



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

SHADMAN RAZZAQ SIDDIQUI

USE OF ADVANCE DRIVER ASSISTANCE SYSTEM SENSORS
FOR HUMAN DETECTION AND WORK MACHINE ODOMETRY

Master of Science Thesis

Examiner: Dr. Reza Ghabcheloo and
Mika Hyvönen
Examiner and topic approved on
26th April 2017

ABSTRACT

Shadman Razzaq Siddiqui: Use of Advance Driver Assistance System Sensors for Human Detection and Work Machine Odometry

Tampere University of technology

Master of Science Thesis, 53 pages, 06 Appendix pages, May 2017

Master's Degree Programme in Automation Engineering

Major: Factory Automation and Industrial Informatics

Minor: Learning and Intelligent Systems

Examiner: Dr. Reza Ghabcheloo and Mika Hyvönen

Keywords: ADAS, solid-state lidar sensor, automotive radar sensor, ROS visualization, XPC target, real-time Simulink, ego-motion, radar odometry

This master thesis covers two major topics, the first is the use of Advance driver assistance system (ADAS) sensors for human detection, and second is the use of ADAS sensors for the odometry estimation of the mobile work machine. Solid-state lidar and automotive radar sensors are used as the ADAS sensors. Real-time Simulink models are created for both the sensors. The data is collected from the sensors by connecting the sensors with the XPC target via CAN communication. Later the data is later sent to Robot Operating System (ROS) for visualization. The testing of the solid-state lidar and automotive radar sensors has been performed in different conditions and scenarios, it isn't limited to human detection only. Detection of cars, machines, building, fence and other multiple objects have also been tested. Moreover, the two major cases for the testing of the sensors were the static case and the dynamic case. For the static case, both the sensors were mounted on a stationary rack and the moving/stationary objects were detected by the sensors. For the dynamic case, both the sensors were mounted on the GIM mobile machine [21], and the machine was driven around for the sensors to detect an object in the environment. The results are promising, and it is concluded that the sensors can be used for the human detection and for some other applications as well.

Furthermore, this research presents an algorithm used to estimate the complete odometry/ego-motion of the mobile work machine. For this purpose, we are using an automotive radar sensor. Using this sensor and a gyroscope, we seek a complete odometry of the GIM mobile machine, which includes 2-components of linear speed (forward and side slip) and a single component of angular speed. Kinematic equations are calculated having the constraints of vehicle motion and stationary points in the environment. Radial velocity and the azimuth angle of the objects detected are the major components of the kinematic equations provided by the automotive radar sensor. A stationary environment is a compulsory clause in accurate estimation of radar odometry. Assuming the points detected by the automotive radar sensor are stationary, it is then possible to calculate all the three unknown components of speed. However, it is impossible to calculate all the three components using a single radar sensor, because the latter system of equation becomes singular. Literature suggests use of multiple radar sensors, however, in this research, a vertical gyroscope is used to overcome this singularity. GIM mobile machine having a single automotive radar sensor and a vertical gyroscope is used for the experimentation. The results have been compared with the algorithm presented in [32] as well as the wheel odometry of the GIM mobile machine. Furthermore, the results have also been tested with complete navigation solution (GNSS included) as a reference path.

PREFACE

This dissertation has been carried out at the Laboratory of Automation and Hydraulics (AUT), Tampere University of Technology (TUT), Finland. First of all, I would like to thank my mentor and supervisor Dr. Reza Ghabcheloo, for always being supportive and guiding me so patiently even in difficult times. I will also like to thank Mika Hyvönen and Antti Kolu, who helped me throughout my thesis especially during the testing phase. Special thanks to (Forum of Intelligent Machines) FIMA for funding the human detection project. The software implementation of the automotive radar sensor for ROS in Chapter 2 is developed by Antti Kolu, doctoral student at AUT, TUT.

Finally, my Parents and Brother Faizan, deserve my deepest thanks because without their love and support, I would never have completed my studies. Last but not the least, I would like to thank my friends Ihtisham Ali, Usman Khan, Zeeshan Waheed, Umer Iftikhar and Abdul Mannan for supporting me during my Masters and my thesis work

Tampere, May.31st.2017

Shadman Razzaq Siddiqui

CONTENTS

1.	INTRODUCTION	1
1.1	Human Detection using Advance Driver Assistance System	2
1.2	Solid-state Lidar Sensor (Leddar)	3
1.2.1	Architecture.....	3
1.2.2	CAN Communication	5
1.3	Automotive Radar Sensor	6
1.3.1	Architecture.....	6
1.3.2	CAN Communication	7
1.4	Odometry Estimation of Work Machine.....	8
1.5	GIM Mobile Machine.....	10
2.	SOFTWARE IMPLEMENTATION FOR SENSOR READING AND PARSING	12
2.1	Real-Time Simulink	12
2.1.1	Simulink model for Solid-state Lidar sensor	12
2.1.2	Simulink Model for Automotive Radar Sensor	14
2.2	xPC Target.....	16
2.3	Robot operating system (ROS)	17
2.3.1	Solid State Lidar MATLAB model for sending data to ROS.....	18
2.3.2	Automotive Radar program for receiving data at ROS.....	19
2.4	ROS launch	23
3.	EXPERIMENTATION OF ADAS SENSORS FOR HUMAN DETECTION.....	25
3.1	Static Case.....	27
3.2	Dynamic Case	33
4.	RADAR ODOMETRY KINEMATICS	36
4.1	Complete Odometry including Side-slip.....	38
4.2	Odometry Excluding Side-slip.....	40
5.	RADAR ODOMETRY EXPERIMENTATION	41
5.1	Odometry Estimation excluding side slip	42
5.1.1	GIM mobile machine Driven Straight	42
5.1.2	GIM mobile machine Driven in circular path.....	43
5.2	Complete Odometry Estimation Including side slip	44
5.2.1	GIM mobile machine Driven Straight	45
5.2.2	GIM mobile machine Driven in circular path.....	47
6.	CONCLUSION.....	51
6.1	Human detection using ADAS sensors	51
6.2	Radar Odometry for Mobile Work Machine.....	53
	REFERENCES.....	54
	APPENDIX A: THERMAL SENSOR	59
	APPENDIX B: KELLNER’S APPROACH FOR ODOMETRY ESTIMATION	64

LIST OF FIGURES

<i>Figure 1: Solid-state lidar Sensor Evaluation kit</i>	3
<i>Figure 2: Beam Pattern width and height of Solid-state Lidar Sensor</i>	4
<i>Figure 3: a) Automotive Radar Sensor, b) field of view of radar sensor</i>	6
<i>Figure 4: Radar sensor connector pin configuration [19]</i>	7
<i>Figure 5: Generic Intelligent Machine (GIM)</i>	11
<i>Figure 6: Sending Function Code to Solid-state Lidar sensor</i>	13
<i>Figure 7: Receiving data from the Lidar sensor and sending it to the Host computer running MATLAB</i>	14
<i>Figure 8: Receiving data from the Radar sensor and sending it to the ROS for visualization</i>	15
<i>Figure 9: xPC target used for experimentation</i>	17
<i>Figure 10: Receiving sensor data from Target PC and sending it to ROS for visualization</i>	19
<i>Figure 11: a) Solid State Lidar Sensor, b) Automotive Radar sensor, c) Sensors mounted in front of the GIM mobile machine</i>	26
<i>Figure 12: Frames for the sensors and camera in the visualization</i>	26
<i>Figure 13: a) Cubic marker for Solid state Lidar sensor, b) sphere and arrow marker for Automotive Radar sensor</i>	27
<i>Figure 14: Test 1, Human detection near the sensor</i>	28
<i>Figure 15: Test 2, Person moving towards and away from the sensor</i>	28
<i>Figure 16: Test 3, Finding the Maximum distance till which a person is detected by the sensors.</i>	29
<i>Figure 17: Test 4, Human detection among multiple objects</i>	30
<i>Figure 18: Test 5, Detecting a moving GIM mobile Machine using ADAS sensors</i>	30
<i>Figure 19: Test 6, Detecting a moving car using ADAS sensors</i>	31
<i>Figure 20: Test 7, Detecting a moving bicycle using ADAS sensors</i>	31
<i>Figure 21: Test 8, Detecting multiple objects using ADAS sensors</i>	32
<i>Figure 22: Test 9, Effect of angle of ADAS sensors on detection</i>	32
<i>Figure 23: Test 10, Human detection in Dynamic conditions using ADAS sensors</i>	33
<i>Figure 24: Test 11, Detection of different mobile machines, cars and fence via ADAS sensors</i>	35
<i>Figure 25: Test 12, Detection of a Building using ADAS sensors</i>	35
<i>Figure 26: Test 13, Testing the sensors in snowy environment</i>	35
<i>Figure 27: Point p_i represented in radar frame R, body frame R and world frame W</i>	37
<i>Figure 28: Comparison of Linear v_B and angular ω_B speed of Radar odometry with Wheel odometry and Gyroscope respectively</i>	43

Figure 29: Comparison of Linear v_B and angular ω_B speed of Radar odometry with Kellner's Work [33]	43
Figure 30: Comparison of Radar odometry with wheel odometry and gyroscope, for the GIM machine moving in a circular path	44
Figure 31: Comparison of Radar odometry with Kellner's Work [33], for the GIM machine moving in a circular path	44
Figure 32: a) Linear speed v_B of the GIM machine compared with wheel odometry, b) angular speed ω_B of the GIM mobile machine measured with gyroscope, c) Side Slip v_d of the GIM machine measured with radar sensor	46
Figure 33: Comparison of Radar odometry with navigation data and wheel odometry for X and Y direction	47
Figure 34: RMS Error estimation of Radar odometry and Wheel odometry relative to Complete Navigation system for straight line test a) in X direction, b) in Y direction	47
Figure 35: a) Linear speed v_B of the GIM machine compared with wheel odometry for circular path, b) angular speed ω_B of the GIM mobile machine measured with gyroscope for circular path, c) Side Slip v_d of the GIM machine measured with radar sensor for circular path	48
Figure 36: Comparison of Radar odometry with navigation data and wheel odometry for X and Y direction for the GIM mobile machine moving in a circular path	49
Figure 37: RMS Error estimation of Radar odometry and Wheel odometry relative to Complete Navigation system for Circular path test a) in X direction, b) in Y direction	50
Figure 38: a) Field of view of Solid-state Lidar sensor for Human Detection, b) Field of view of Automotive Radar sensor for Human detection	52
Figure 39: 8 segment Thermal Sensor	59
Figure 40: Omron thermal Sensor	59
Figure 41: Receiving Data from Arduino and sending Data to ROS	61
Figure 42: Receiving Data from Arduino and sending Data to ROS	62

LIST OF SYMBOLS AND ABBREVIATIONS

ADAS	Advance Driver Assistance System
ATA	Automation Test Area
AUT	Laboratory of Automation and Hydraulics
CAN	Controlled area network
CPU	Central Processing Unit
EMI	Electromagnetic Interference
FMCW	Frequency modulated continuous wave
GIM	Generic Intelligent Machine
GND	Ground
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
LED	Light emitting diode
Lidar	Light detection and ranging
NDT	Normal Distribution Transform
Odometry	Estimation of the position of vehicle over time
Radar	Radio detection and ranging
RANSAC	Random Sample Consensus
ROS	Robot Operating system
Rviz	ROS visualization
${}^W\mathbf{r}_p$	vector between world frame and the point, w.r.t world frame
${}^W\mathbf{r}_B$	vector between origin of world and base frames w.r.t world frame
${}^B\mathbf{r}_R$	vector between origin of base and radar frames w.r.t base frame
${}^R\mathbf{r}_E$	vector between origin of radar and end frames w.r.t radar frame
${}^E\mathbf{r}_p$	vector between end frame and the point, w.r.t world frame
${}^W\mathbf{R}_B$	rotational matrix to transform body frame to world frame
${}^B\mathbf{R}_R$	rotational matrix to transform radar frame to body frame
${}^R\mathbf{R}_E$	rotational matrix to transform end frame to radar frame
SLAM	Simultaneous Localization and Mapping
TUT	Tampere University of Technology
UDP	User datagram protocol
USB	Universal serial bus

1. INTRODUCTION

There are two problems that are addressed in this research, first is to make a mobile work machine able to detect humans in the surrounding efficiently, so that the environment can be made safer to work for both the machines and humans. For this problem, our aim is to use Advance Driver Assistance System (ADAS) sensors. Solid-state lidar sensor and automotive radar sensors are the two ADAS sensors used for this purpose. Chapter 1 gives the introduction of these two sensors, by presenting the architecture and the communication platform for both the sensors. Later in Chapter 2, the software implementation for reading and parsing the data from both the sensors is presented. It includes Real-time Simulink, xPC target, and ROS visualization of data. After that, in Chapter 3 the experimental results are shown for the static case and the dynamic case of both the sensors. In the end, chapter 6 concludes the performances of the sensors.

