them at

a longer range. Further measurement conditions and operating modes are related to the specific IUT and to its functionalities, thus they will be discussed in Section 2.2. An initial pointcloud acquisition and a planar fit of the detected target points must be performed for each different measurement condition and operating mode to ensure that the system is aligned and that the target is orthogonal to the optical axis.

Data acquisition is performed over a range of 49 centimeters, in order to characterize the behavior of the IUT over distance: 25 individual pointclouds are acquired at a fixed position and then the target is moved 10 millimeters away from the LiDAR. The acquisition process described is repeated 50 times (thus over a total range of 49 centimeters) meaning that, at the end of data collection, $25 \cdot 50 = 1250$ pointclouds are available for processing, specifically 25 pointclouds for each of the 50 different distance ranges. During acquisition, each pointcloud is spatially filtered in order to extract the information on the target, as described in the procedure for Warm-Up Time and Stability.

Again, the information belonging to one single channel of the IUT that hits the target is extracted, and then the x, y and z data is converted in spherical coordinates. This is done because the LiDAR actually measures the axial distance from its laser source/detector couple, mapping the environment on concentric spheres with quantized radii, whereas x, y and z coordinates depend on the azimuth and elevation angles at each acquisition and do not allow to appreciate the quantization performed by the IUT. In order to characterize Quantization, for each acquisition n carried out at a distance d , the number of points $N(n, d, r_{bin})$ falling in each specific radial bin r_{bin} and belonging to the extracted channel is evaluated. For each position d , its average value \bar{N} and the experimental standard deviation of the mean $s_{\bar{N}}$ are evaluated according to eq. 2.3 and eq. 2.4, where n_{PC} is the number of pointclouds available at each distance (25).

$$\bar{N}(d, r_{bin}) = \frac{1}{n_{PC}} \cdot \sum_{n=1}^{n_{PC}} N(n, d, r_{bin}) \tag{2.3}$$

$$s_{\bar{N}}(d, r_{bin}) = \sqrt{\frac{\sum_{n=1}^{n_{PC}} [N(n, d, r_{bin}) - \bar{N}(d, r_{bin})]^2}{n_{PC}(n_{PC} - 1)}} \tag{2.4}$$

In order to characterize Range Error, the distance d_{IUT} estimated by the instrument under test is calculated with the expression shown in eq. 2.5, where n_{points} is the total number of points detected in each single acquisition and $n_{r_{bin}}$ is the total number of bins considered

for the measurement.

$$d_{IUT}(n, d) = \frac{1}{n_{points}} \cdot \sum_{i=1}^{n_{rbin}} [N(n, d, r_{bin}) \cdot r_{bin}] \quad (2.5)$$

For each target distance d , the mean distance \bar{d}_{IUT} and the respective experimental standard deviation of the mean are obtained with eq. 2.6 and eq. 2.7.

$$\bar{d}_{IUT}(d) = \frac{1}{n_{PC}} \cdot \sum_{n=1}^{n_{PC}} d_{IUT}(n, d) \quad (2.6)$$

$$s_{\bar{d}_{IUT}}(d) = \sqrt{\frac{\sum_{n=1}^{n_{PC}} [d_{IUT}(n, d) - \bar{d}_{IUT}(d)]^2}{n_{PC}(n_{PC} - 1)}} \quad (2.7)$$

Further elaboration of the information obtained from this analysis, as well as the results of Range Error and Quantization measurements are presented in Chapter 3.

2.2 Measurement Setup

In order to implement the characterizations described in Chapter 2, a custom measurement setup was designed, developed and finally realized. A high level schematic of the complete setup is shown in Fig. 2.8. The common setup exploited in all the characterizations performed in this work consists of several elements:

- the instrument under test (IUT) with its mounting bracket, shown in Fig. 2.1;
- a custom 21 meters long rail system based on aluminum extrusion profiles, a portion of which is shown in Fig. 2.2, that allows to change the distance between target and IUT while maintaining the target orthogonal to the optical axis;
- a custom multi-axis stage that allows to position the IUT and to finely tune its orientation;
- a planar reflective target mounted on a sliding carriage, shown in Fig. 2.3;
- a PC used for collecting the pointclouds from the IUT.

The IUT characterized in this thesis work is the MRS6000 3D scanning LiDAR by SICK. The operating mode of the MRS6000 cannot be changed: the device evaluates up



Figure 2.1: Instrument under test (SICK MRS6000) mounted on its bracket.

to four echo signals for each measuring beam to deliver reliable measurement results even under adverse conditions. The only degree of freedom in the evaluation of the output of the measurement is the selection of the software echo filter to be applied, between the available ones. The echo filter screens out unwanted measurement data and signals: it can be chosen whether the first, the last, or all echoes are output. In order to fully characterize the IUT, all of these configurations have been examined in this work.

The rail system allows to change the distance of the target in a range of 21 meters while maintaining the alignment with the IUT, as well as other optical instruments that might be adopted to evaluate the performance of the IUT itself. The vertical aluminum extrusion profiles that sustain the structure are not fixed to the ground, they rely on screws that can be moved to finely adjust rail height and rail tilting. The horizontal aluminum extrusion profiles, on the other hand, are fixed to the underlying structure and present a central slot that allows to connect and rigidly translate the sliding carriage over the whole length of the rail.

The custom multi-axis stage allows to fix the IUT or any other instrument for range measurement to it and to finely tune the whole block's orientation. The stage has a total of six degrees of freedom in its movement: it is not fixed to the ground, thus it can be

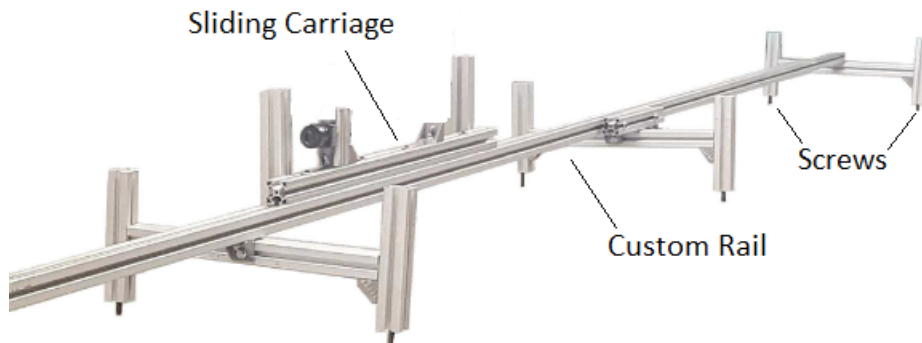


Figure 2.2: Portion of the custom rail used to align the target to the optical axis of the system.

translated along two directions and oriented with the desired yaw angle; it exploits the same screw system of the rail, thus its height, as well as roll and pitch angles can be modified.

The planar reflective target adopted is the 24" by 24" hardboard model TB4 by Thorlabs, which is fixed on a sliding carriage that can be rigidly translated along the rail to modify its distance from the IUT while maintaining the alignment to the optical axis of the system. The target absorbs visible light (this is why it looks black), but actually shows a reflective behavior to infrared light, that is the one used by the majority of LiDARs. In particular, it is a Lambertian reflective target for infrared light.

2.2.1 Warm-Up and Stability

As specified in Section 2.1, Warm-Up and Stability measurements require a long acquisition time to be carried out, thus temperature variations and lighting conditions must be controlled, or at least monitored, to prevent them from affecting measurement results. The thermometer used to monitor the temperature of the environment throughout the whole duration of the measurements is the Fluke 51 II thermocouple thermometer shown in Fig. 2.4. This instrument allows to keep track of the maximum, minimum and average temperatures registered during the 15 hour long measurement period. The setup is located in a closed environment, isolated from external light, thus all the measurements are performed with the same lighting conditions. Fig. 2.5 shows a high level schematic of the setup used for Warm-Up Time and Stability measurements, where the only missing blocks are the thermometer and the PC used for data acquisition.



Figure 2.3: Planar reflective target mounted on a sliding carriage.



Figure 2.4: Fluke 51 II thermocouple thermometer.

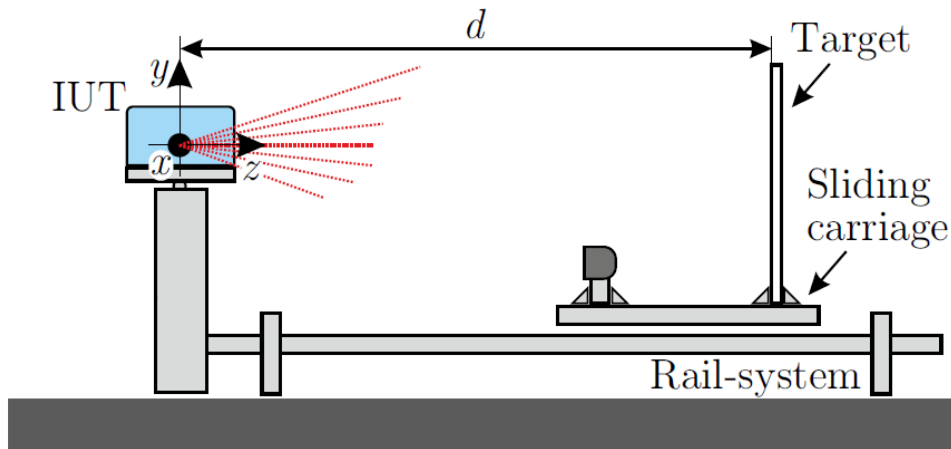


Figure 2.5: High level schematic of the Measurement Setup used for Warm-Up and Stability characterization.

2.2.2 Range Error and Quantization

Range error measurements require an instrument that provides a higher degree of accuracy than the IUT to be carried out. In addition to the common setup described before, the interferometer shown in Fig. 2.6 is thus exploited to obtain an accurate distance reference.

The laser head emits a beam that goes through the beam splitter, where a small fraction of the signal is deflected to the laser detector, which uses it as a reference. A significant part of the original beam is not influenced by the beam splitter and hits the reference target, the corner cube depicted in Fig. 2.7. The corner cube reflects the original beam back to the beam splitter, that redirects part of the received signal to the laser detector. The interferometric signal generated by the interaction between original beam and reflected beam allows the measurement of relative distances with a degree of precision down to nanometers.

Fig. 2.8 shows a high level schematic of the setup used for Range Error and Quantization measurements. Since the origin of the IUT and interferometer reference frames does not coincide, there is a fixed and unknown offset between the distances measured by the two. This was estimated as the intercept of the linear interpolation of the values provided by the IUT and INT translating the target along the rail. By doing so, however, any potential offset error affecting the IUT has been compensated.

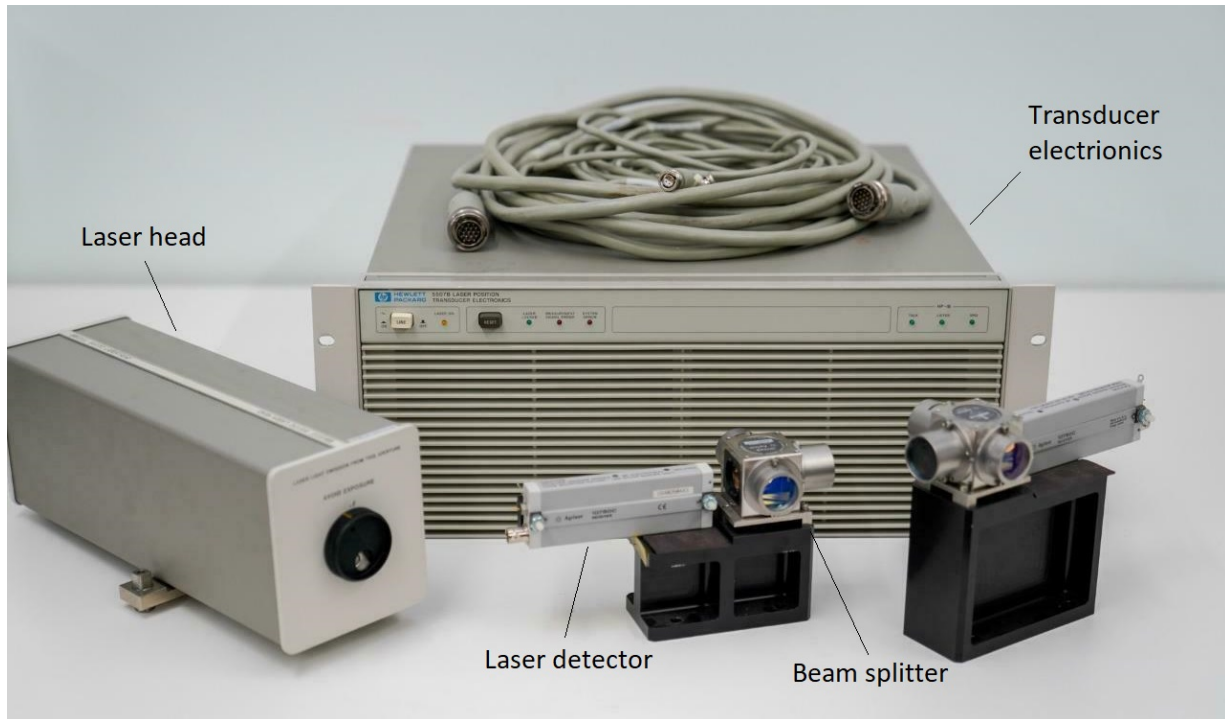


Figure 2.6: HP5527A interferometer used for accurate distance measurements. The main components (laser head, laser detector, beam splitter and transducer electronics) are highlighted.



Figure 2.7: Newport M-BBR 1-1I corner cube used as reflective target for the laser beam emitted by the interferometer.

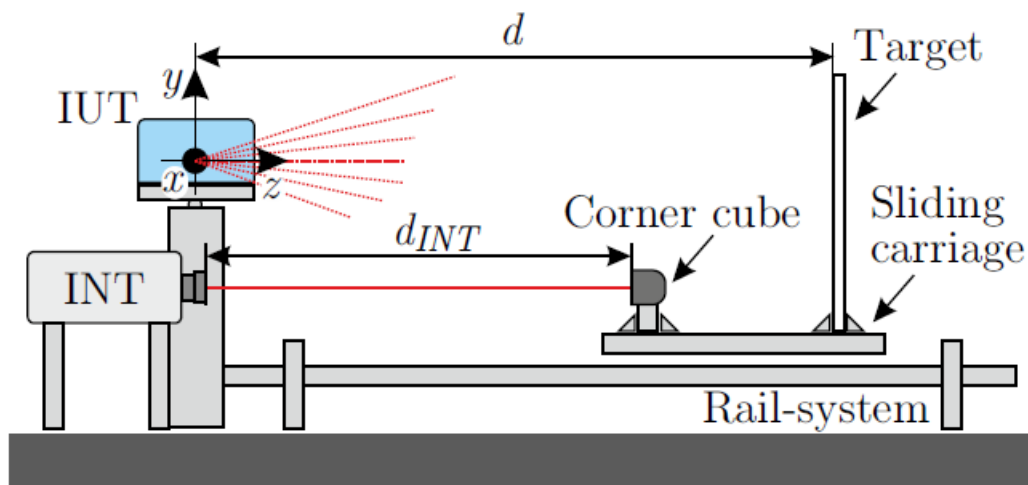


Figure 2.8: High level schematic of the Measurement Setup used for Range Error and Quantization measurements.

2.3 Software and Firmware

This Section gives a brief introduction to the software realized to interface the measurement setup described in Section 2.1 with the PC used to collect and analyze the pointcloud data extracted from the IUT, as well as the software tools used to communicate with the instrumentation. All the software introduced in this Section is fully available in the Appendices.

The operating system used to interface the PC with the LiDAR is Ubuntu, version 16.04 "Xenial Xerus" in particular. The adoption of this operating system was performed because it allows to exploit the powerful features of the Robot Operating System (ROS), a set of software libraries and tools designed to communicate with electro-mechanical systems, to generate a network where each node is a device, to issue commands and to collect data. Moreover, specific software libraries that allow to interface ROS with the SICK MRS6000 are available for Linux powered devices. Appendix A contains the software guides produced during this thesis work to correctly setup a PC for interacting with the custom measurement setup proposed. The HP5527A interferometer described above was interfaced with the PC through a GPIB-USB-B connector and specific software libraries available for Linux systems, also described in Appendix A.

Since pointcloud data can be easily represented and stored in matrices, MATLAB has been used to acquire the data from the IUT and process the information to obtain the metrics

used for the characterization displayed in Chapter 3. MATLAB's ROS Toolbox, in particular, allowed to interface the specific software realized for extracting data with the LiDAR, enabling the automatic collection of static measurements (realized thus for Warm-Up and Stability assessment). Appendix B contains the MATLAB software written to perform all of the acquisition and elaboration required to obtain the results proposed in the following. An important result of this work is the complete automation of the measurement process for what concerns static measurements (Warm-Up and Stability), which don't require the target to be moved along the rail system. The possibility of realizing a fully automatic dynamic characterization (Range Error and Quantization) is discussed between the improvements presented in Chapter 4.

Chapter 3

Results

This chapter presents measurement results obtained using the methods, materials and software described in Chapter 2. In particular, Section 3.1 provides the results of the Warm-Up and Stability characterization performed in different operating conditions, Section 3.2 presents the results of Range Error and Quantization characterization. The IUT that provided these results is the SICK MRS6000 described in Section 2.2.

3.1 Warm-Up and Stability

Table 3.1 summarizes target distances and LiDAR operating conditions analyzed for assessing the Warm-Up Time and Stability features of the IUT.

Target distance	Echo filter setting	Number of measurements performed
7 m	First echo	3
7 m	All echoes	3
7 m	Last echo	3
21 m	First echo	3
21 m	All echoes	3
21 m	Last echo	3

Table 3.1: Number of Warm-Up and stability measurements performed for each different measurement condition, defined by target distance and LiDAR operating mode (selected echo filter).

The metric obtained for IUT characterization according to Section 2.1, i.e. measured target distance $\bar{z}(t)$ is used to obtain a new metric, the difference between measured target distance at instant t and the initial measured target distance (at $t = t_0$), normalized by the

initial measured target distance: $\frac{\bar{z}(t) - \bar{z}(t_0)}{\bar{z}(t_0)}$. This metric allows to assess the Warm-Up Time of the instrument, once the Tolerance Interval related to the measurement is defined. The metric used for assessing the stability of the IUT is the standard deviation of the mean of measured distances, evaluated starting when the Warm-Up Time has elapsed. Assuming that warm-up ends in the first n_w samples, Stability is calculated as the standard deviation of the mean shown in eq. 3.1, where n_{TOT} is the total number of recorded pointclouds (900) and \bar{z}_{steady} is the mean distance evaluated according to eq. 3.2

$$s_{\bar{z}_{steady}} = \sqrt{\frac{1}{n_{TOT} - n_w} \sum_{i=n_w}^{n_{TOT}} [\bar{z}(i) - \bar{z}_{steady}]^2} \quad (3.1)$$

$$\bar{z}_{steady} = \frac{1}{n_{TOT} - n_w + 1} \sum_{i=n_w}^{n_{TOT}} \bar{z}(i) \quad (3.2)$$

The following Subsections describe the results of the measurements, classified by target distance and LiDAR operating mode.

3.1.1 7 m Distance

The first measurement performed once the target was placed at a distance of around 7 meters includes the planar fit of the points acquired by the target, to ensure the correct alignment of the system and no tilting with respect to the optical axis (as described in the measurement procedure developed in Section 2.1). The top view of the planar fit of the target is shown in Fig. 3.1. All the measurements described in this Subsection have been performed without moving the target, the only thing that differentiates them is the operating condition of the IUT, defined by the echo filter applied to the beams.

"First" echo filter

As mentioned in Section 2.1 the data analyzed and presented in this Chapter belongs to a single channel of the IUT. Fig. 3.2 shows the behavior of measured target distance over time, during the 15 hour long acquisition period. The resulting behavior displayed by the LiDAR is an underestimation of target distance during warm-up, followed by a stabilization around the 100 minute mark. Fig. 3.3 shows the plot of the normalized difference with respect to the initial acquisition $\frac{\bar{z}(t) - \bar{z}(t_0)}{\bar{z}(t_0)}$.

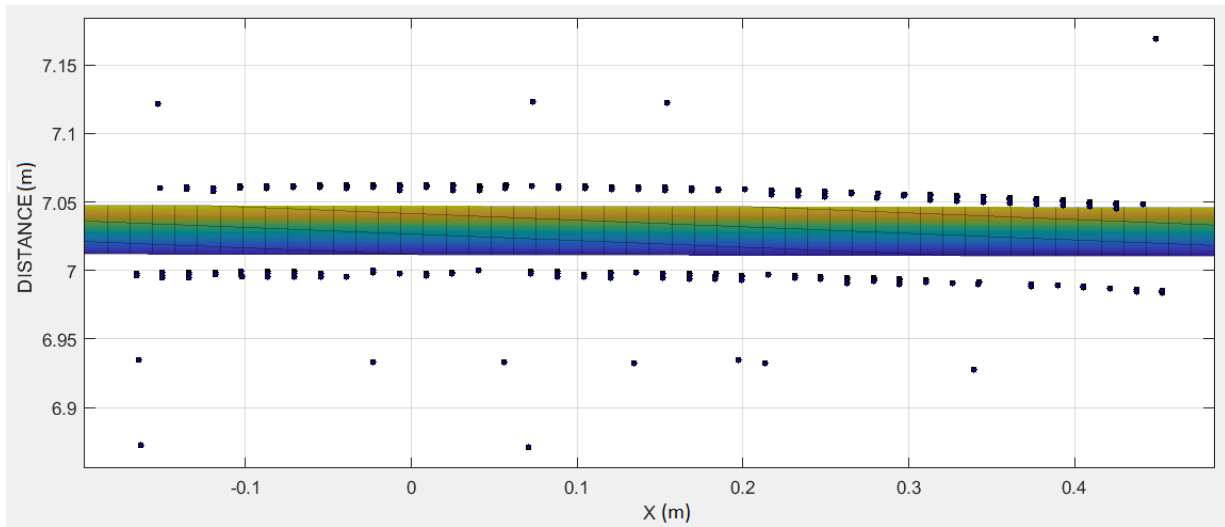


Figure 3.1: Top view of the plane that fits a set of target points detected by the IUT. The x axis shows the horizontal displacement with respect to the LiDAR (x coordinate of the defined reference frame). The y axis shows the distance (z coordinate of the reference frame). The fit shows that the IUT and the target are aligned (the target is orthogonal to the optical axis of the system, i.e. orthogonal to the z axis defined in the reference frame). It is possible to see that the distance value is about 7 meters.

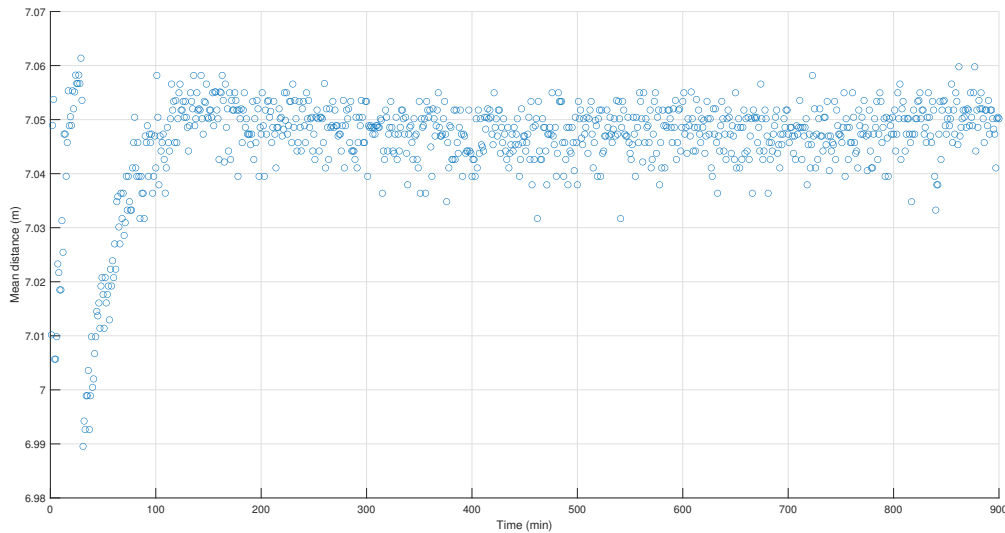


Figure 3.2: Mean distance measured by one channel of the LiDAR over time with echo filter set to "first echo". The measurements qualitatively stabilize after a period of around 100 minutes at a value of about 7.05 meters.

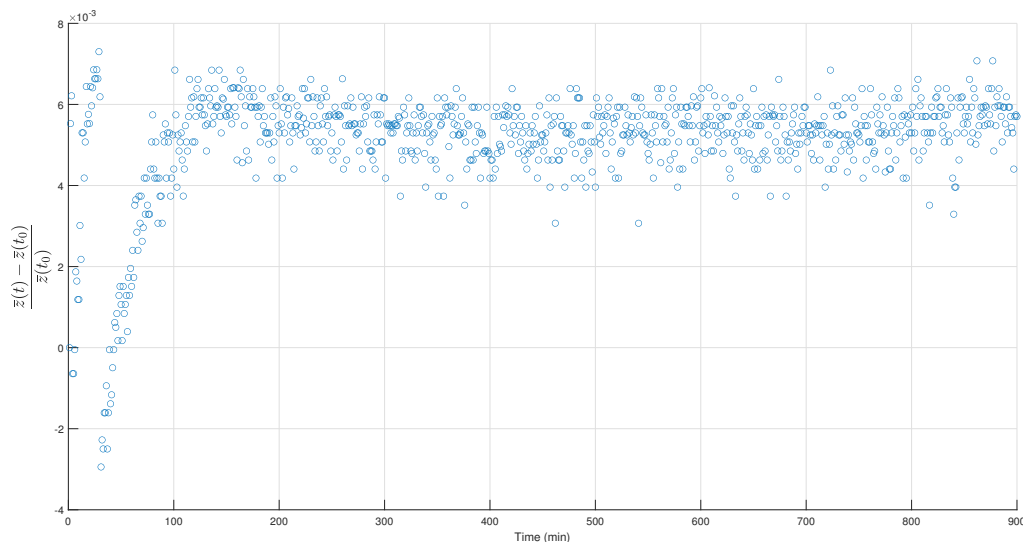


Figure 3.3: MRS6000 difference between the mean z value at instant t and mean z value at instant t_0 normalized by the mean z value at t_0 for the target at a distance of around 7 meters, with echo filter "first".

The application of the decision rule mentioned in Section 2.1, performed selecting specific Tolerance Limits TU and TL around the steady state value \bar{z}_{steady} (calculated according to eq. 3.2 with $n_w = 90$ samples) and considering guard bands dependent on the standard deviation of the mean of measured distances at each time stamp, allows to determine the Warm-Up Time of the instrument [33]. This is obtained as the time it takes to measured distance to stay within the most restricted acceptance interval, the one delimited by the portion of guard bands that falls inside the acceptance interval. Fig. 3.4 shows the results of the characterization by means of the ratio $\frac{\bar{z}}{\bar{z}_{steady}} \cdot 100$.

As stated in Table 3.1, the measurement has been carried out for a total of 3 times under repeatability conditions, and the estimation of the Warm-Up Time of the IUT for the described operating conditions that satisfies the defined requirements can be safely assessed at a value of 90 minutes.

The Stability estimated according to eq. 3.1 is $\frac{S_{\bar{z}_{steady}}}{\bar{z}_{steady}} = 0.12\%$.

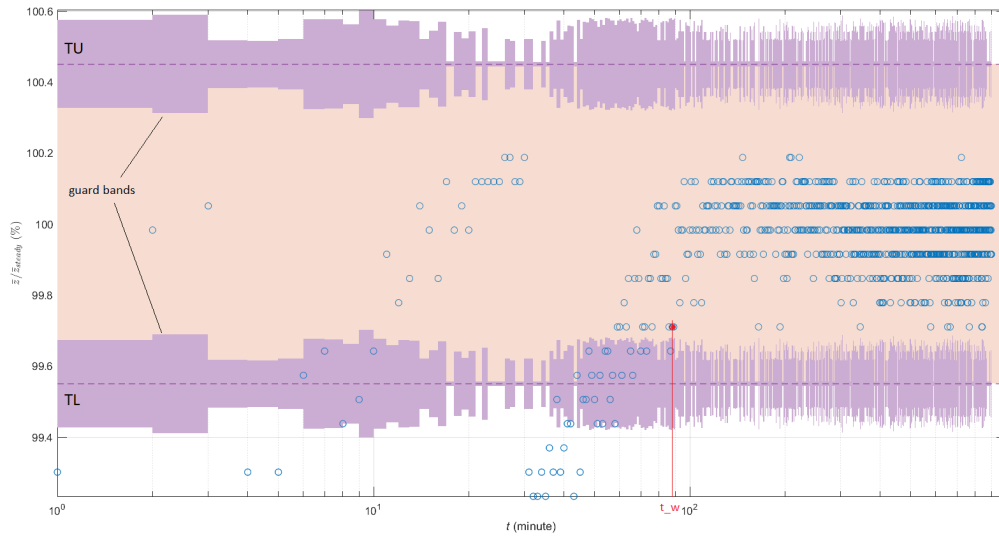


Figure 3.4: Result of one Warm-Up Time measurement performed in the specified operating conditions. TU and TL are the limits of the tolerance interval defined around the steady state value \bar{z}_{steady} . The guard bands depend on the standard deviation of the mean of measured distances at each time stamp and are represented by purple regions. The value t_w is obtained as the time of the red sample, which is the time it takes to measured distance to stay within the orange portion of the plot (tolerance interval restricted by the guard bands). In this case the value t_w is 88 minutes.