The second problem addressed in this research is regarding the odometry of a mobile work machine. In general, odometry is computed using wheel speed sensors, but the major issue using these sensors is that they are not able to detect side slip or wheel slippage, and thus there is a need for an alternative better solution for odometry estimation of any work mobile machine. For this purpose, we are using automotive radar sensor and a vertical gyroscope (already installed in GIM). Chapter 1 contains the introduction of the odometry estimation for a mobile work machine. Research suggest that the radar sensors are already being used for odometry estimation. However, more than one radar sensors were used for complete odometry estimation, which includes two components of Linear speed (forward and side slip) and a single component of angular speed. While using a single radar sensor for odometry estimation, the side slip has been neglected. In our research, instead of multiple radar sensors, we are using a single radar sensor (for two components of linear speed) fused with a vertical gyro (for angular speed). Chapter 4 presents the kinematic equations derived for the odometry estimation of the mobile work machine, arising from the constraints among the vehicle motion and a stationary point in the environment. Radial speed and the azimuth angle of multiple objects detected by the automotive radar sensors are the main variables of the algorithm. Later in Chapter 5 the experimental results of the radar odometry are presented, and the results are compared with the already installed wheel odometry of the GIM machine, and with Kellner's [32] work. Dead reckoning trajectories are also computed to check the performance of the radar odometry with complete navigation solution as a reference (including GNSS). Lastly, chapter 6 concludes the radar odometry.

1.1 Human Detection using Advance Driver Assistance System

Advanced Driver Assistance System is based on the fusion of different sensors. The demands of safety applications require trustable technologies of sensing, and fusion of different sensors fulfill this demand [36]. For specific case of Human detection near the work machine, two range sensors are used 1) Solid-state Lidar and 2) Automotive Radar sensor. The evaluation of the performance is based on the maximum range at which human can be detected, also to test the sensors in different environmental conditions like, sunny, rainy, cloudy and during snow falling. Furthermore, to test the sensors in a static condition while mounted on a stationary rack and detecting stationary/moving objects, and in a dynamic condition while mounted on a mobile work machine and detecting stationary/moving objects in the surrounding.

In the field of Robotics, Automation and Mapping, Laser Range Scanners and Radars have a wide area of applications. Typically, radar sensors are used in but not limited to healthcare system [1], bistatic radar (ground to air missile) [2], for meteorology [3], In automotive industry [4], Localization [5], Aviation [6], and for security purposes [7]. However, Laser Range scanners have much widespread application due to the reason of low cost sensors and extensive improvement in this technology [8]. Mostly the applications cover areas which include, Aerial and UAV's [9], crane positioning [10], Object profiling [11], thickness measurement [12], vehicle safety system [13], arc welding gap detection [14], Simultaneous Localization and Mapping (SLAM) [15], Airborne applications [16] and many others.

Radars and Laser Range scanners are mostly complementing sensing devices in any given application, However, regarding their performances, there has not been done considerable research before. In comparison with radar sensors, Laser range sensors provide much more precision with a significantly compact beam, making laser sensors a preferable choice for localization and mapping [8]. Furthermore, the technology of these sensors has been matured and is still progressing with time. However, in the case of Radar sensor dust particles and water vapors do not effect measurements and they are mostly immune to it, also the Radars can measure radial velocities of the objects detected by it. A larger field of view in terms of angle and the distance is also an advantage of radars over laser range scanners. However, the radar beams are mostly transparent to some common materials [8]. According to the IEC 60825 and IEEE C95.119911, both technologies don't pose any threat to health and are accepted for being safe.

The two sensors exercised during this research are 1) LeddarTech Solid-State LiDAR sensor commonly known as Leddar Sensor with range up to 100 m, a field of view of 45°, 16 independent beams for detection, simultaneous acquisition with multiple object detection capabilities [17]. 2) Doppler/automotive Radar 76-77 GHz, with detection range up to 100 m, a field of view of 150°, can detect 32 object in a single measurement

cycle, direct measurement of radial distance and radial velocity, and is Robust to weather behavior (snow, fog, rain, dust, illumination).

1.2 Solid-state Lidar Sensor (Leddar)

Solid-state lidar is based on LiDAR technology, it can detect the position of the object present in the field of view of the sensor. In terms of remote sensing, it belongs to the active time of flight category which also includes radar and ultrasound sensors, further within this category, Leddar is a part of direct optical time-of-flight [17]. The sensor operates by emitting pulses of invisible light beams, which makes the sensor insensitive to weather conditions; rain, fog and snow doesn't reduce the performance of the sensor. Using a photodetector, the reflecting beams are recorded and are converted to digital signals for further processing. The sensor is immune to ambient light and the operating temperature for the sensor is from -40° to 100° . There are total 16 independent regions/segments in the field of view of the sensors. Each segment can detect 6 objects in a single measurement cycle. The segments are numbered and hold a specific position in the field of view, which is used to get the accurate position of the object detected. [17]

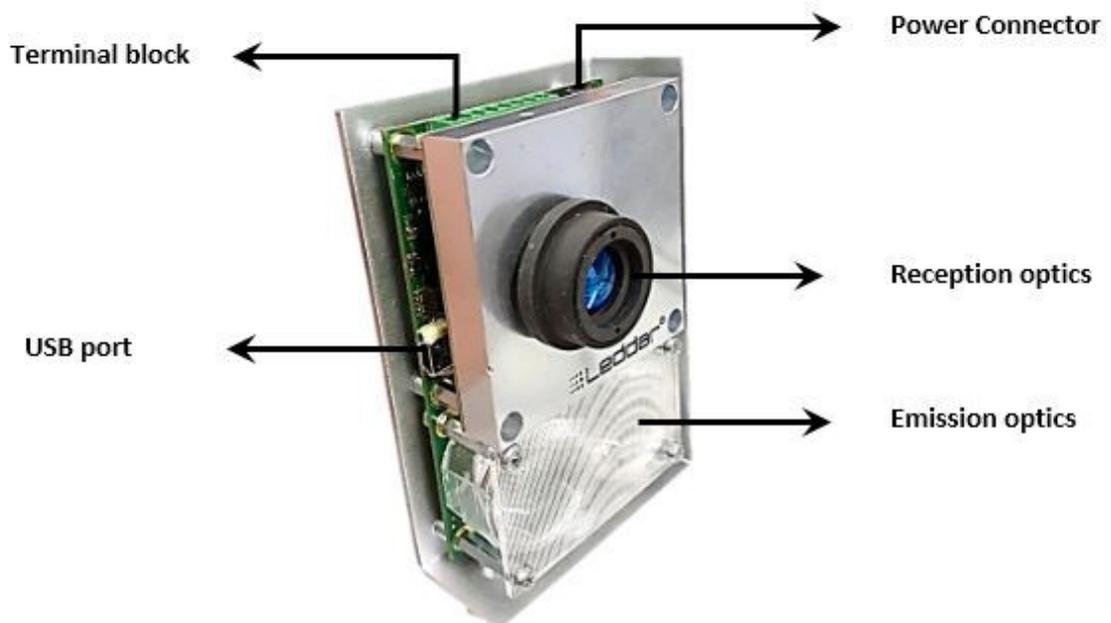


Figure 1: Solid-state lidar Sensor Evaluation kit

1.2.1 Architecture

The solid-state lidar sensor evaluation kit used for this research is shown in Figure 1. The Terminal block shown in Figure 1 contains 8 connecting pins for CAN, RS-485, and power connectivity. LeddarTech provides a Leddar configurator software to set parameters of the sensor while connected to a computer via USB port. Emissions optic are LEDs which illuminate the area at 100 kHz and the Reception optics receives the

reflected light of the LED. As mentioned above there are in total 16 independent segments which detect objects and each segment can detect up to six objects, given the objects in front do not obscure the objects behind them. Table 1 shows the specification of solid-state lidar sensor. We are using optics with a beam width of 45° and there are 16 segments, so each segment is 2.8° and height of each beam is 7.5° . Figure 2a shows the height and width of the beam for the 45° optic beam. Figure 2b shows the 6 objects detected by the solid-state lidar sensor. Object 1 shows the maximum distance till which the object is detected. Object 2 is detected by three neighboring beams. For the objects 3, 4, and 5. It can be seen that single beam can detect multiple objects, and the objects can be of any shape and size, given that the object closer to the sensor doesn't completely block the beam. For the sixth case, the object in any orientation can be detected by the sensor.

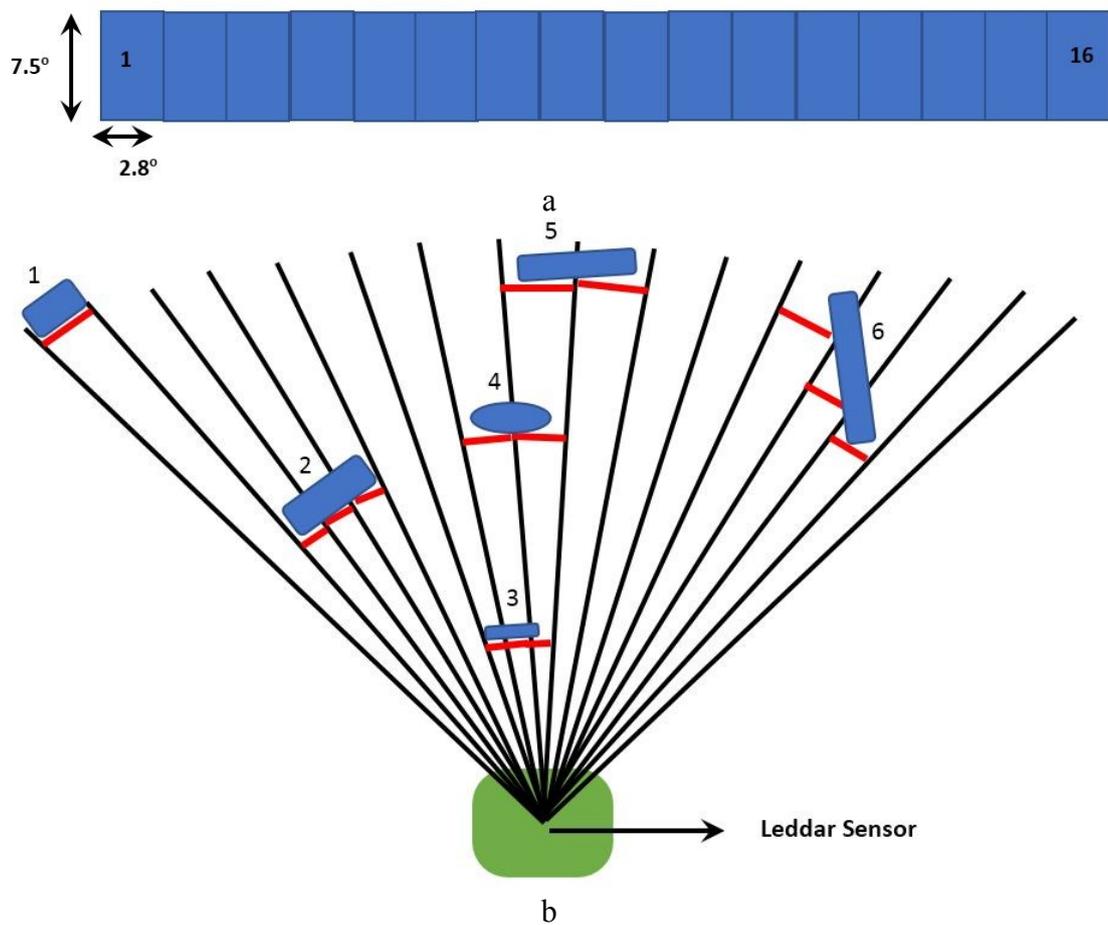


Figure 2: Beam Pattern width and height of Solid-state Lidar Sensor

Table 1: Solid-state Lidar Sensor specifications [17]

SOLID-STATE LIDAR SENSOR		
Features	Beams	9°, 18°, 24°, 34°, 45°, 95°
	Interface	USB, RS-485, CAN, UART
	wavelength	940 nm
	Power supply	12 or 24 VDC
System Performance	Detection Range	0 to 100 m (depends on optics)
	Accuracy	5 cm
	Power Consumption	4 W
	Distance Resolution	10 mm

1.2.2 CAN Communication

The terminal block contains CAN-H and CAN-L pins for CAN communication. The CAN port uses 1874 to 1970 ID to send messages in multiple message mode with a flag information, and each CAN message contain information regarding one detection. While sending the detection, one message on ID 1873 will be sent followed by messages starting from 1874 to as many as needed in the multiple message mode. For setting up the mode of the sensor a single 8-byte message is sent, which contains the information for whether the sensor should send a single message or should it send the multiple message, and when to stop sending the message [17].

1873 message ID (Tx)

This is the 8-byte message, which the sensor sends before sending the detection messages, and it contains the information of the number of detection messages that will be sent and some additional information of LED power and timestamp of the acquisition. The timestamp contains the number of milliseconds from the time the sensor was started.