"All" echo filter

The following results were obtained by changing the operating mode of the LiDAR through the modification of the echo filter applied, whereas the rest of the setup remained unchanged. Fig. 3.5 shows the plot of the normalized difference with respect to the initial acquisition $\frac{\bar{z}(t) - \bar{z}(t_0)}{\bar{z}(t_0)}$. Fig. 3.6 shows the results of the characterization by means of the ratio $\frac{\bar{z}}{\bar{z}_{steady}} \cdot 100$.

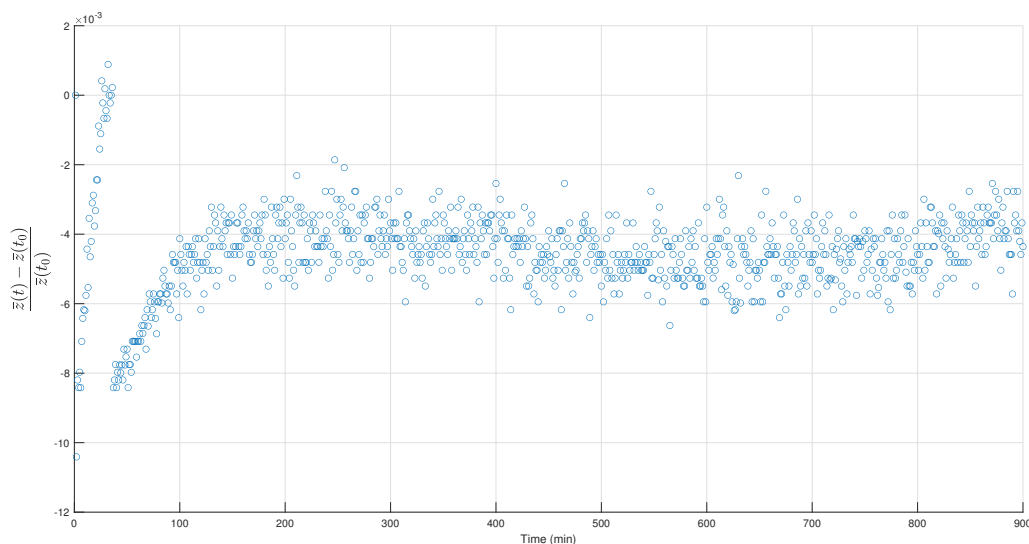


Figure 3.5: MRS6000 difference between the mean z value at instant t and mean z value at instant t_0 normalized by the mean z value at t_0 for the target at a distance of around 7 meters, with echo filter "all".

As stated in Table 3.1, the measurement has been carried out for a total of 3 times under repeatability conditions, and the estimation of the Warm-Up Time of the IUT for the described operating conditions that satisfies the defined requirements can be again safely assessed at a value of 90 minutes. Occasionally, some mean distance measurement fall outside of the acceptance interval, it is not trivial to understand why this happens, but reasonably it should not happen because of the instrument not having reached Warm-Up Time.

The Stability estimated according to eq. 3.1 is $\frac{s_{\bar{z}_{steady}}}{\bar{z}_{steady}} = 0.12\%$.

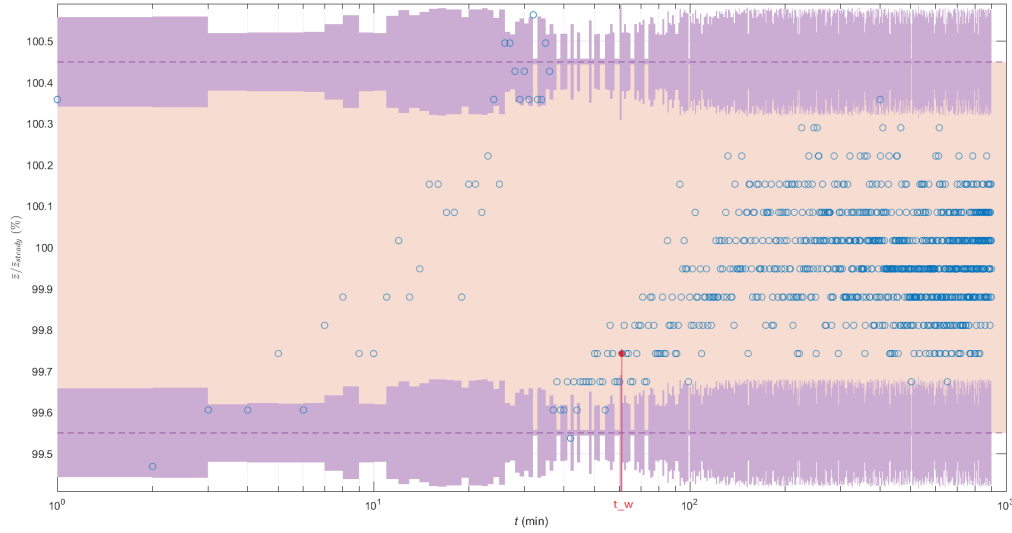


Figure 3.6: Result of one Warm-Up Time measurement performed in the specified operating conditions. The value t_w required for measured distance to stay within the orange portion of the plot is 61 minutes, but the plot shows clearly that the distance value stabilizes around the 100% ratio at about 90 minutes.

"Last" echo filter

The following results were obtained by changing the operating mode of the LiDAR through the modification of the echo filter applied, whereas the rest of the setup remained unchanged. Fig. 3.7 shows the plot of the normalized difference with respect to the initial acquisition $\frac{\bar{z}(t) - \bar{z}(t_0)}{\bar{z}(t_0)}$. Fig. 3.8 shows the results of the characterization by means of the ratio $\frac{\bar{z}}{\bar{z}_{steady}} \cdot 100$.

As stated in Table 3.1, the measurement has been carried out for a total of 3 times under repeatability conditions, and the estimation of the Warm-Up Time of the IUT for the described operating conditions that satisfies the defined requirements can be again safely assessed at a value of 90 minutes. Mode "last" shows an higher Warm-Up Time in average, but this seems due to possible measurement errors or background noise, rather than the instrument not having completed its warm-up.

The Stability estimated according to eq. 3.1 is $\frac{s_{\bar{z}_{steady}}}{\bar{z}_{steady}} = 0.11\%$.

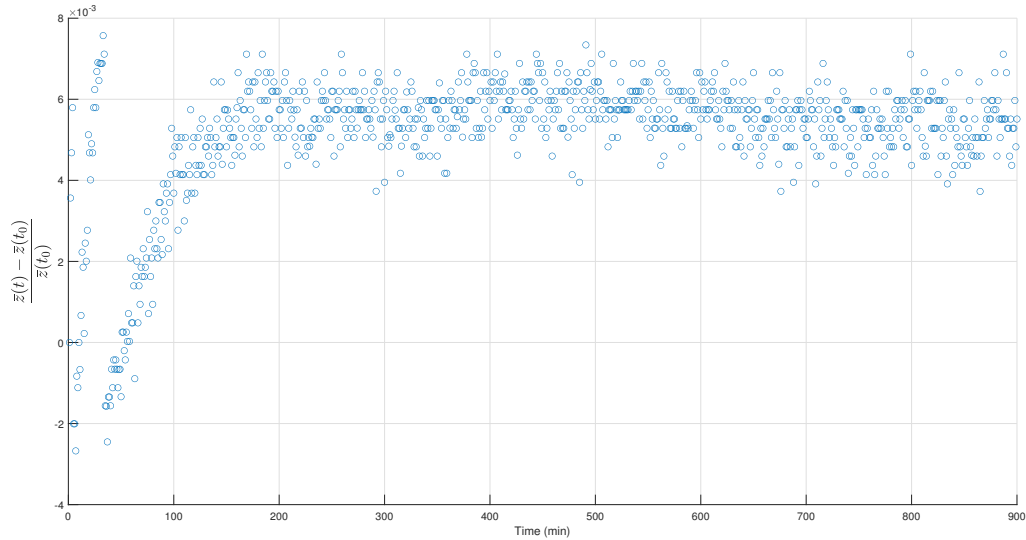


Figure 3.7: MRS6000 difference between the mean z value at instant t and mean z value at instant t_0 normalized by the mean z value at t_0 for the target at a distance of around 7 meters, with echo filter "last".

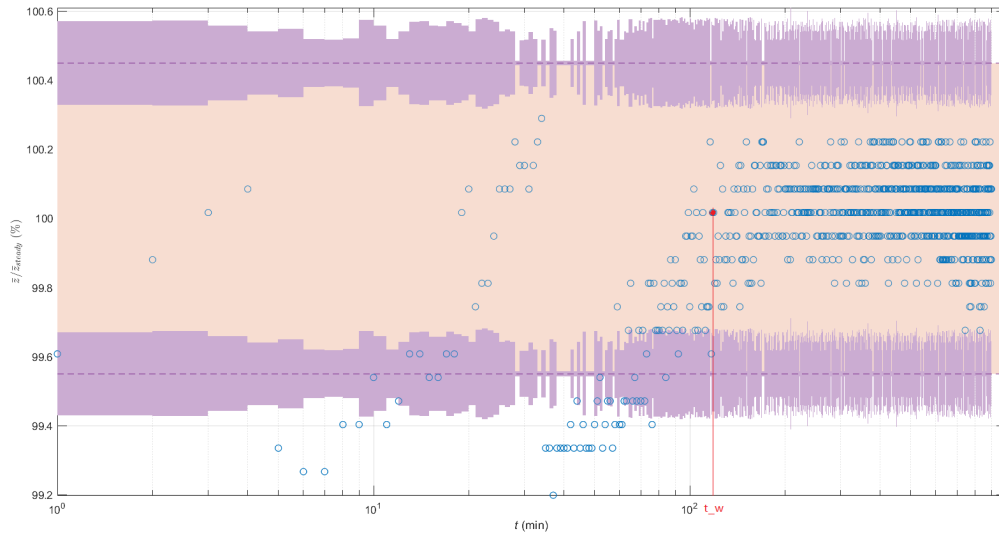


Figure 3.8: Result of one Warm-Up Time measurement performed in the specified operating conditions. The value t_w required by measured distance to stay within the orange portion of the plot is 118 minutes, but once again the distance value stabilizes around the 100% ratio at about 90 minutes.

3.1.2 21 m Distance

Warm-Up Time and Stability measurements performed at around 21 meters were carried out following the same procedure used for the 7 meters distance described above. Fig. 3.9 shows an example behavior of the mean distance over time. With respect to Fig. 3.2, the distance measurement presents a wider range of uncertainty, however the measure begins to oscillate around the steady state value at a time of about 90 minutes once again. All

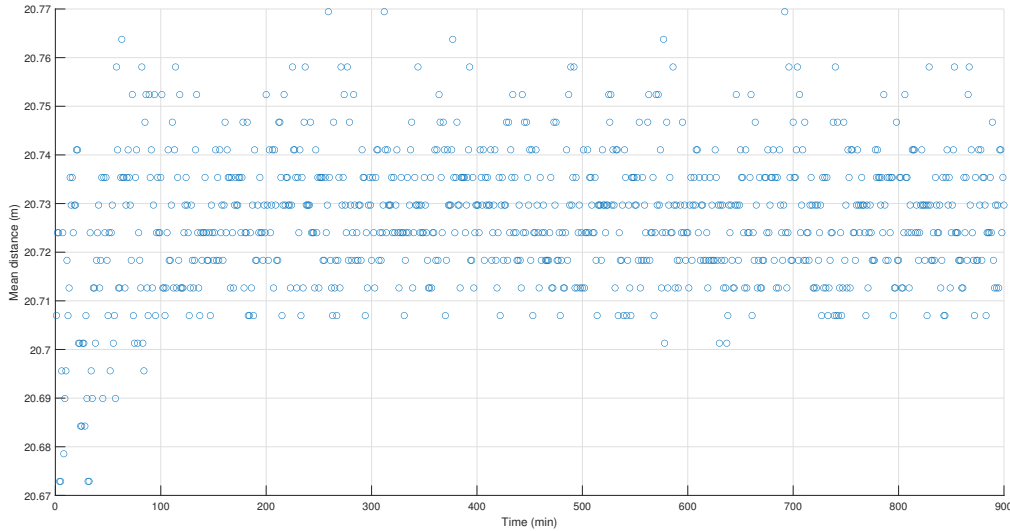


Figure 3.9: Mean distance measured by one channel of the LiDAR over time with echo filter set to "first echo". The measurements qualitatively stabilize after a period of about 90 minutes, oscillating around the value of 20.725 meters.

the measurements described in this Subsection have been performed without moving the target, the only thing that differentiates them is the operating condition of the IUT, defined by the echo filter applied to the beams. As described in the previous Section, the Warm-Up Time of the instrument was obtained selecting specific Tolerance Limits TU and TL (a larger Tolerance Interval, due to the bigger uncertainty of the measurement) around the steady state value \bar{z}_{steady} (calculated according to eq. 3.2 with $n_w = 90$ samples) and considering guard bands dependent on the standard deviation of the mean of measured distances at each time stamp, hence larger than those seen at 7 meters. It is important to mention, however, that some points are randomly not detected by the LiDAR at certain time stamps, causing a non-homogeneity of the data to be analyzed that doesn't allow to always obtain a clear plot of the results of the measurements.

"First" echo filter

Fig. 3.10 shows the results of the characterization by means of the ratio $\frac{\bar{z}}{\bar{z}_{steady}} \cdot 100$.

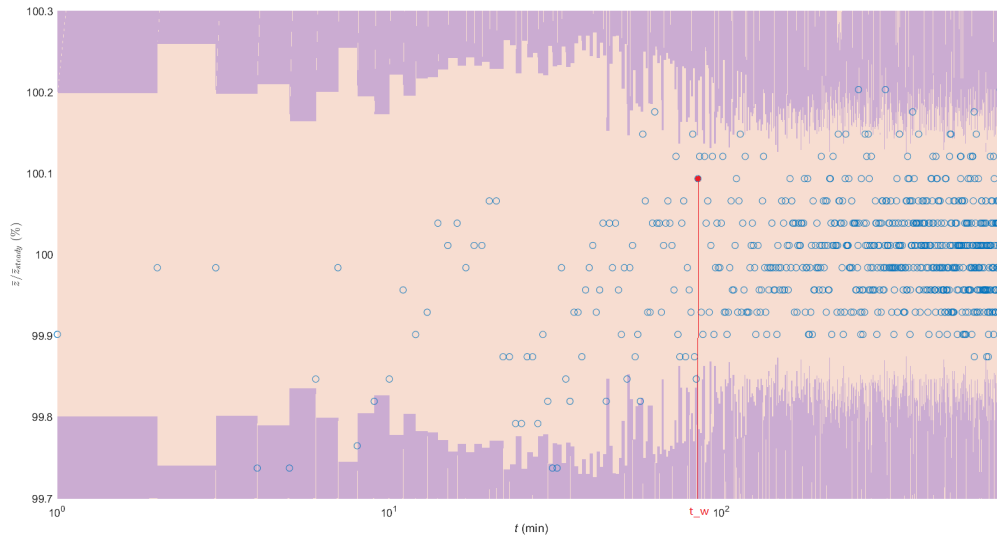


Figure 3.10: Result of one Warm-Up Time measurement performed in the specified operating conditions. The value t_w is obtained as the time of the red sample, which is the time it takes to measured distance to stay within the orange portion of the plot (tolerance interval restricted by the guard bands), excluding isolated overestimation that affects the measurements. In this case the value t_w is 85 minutes.

As stated in Table 3.1, the measurement has been carried out for a total of 3 times under repeatability conditions, and the estimation of the Warm-Up Time of the IUT for the described operating conditions that satisfies the defined requirements can be safely assessed at a value of 90 minutes.

The Stability estimated according to eq. 3.1 is $\frac{S_{\bar{z}_{steady}}}{\bar{z}_{steady}} = 5.8917 \cdot 10^{-2}\%$.

"All" echo filter

Fig. 3.11 shows the plot of the normalized difference with respect to the initial acquisition $\frac{\bar{z}(t) - \bar{z}(t_0)}{\bar{z}(t_0)}$. As shown in the plot, the dispersion of measurements is very broad for the whole 900 minutes long period, making the evaluation of the Warm-Up Time difficult to

be carried out. The two time stamps highlighted in the Figure, belonging to the 5-30 min temporal range, show the end of the initial overshoot of the measurement (around 16 minutes) and one of the first samples that measure the steady state value (around 36 minutes), respectively. It is important to notice that the manufacturer declares a Warm-Up Time of 5-30 minutes for the IUT, meaning that this estimate could be obtained to the evaluation of conditions similar to those just used for locating the two highlighted points in Fig. 3.11. An interesting fact is that this type of behavior around the 5-30 minutes time range is noticeable for each one of the measurements performed, independently from target distance.

According to the definition of Warm-Up Time given in this thesis work, the metric obtained evaluating this operating condition is not relevant, since LiDAR measurements could be considering undesired echoes and mixing relevant target information with environmental noise due to the non-ideal measurement environment.

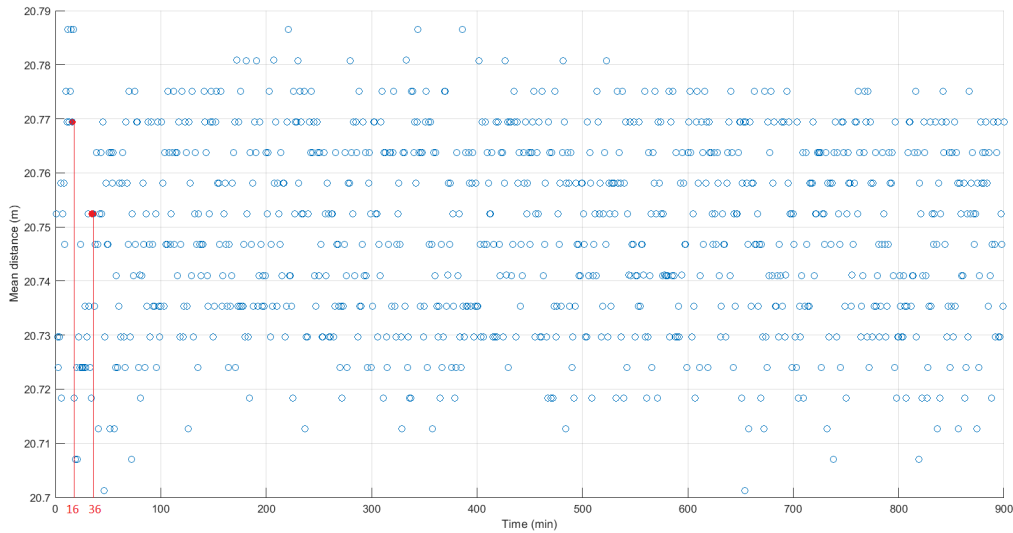


Figure 3.11: MRS6000 difference between the mean z value at instant t and mean z value at instant t_0 normalized by the mean z value at t_0 for the target at a distance of around 21 meters, with echo filter "all".

The Stability estimated according to eq. 3.1 is $\frac{S_{\tilde{z}_{steady}}}{\tilde{z}_{steady}} = 8.2749 \cdot 10^{-2}\%$.

"Last" echo filter

Fig. 3.12 shows the normalized difference with respect to the initial acquisition $\frac{\bar{z}(t) - \bar{z}(t_0)}{\bar{z}(t_0)}$. As for the case of the "all" echo filter, the dispersion of measurements is very broad for the whole 900 minutes long period. The sample highlighted at 90 minutes is used as a reference starting point for the oscillation of measured distances around the steady state value. Once again, the measurement has been carried out for a total of 3 times under repeatability conditions, and it is safe to state a Warm-Up Time of 90 minutes for this operating mode, according to the definition provided in Section 2.1. According to the definition given by the manufacturer, however, the IUT can be considered ready for operation after about 5 to 30 minutes.

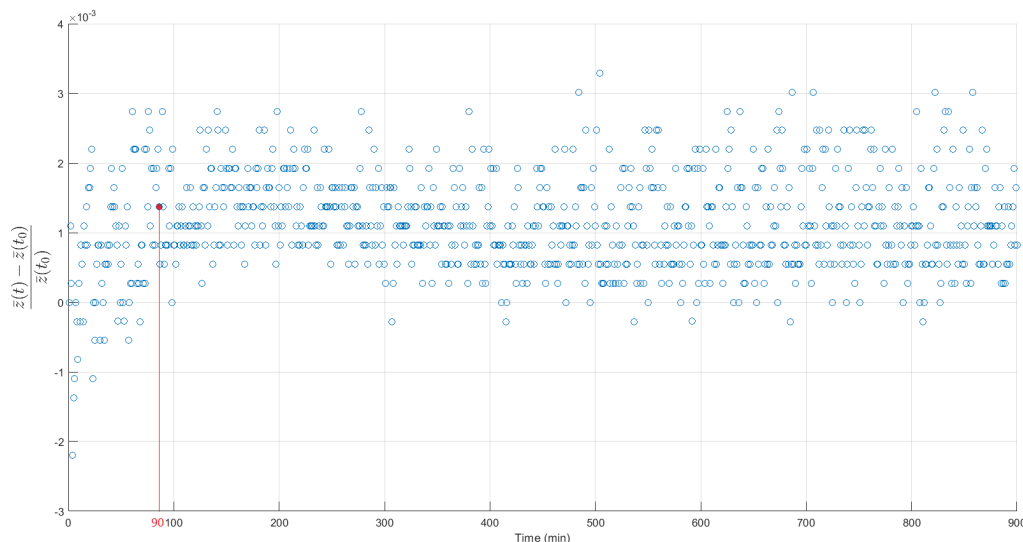


Figure 3.12: MRS6000 difference between the mean z value at instant t and mean z value at instant t_0 normalized by the mean z value at t_0 for the target at a distance of around 21 meters, with echo filter "last".

The Stability estimated according to eq. 3.1 is $\frac{S_{\bar{z}_{steady}}}{\bar{z}_{steady}} = 6.4898 \cdot 10^{-2}\%$.

3.2 Range Error and Quantization

Table 3.2 summarizes target distance ranges and LiDAR operating conditions analyzed for assessing the Range Error and Quantization features of the IUT.

Target distance (49 cm range)	Echo filter setting	Number of measurements
1.35 m - 1.84 m	First echo	1
1.35 m - 1.84 m	Last echo	1
6.86 m - 7.25 m	First echo	1
6.86 m - 7.25 m	Last echo	1

Table 3.2: Number of Range Error and Quantization measurements performed for each different measurement condition, defined by target distance ranges and LiDAR operating mode (selected echo filter).

The motivation behind Quantization measurements lies in the method that 3D ToF scanning LiDARs use to map objects. As shown in Fig. 3.13, the planar target fixed at a distance of around 1.4 meters is mapped on concentric spheres having quantized radii. This mapping method is independent from distance: the IUT measures the Time-of-Flight of the beam with a quantization related to its Time to Digital Converter and to its electronics' operating frequency. When the target is far from the LiDAR, however, the small portion of spherical surface on which the points are mapped approximates a plane, and this effect is less noticeable. Considering a single channel of the IUT, Fig. 3.14 shows a schematic representation of how the IUT scans its surroundings, giving a top view of how points belonging to a planar target are mapped on concentric circles (concentric spheres when considering multiple channels). Analyzing raw pointclouds, converted to spherical coordinates in order to have the same data representation of the IUT, the quantization step that separates two adjacent radial bins r_{bin} , δd , was evaluated. In particular, the value of this quantization step is $\delta d = 0.0625$ m or, equivalently, $\delta d = 6.25$ cm.

The metric analyzed in order to characterize the quantization performed by the IUT is the distribution of the occurrences of a reduced set of points belonging to the target in the different radial bins, while finely changing the position of the target (moving it by 1 cm steps) as mentioned in Section 2.1. The results of the measurement are obtained from the distribution of the average number of points, $\bar{N}(d, r_{bin})$, as a function of target position d . Quantization was evaluated for a total of 4 different operating conditions, as summarized in Table 3.2. The results obtained for the first operating condition are shown in Fig. 3.15, those obtained for the second operating condition are shown in Fig. 3.16, the metrics relative to the two different operating conditions characterized at a set of distances around 7 meters are shown in Fig. 3.17 and Fig. 3.18.

Range Error $\epsilon(d)$ was characterized for each target position with the expression displayed in eq. 3.3, where $d_{INT}(d)$ is the distance measured by the interferometer when the

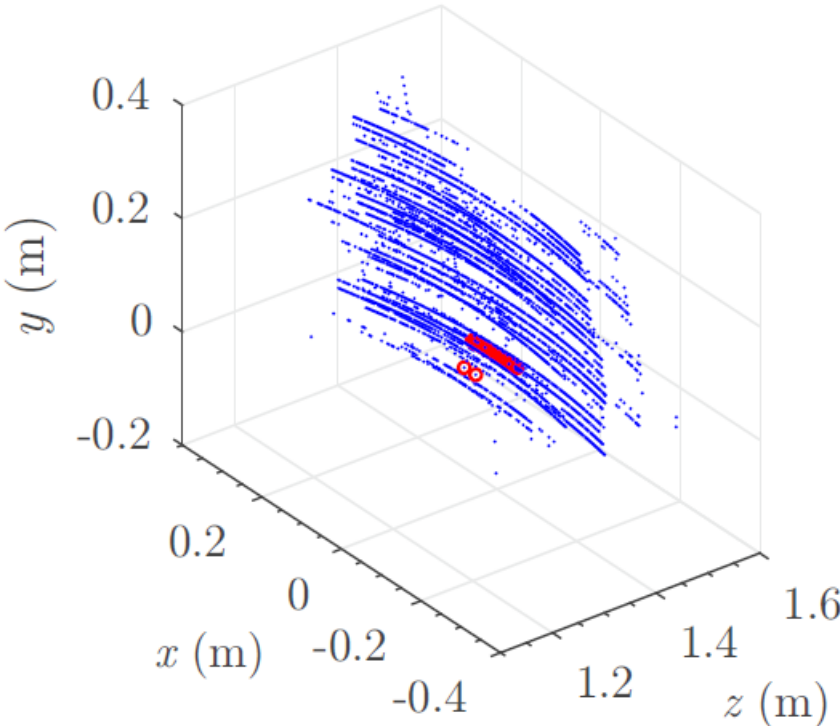


Figure 3.13: Example of point-cloud acquired by the IUT once the target was at a distance of about 1.4 meters. The blue dots represent all the points that belong to the target. The red circles represent the points belonging to a single channel. The planar target undergoes a deformation due to the representation on spherical surfaces.

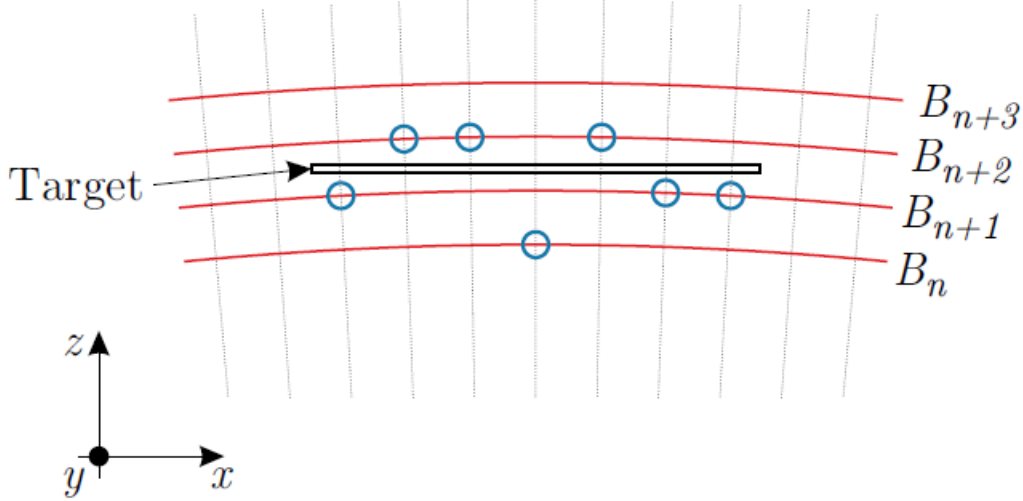


Figure 3.14: Schematic representation of how the IUT scans its surroundings (top view). The red semicircles represent the quantization in the distance estimate. Each of these semicircles constitutes a bin, and its radius differs by one quantization step from the adjacent semicircles. Blue circles represent the result of the sampling of the target performed by one channel of the IUT. Due to quantization and measurement errors, the samples are distributed over several bins.

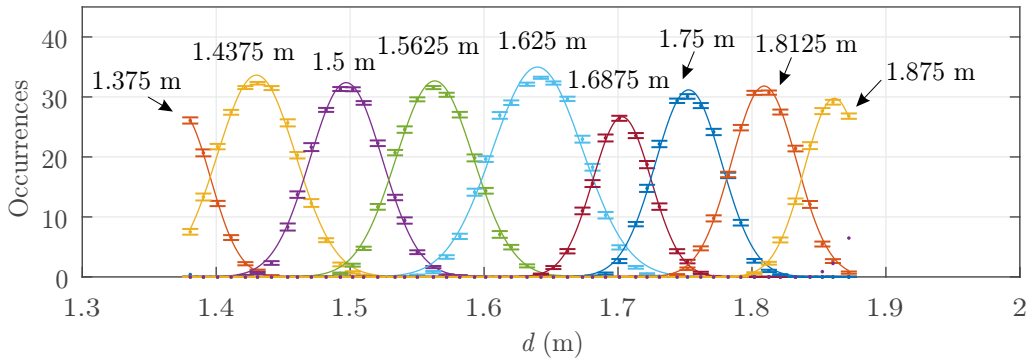


Figure 3.15: Results of Quantization characterization for $d \in (1.3, 1.9)$ m and echo filter “first”. Each curve represents the Gaussian function obtained by fitting the $\bar{N}(d, r_{bin})$ values relative to a specific bin (identified by a different color); the numbers on top of the curves represent the r_{bin} values. For example, when the target was at $d = 1.502$ m, almost all the points of the channel fell on bin $r_{bin} = 1.5$ m. When the target was at $d = 1.782$ m, instead, the points of the channel detecting it distributed among the bins $r_{bin} = 1.75$ m and $r_{bin} = 1.8125$ m. The error bars represent the experimental standard deviations of the mean $s_{\bar{N}}(d, r_{bin})$.