1874-1970 message ID (Tx)

Each message ID is an 8-byte message, which the sensor sends. It is not necessary that the sensor sends all the 96 messages but it depends on the number of detection the sensor makes. A single message of eight bytes contains the distance at which the object is detected in first two bytes, the third and fourth byte contains the amplitude of reflection from the object, the fifth byte contains the flag information which will be later discussed, and the sixth byte contains the segment number information.

- **Flag information:** It is a 1 byte message and it contains the information regarding the validity of the detection. The first bit is set if the detection is valid. Second bit presents if the object was the result of object demerging. The fourth bit presents if the detection is saturated or not.

- **Segment number:** It provides information regarding the angle at which the object is present. As discussed above each segment is 2.8° in width and 7.5° in height, therefore using this information the angle can be easily achieved.

1.3 Automotive Radar Sensor

Radar is an acronym for Radio Detection and Ranging. Automotive radars use active scanning method for measurement, i.e. the sensor sends radio signals depending on the field of view, and analyses the reflected radio signals. Automotive radars are low-cost sensors and are resistant to different weather. The major output of automotive radar sensors is that it detects the position of the object, speed of the object which in this case is the radial speed, and the angle at which the object is detected, all relative to the sensor [18]. However, this isn't the only information provided by the sensor. It will be later discussed in the 'CAN communication' section

1.3.1 Architecture

For this research, Mid-range rear automotive radar sensor is used. The detection range of this sensor is up to 100 m with the field of view approximately 150° and angular accuracy $\pm 0.8^\circ$ as shown in Figure 3b. The cycle time of the radar sensor is 60 ms. The sensor can detect a maximum of 32 objects in a single measurement cycle. Table 2 presents the specifications for the automotive radar sensor. The sensor differentiates between objects based on velocity/distance measurement. Moreover, it is robust to weather behavior (snow, fog, rain, illumination) [19]. Figure 3a shows an automotive radar sensor.

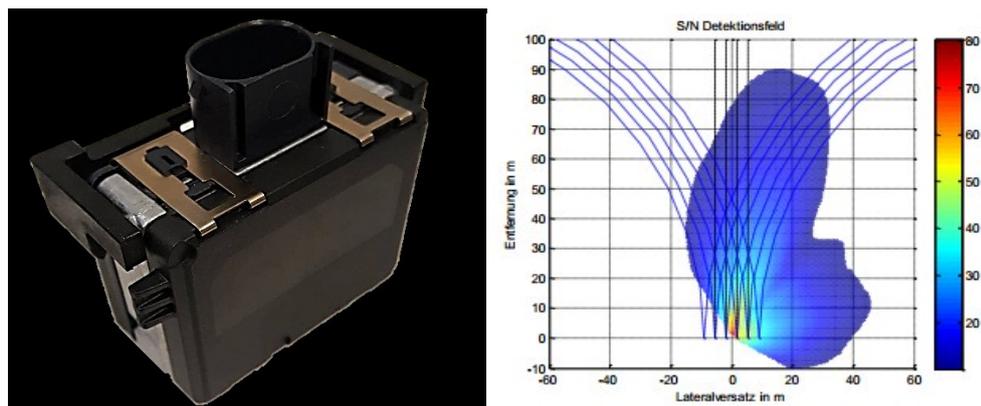


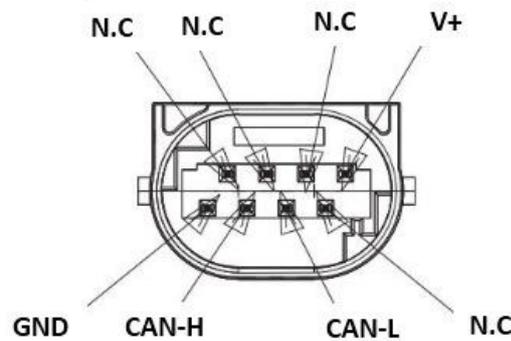
Figure 3: a) Automotive Radar Sensor, b) field of view of radar sensor

Table 2: Automotive Radar sensor specifications [19]

Features	Automotive Radar Sensor
Field of View	150°
Interface	CAN
Cycle Time	60 ms
Power supply	12 VDC
Detection Range	0 to 100 m (depends on optics)
Angular Accuracy	+/- 0.8°
Distance Resolution	10 mm
Maximum number of detections	32

1.3.2 CAN Communication

The sensor uses CAN communication for data transferring. Figure 4 gives a complete description of the wiring of the sensor. N.C is for no connection. CAN-H and CAN-L are for the CAN communication, and GND and V+ are for the 12V power supply

**Figure 4: Radar sensor connector pin configuration [19]**

As mentioned before, automotive radar sensor is not limited to distance, speed, and the angle. But it also provides a list of other information regarding the detection to filter out the data. Table 3 gives information of 11 variables, which the sensor sends for each detection in a single 8-byte CAN message. It sends the data, starting from CAN message ID of 1011 and using only the odd numbers up to 1073.

Table 3: CAN message of Automotive Radar Sensor for single object detection [19]

Signal	Description	unit
Object_vr	Radial velocity measured with respect to radar. Oncoming objects are measured with a negative sign	m/s
Object_dr	Radial distance measured with respect to the radar	m
Object_wExist	Existence probability of the object	-
Object_countAlive	Alive counter. Counts for each consecutive cycle that the object exists, remaining at the maximum number of 7.	-
Object_dbPower	Reflected power from this object, measured in db.	dB
Object_phiSdv	Standard deviation of azimuth angle.	rad

Object_phi	Azimuth angle, measured from the center of the sensor. Left side is positive.	rad
Object_flagHist	Object History flag. 0: new object in this cycle, 1: object existed in previous cycle	-
Object_flagMeas	The measured flag is a bit which states whether measured data has contributed to this object in this cycle. 0: object not measured in this cycle, 1: object measured in this cycle	-
Object_flagValid	Flag if this CAN-position contains a valid object. 0: object invalid (CAN position does not contain object), 1: object valid	-
Object_MessAconsistBit	Indicator for data consistency.	-

1.4 Odometry Estimation of Work Machine

In general, odometry is the use of motion or range sensors to compute the change in the position of a vehicle over time. It plays a key role in the field of mobile robotics and automation. Most common type of odometry is the wheel odometry, where the wheel speed sensors are used to estimate the odometry of a vehicle or a mobile machine [25]. Wheel speed sensors basically convert the rotational speed of the wheel to the linear speed of the vehicle. However, this conversion can have errors, which can be categorized into two groups; Systematic error and Non-systematic error. Systematic error is due to the kinematic imperfections/ faultiness of the vehicle. Whereas non-systematic error arises due to different types of terrain on which the vehicle is driven. Side-slip, crack bumps and wheel slippage are some of the non-systematic errors affiliated with the wheel speed sensors. Odometry is not completely dependent on wheel speed sensors, because a standard odometry is a fusion of wheel speed sensors with IMUs and GNSS sensors [23]. These absolute sensors are used to compensate the errors caused by the wheel speed sensors, but a major non-systematic or a systematic error cannot save the system from failure [20].

Odometry estimation is not limited to wheel speed sensors, it can also be estimated using relative or exteroceptive sensors such as visual odometry [22]. The focus of this research is to use a relative sensor (automotive radar sensor) to compute odometry of the GIM mobile machine and to check the performance of this algorithm. The developed algorithm can be applied to any mobile machine or a vehicle (Ackermann and non-Ackermann platforms), however for this research we are using GIM mobile machine, available at the Laboratory of Automation and Hydraulics (AUT), Tampere University of Technology (TUT). The GIM mobile machine is actually a wheel loader shown in Figure 5 (pg. 11). It was designed under the project of Generic Intelligent Machine [21]. It is currently being used for different projects, we are using it to demonstrate the algorithm of odometry estimation.

For a vehicle moving on a plane, the odometry will contain three vectors. Two of the vectors will be for the linear speed (forward and side slip) and one vector will be of angular speed. The main purpose of this research is to calculate the complete odometry

of the GIM mobile machine using a single automotive radar sensor fused with the vertical gyroscope. However, there are cases where the side slip of the mobile machine can be neglected and we have also worked with this case as well. If the side slip is neglected then the single component of both linear and angular speed can be calculated using a single radar sensor and there is no further use of a vertical gyroscope. The algorithm proposed in Chapter 5 discusses the use of radial velocity and azimuth angle of the object detected by the automotive radar sensor, for odometry estimation. The automotive radar sensor is capable of providing the information of the object, previously shown in table 3 (pg. 7).

In [23] autonomous motion control of the GIM is presented. Wheel speed sensors are utilized for odometry. The kinematic model proposed is under the assumption that there is no side-slip, neither there is any wheel slippage. Wheel odometry calculates the rotational speed of the wheel by generating pulses proportional to the frequency of the rotation. However, due to side slip and wheel slippage in different terrains the odometry data generates an error, which accumulates over time as the vehicle moves. Later in [24] the odometry estimation of the GIM mobile machine was improved using Hall sensors. The sensors were installed on all of the four wheels, the problem was that it was impossible to control the hydraulics flow of all the four wheels independently, which causes the tires to slip on slippery terrain. To remove these sorts of errors from the odometry, a least-square estimator was utilized, which caused much improvement in terms of angular speed. However, only slight improvement was recorded in linear speed when compared with simple odometry. In [26] [27] [28] and [29] wheel speed sensors are used for odometry estimation.

A rotating radar sensor K2P1 has been used in [30] by Vivet et al for odometry estimation of a vehicle. The author proposed the use of distortion, which is generally available in the data collected by the rotating sensor, as a source for odometry estimation. The drawback of this technique is that it is limited to slow moving vehicles and cannot estimate odometry of fast moving cars. Paul Checchin in [31] used Simultaneous Localization and Mapping (SLAM) with the rotating radar sensor for odometry estimation. The author used Fourier Mellin Transform to form a sequence of images, created by the radar sensor. Moreover, this technique does not require linking of landmarks.

Coming towards the automotive radar sensors; Kellner et al in [32] proposed an algorithm based on the Ackermann conditions of the vehicle using single automotive radar. Random Sample consensus (RANSAC) is utilized to remove errors or outliers from the data. The error or outlier in the data are mostly the moving objects detected by the radar sensor. Kellner et al. used a single radar sensor to estimate the linear and angular speed of the vehicle, however, he neglected the side slip, which is not possible to estimate using a single radar sensor. In [33] Kellner proposed a technique of using multiple radar sensors to calculate the complete odometry of the vehicle. The drawback of this technique is that it does not provide optimum results for non-Ackermann platforms.

In [34], a kinematic equation based approach for odometry estimation is presented, using single and multiple automotive radar sensors. For optimum position matching of radar sensor Normal Distribution Transform (NDT) is used. Furthermore, a clustering technique is used to get a Gaussian output instead of discontinuities of NDT. Multiple radar sensors are used for experimentation and the results are compared with the wheel odometry and IMUs. A similar work is presented in [35] by Coromina-Mutra et al. According to the author placement of multiple automotive radar sensor's is very much important for optimum estimation of odometry. Moreover, the author proved that it is impossible to use a single radar sensor for complete odometry estimation.

1.5 GIM Mobile Machine

GIM is a combine research project of Department of Automation and Systems Technology at Alto University and Laboratory of Automation and Hydraulics (AUT) at Tampere University of Technology. The main goal of this research is to develop skills and technologies for future autonomous mobile working machines and service robots. [21]

The vehicle for this research is a wheel loader based on Avant 635. This machine is hereafter referred to as the GIM mobile machine. As shown in Figure 5. GIM mobile machine contains sensors like SICK laser Scanner, Inertial measurement Unit (IMUs), Global Navigation Satellite system (GNSS) sensors, and Hall sensors [22,23,24]. The machine also has a xPC target and a Linux PC as shown in Figure 5, which are used to operate different sensors and actuators in the GIM mobile machine. For this research, solid-state lidar and automotive radar sensors are mounted in front of the machine, as shown in Figure 5. A camera is also attached, for visualization of the data.