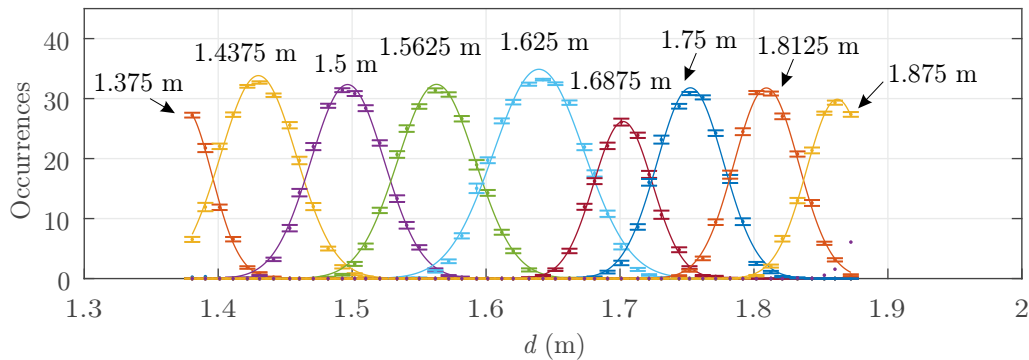


Figure 3.16: Results of Quantization characterization for $d \in (1.3, 1.9)$ m and echo filter “last”. Each curve represents the Gaussian function obtained by fitting the $\bar{N}(d, r_{bin})$ values relative to a specific bin (identified by a different color); the numbers on top of the curves represent the r_{bin} values. The error bars represent the experimental standard deviations of the mean $s_{\bar{N}}(d, r_{bin})$.

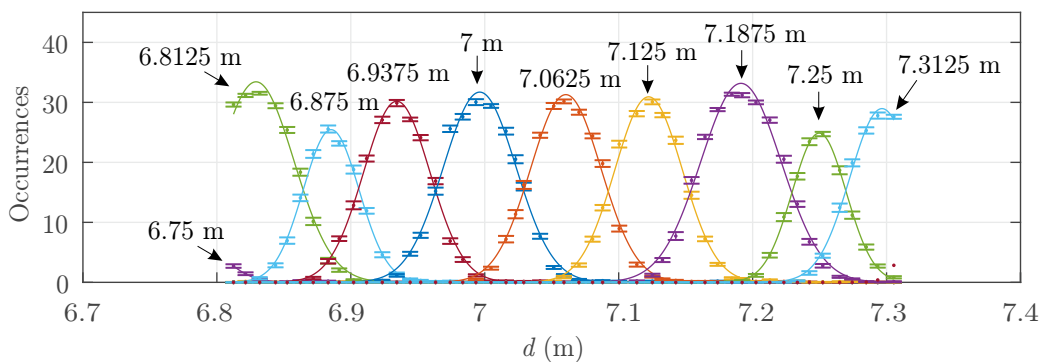


Figure 3.17: Results of Quantization characterization for $d \in (6.8, 7.4)$ m and echo filter “first”. Each curve represents the Gaussian function obtained by fitting the $\bar{N}(d, r_{bin})$ values relative to a specific bin (identified by a different color); the numbers on top of the curves represent the r_{bin} values. The error bars represent the experimental standard deviations of the mean $s_{\bar{N}}(d, r_{bin})$.

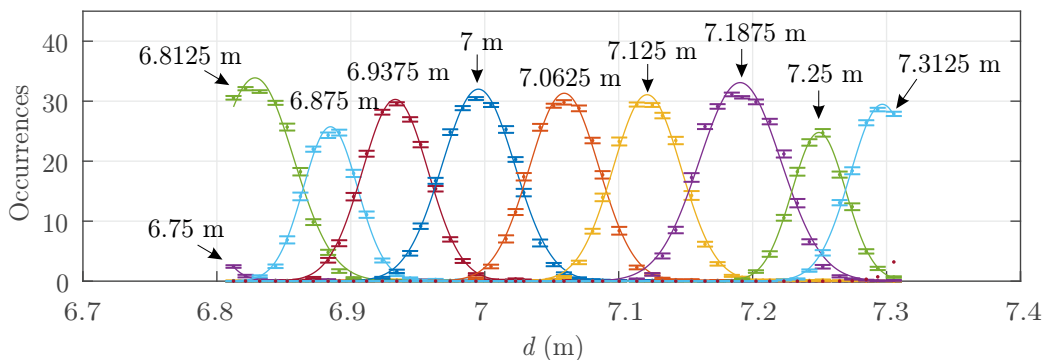


Figure 3.18: Results of Quantization characterization for $d \in (6.8, 7.4)$ m and echo filter “last”. Each curve represents the Gaussian function obtained by fitting the $\bar{N}(d, r_{bin})$ values relative to a specific bin (identified by a different color); the numbers on top of the curves represent the r_{bin} values. The error bars represent the experimental standard deviations of the mean $s_{\bar{N}}(d, r_{bin})$.

target is placed in position d .

$$\epsilon(d) = d_{INT}(d) - \bar{d}_{IUT}(d) . \quad (3.3)$$

Fig. 3.19 and Fig. 3.20 show the results of Range Error characterization for the set of distances around 1.6 m and the set of distances around 7 m respectively. The plots show the distributions of the errors, the height of each bar being the probability that the error fell inside the range defined by the width of the bar itself.

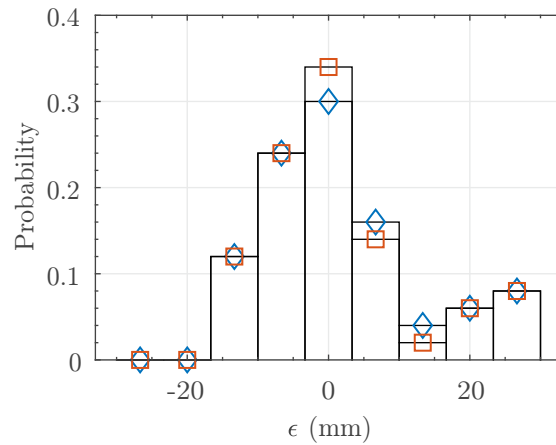


Figure 3.19: Distribution of the errors ϵ for $d \in (1.3, 1.9)$ m. The height of each bar represents the probability that the error fell within the range defined by the width of the bar. For each bar, the blue diamond represents the probability evaluated for the “first” echo mode and the orange square represents the probability evaluated for the “last” echo mode.

The characterization of Range Error was ultimately performed by considering one single point instead of an entire channel, to evaluate the axial measurement performance of the IUT without the possibility of averaging out errors by considering a greater number of points. The procedure followed for single-point characterization is exactly the same described in Section 2.1 for a reduced set of points belonging to a channel, and once again the error was defined with the expression shown in eq. 3.3. The metric corresponding to the $N(n, d, r_{bin})$, describing the occurrence of points in different radial bins r_{bin} , was defined as $\Gamma(n, d, r_{bin})$ for single point analysis, and the Gaussian fitting of occurrences was performed once again. Fig. 3.21, Fig. 3.22, Fig. 3.23, Fig. 3.24 show the distribution of errors in each one of the different measurement conditions defined by Table 3.2. It is important to notice that the σ values obtained from Gaussian fittings are compatible with the range error declared by the manufacturer in the datasheet, which is equal to 30 mm.

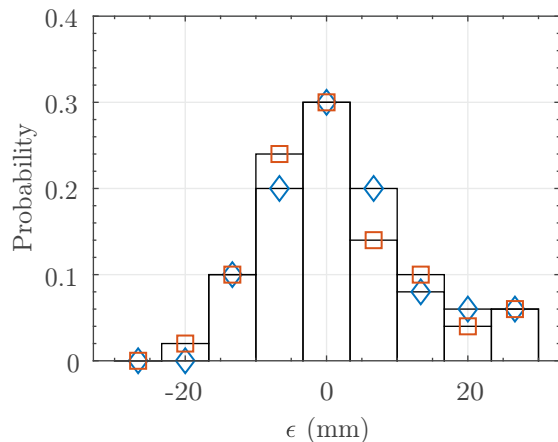


Figure 3.20: Distribution of the errors ϵ for $d \in (6.8, 7.4)$ m. The height of each bar represents the probability that the error fell within the range defined by the width of the bar. For each bar, the blue diamond represents the probability evaluated for the “first” echo mode and the orange square represents the probability evaluated for the “last” echo mode.

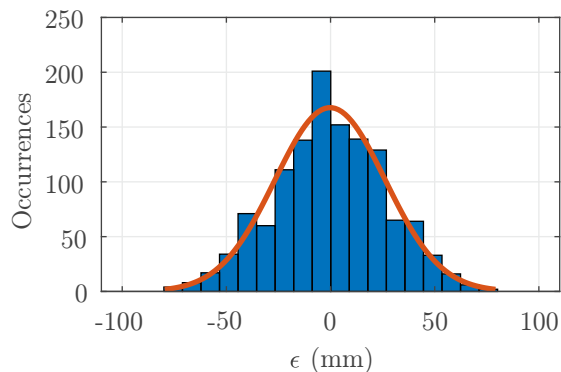


Figure 3.21: Distribution of the single-point analysis errors ϵ for $d \in (1.3, 1.9)$ m and echo filter “first”. The height of each bar represents the probability that the error fell within the range defined by the width of the bar. The orange curve represents the result of the fitting with a Gaussian function. Values obtained from the Gaussian fitting: $\mu_\epsilon = -2.08$ mm, $\sigma_\epsilon = 27.77$ mm.

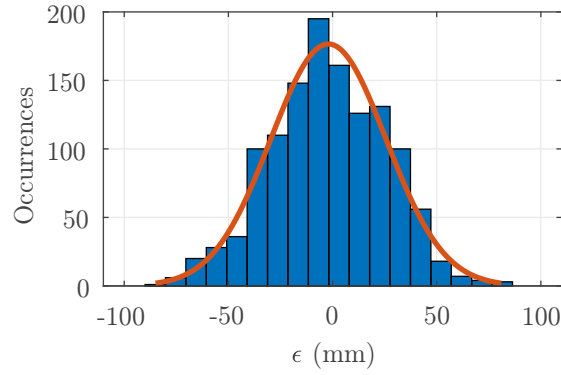


Figure 3.22: Distribution of the single-point analysis errors ϵ for $d \in (1.3, 1.9)$ m and echo filter “last”. The height of each bar represents the probability that the error fell within the range defined by the width of the bar. The orange curve represents the result of the fitting with a Gaussian function. Values obtained from the Gaussian fitting: $\mu_\epsilon = -1.98$ mm, $\sigma_\epsilon = 27.67$ mm.

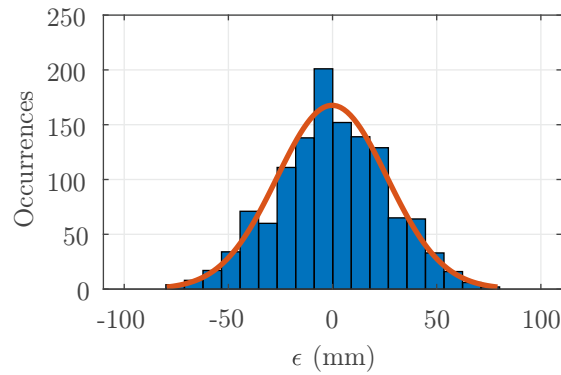


Figure 3.23: Distribution of the single-point analysis errors ϵ for $d \in (6.8, 7.4)$ m and echo filter “first”. The height of each bar represents the probability that the error fell within the range defined by the width of the bar. The orange curve represents the result of the fitting with a Gaussian function. Values obtained from the Gaussian fitting: $\mu_\epsilon = -0.21$ mm, $\sigma_\epsilon = 26.47$ mm.

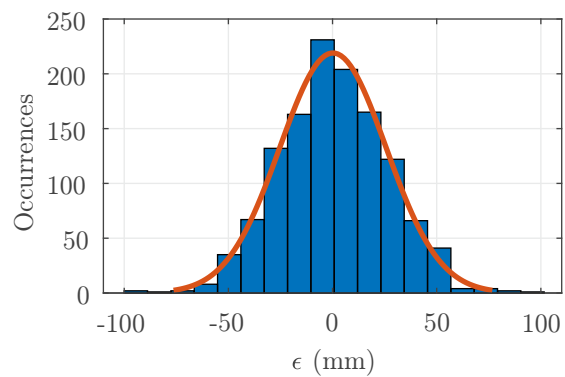


Figure 3.24: Distribution of the single-point analysis errors ϵ for $d \in (6.8, 7.4)$ m and echo filter “last”. The height of each bar represents the probability that the error fell within the range defined by the width of the bar. The orange curve represents the result of the fitting with a Gaussian function. Values obtained from the Gaussian fitting: $\mu_\epsilon = 0.19$ mm, $\sigma_\epsilon = 25.52$ mm.

Chapter 4

Conclusions

The increasing importance of LiDARs, especially for what concerns cutting-edge technological applications such as ADAS and autonomous driving, is focusing research efforts towards effective, standardized characterizations of these instruments in a lot of different application fields. For what concerns automotive in particular, the current lack of national/international standards and the non-homogeneity of the data provided by manufacturers constitutes a relevant issue for ADAS designers and vehicle manufacturers worldwide. Despite of the significant number of new studies, research activities and overall investigations, some relevant characteristics that define the behavior of these sensors are still not being highlighted, leaving some gaps in the knowledge we currently have of their behavior.

The purpose of this work was to effectively identify some relevant parameters that allow to characterize any type of LiDAR and to provide a general methodology focused on investigating specifically defined metrological properties of these instruments. The study of both LiDAR working principles and research literature allowed to identify four metrics that are currently not being investigated, but that are of paramount relevance for characterizing the behavior of LiDAR systems: warm-up time, stability, range error, quantization. The first two metrics allow to assess the performance of the instrument over time, whereas the other two allow to quantify the measurement error due to how real world information is sensed and represented by the instrument. In order to achieve the proposed goal, formal definitions of the measurands identified through the investigation and of the measurement procedures designed to guarantee the repeatability of the characterization have been provided. A modular custom measurement setup has been then designed and realized to enable the execution of the measurement procedures, and fully characterize ideally any type of LiDAR in a various range of operating conditions. The software used for characterization

has been designed and developed, allowing to perform complete static characterizations automatically. The validation of the research activity, measurement procedures and measurement setup altogether has been ultimately performed by characterizing a state of the art 3D ToF scanning LiDAR, the MRS6000 by SICK. The results proposed in Chapter 3 give a significant display of the ability of the overall system (constituted by measurement procedures, setup and software) to characterize the relevant features of the IUT.