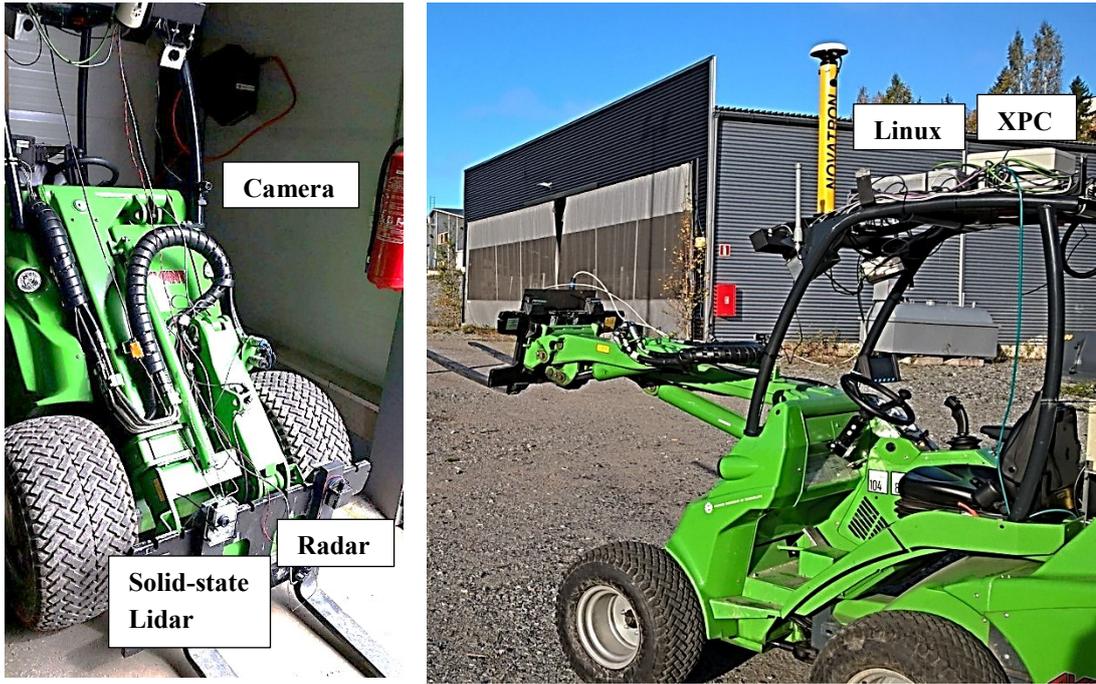


Figure 5: Generic Intelligent Machine (GIM)

2. SOFTWARE IMPLEMENTATION FOR SENSOR READING AND PARSING

2.1 *Real-Time Simulink*

MATLAB Simulink is graphical modeling environment available for simulation and model based design. In addition to this, MATLAB Simulink is also used for C/C++ code generation and continuous testing and verification of embedded systems [37]. The models can be created using a Library of different elements called blocks. Using subsystems, signals and buses; Simulink allows blocks to connect with each other and have a hierarchical relation between them.

Real-Time Simulink enables Simulink models to be run on embedded systems or XPC target computers for real time applications. It contains driver blocks which include I/O for analog and digital, different encoders, transformers, serial, audio share memory and many others. It also includes driver blocks for different protocols and data buses, which includes RAW Ethernet, UDP, CAN, and USB video. [38]

Automotive radar sensor and solid-state lidar sensor, both use CAN for communication of data. A Real-time Simulink model is created for both sensors so that the data can be read and then sent for further processing and visualization. In the following section, the models have been discussed for both the sensors.

2.1.1 Simulink model for Solid-state Lidar sensor

For the sensor to start sending data, first an 8-byte message is sent to 1856 CAN message ID. There are 3 function codes used to operate the sensor. By sending the first byte with a value of '1' the sensor will send only 1 detection and will then stop. By sending the first byte with a value of '2', the sensor will start sending the values continuously and will only stop when the value of '3' is sent in the first byte. The model for sending the value and the configuration is shown in Figure 6. We are using the sensor in the multi message sending mode.

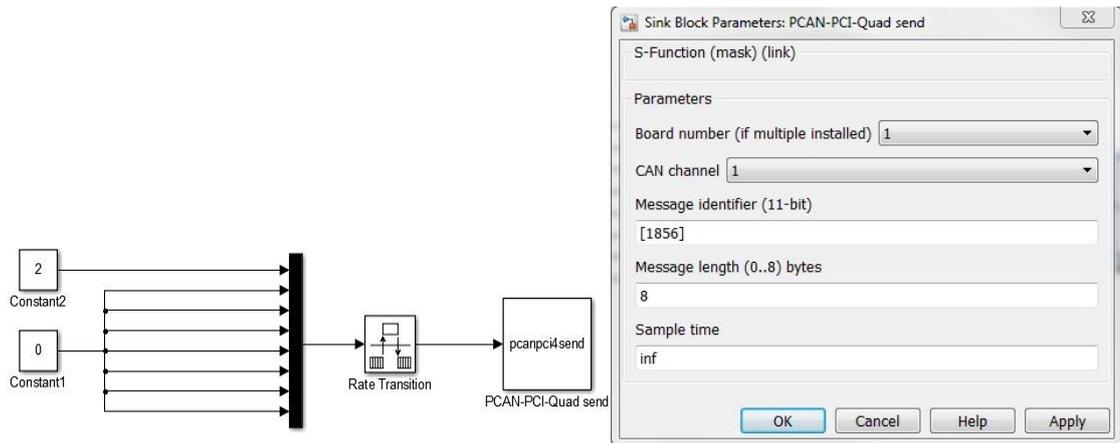


Figure 6: Sending Function Code to Solid-state Lidar sensor

Once this message is received by the sensor, it will start sending detections from CAN message ID of 1875 to a maximum up to 1970 depending on the number of detection, as discussed before the sensor can send up to 96 detections. Except from ID 1875 each message ID contains first two bytes of distance measurement, then two bytes of amplitude of light, after that a single byte containing flag information and last having the segment number. Figure 7 presents the complete model of reading the sensor and sending the data via UDP.

Figure 7a shows the ‘CAN receive’ block, for this case we are considering a maximum of 32 segments, therefore the message CAN ID is from 1875 to 1907. 1st 16 segments are processed separately and the next 16 segments are processed separately. In most of the testing, done at Automation Test Area (ATA/TUT) the sensor detects no more than 32 segments. Therefore, we limit the segments to 32. By increasing the number of processing blocks the segments processing can be increased. CAN message ID 1875 contains the number of detections, this ID is always sent first and the rest of the detections are sent. In our case, we are using this information as a heartbeat, Figure 7b explains how this information is used to send the UDP package. It is also used to rest the data which is once sent by adding a 2-msec delay. RX status in the ‘CAN receive’ block gives as many output bits as there are CAN message ID’s. In this case, it is 33, which is 1 for the heartbeat and remaining having the detection information. The bit goes high for a specific CAN message ID when it receives any data. Using this information that specific block in Figure 7c is triggered. In every block, the data is analyzed and converted to double. Only the distance information and the segment number are proceeded for further processing. The data is rest once the sending of a single measurement cycle is completed, as shown in Figure 7d. Figure 7e shows the sending of the data, all the data is mux into pack and is sent via UDP. This model is run on a target XPC therefore the sending of the UDP is to the MATLAB where it is received

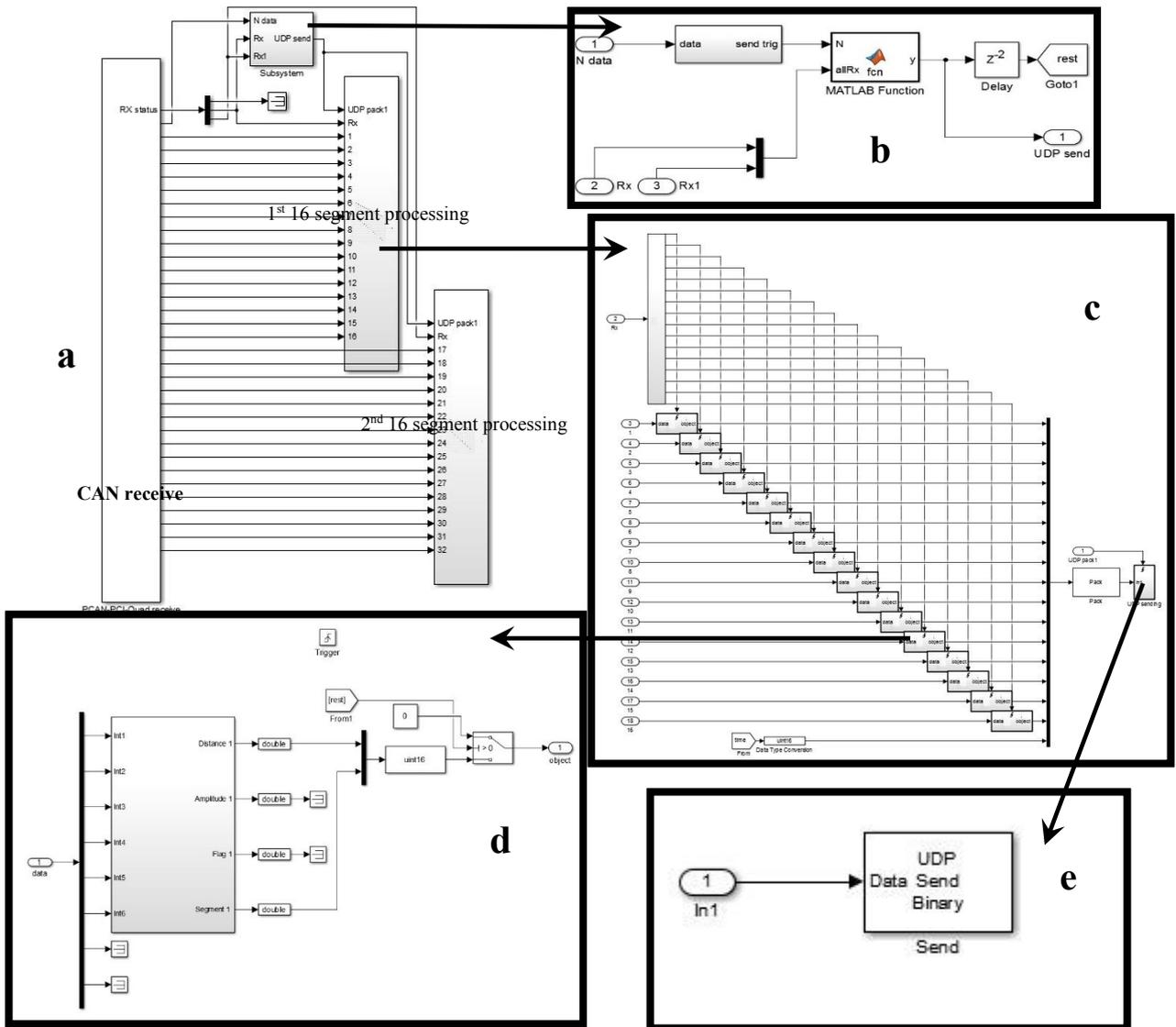


Figure 7: Receiving data from the Lidar sensor and sending it to the Host computer running *MATLAB*

2.1.2 Simulink Model for Automotive Radar Sensor

Unlike solid-state lidar sensor, automotive radar sensor is not required to be triggered or configured for sending the data. Once connected, radar starts sending data from CAN message ID of 1011 to 1073 with only the odd number IDs. As discussed before, the sensor can detect 32 objects at a time. However, it sends 32 CAN messages, even if there is not any detection of an object in a specific message, in such case the message is empty, as shown in Figure 8a, the data sent by the sensor is received and then each message is processed separately. Each subsystem in Figure 8a uses 8-bytes of data and it contains all the information which already has been explained in table 3 (pg-7). Figure 8c shows the actual processing of the data. Later the data is mux and converted in double as shown in Figure 8d. the data for each object is separately packed as shown in Figure 8e and is then sent via UDP to Robotic Operating system (ROS) for visualization.

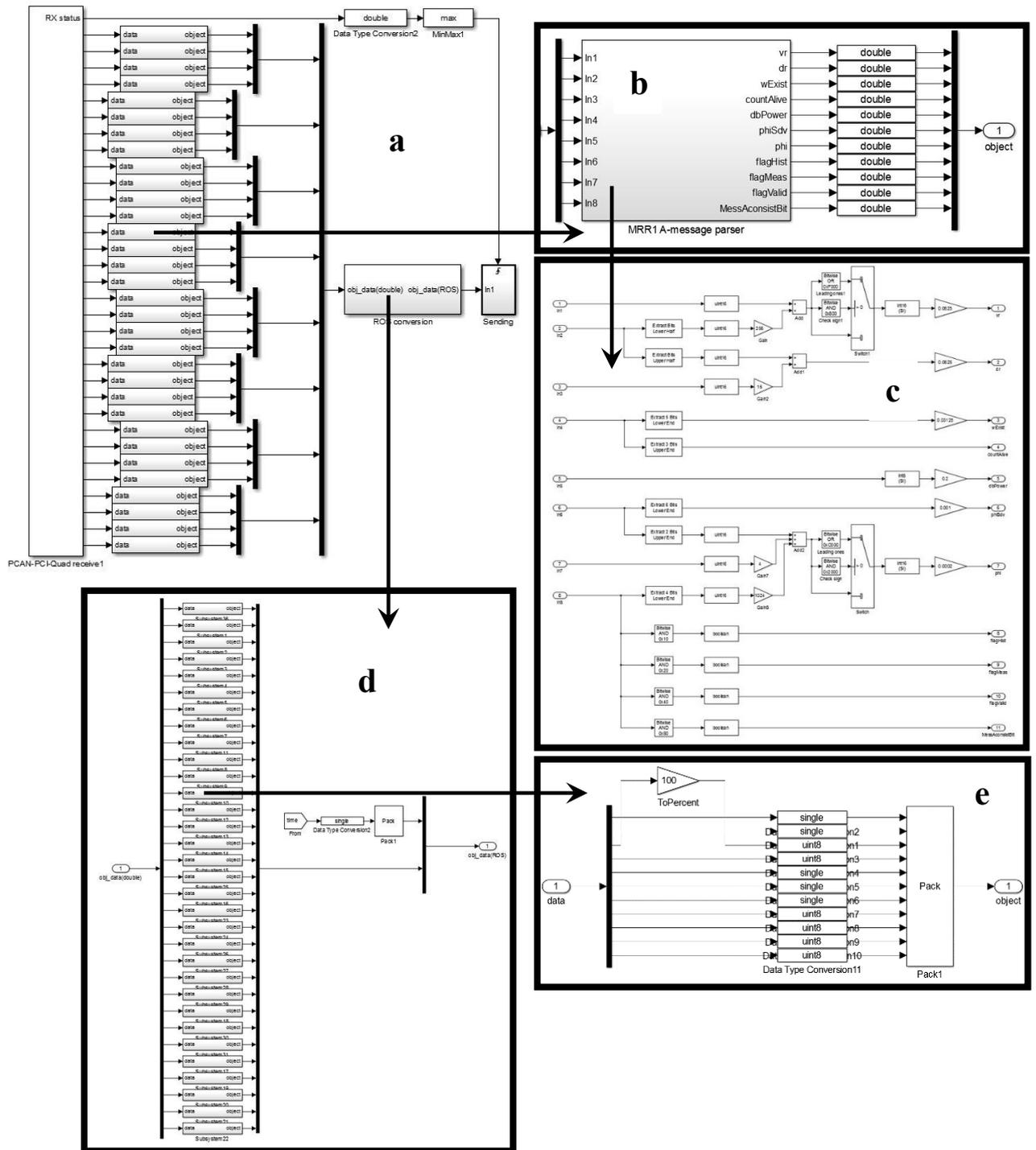


Figure 8: Receiving data from the Radar sensor and sending it to the ROS for visualization

2.2 xPC Target

xPC target is PC hardware which is generally used for testing hardware, deploying real-time systems and in few cases for prototyping as well. The xPC target works as an operating system to run only the real-time applications on a target PC, separate from a host PC. In our case, we are using xPC target for connecting Real-time Simulink with actual hardware, which in this case are the sensors. The target PC can be a desktop computer, a laptop, a workstation, PC104/PC104+, or an embedded PC, it depends on the application for which it is used. Moreover, xPC target provides I/O compatibilities and real-time target on any PC given that it contains that specific I/O card installed. There are three main steps to select the best target PC for a certain application. 1) selecting a form factor, 2) selecting the I/O boards, 3) selecting the CPU [39]

Selecting the form factor means to analyze the environment in which the target PC will work. That is, selecting specific space for the system, temperature range, mechanical vibration and shock, dust and water resistance, and Electromagnetic interference (EMI). Keeping in mind these form factors Desktop PC can be used in Laboratories or offices where there is not much vibration, dust and EMI. For harsh conditions PC/104 and embedded PCs in black box enclosure are used, which can be portable, tolerates vibration or shock, water and dust proof, and EMI resistant [39]. In our case, we are using PC/104

The next step is to select the I/O boards, there are different types of I/O boards commercially available, such as PCI, ISA, PC/104 and PMC. Using PCI gives an advantage of having a large number of I/O types available, moreover, it has the fastest available bus for PC systems, also it has plug and play operations. However, it can only be used for desktops and it is not rugged. ISA on the other hand has many generic I/O types still the bus is slow and modern desktops do not have ISA slots. Whereas, PC/104 and PC/104+ have very small physical dimension, plus it uses PCI bus and it can also be used for harsh environment [39]. It also has compatible I/O available for CAN card, therefore, we are using it.

Lastly the major step is to select an appropriate CPU for the target PC. It depends on the complexity of computation. The performance should be enough to execute the model within the given sample time without overloading. However, the performance is also dependent on the I/O complexities [39]. For our case, we do not need high performance CPUs, therefore the CPU is Pentium 3. Figure 9 shows the target PC used for this research, as discussed before it is PC/104 and enclosed in a box for harsh conditions.

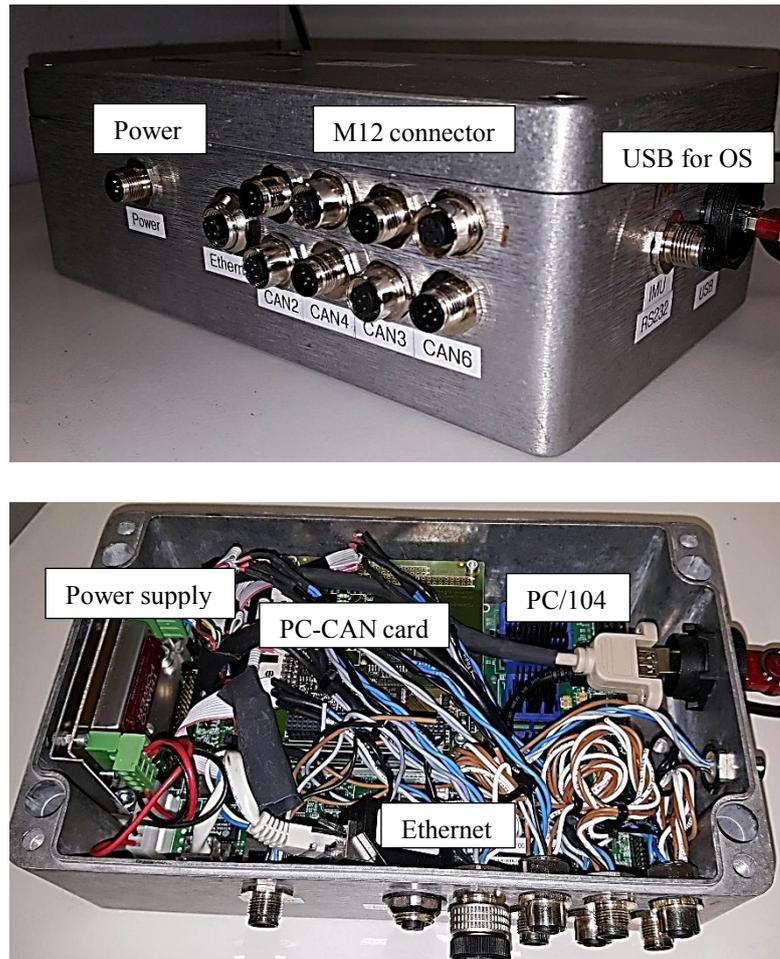


Figure 9: xPC target used for experimentation

2.3 Robot operating system (ROS)

ROS is a widely-used software in the field of Robotics. It is an open-source, meta-operating system, the services provided by ROS include hardware abstraction, implementation of commonly used functionality, visualization, low-level device control, message passing between processes, and package management [40]. Unix-based platforms such as Linux, Ubuntu, Gentoo, Arch Linux and other Linux platforms are required to run ROS. There is currently no version available for Microsoft Windows.

To launch ROS, the first step is to run ‘roscore’. It is a collection of different nodes and programs which are required to run ROS system. The main purpose of roscore is to enable ROS to communicate with other devices. By using the roscore command in the terminal, it will start up a ‘ROS master’ a ‘ROS parameter server’ and a rosout logging node [42]. After configuring ROS master the next step is to launch ‘rviz’ or ‘ROS visualization’

For our case, ‘Rviz’ is used as for visualization of sensors and monocular camera data. It is the 3D visualizer for ROS and it can display the data coming from the sensor in real-time [41]. In our case, we are using two different approaches for both the sensors. For

automotive radar sensor, the target PC sends the data directly to ROS where it is processed and visualized in rviz. Whereas, for solid-state lidar, the target PC sends the data to MATLAB, using Robotics toolbox a link is made between ROS and MATLAB, the data received in MATLAB is further processed and only the markers are sent to rviz for visualization.

In rviz there are different types of markers that are used for visualization of the data. There are in total 12 types of markers available in rviz, few of which are arrow, cube, sphere, cylinder, and line strip. In our case, we are using cube markers for the Solid-state lidar, sphere markers for radial distance and arrow markers for radial speed measured by the automotive Radar sensor. All of the markers require some special parameters for visualization, some of the major parameters are: 1) Marker.type, 2) Marker.action, 3) Marker.position in X, Y, and Z 4) Marker scale in X, Y, and Z, 5) Marker.color and 6) Marker.mesh_resource. For better understanding of the data a monocular camera is also calibrated for rviz, so that the markers can be distinguished in the camera stream.

2.3.1 Solid State Lidar MATLAB model for sending data to ROS

Figure 10 shows the complete model used to receive solid-state lidar data from the target PC and sending it to ROS for visualization. Data is first received via ‘UDP receive binary’ block, then it is unpacked as shown in Figure 10a. later each segment is processed separately as shown in Figure 10b. UDP data contains the distance detected by the sensor and the segment number. As shown in Figure 10c the x position and y position of the segment are dependent on the segment number. MATLAB functions to check the segment number and assign a dedicated angle at which the segment should be. Using this angle, the x and y position are specified for visualization. As discussed before each segment is of 3 degree in width, therefore this width is used for z scale of the marker. In Figure 10d, the subsystem of publishing the marker is shown. A blank message of visualization_msg/marker is used for this purpose. A specific frame id is assigned to the message and current and received lengths are specified. X and Y positions and are provided by the sensor data, including the scale of the markers. After that the type of the marker, mesh resource and color of the marker are selected. A MATLAB code is made for the action part, where the marker is deleted soon after the detection for the segment is zero or inf. This data is then send to the ROS. The linking of the MATLAB and ROS is done in the publisher block, where the network addresses are configured between the two-software running on separate computers via ROS master’s IP addresses.

The program has been written in C++ by Antti Kolu¹. As shown in Program 1, the first step is to include the libraries and initializing the variables that will be used later.

```

1  #include <ros/ros.h>
2  #include <visualization_msgs/MarkerArray.h>
3  #include "tf/transform_datatypes.h"
4  #include "udpreceiver.h"
5  #include "auto_radar_sensor/RadarObjectsVec.h"
6
7  struct AutoObject{
8      float radial_velocity;
9      float range;
10     unsigned char propability;
11     unsigned char count_alive;
12     float power;
13     float angle_distribution;
14     float angle;
15     unsigned char hist;
16     unsigned char meas;
17     unsigned char valid;
18     unsigned char consistency;
19 } __attribute__((packed));

```

Program 1. *Introducing the Libraries and initializing the variables*

After the above step the main program is created (Program 2), it includes initializing the node to perform any ROS arguments and name remapping provided in the command line, setting the port at which the data will be received, and initializing the filters.

```

1  int main( int argc, char** argv )
2  ros::init(argc, argv, "auto_radar_sensor_node");
3  ros::NodeHandle n;
4  ros::NodeHandle n_priv("~");
5  ros::Rate r(100);
6
7  std::string marker_topic_name="/auto_radar_sensor_markers";
8  std::string  marker_topic_name_speed="/auto_radar_sensor_speed_arrows";
9
10 std::string radar_frame = "/world";
11 std::string radar_ip = "255.255.255.255";
12 std::string image_ip = "255.255.255.255";
13 int radar_port = 25004;
14 int image_port = 25005;
15 int count_alive_filter = 0;
16 int valid_filter = 0;
17 int probability_filter = 0;

```

Program 2. *Initializing the node, setting the port, and initializing the filters*

Next step is to use the strings and ints initialized above to get parameters, and create the publishers. The advertise() function in the Program 3 is used to publish the given topic name and it also invokes a call to ROS master node to keep the registry of publishing.