In particular, the proposed methodology allowed to assess the Warm-Up Time of the instrument, which resulted independent on distance and operating mode, and the Stability of the measurement, providing a relevant characterization of the behavior of the LiDAR over time; moreover, the system proved to be suitable for Range Error and Quantization measurements, allowing to investigate "how" the LiDAR works, how it maps the surrounding environment and how it encapsulates spatial information inside discrete pointclouds. One incredible quality of the applied methodology is, without doubt, modularity, both in the measurement procedures and in the measurement setups, that makes the overall system easy to expand, in order to overcome its current limitations or to enrich it with further characterization possibilities. Some improvements that can be achieved because of modularity are:

- extension of the rail, to allow long range characterizations at a distance above 21 m (if desired);
- automation of the dynamic characterization (Range Error and Quantization);
- definition and realization of additional characterizations.

The possibility of realizing automatic dynamic characterization has been investigated during this thesis work, leading to the realization of the prototype motor unit shown in Fig. 4.1: this unit could be directly linked to the target's sliding carriage and remotely controlled in order to make the movement of the target along the rail system completely automatic, thus enabling automatic Range Error and Quantization measurements.

In conclusion, not only the presented methodologies are able to give a useful characterization of important LiDAR features that are currently not being investigated, they also constitute a basic framework that can be easily expanded to allow further characterizations according to the specific needs. Moreover, the system allows direct comparison between different LiDARs, making them undergo the same characterization procedures and obtaining comparable metrics that define their performance and suitability for a specific application

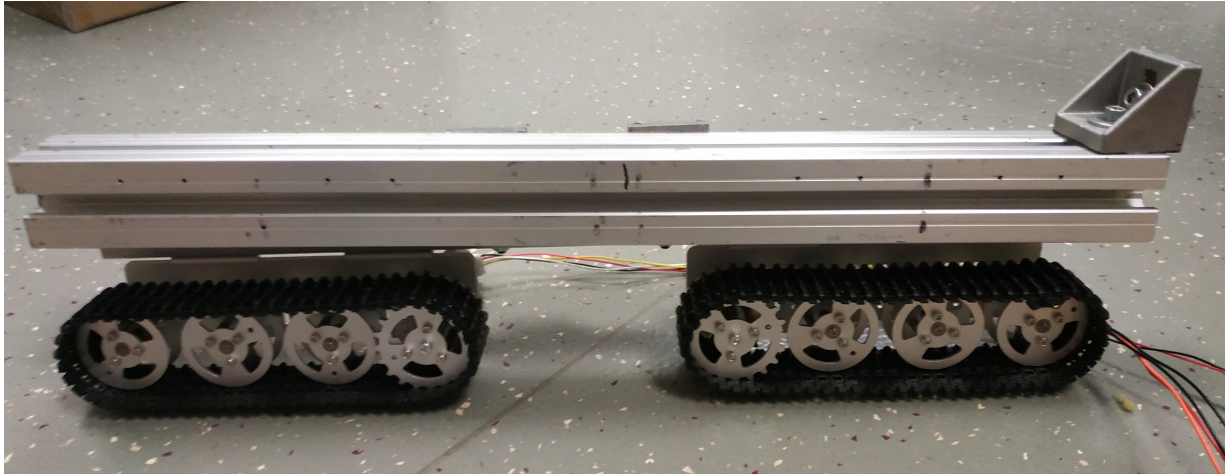


Figure 4.1: Prototype motor unit realized for automatically moving the sliding carriage of the target via remote control. This unit can be directly linked to the sliding carriage thanks to the modularity of the system.

field. This is especially useful in automotive applications, where the ability to correctly characterize sensors enables far more powerful and innovative technologies.

Acknowledgements

Ringrazio innanzitutto i miei genitori e i miei nonni, che mi hanno sempre sostenuto e incoraggiato credendo in me anche quando io stesso non riuscivo a farlo. Sono felice di condividere con voi questo traguardo o, meglio, questo inizio.

Ringrazio anche Stefano e il prof. Rovati, che mi hanno concesso l'opportunità di lavorare con loro in questi mesi difficili, permettendomi di ampliare le mie conoscenze e competenze oltre quanto mi sarei aspettato. Grazie anche a Andrea, Davide, Ettore, Giovanni e Tommy per aver contribuito a rendere ogni giorno piacevole e allo stesso tempo interessante.

Grazie a Leonardo, Mattia, Oliviero e Samuele per essere dei fratelli, più che degli amici. Un sentito grazie va a chi, anche se per pochi anni, ha voluto condividere con me una parte del suo percorso: grazie Irene, Marco, Michele, Nicolò, Vincenzo, senza di voi niente sarebbe stato lo stesso.

Infine un grazie speciale va a Chiara, che è sempre riuscita a farmi sentire il suo sostegno e la sua presenza nonostante tutto, sconfiggendo ogni distanza.

I miei ringraziamenti più sinceri vanno a tutti voi,

Giorgio.

Acknowledgements

Appendix A - Software Guides

I Ubuntu 16.04 LTS (Xenial Xerus) guide

Ubuntu 16.04 LTS (Xenial Xerus) guide

1. Resources

The operating system's image file is available for download at the following web page

https://releases.ubuntu.com/16.04.7/?_ga=2.80586583.1346164706.1606991737-293788104.1605203282

The installation guide is available at the following web page

<https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview>

The list of ROS distributions is available at the following web page

http://wiki.ros.org/Distributions#Distribution_Details

2. Compatibility

Ubuntu 16.04 is compatible with ROS Kinetic Kame (EOL date visible on ROS distributions' webpage).

Ubuntu 16.04 is compatible with MATLAB (ROS toolbox available for version R2020b).

II ROS for Ubuntu 16.04 guide

ROS guide (Ubuntu 16.04)

1. Resources

ROS distributions available for download at the following web page

<http://wiki.ros.org/Distributions>

ROS documentation available at the following web page

<http://wiki.ros.org/>

SICK GitHub repository

https://github.com/SICKAG/sick_scan

2. ROS setup guide

Open the terminal.

Execute the following commands:

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > ...  
/etc/apt/sources.list.d/ros-latest.list' (Setup the computer to accept software from pack-  
ages.ros.org)
```

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key ...  
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654 (Setup your keys)
```

```
sudo apt-get update (make sure Debian package index is up to date)
```

```
sudo apt-get install ros-kinetic-desktop-full (full install of ROS kinetic)
```

3. ROS setup for MRS6000

Open the terminal. NOTE: some of these instructions might require to issue sudo because they can only be executed by a system administrator.

Download the sick_scan packet using the following instructions:

```
source /opt/ros/kinetic/setup.bash
```

```
mkdir -p /ros_catkin_ws/src/
```

```
cd /ros_catkin_ws/src/  
git clone -b devel --single-branch  
git://github.com/SICKAG/sick_scan.git  
The last instruction might not work. If it doesn't work, manually download the packet  
from  
https://github.com/SICKAG/sick\_scan
```

Once the packet is downloaded, it must be copied in its destination directory:

```
mv sick_scan-master /ros_catkin_ws/src/
```

When the process is over, issue:

```
cd ..  
catkin_make install  
source /ros_catkin_ws/install/setup.bash
```

Before launching the packet, the IP address of the device (the MRS6000 LiDAR in this case) must be configured so that it shares the same subnetmask of the PC.

When a window with the message “Connected” is displayed, ROS can be launched with the following command:

```
roslaunch sick_scan sick_mrs_6xxx.launch
```

4. How to use ROS

Make sure that the device to be controlled is plugged to the PC and that drivers for this device are installed. See device-specific guides for detailed information on how to get and install drivers.

Issue `roscpp` to initialize the ROS master and global node.

Use `rostopic list` to see the full list of available topics.

Online documentation is available for detailed information on how to control specific devices using ROS.

When finished using the device, issue `roscpp shutdown` to shut down the ROS master and delete the global node.

III MATLAB ROS guide

MATLAB ROS guide

1. Resources

MATLAB version R2020b is available for download at the following web page

https://it.mathworks.com/downloads/web_downloads/select_release?mode=gwylf

ROS toolbox documentation available at the following web page

<https://it.mathworks.com/help/ros/>

2. ROS Toolbox installation guide

Open the terminal inside the directory where the MATLAB executable is located.

Execute the command `./matlab` (for AMD powered PCs, `./matlab -softwareopengl` might be required).

Under the “HOME” tab, click on “Add-Ons”.

Search “ros toolbox”.

Open “ROS Toolbox” by MathWorks.

Click on “Install”.

3. How to use ROS

Make sure that the device to be controlled is plugged to the PC and that drivers for this device are installed. See device-specific guides for detailed information on how to get and install drivers.

Open MATLAB R2020b, open MATLAB’s Command Window in order to issue commands.

Issue “`rosinit`” to initialize the ROS master and global node.

Use “`rostopic list`” to see the full list of available topics.

See device-specific guides for detailed information on how to control them using ROS for MATLAB.

When finished using the device, issue “`roshutdown`” to shut down the ROS master and delete the global node.

IV MRS6000 guide

MRS6000 guide

1. Resources

MRS6000 documentation available at the following web page

<https://www.sick.com/it/it/soluzioni-di-misurazione-e-rilevamento/sensori-3d-lidar/mrs6000/c/g448151>

ROS documentation available at the following web page

<http://wiki.ros.org/>

2. MRS6000 setup guide

Power on the MRS6000 and connect it to the PC via Ethernet.

Open connection settings in order to configure the Ethernet connection between PC and LiDAR.

Manually set the PC's IPv4 address for this connection to 192.168.0.175, the netmask to 24 (255.255.255.0) and the gateway to 0.0.0.0

The next step requires to know the IP address of the MRS6000 (192.168.0.150). Open the terminal and input the following commands:

```
sudo ifconfig enp2s0 192.168.0.175
```

```
sudo route add 192.168.0.150 enp2s0
```

Now the web page at address 192.168.0.150 (IP address of the MRS6000) should be accessible.

For the following steps, make sure that ROS Kinetic Kame is already installed on the system and refer to the “ROS for Ubuntu 16.04 guide”.

3. MRS6000 control through MATLAB (ROS Toolbox)

Issue `rosinit` to initialize the ROS master and global node.

Use `rostopic list` to see the full list of available topics.

Use the `rossubscribe` command to subscribe to the desired topic (can receive and examine

the messages that are sent).

Use the receive command to get data from the subscriber (to store the data, issue “data = receive(<sub>, <timeout>)”).

When finished using the device, issue roshutdn to shut down the ROS master and delete the global node.

Example: code for storing one frame of the scan performed by the MRS6000 as a point-cloud and plotting it in 3D.

```
rosinit
camera = rossubscriber('/cloud')
scandata1 = receive(camera,1)
X1 = readField(scandata1,'x');
Y1 = readField(scandata1,'y');
Z1 = readField(scandata1,'z');
plot3(X1,Y1,Z1, 'b.', 'MarkerSize', 4)
roshutdn
```

V Interferometer (HP5527A) guide

Interferometer guide (HP5527A)

1. Resources

GPIB-USB-B drivers

<https://www.ni.com/it-it/support/downloads/drivers/download.ni-488-2.html#306147>

Pyvisa documentation available at the following web page

<https://pyvisa.readthedocs.io/en/latest/>

Python releases available at the following web page

<https://www.python.org/downloads/windows/>

2. Interferometer setup guide

Make sure to have Python installed on the system (both Python 2.7.18 and Python 3.9.0 are suitable to interact with the interferometer, link to the website in “Resources”).

Install the required packets using the following instructions:

```
sudo apt-get install libelf-dev
```

```
apt install python-pip
```

```
pip install -U pyvisa
```

```
pip install -U pyvisa-py
```

```
apt install fxload
```

Download the following packets required for operating National Instruments’ GPIB-USB-B:

```
linux-gpib-4.2.0
```

```
PyVISA-py-0.3.1
```

```
gpib_firmware-2008-08-10
```

Execute the command

```
sudo passwd root
```

Open the linux-gpib-4.2.0 folder and extract the two folders it contains (user and kernel).

Open the gpib.conf file and edit the interface type to be “ni_usb_b”.

Check that these values are equal to those written in the file:

```
minor = 0 /* board index, minor = 0 uses /dev/gpib0, minor = 1 uses /dev/gpib1, etc.
*/
board_type = "ni_usb_b" /* type of interface board being used */
name = "gpib0" /* optional name, allows you to get a board descriptor using ibfind() */
pad = 0 /* primary address of interface */
sad = 0 /* secondary address of interface */
timeout = T3s /* timeout for commands */
```

For both folders execute the following steps:

```
./configure
```

```
make
```

Now two symlinks must be created in order to avoid an error in the execution:

```
sudo ln -s /usr/src/linux-headers-4.15.0-58-generic/include/generated/autoconf.h...  
/usr/src/linux-headers-4.15.0-58-generic/include/linux
```

```
sudo ln -s /usr/src/linux-headers-4.15.0-58-generic/include/config/auto.conf...  
/usr/src/linux-headers-4.15.0-58-generic/include/linux
```

```
make install
```

Now check if the gpib is working by executing:

```
ldconfig  
gpib_conf
```

Now access the folder “gpib_firmware-2008-08-10” and open the “ni_gpib_usb_b” folder which contains two .hex files that must be loaded on the GPIB. Copy the “ni_gpib_usb_b” folder in /lib/udev, renaming it as “ni_usb_gpib”.

The next step is to go in /lib/udev/ni_usb_gpib and check that the file “ni_gpib_usb” looks like this:

```
GPIB_CONFIG_OPTIONS="-minor 0"  
DATADIR=/lib/firmware/ni_usb_gpib  
FXLOAD=fxload  
FXLOAD_OPTIONS="-D fx"
```

In /lib/udev/rules.d create the file “99-ni_usb_gpib.rules” that contains:

```
SUBSYSTEM=="usb", ACTION=="add", ENVDEVTYPE=="usb_device", ATTRidVen-  
dor=="3923", ATTRidProduct=="702[ab]", ENVDEVICE="$devnode", ENVSERIAL=  
"$attrserial", RUN+="/usr/local/etc/hotplug/usb/ni_usb_gpib"  
SUBSYSTEM=="usb", ACTION=="add", ENVDEVTYPE=="usb_device", ATTRidVen-  
dor=="3923", ATTRidProduct=="713b", ENVDEVICE="$devnode", ENVSERIAL= "$at-  
trserial", RUN+="/usr/local/etc/hotplug/usb/ni_usb_gpib"  
SUBSYSTEM=="usb", ACTION=="add", ENVDEVTYPE=="usb_device", ATTRidVen-  
dor=="3923", ATTRidProduct=="709b", ENVDEVICE="$devnode", ENVSERIAL= "$at-  
trserial", RUN+="/usr/local/etc/hotplug/usb/ni_usb_gpib"  
SUBSYSTEM=="usb", ACTION=="add", ENVDEVTYPE=="usb_device", ATTRidVen-
```

```
dor=="3923", ATTRidProduct=="7618", ENVDEVICE="$devnode", ENVSERIAL= "$at-  
trserial", RUN+="/usr/local/etc/hotplug/usb/ni_usb_gpib"
```

Check the presence of the “ni_usb_gpib” folder inside the /lib/firmware directory and that it contains firmware files. Now issue lsusb, that allows us to see if the GPIB is connected to the PC and to get Device and Bus numbers.

```
Load the driver with fxload -D /dev/bus/usb/001/002 -I niusbb_firmware.hex -s ...  
niusbb_loader.hex
```

copy the .ko file located in /lib/modules/LINUXKERNELFOLDER/ to the /lib/modules/LINUXKERNELFOLDER/gpib/ni_usb folder.