¹ Antti Kolu Doctoral Student at Laboratory of Automation and Hydraulics (AUT), Tampere University of Technology (TUT)

There are two visualization messages shown in the Program 3, 1st is for the distance measurement and 2nd is for the radial speed measured by the radar sensor.

```

1  n_priv.param("MarkerTopic",marker_topic_name,
2  std::string("/auto_radar_sensor_markers"));
3
4  n_priv.param("MarkerTopicSpeed",marker_topic_name_speed,
5  std::string("/auto_radar_sensor_speed_arrows"));
6
7  n_priv.param("RadarFrame", radar_frame, std::string("/world"));
8  n_priv.param("count_alive_filter", count_alive_filter, 0);
9  n_priv.param("valid_filter", valid_filter, 0);
10 n_priv.param("probability_filter", probability_filter, 0);
11 n_priv.param("LocalNetworkIP",radar_ip,
12 std::string("255.255.255.255"));
13
14 n_priv.param("RadarPortToReceiveFrom", radar_port, 25004);
15
16 ros::Publishermarker_pub=n.advertise<visualization_msgs::MarkerArray>(m
17 arker_topic_name.c_str(),1);
18
19 ros::Publisherspeed_arrows_pub=n.advertise<visualization_msgs::MarkerAr
20 ray>(marker_topic_name_speed.c_str(), 1);
21
22 ros::Publisherauto_data_pub=n.advertise<auto_radar_sensor::RadarObjects
23 Vec>("/auto_radar_data_vec", 1);

```

Program 3. *Getting the parameters and initializing the publishers*

After that the next step is to receive the data from the UDP port and resize it, as shown in Program 4

```

1  UDPReceiver receiver;
2  if(!receiver.init(radar_port, radar_ip)){
3      ROS_ERROR("Could not initialize socket to receive data! Local
4  interface=%s port=%i", radar_ip.c_str(), radar_port);
5      return false;
6  }
7
8  int count = 0;
9  AutoObjectsVec auto_data_vec;
10 auto_radar_sensor::RadarObjectsVec auto_msgs;
11 auto_msgs.objects.resize(32);
12
13 visualization_msgs::MarkerArray marker_array;
14 visualization_msgs::MarkerArray arrow_array;

```

Program 4. *Receiving the data via UDP, checking for incoming data and resizing it*

Then the size of the data is checked if it is 0, the system waits for 100 ms and checks it again for the data else it saves the data count sets parameters for the markers to be visualized. As shown in Program 5 the data is also filtered out using the filters provided by the automotive radar sensor. For the radial distance ‘sphere’ shape marker is used where its x, y positions are figured from the distance and azimuth angle provided by the sensor, the scale in the y direction depends on the angle of distribution and the scale in z

depends on the power of reflection. The marker color is chosen to be 'blue'. For the radial speed, 'arrow' shaped marker is used, the scale in x direction is the radial speed provided for each object by the sensor. The marker color chosen for radial speed is 'red'. At the end, all the data is published, and the node is turned off to repeat the loop.

```

1 while (ros::ok())
2   {
3     int    packet_size    =    receiver.receive(&auto_data_vec,
4 sizeof(AutoObjectsVec), 1000000);
5     if(0 >= packet_size) { // wait 100ms
6       ROS_INFO("%lu", sizeof(AutoObjectsVec));
7       ros::spinOnce();
8       r.sleep();
9       continue;
10    }
11    count++;
12    ROS_INFO_THROTTLE(1, "Received total %i packets from radar.", count);
13
14    for(unsigned i = 0; i < 32; i++){
15      if(count == 1){
16        break;
17      }
18
19      visualization_msgs::Marker marker;
20      if(auto_data_vec.objects[i].valid < valid_filter ||
21 auto_data_vec.objects[i].propability < probability_filter ||
22 auto_data_vec.objects[i].count_alive < count_alive_filter){
23        marker.action = visualization_msgs::Marker::DELETE;
24      }else{
25        marker.action = visualization_msgs::Marker::ADD;
26      }
27      marker.header.frame_id = radar_frame;
28      marker.header.stamp = ros::Time::now();
29      marker.ns = "auto";
30      marker.id = i;
31      marker.type = shape;
32
33      tf::Quaternion quat = tf::createQuaternionFromYaw(-
34 auto_data_vec.objects[i].angle);
35
36      marker.pose.position.x =
37 cos(auto_data_vec.objects[i].angle)*auto_data_vec.objects[i].range;
38      marker.pose.position.y = sin(-
39 auto_data_vec.objects[i].angle)*auto_data_vec.objects[i].range;
40      marker.pose.position.z = 0;
41      marker.pose.orientation.x = quat.getX();
42      marker.pose.orientation.y = quat.getY();
43      marker.pose.orientation.z = quat.getZ();
44      marker.pose.orientation.w = quat.getW();
45
46      marker.scale.x = 0.2;
47      marker.scale.y =
48 sin(auto_data_vec.objects[i].angle_distribution*6)*auto_data_vec.object
49 s[i].range;
50      marker.scale.z = auto_data_vec.objects[i].power; // scale to power
51
52      marker.color.r = 0.0f;
53      marker.color.g = 0.0f;

```

```

54 marker.color.b = 1.0f;
55 marker.color.a = 0.5;
56
57     marker_array.markers.push_back(marker);
58
59     visualization_msgs::Marker arrow = marker;
60     arrow.type = visualization_msgs::Marker::ARROW;
61     arrow.scale.x = auto_data_vec.objects[i].radial_velocity;
62     arrow.scale.y = 0.2;
63     arrow.scale.z = 0.2;
64
65     arrow.color.r = 1.0f;
66     arrow.color.g = 0.0f;
67     arrow.color.b = 0.0f;
68     arrow.color.a = 1.0;
69     arrow_array.markers.push_back(arrow);
70
71
72 auto_msgs.header.frame_id = "radar";
73 auto_msgs.header.stamp = ros::Time::now();
74 auto_msgs.objects.at(i).pose = marker.pose;
75 auto_msgs.objects.at(i).scale = marker.scale;
76 auto_msgs.objects.at(i).velocity.x=
77 auto_data_vec.objects[i].radial_velocity;
78 auto_msgs.objects.at(i).propability=
79 auto_data_vec.objects[i].propability;
80 auto_msgs.objects.at(i).count_alive=
81 auto_data_vec.objects[i].count_alive;
82 auto_msgs.objects.at(i).power = auto_data_vec.objects[i].power;
83 auto_msgs.objects.at(i).hist = auto_data_vec.objects[i].hist;
84 auto_msgs.objects.at(i).meas = auto_data_vec.objects[i].meas;
85 auto_msgs.objects.at(i).valid = auto_data_vec.objects[i].valid;
86     }
87
88     marker_pub.publish(marker_array);
89     speed_arrows_pub.publish(arrow_array);
90     marker_array.markers.clear();
91     arrow_array.markers.clear();
92     auto_data_pub.publish(auto_msgs);
93
94     ros::spinOnce();
95     r.sleep();
}

```

Program 5. *Checking the receiving data and publishing the markers.*

2.4 ROS launch

To start the visualization of data in ‘rviz’ a launch file is required which specifies the frames and their positions for the visualization. Program 6 shows the launch file that was made for the solid state lidar sensor and automotive radar sensor. As discussed before a monocular camera/USB Cam was also utilized to get a better understanding of the environment that is detected by the sensors. The values of the filters are defined in the Launch File, as shown in Program 6, the count alive value is set to 2, valid filter value to

1, and probability filter value to 20%. The frame of the sensor is also defined here, and the topics as well. From line 16-21, the positions and orientations of the frames are defined relative to the world frame. The position values specified in this launch file are for the GIM mobile machine shown in Figure 5. Calibration of the monocular camera is done prior to use of it.

```

1 <launch>
2
3   <node      name="auto_radar_driver_node"      pkg="auto_radar_driver"
4 type="auto_radar_driver_node" output="screen">
5 <param name="MarkerTopic" value="/auto_radar_sensor_markers" />
6 <param name="MarkerTopicSpeed" value="/auto_radar_speed_arrows" />
7 <param name="BoshRawTopic" value="/auto_raw_data"/>
8 <param name="RadarFrame" value="/radar_frame" />
9 <param name="count_alive_filter" value="2" />
10 <param name="valid_filter" value="1" />
11 <param name="probability_filter" value="20" />
12 <param name="LocalNetworkIP" value="255.255.255.255" />
13 <param name="RadarPortToReceiveFrom" value="25004" />
14
15   </node>
16 <node pkg="tf" type="static_transform_publisher" name="radar_frame"
17 args="0 -0.1 0.05 0 0 0 /world /radar_frame 10"/>
18 <node pkg="tf" type="static_transform_publisher" name="camera_frame"
19 args="0 0 0 -1.57 0 -1.57 /world /head_camera 10"/>
20 <node pkg="tf" type="static_transform_publisher" name="leddar_frame"
21 args="0 0.1 0.05 0 0 0 /world /utm 10"/>
22
23 <node pkg="usb_cam" type="usb_cam_node" name="camera">
24 <param name="video_device" value="/dev/video1" />
25   </node>
26
27 </launch>

```

Program 6. *Launch File for both the sensors and the camera.*

3. EXPERIMENTATION OF ADAS SENSORS FOR HUMAN DETECTION

Our first objective is to detect humans near the automated mobile work machine, using ADAS sensors. For this reason, solid-state lidar and automotive radar sensors are chosen. Chapter 2 was regarding software implementation for sensor reading and parsing, now the next step is to test these sensors for human detection. The experimentation is not restricted to human only, detection of car, bicycle, GIM mobile machine, buildings, containers, fence and other heavy machines are also being performed. Mainly the test involves two conditions: static testing and dynamic testing.

For Static testing the sensor is in the static position and it detects moving objects in a different direction or static objects with different sizes. For dynamic testing, the sensor will be mounted on the GIM mobile machine and the machine will be driven so that the sensor can detect the moving or static environment. With these tests, we will be able to get the maximum distance for detection of objects, maximum numbers of objects detected, maximum angle for detection, also the false detections by the sensors and lastly the effect of the environment on the sensors.

The solid-state lidar sensor used for these tests is not waterproof therefore a waterproof cover is designed for it so that the sensor can work in the rain as well, polycarbonate sheet has been used for the lens so that the efficiency of the sensor is not decreased as shown in Figure 11a. On the other hand, automotive radar sensor is waterproof and doesn't need any external covering as shown in Figure 11b. Figure 11c shows the covered solid-state lidar sensor and automotive radar sensor mounted in front of the GIM mobile machine.

*a**b*

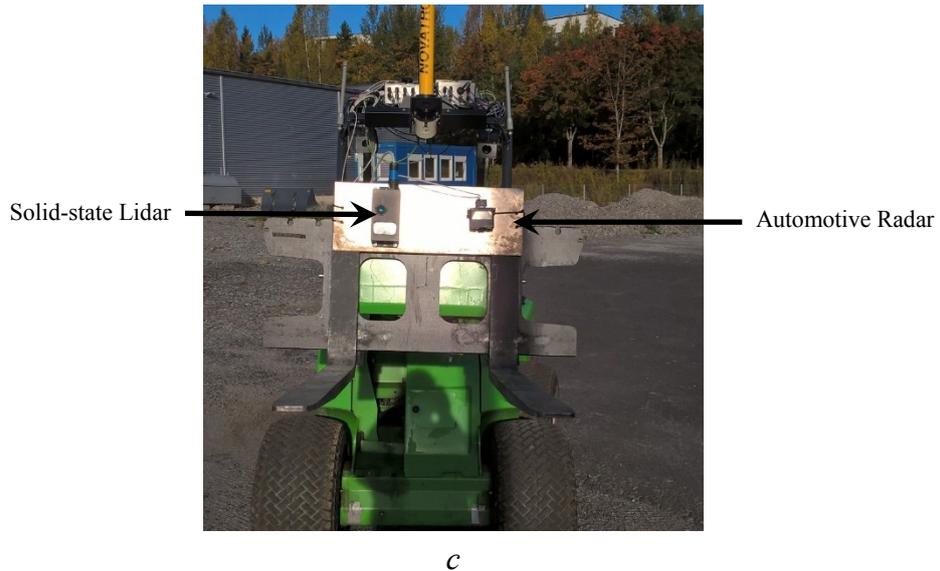


Figure 11: a) Solid State Lidar Sensor, b) Automotive Radar sensor, c) Sensors mounted in front of the GIM mobile machine

For the accurate visualization of the data it is very important that the origin frames for solid-state lidar, automotive Radar and the camera are at the right position and with right orientation. For the sensors, the distance has been measured in X and Y direction where as for the camera the frame has been rotated because it visualizes in Z direction. The orientation and the position of the frames are shown in Figure 12. In the Figure, red line shows the X direction, green shows the Y direction and blue represents Z direction of the frame.



Figure 12: Frames for the sensors and camera in the visualization

As discussed before cube markers have been used for visualization of solid-state lidar data, sphere markers for radial distance and arrows for the radial velocity of the object measured by the automotive radar sensor. Figure 13 gives a clear view of these markers. Figure 13a shows the cube markers used for solid-state lidar sensor. It is clearly visible that the width and height of the marker is increased as the distance between the object and the sensor increased. In Figure 13b, the sphere marker of automotive radar sensor is visible with the arrow marker. The height in Z direction is due to the power of reflection of the object measured by the sensor. The width in Y direction is due to the distribution

angle relative to the distance at which the object is detected. For the case of the arrow, the sensor provides the radial velocity either positive or negative. If it is negative it means the object is coming towards the sensor, therefore the direction of the arrow is towards the origin frame of the automotive radar sensor and vice versa. The length of the arrow depends on the amplitude of the radial velocity. For the reference, the grid lines are used, each grid size is 1m x 1m as shown in Figure 13.

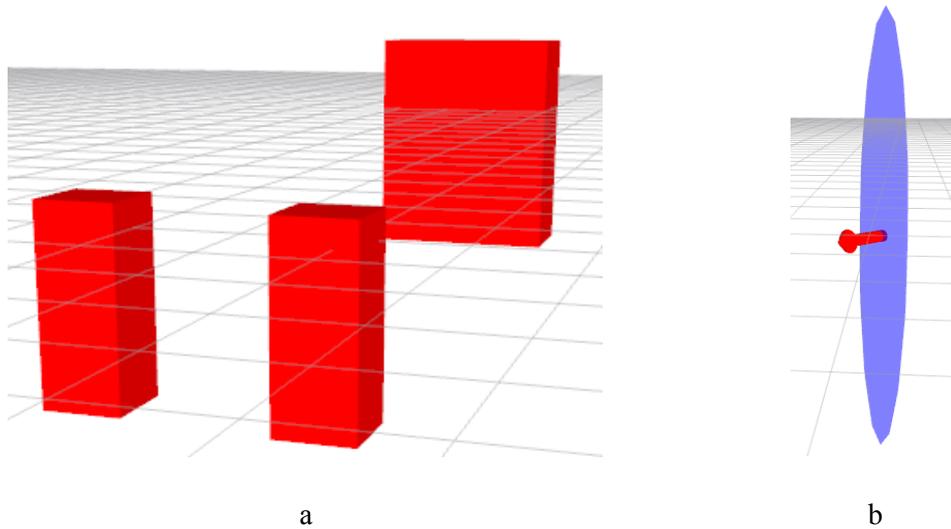


Figure 13: a) Cubic marker for Solid state Lidar sensor, b) sphere and arrow marker for Automotive Radar sensor

3.1 Static Case

This section is regarding the sensor testing in a static state in outdoor conditions, with objects either stationary or in moving conditions. The main goal of these experiments is to analyze the performance of the sensors. The tests will involve detection of human, car, fence, GIM mobile machine, multiple objects, and buildings.

First test is detecting a human near the sensors. Figure 14 shows human detected by both the sensors. It is visible that only 2 segments of solid-state lidar are detecting the human, the segment size increases as the distance between the object and the sensor increases so the same object that is detected using 2 segments near the sensor will be detected by the sensor with only 1 segment, if it is away from the sensor. The orange circle is not the part of the visualization but it is just to show the current focus point

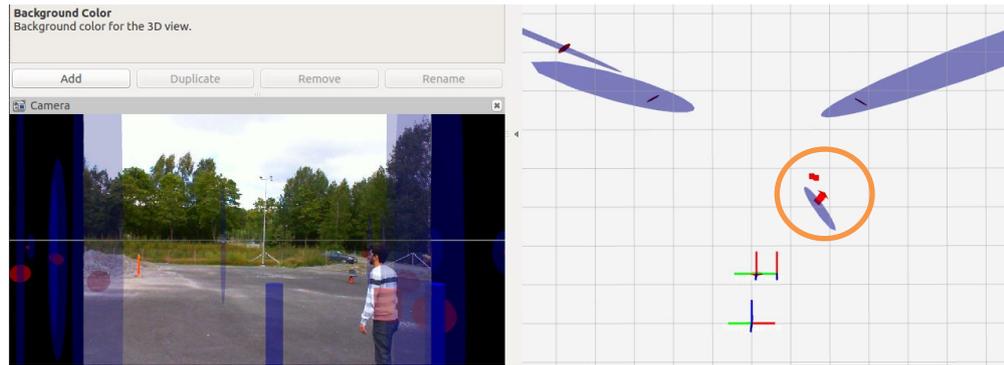


Figure 14: Test 1, Human detection near the sensor

The automotive radar sensor is able to measure the radial speed of the object detect. Figure 15a and 15b shows the human detection by both the sensors. In Figure 15a, the radial velocity is towards the sensor, which shows the person is moving towards the sensor. Similarly, in Figure 15b the radial velocity arrow is directed away from the sensor, which shows the person is moving away from the sensor. the radial velocity only shows the amplitude of the velocity and not the velocity vector direction, we do not have that information, arrows are plotted for information and always along the line passing the object center and the sensor center

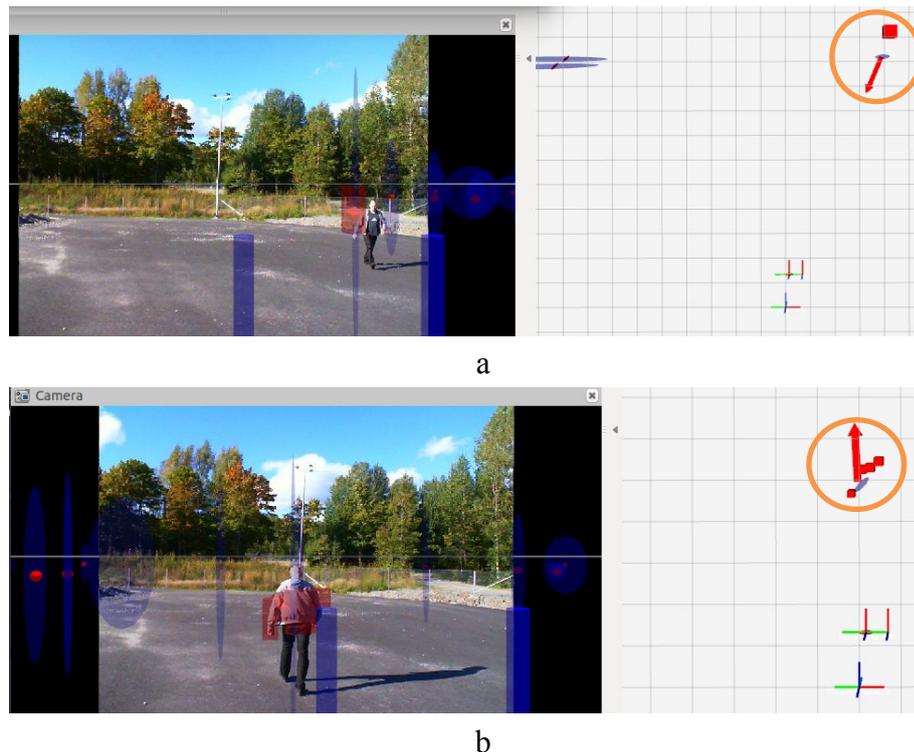


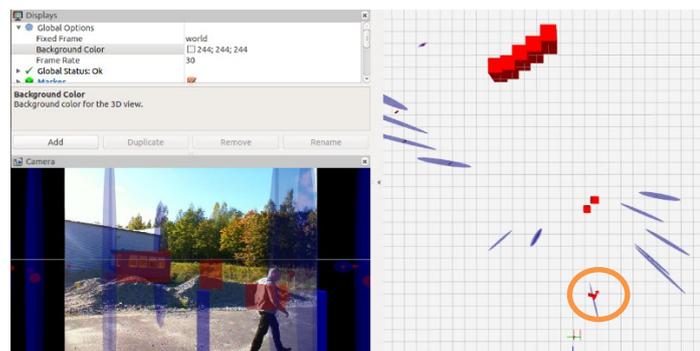
Figure 15: Test 2, Person moving towards and away from the sensor

After experimentation, it was concluded that the maximum distance till which the solid-state lidar sensor detects human is approx. 20 m and for the automotive radar sensor it is approx. 18 m. Figure 16 shows the human detection by both the sensors till 17 m of distance.



Figure 16: Test 3, Finding the Maximum distance till which a person is detected by the sensors.

The next test is to for detecting human among different other objects. The solid-state lidar sensor is able to detect up to 96 objects, depending upon the object's placement in each segment. In Figure 17a, a moving person is detected by both the sensors, the person is near to the sensor and is using 2 segments from the solid-state lidar sensor. In Figure 17b, the person detected by the solid-state lidar sensor is behind the pile of sand. The sensor is able to detect the pile of sand and the container behind the person, including the person itself. Whereas the automotive Radar sensor is not able to detect the human. In Figure 17c the person is standing next to the fence. Both the sensors are able to detect the fence but only the solid-state lidar is able to detect the static human due to the ability of detecting multiple objects per segments. The automotive Radar sensor sometimes mixes the human with the fence.



a

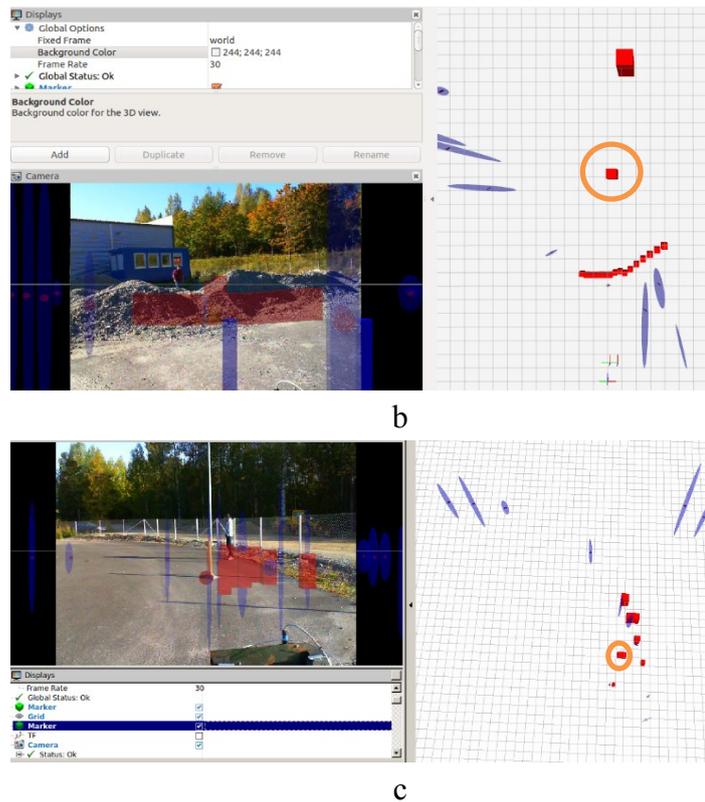


Figure 17: Test 4, Human detection among multiple objects

In Figure 18, GIM mobile machine is being detected by the sensors. The maximum distance till which the sensors are able to detect this machine is 22 m and speed at which the machine is driven for the testing, is 15 km/hr.

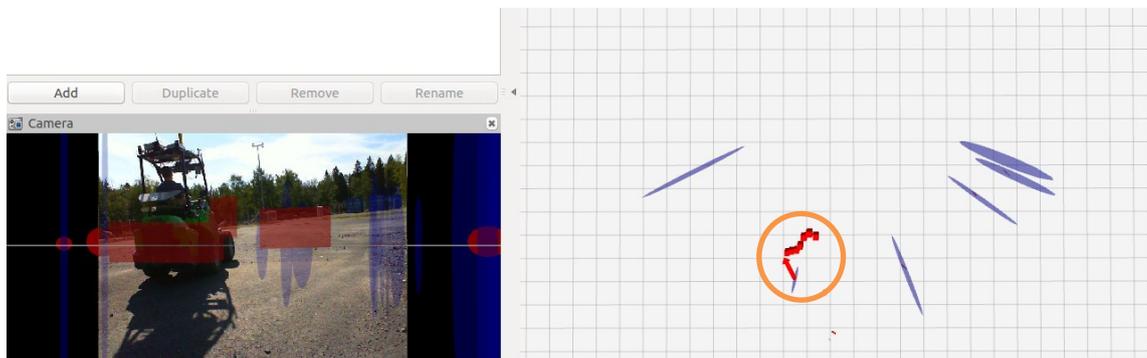


Figure 18: Test 5, Detecting a moving GIM mobile Machine using ADAS sensors

The next test is regarding detection of a moving car in front of the sensors. Figure 19 shows the car been detected by the sensors. The maximum distance till which the car is detected is approx. 22 m and the speed at which the car is driven for the test is 20 km/hr.

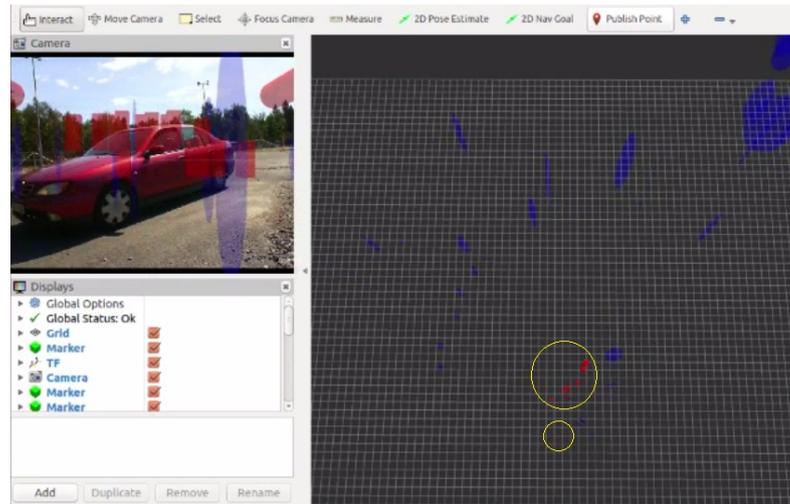


Figure 19: Test 6, Detecting a moving car using ADAS sensors

In Figure 20, a moving bicycle with a rider is detected by the sensors. There is some lagging seen in this Figure. It is due to the data communication delay. Sensors send the data to xPC target and then the data is sent to MATLAB or ROS for solid-state lidar and automotive radar respectively. After that, the MATLAB sends the data to ROS. Therefore, it causes a bit delay and it is not the sensor's problem, because the sensors update the value every 60 ms [17,19].