Execute:

```
lsusb  
gpib_config  
modprobe ni_usb_gpib  
modinfo ni_usb_gpib  
ibtest
```

Now everything should work.

Switch the interferometer on.

Open command prompt as administrator (right click + run as administrator).

Run Python by issuing the “python” command.

Execute the following commands:

```
» import pyvisa  
» rm = pyvisa.ResourceManager()  
» rm.list_resources()
```

Something like (u'GPIB0::3::INSTR') should be detected

```
» inst = rm.open_resource('GPIB0::3::INSTR')
```

After this command the interferometer setup is completed and the device is ready to communicate. In order to communicate with the device, issue queries through the inst.query command. For the full list of queries refer to the manual in “Resources” or execute the » print(inst.query(“INST?")) command.

Example: issuing a query to obtain a distance measurement

```
» print(inst.query("XPOS?"))
```

Appendix B - MATLAB code

I Scan example

```
%MATLAB example code to collect a pointcloud from the LiDAR and obtain its 3D plot
rosinit
camera = rossubscriber('/cloud')
scandata1 = receive(camera,1)
X1 = readField(scandata1,'x');
Y1 = readField(scandata1,'y');
Z1 = readField(scandata1,'z');
plot3(X1,Y1,Z1, 'b.', 'MarkerSize', 4)
```

II Crop and Fit example

```
%MATLAB example code to spatially crop a stored pointcloud and obtain mean distance,
standard deviation of the distance and planar fit.
Y1min=-0.23;
Y1max=-0.09;
Z1min=0.1229;
Z1max=0.2035;
index = ((Y1 > Y1min) & (Y1 < Y1max) & (Z1 > Z1min) & (Z1 < Z1max));
Y1crop = Y1(index);
Z1crop = Z1(index);
X1crop = X1(index);
plot3(X1crop, Y1crop, Z1crop, 'b.', 'MarkerSize', 4);
```

```

mean(X1crop)
std(X1crop)
plan = fit([Y1crop Z1crop], X1crop, poly11);
plot(plan, [x y], z)

```

III Warm-Up and Stability, 7m acquisition

%MATLAB code to automatically collect the data required for Warm-Up Time and Stability measurements at 7 m (900 pointclouds, 1 each minute for 900 minutes).

```

Xmin = 6;
Xmax = 8;
Ymin = -1;
Ymax = 1;
Zmin = -1;
Zmax = 0.35;
acquisitioninterval = 60;
totaltime = 900;
camera = rossubscriber('/cloud');
for i = 1:1:totaltime
scandata = receive(camera,1);
XYZtemp = readXYZ(scandata);
Itemp = readField(scandata, 'intensity');
index = ((XYZtemp(:,1) > Xmin) & (XYZtemp(:,1) < Xmax) & (XYZtemp(:,2) > Ymin)
& (XYZtemp(:,2) < Ymax) & (XYZtemp(:,3) > Zmin) & (XYZtemp(:,3) < Zmax));
data(i).XYZ = XYZtemp(index, 1:3);
data(i).intensity = Itemp(index);
pause(acquisitioninterval);
end
disp('Acquisition completed.');
```

IV Warm-Up and Stability, 7m plot

%MATLAB code to plot the statistical data related to a single channel selected by the user.

```
Zbase = -0.146;
deltaZ = 0.077;
numchannel = input('Channel number (1 to 6)? ');
Zmintemp = Zbase + (deltaZ*(numchannel-1)) - (deltaZ/2);
Zmaxtemp = Zbase + (deltaZ*(numchannel-1)) + (deltaZ/2);
for i = 1:1:totaltime
index = ((data(i).XYZ(:,3) > Zmintemp) & (data(i).XYZ(:,3) < Zmaxtemp));
datachannel(i).XYZ = data(i).XYZ(index, 1:3);
datachannel(i).intensity = data(i).intensity(index);
meandistancechannel(i) = mean(datachannel(i).XYZ(:,1));
end
j = [1:1:totaltime];
figure;
plot3(data(1).XYZ(:,1), data(1).XYZ(:,2), data(1).XYZ(:,3), 'b.', 'MarkerSize', 7)
grid on;
figure;
plot3(data(720).XYZ(:,1), data(720).XYZ(:,2), data(720).XYZ(:,3), 'b.', 'MarkerSize', 7)
grid on;
figure;
plot3(datachannel(1).XYZ(:,1), datachannel(1).XYZ(:,2), datachannel(1).XYZ(:,3), 'b.', ...
'MarkerSize', 7)
grid on;
figure;
plot3(datachannel(720).XYZ(:,1), datachannel(720).XYZ(:,2), datachannel(720).XYZ(:,3),
'b.', 'MarkerSize', 7)
grid on;
figure;
scatter(j, meandistancechannel);
grid on;
xlabel('Time (min)')
```

```

ylabel('Mean distance (m)')
figure;
scatter(j, (meandistancechannel - meandistancechannel(1))./meandistancechannel(1));
grid on;
xlabel('Time (min)')
ylabel('frac{z(t) - z(t_0)}{z(t_0)}', 'interpreter', 'latex', 'FontSize', 16)

```

V Warm-Up and Stability, 7m elaboration

%MATLAB code to perform the statistical elaboration related to a single channel selected by the user.

```

M=[]; STDD=[]; STDDN=[];
Zbase = -0.146;
deltaZ = 0.077;
numchannel = input('Channel number (1 to 6)? ');
Zmintemp = Zbase + (deltaZ*(numchannel-1)) - (deltaZ/2);
Zmaxtemp = Zbase + (deltaZ*(numchannel-1)) + (deltaZ/2);
for i = 1:1:totaltime
index = ((data(i).XYZ(:,3) > Zmintemp) & (data(i).XYZ(:,3) < Zmaxtemp));
X = data(i).XYZ(index, 1);
Y = data(i).XYZ(index, 2);
Z = data(i).XYZ(index, 3);
I = data(i).intensity(index);
Mtemp = mean(X);
M = [M, Mtemp];
STDDtemp = std(X);
STDD = [STDD, STDDtemp];
STDDNtemp = std(X)/(sqrt(numel(X))*M(1));
STDDN = [STDDN, STDDNtemp];
end
j = 1:1:totaltime;
figure;

```

```

scatter(j, M);
grid on;
xlabel('Time (min)')
ylabel('Mean distance (m)')
figure;
scatter(j, (M - M(1))/M(1));
grid on;
xlabel('Time (min)')
ylabel('  $\frac{\bar{z}(t) - \bar{z}(t_0)}{\bar{z}(t_0)}$  ', 'interpreter', 'latex', 'FontSize', 16)
figure;
scatter(j, STDDN);
grid on;
xlabel('Time (min)')
ylabel('  $\frac{S_z(t)}{\bar{z}(t_0)\sqrt{(N)}}$  ', 'interpreter', 'latex', 'FontSize', 16)

```

VI Warm-Up and Stability, 21m acquisition

%MATLAB code to automatically collect the data required for Warm-Up Time and Stability measurements at 21 m (900 pointclouds, 1 each minute for 900 minutes).

```

rosinit
Ymin = -1;
Ymax = 1;
Zmin = -1;
Zmax = 0.35;
acquisitioninterval = 60;
totaltime = 900;
camera = rossubscriber('/cloud');
for i = 1:1:totaltime
scandata = receive(camera,1);
XYZtemp = readXYZ(scandata);
Itemp = readField(scandata, 'intensity');
index = ((XYZtemp(:,2) > Ymin) & (XYZtemp(:,2) < Ymax) & (XYZtemp(:,3) > Zmin)

```

```

& (XYZtemp(:,3) < Zmax));
data(i).XYZ = XYZtemp(index, 1:3);
data(i).intensity = Itemp(index);
pause(acquisitioninterval);
end
disp('Acquisition completed.');
```

VII Warm-Up and Stability, 21m plot

%MATLAB code to plot the statistical data related to a single channel selected by the user for 21 m acquisitions.

```

Zbase = -0.205;
deltaZ = 0.225;
numchannel = input('Channel number (1 to 6)? ');
Zmintemp = Zbase + (deltaZ*(numchannel-1)) - (deltaZ/2);
Zmaxtemp = Zbase + (deltaZ*(numchannel-1)) + (deltaZ/2);
for i = 1:1:totaltime
index = ((data(i).XYZ(:,3) > Zmintemp) & (data(i).XYZ(:,3) < Zmaxtemp));
datachannel(i).XYZ = data(i).XYZ(index, 1:3);
datachannel(i).intensity = data(i).intensity(index);
meandistancechannel(i) = mean(datachannel(i).XYZ(:,1));
end
j = [1:1:totaltime];
figure;
plot3(data(1).XYZ(:,1), data(1).XYZ(:,2), data(1).XYZ(:,3), 'b.', 'MarkerSize', 7)
grid on;
figure;
plot3(data(720).XYZ(:,1), data(720).XYZ(:,2), data(720).XYZ(:,3), 'b.', 'MarkerSize', 7)
grid on;
figure;
plot3(datachannel(1).XYZ(:,1), datachannel(1).XYZ(:,2), datachannel(1).XYZ(:,3), 'b.',...
'MarkerSize', 7)
```

```
grid on;
figure;
plot3(datachannel(720).XYZ(:,1), datachannel(720).XYZ(:,2), datachannel(720).XYZ(:,3),
'b.', 'MarkerSize', 7)
grid on;
figure;
scatter(j, meandistancechannel);
grid on;
xlabel('Time (min)')
ylabel('Mean distance (m)')
figure;
scatter(j, (meandistancechannel - meandistancechannel(1))./meandistancechannel(1));
grid on;
xlabel('Time (min)')
ylabel('frac{z(t) - z(t_0)}{z(t_0)}', 'interpreter', 'latex', 'FontSize', 16)
```

VIII Warm-Up and Stability, 21m elaboration

%MATLAB code to perform the statistical elaboration related to a single channel selected by the user for 21 m acquisitions.

```
M=[]; STDD=[]; STDDN=[];
Zbase = -0.205;
deltaZ = 0.225;
numchannel = input('Channel number (1 to 3)? ');
Zmintemp = Zbase + (deltaZ*(numchannel-1)) - (deltaZ/2);
Zmaxtemp = Zbase + (deltaZ*(numchannel-1)) + (deltaZ/2);
for i = 1:1:totaltime
index = ((data(i).XYZ(:,3) > Zmintemp) & (data(i).XYZ(:,3) < Zmaxtemp));
X = data(i).XYZ(index, 1);
Y = data(i).XYZ(index, 2);
Z = data(i).XYZ(index, 3);
I = data(i).intensity(index);
```

```

Mtemp = mean(X);
M = [M, Mtemp];
STDDtemp = std(X);
STDD = [STDD, STDDtemp];
STDDNtemp = std(X)/(sqrt(numel(X))*M(1));
STDDN = [STDDN, STDDNtemp];
end
j = 1:1:totaltime;
figure;
scatter(j, M);
grid on;
xlabel('Time (min)')
ylabel('Mean distance (m)')
figure;
scatter(j, (M - M(1))/M(1));
grid on;
xlabel('Time (min)')
ylabel('frac{z(t) - z(t_0)}{z(t_0)}', 'interpreter', 'latex', 'FontSize', 16)
figure;
scatter(j, STDDN);
grid on;
xlabel('Time (min)')
ylabel('frac{S_z(t)}{z(t_0) sqrt(N)}', 'interpreter', 'latex', 'FontSize', 16)

```

IX Angular crop and data elaboration

%MATLAB code to crop and elaborate data based on spherical coordinates (according to how the LiDAR performs measurements).

```

deltaaz = 0.13;
deltaze = 0.625;
minaz = 1;
minze = 0;

```

```

maxaz = 2.5;
maxze = 0.8;
Npoints = 11;
Nchannels = 2;
Yhist = NaN(totaltime, Npoints*Nchannels);
Zhist = NaN(totaltime, Npoints*Nchannels);
MediaMRS7 = zeros(totaltime, Nchannels);
stdevMRS7 = zeros(totaltime, Nchannels);
stdevMRS7N = zeros(totaltime, Nchannels);
radiustemp = NaN(totaltime, 1);
h = NaN(Nchannels, Npoints);
[XYZsph.azimut, XYZsph.zenit, XYZsph.radius] = ...
cart2sph(XYZtemp(:,1), XYZtemp(:,2), XYZtemp(:,3));
for i= 1:1:totaltime
[datasph(i).azimut, datasph(i).zenit, datasph(i).radius] = ...
cart2sph(data(i).XYZ(:,1), data(i).XYZ(:,2), data(i).XYZ(:,3));
index = ((rad2deg(datasph(i).azimut)>minaz)&(rad2deg(datasph(i).azimut)<maxaz)...
& (rad2deg(datasph(i).zenit)>minze)&(rad2deg(datasph(i).zenit)<maxze)));
datacropsph(i).azimut = datasph(i).azimut(index);
datacropsph(i).zenit = datasph(i).zenit(index);
datacropsph(i).radius = datasph(i).radius(index);
datacrop(i).XYZ = data(i).XYZ(index, 1:3);
datacrop(i).intensity = data(i).intensity(index);
Yhist(i, :) = datacrop(i).XYZ(:, 2);
Zhist(i, :) = datacrop(i).XYZ(:, 3);
end
figure()
for i = 1:1:totaltime
plot3(rad2deg(datacropsph(i).azimut), rad2deg(datacropsph(i).zenit),...
datacropsph(i).radius, 'b.', 'MarkerSize', 8)
hold on
end
xlabel('Azimut')
ylabel('Zenit')

```

```

xlabel('Radius')
xlim([minaz maxaz])
ylim([minze maxze])
grid on
for i = 1:1:totaltime
for row = 1:1:Nchannels
X = datacrop(i).XYZ(1+Npoints*(row-1):Npoints*row, 2);
Y = datacrop(i).XYZ(1+Npoints*(row-1):Npoints*row, 3);
Z = datacrop(i).XYZ(1+Npoints*(row-1):Npoints*row, 1);
I = datacrop(i).intensity(1+Npoints*(row-1):Npoints*row);
MediaMRS7(i, row) = mean(Z);
stdevMRS7(i, row) = std(Z);
stdevMRS7N(i, row) = std(Z)/(sqrt(Npoints)*MediaMRS7(1));
end
end
j = 1:1:totaltime;
for row = 1:1:Nchannels
figure;
scatter(j, MediaMRS7(:, row));
grid on;
xlabel('Time (min)')
ylabel('Mean distance (m)')
figure;
scatter(j, (MediaMRS7(:, row) - MediaMRS7(1, row))/MediaMRS7(1, row));
grid on;
xlabel('Time (min)')
ylabel('  $\frac{\bar{z}(t) - \bar{z}(t_0)}{\bar{z}(t_0)}$  ', 'interpreter', 'latex', 'FontSize', 16)
figure;
scatter(j, stdevMRS7N(:, row));
grid on;
xlabel('Time (min)')
ylabel('  $\frac{S_z(t)}{\bar{z}(t_0)\sqrt{N}}$  ', 'interpreter', 'latex', 'FontSize', 16)
end

```

```

for row = 1:1:Nchannels
for column = 1:1:Npoints
for i = 1:1:totaltime
if (rad2deg(datacropsph(i).azimut(column+Npoints*(row-1)))) > ...
(minaz+(column-1)*deltaaz) & ...
(rad2deg(datacropsph(i).azimut(column+Npoints*(row-1)))) < ...
(minaz+(column)*deltaaz) & ...
(rad2deg(datacropsph(i).zenit(column+Npoints*(row-1)))) > ...
(minze+(row-1)*deltaze) & ...
(rad2deg(datacropsph(i).zenit(column+Npoints*(row-1)))) < ...
(minze+(row)*deltaze)
radiustemp(i) = datacropsph(i).radius(column+Npoints*(row-1));
end
end
figure;
histogram(radiustemp);
title(sprintf('Histogram for row %d and column %d', row, column))
end
end
figure
histogram2(Yhist, Zhist, [Npoints Nchannels]);

```

X Range Error and Quantization acquisition

%MATLAB code to collect the data required for Range Error and Quantization measurements.

```

%Start comment after first measurement
numerodistanze = 0;
numeromisure = 0;
numeroechofilter = 1;
Ymin = -1;
Ymax = 1;

```

```

Zmin = -1;
Zmax = 0.35;
totaltime = 25;
%End comment after first measurement.
camera = rossubscriber('/cloud');
for i = 1:1:totaltime
scandata = receive(camera,1);
XYZtemp = readXYZ(scandata);
Itemp = readField(scandata, 'intensity');
index = ((XYZtemp(:,2) > Ymin) & (XYZtemp(:,2) < Ymax) & (XYZtemp(:,3) > Zmin)
& (XYZtemp(:,3) < Zmax));
switch numeroechofilter
case 1
datafirst(i).XYZ = XYZtemp(index, 1:3);
datafirst(i).intensity = Itemp(index);
case 2
dataall(i).XYZ = XYZtemp(index, 1:3);
dataall(i).intensity = Itemp(index);
case 3
datalast(i).XYZ = XYZtemp(index, 1:3);
datalast(i).intensity = Itemp(index);
otherwise
disp('Error acquiring data')
end
end
switch numeroechofilter
case 1
pcloudsfirst(nerodistanze, :) = datafirst;
disp('pointcloud first acquisita')
case 2
pcloudsall(nerodistanze, :) = dataall;
disp('pointcloud all acquired')
case 3
pcloudslast(nerodistanze, :) = datalast;

```

```

disp('pointcloud last acquired')
otherwise
disp('Error acquiring pointcloud')
end
if numeroechofilter == 3
numerodistanze = numerodistanze + 1
disp('Acquisition at current distance completed, move the target (1 cm forward) ')
end
numeroechofilter = mod(numeroechofilter, 3) + 1
numeromisure = numeromisure + totaltime
if numerodistanze == 50
disp('Acquisition completed for the 50 cm long interval ')
end

```

XI Range Error and Quantization elaboration

%MATLAB sample code to preliminary elaborate the data acquired for Range Error and Quantization measurements.

```

deltaaz = 0.13;
deltaze = 0.625;
minaz = 1;
minze = 0;
maxaz = 2.5; maxze = 0.8;
Npoints = 11;
Nchannels = 1;
nbins = 10;
deltabins = 0.0625;
centre = [0 : deltabins : nbins * deltabins] + 6.8125;
asseX = [0.000 5.805 12.066 17.586 22.454 28.343 33.638 37.643 42.071 48.486 51.647
56.541 62.701 66.783 71.311 77.067 81.432 86.826 92.021 96.463 102.642 107.596 111.866
115.907 122.384 126.333 131.193 135.454 140.433 146.442 151.407 156.358 161.921 167.046
172.342 177.069 181.630 186.788 192.264 192.264+5.346 192.264+10.818 192.264+16.390

```

```

192.264+20.725 192.264+26.777 192.264+31.335 192.264+35.900 192.264+41.464 ...
192.264+45.959 192.264+51.151 192.264+55.627];
numerooccorrenze = zeros(nbins, numerodistanze);
radiustemp = zeros(numerodistanze, totaltime, 1);
for distance = 1:1:numerodistanze
for i= 1:1:totaltime
[datasph(distance, i).azimut, data_sph(distance, i).zenit, data_sph(distance, i).radius] = ...
cart2sph(pointclouds(distance, i).XYZ(:,1), pointclouds(distance, i).XYZ(:,2),...
pointclouds(distance, i).XYZ(:,3));
index = ((rad2deg(datasph(i).azimut)>minaz)&(rad2deg(datasph(i).azimut)<maxaz)...
& (rad2deg(datasph(i).zenit)>minze)&(rad2deg(datasph(i).zenit)<maxze)));
datacropsph(distance, i).azimut = datasph(distance, i).azimut(index);
datacropsph(distance, i).zenit = datasph(distance, i).zenit(index);
datacropsph(distance, i).radius = datasph(distance, i).radius(index);
end
end
figure;
for bin = 1:1:nbins
for distance = 1:1:numerodistanze
count = 0;
for i = 1:1:totaltime
for row = 1:1:Nchannels
for column = 1:1:Npoints
if (rad2deg(datacropsph(distance,i).azimut(column+Npoints*(row-1)))) >...
(minaz+(column-1)*deltaaz) &...
(rad2deg(datacropsph(distance, i).azimut(column+Npoints*(row-1)))) <...
(minaz+(column)*deltaaz) &...
(rad2deg(datacropsph(distance, i).zenit(column+Npoints*(row-1)))) >...
(minze+(row-1)*deltaze) &...
(rad2deg(datacropsph(distance, i).zenit(column+Npoints*(row-1)))) <...
(minze+(row)*deltaze)
if datacropsph(distance, i).radius(column+Npoints*(row-1)) >=...
(centre(bin)-deltabins/2) &...
datacropsph(distance, i).radius(column+Npoints*(row-1)) <=...

```

```
(centre(bin)+deltabins/2)
count = count + 1;
end
end
end
end
end
numerooccorrenze(bin, distance) = count;
end
f = fit(asseX', numerooccorrenze(bin, :)', 'gauss1');
plot(f, asseX, numerooccorrenze(bin, :));
xlabel('Distance measured by the interferometer')
ylabel('Number of occurrences')
hold on
end
```


Bibliography

- [1] ASTM E2938-15, “Standard test method for evaluating the relative range measurement performance of 3D imaging systems in the medium range,” ASTM International, West Conshohocken, PA, USA, Tech. Rep. DOI:10.1520/E2938-15, 2015, <http://www.astm.org/cgi-bin/resolver.cgi?E2938>
- [2] ASTM E3125-17, “Standard test method for evaluating the point-to-point distance measurement performance of spherical coordinate 3D imaging systems in the medium range,” ASTM International, West Conshohocken, PA, USA, Tech. Rep. DOI:10.1520/E3125-17, 2017, <http://www.astm.org/cgi-bin/resolver.cgi?E3125-17>.
- [3] ISO 17123-1:2014, “Optics and optical instruments — Field procedures for testing geodetic and surveying instruments — Part 1: Theory,” ISO — International Organization for Standardization, Geneva, Switzerland, Tech. Rep., 2014, <https://www.iso.org/standard/64156.html>
- [4] ISO 17123-9:2018, “Optics and optical instruments — Field procedures for testing geodetic and surveying instruments — Part 9: Terrestrial laser scanners,” ISO — International Organization for Standardization, Geneva, Switzerland, Tech. Rep., 2018, <https://www.iso.org/standard/68382.html>
- [5] ISO 17123-4:2012, “Optics and optical instruments — Field procedures for testing geodetic and surveying instruments — Part 4: Electro-optical distance meters (EDM measurements to reflectors),” ISO — International Organization for Standardization, Geneva, Switzerland, Tech. Rep., 2012, <https://www.iso.org/standard/54624.html>
- [6] ISO 17123-6:2012, “Optics and optical instruments — Field procedures for testing geodetic and surveying instruments — Part 6: Rotating lasers,” ISO — International

- Organization for Standardization, Geneva, Switzerland, Tech. Rep., 2012, <https://www.iso.org/standard/56088.html>
- [7] ISO/TS 19159-2:2016, “Geographic information — Calibration and validation of remote sensing imagery sensors and data — Part 2: Lidar,” ISO — International Organization for Standardization, Geneva, Switzerland, Tech. Rep., 2016, <https://www.iso.org/standard/64768.html>
- [8] F. Wang, Y. Zhuang, H. Gu, and H. Hu, “Automatic generation of synthetic LiDAR point clouds for 3-d data analysis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 7, pp. 2671–2673, July 2019.
- [9] S. Xie, D. Yang, K. Jiang, Y. Zhong, “Pixels and 3-D points alignment method for the fusion of camera and LiDAR data,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 10, pp. 3661–3676, 2019.
- [10] R. Ma, M. Liu, H. Zheng, R. Ma, Z. Zhu, “A 66db linear dynamic range, 100 dB transimpedance gain TIA with High Speed PDSH for LiDAR,” *IEEE Transactions on Instrumentation and Measurement*, 2019.
- [11] S. Saponara and B. Neri, “Radar sensor signal acquisition and multidimensional FFT processing for surveillance applications in transport systems,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 4, pp. 604–615, April 2017.
- [12] B. Fu, Y. Wang, X. Ding, Y. Jiao, L. Tang, R. Xiong, “LiDAR-Camera Calibration under Arbitrary Configurations: Observability and Methods,” *IEEE Transactions on Instrumentation and Measurement*, 2019.
- [13] S. Kurtti, J. Jansson, J. Kostamovaara, “A CMOS Receiver-TDC Chip Set for Accurate Pulsed TOF Laser Ranging,” *IEEE Transactions on Instrumentation and Measurement*, 2019.
- [14] C. Ma, Y. Guo, Y. Lei, W. An, “Binary volumetric convolutional neural networks for 3-d object recognition,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 1, pp. 38–48, Jan 2019.
- [15] M. A. Cooper, J. F. Raquet, and R. Patton, “Range information characterization of the hokuyo UST-20LX LIDAR sensor,” *Photonics*, vol. 5, no. 2, 2018. [Online]. Available: <https://www.mdpi.com/2304-6732/5/2/12>

-
- [16] Z. Wang, Y. Liu, Q. Liao, H. Ye, M. Liu, and L. Wang, "Characterization of a RS-LiDAR for 3D perception," *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, July 2018, pp. 564–569.
- [17] P. K. Rachakonda, B. Muralikrishnan, M. Shilling, D. Sawyer, G. Cheok, "An overview of activities at NIST towards the proposed ASTM E57 3D imaging system point-to-point distance standard," *Journal of the CMSC*, vol. 12, no. 2, pp. 1–14, 2017.
- [18] A. G. Kashani, M. J. Olsen, C. E. Parrish, N. Wilson, "A Review of LIDAR Radiometric Processing: From Ad Hoc Intensity Correction to Rigorous Radiometric Calibration," *Sensors*, vol. 15, no. 11, pp. 28 099–28 128, 2015. [Online]. Available: <http://www.mdpi.com/1424-8220/15/11/28099>
- [19] J.-A. Beraldin, D. MacKinnon, and L. Cournoyer, "Metrological characterization of 3D imaging systems: Progress report on standards developments," in *17th International Congress of Metrology, CIM 2015*, 2015.
- [20] G. Guidi, "Metrological characterization of 3D imaging devices," in *Proc. SPIE 8791*, vol. 8791, 2013. [Online]. Available: <https://doi.org/10.1117/12.2021037>
- [21] A. F. Habib, A. P. Kersting, A. Shaker, and W.-Y. Yan, "Geometric calibration and radiometric correction of lidar data and their impact on the quality of derived products," *Sensors*, vol. 11, no. 9, pp. 9069–9097, 2011. [Online]. Available: <http://www.mdpi.com/1424-8220/11/9/9069>
- [22] M. Tsakiri, V. Pagounis, and O. Arabatzi, "Evaluation of a pulsed terrestrial laser scanner based on ISO standards," *Surface Topography: Metrology and Properties*, vol. 3, no. 1, p. 015006, feb 2015. [Online]. Available: <https://doi.org/10.1088%2F2051-672x%2F3%2F1%2F015006>
- [23] B. Muralikrishnan, M. Ferrucci, D. Sawyer, G. Gerner, V. Lee, C. Blackburn, S. Phillips, P. Petrov, Y. Yakovlev, A. Astrelin, S. Milligan, and J. Palmateer, "Volumetric performance evaluation of a laser scanner based on geometric error model," *Precision Engineering*, vol. 40, pp. 139 – 150, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0141635914002013>

- [24] J. Laconte, S. Deschenes, M. Labussiere, and F. Pomerleau, “Lidar measurement bias estimation via return waveform modelling in a context of 3D mapping,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8100–8106.
- [25] S. Cattini, L. Di Cecilia, L. Ferrari, and L. Rovati, “Optical characterization of the beams generated by 3D-LiDARs: proposed procedure and preliminary results on MRS1000,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 7796–7804, 2020. [Online]. Available: <https://doi.org/10.1109/TIM.2020.2984137>
- [26] J. Lambert, A. Carballo, A. M. Cano, P. Narksri, D. Wong, E. Takeuchi, and K. Takeda, “Performance analysis of 10 models of 3d lidars for automated driving,” *IEEE Access*, vol. 8, pp. 131 699–131 722, 2020.
- [27] S. Cattini, D. Cassanelli, L. D. Cecilia, L. Ferrari, and L. Rovati, “A procedure for the characterization and comparison of 3-d lidar systems,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.
- [28] P. McManamon, “Field Guide to Lidar”, *Bellingham: Society of Photo-Optical Instrumentation Engineers (SPIE)*, 2015.
- [29] M. Khader, S. Cherian, “An introduction to automotive LIDAR”, *TI Marketing Whitepapers*, 2020 available at <https://www.ti.com/lit/pdf/slyy150?keyMatch=AN%20INTRODUCTION%20TO%20AUTOMOTIVE%20LIDAR&tisearch=Search-EN-everything>
- [30] BIPM, “JCGM 106:2012, Evaluation of measurement data – The role of measurement uncertainty in conformity assessment”, 2012.
- [31] BIPM, “JCGM 200:2012, International Vocabulary of Metrology – Basic and General Concepts and Associated Terms (VIM)” 3rd edition, 2012.
- [32] J. Laconte, S. Deschenes, M. Labussiere, and F. Pomerleau, “Lidar measurement bias estimation via return waveform modelling in a context of 3D mapping”, *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8100–8106.
- [33] S. Cattini, L. Rovati, L. Di Cecilia, L. Ferrari, “Comparison of the VLP-16 LiDAR system with an absolute interferometer”, *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2020, pp. 1-6.