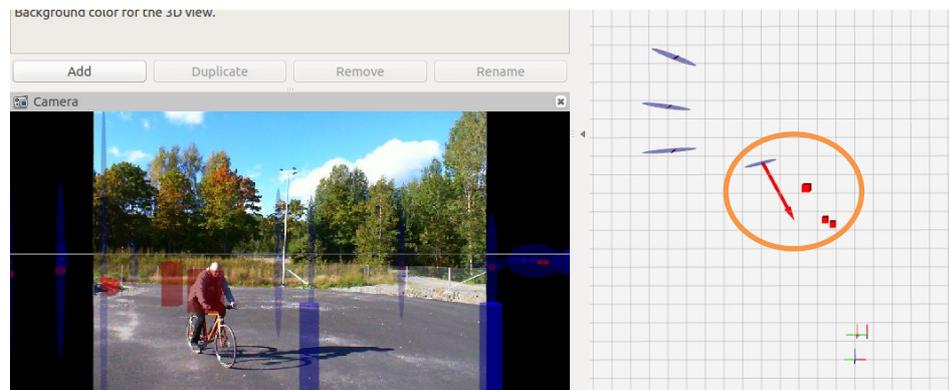


Figure 20: Test 7, Detecting a moving bicycle using ADAS sensors

The test shown in Figure 21 is for the multiple objects. As it can be seen in the figure the pile of sand and the building behind it are detected by the solid-state lidar sensor, whereas for the automotive radar sensor it is difficult to detect the building. In this figure, the building is approx. 20 m away from the sensor.

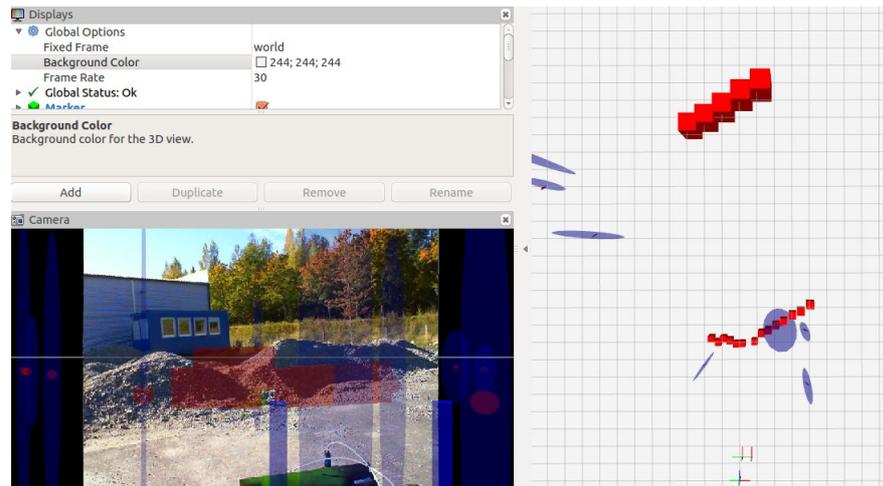


Figure 21: Test 8, Detecting multiple objects using ADAS sensors

The angle at which the sensors are placed also affect the detection. As shown in Figure 22a the boom of the GIM machine is tilted upwards causing sensors not to detect the pole in-front of the GIM mobile machine and it can be seen in Figure 22b, no object is detected. Whereas in Figure 22c when the boom is tilted perpendicular to the pole the sensors can easily detect the pole even in the presence of sunlight behind the pole, and Figure 22d proves it. Therefore, for proper detection it is very important that the sensors are mounted such that the rays can hit the object near perpendicularly.

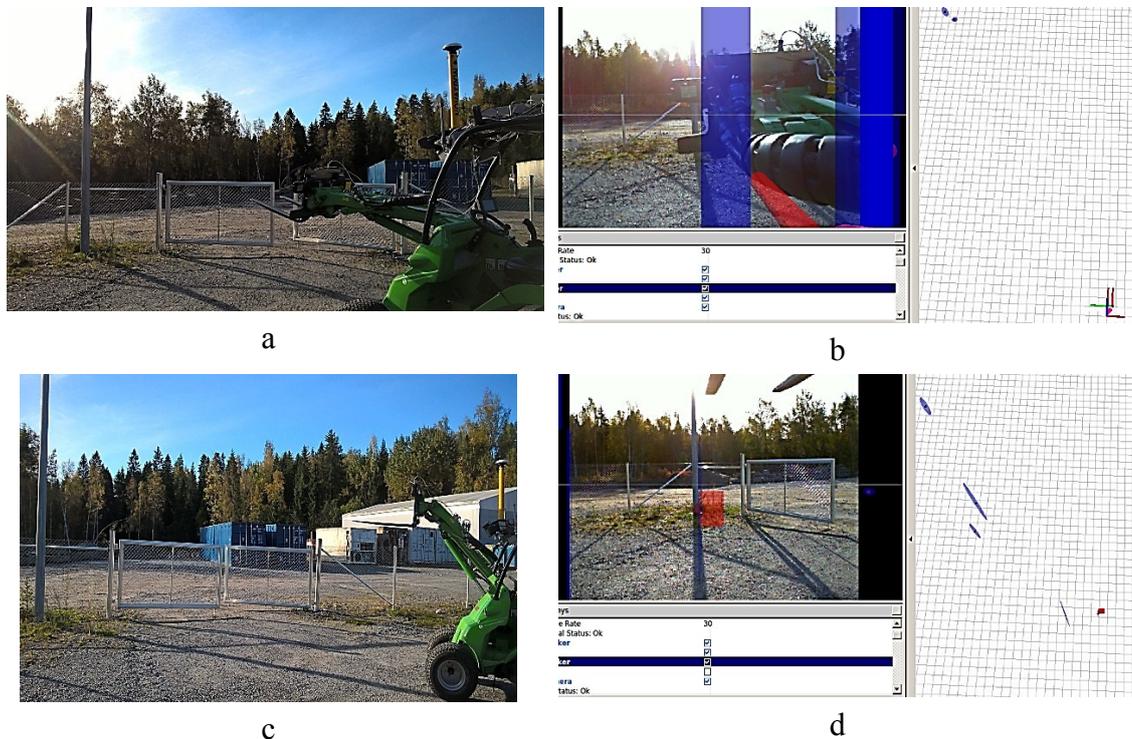


Figure 22: Test 9, Effect of angle of ADAS sensors on detection

3.2 Dynamic Case

For the dynamic case, the sensors mounted on the GIM mobile machine are driven around the TUT Autonomous Test Area (TUT-ATA). So, that the sensors could be tested for harsh environment, and for detecting humans near the moving machine.

The first test is related to the human detection. It was observed that dynamic conditions did not affect the sensor's detection range. The maximum range for detecting a person is still approx. 20 m with solid-state lidar sensor and 18 m for automotive radar sensor. There wasn't any significant change in the performance of the sensors related to human detection, even though the experimentation was performed on different terrain.

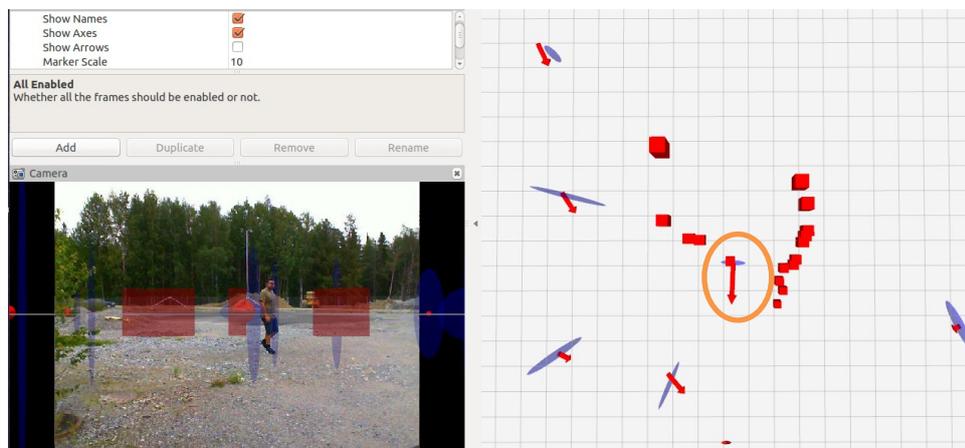
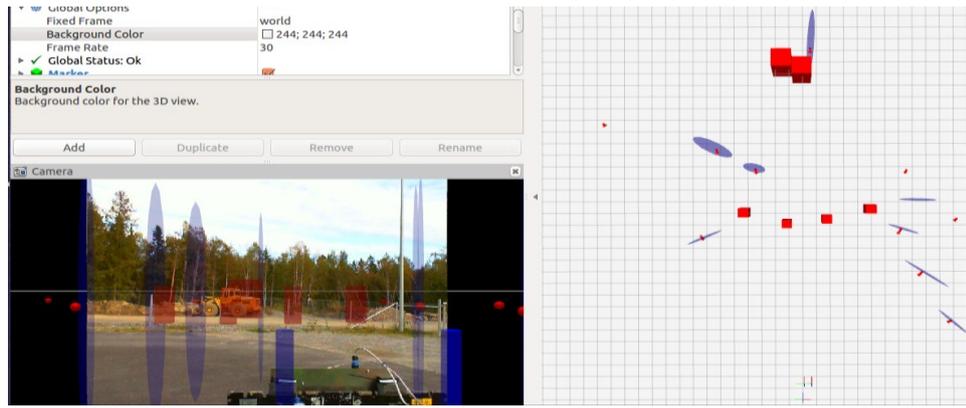
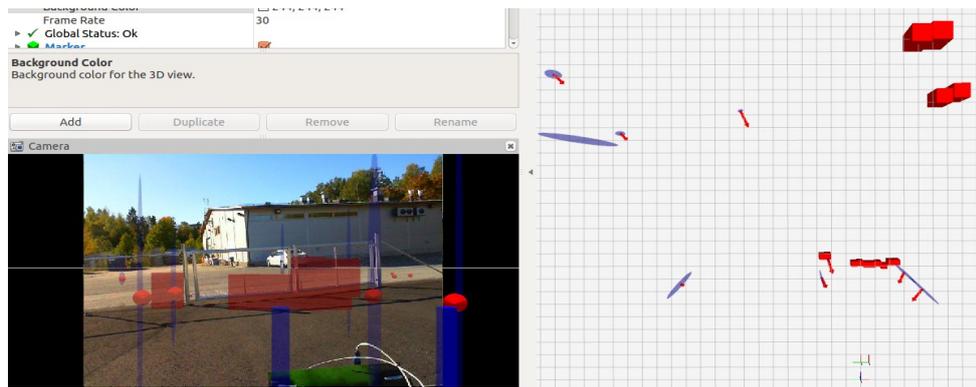


Figure 23: Test 10, Human detection in Dynamic conditions using ADAS sensors

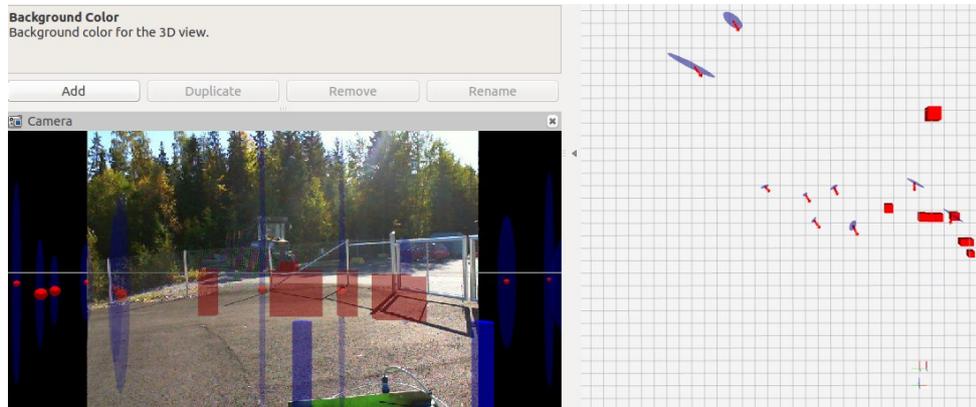
Figure 24 shows the detection of objects such as heavy mobile machines, cars, and buildings behind the fence. In Figure 24a a work machine is detected by both the sensors including the fence between the sensors and the work machine. The speed of the GIM mobile machine on which the sensors are mounted was 10 km/hr. In Figure 24b the building and the car are being detected by solid-state lidar sensor. Whereas the automotive radar sensor is not detecting the car, however, due to much wider field of view of the automotive radar sensor, it is detecting objects which are not visible in the camera. In Figure 24c a forest machine can be seen in the camera beyond the fence. The solid-state lidar is able to detect the machine but the automotive radar is much focused on the fence. However, the automotive radar is detecting some trees behind the fence, as the range of the radar sensor is much greater than that of the solid-state lidar. Figure 24d shows a car parked near the fence. The same as the case before, the solid-state lidar is able to detect the car, but the automotive Radar Sensor is mixing it with the fence as both the objects are very near



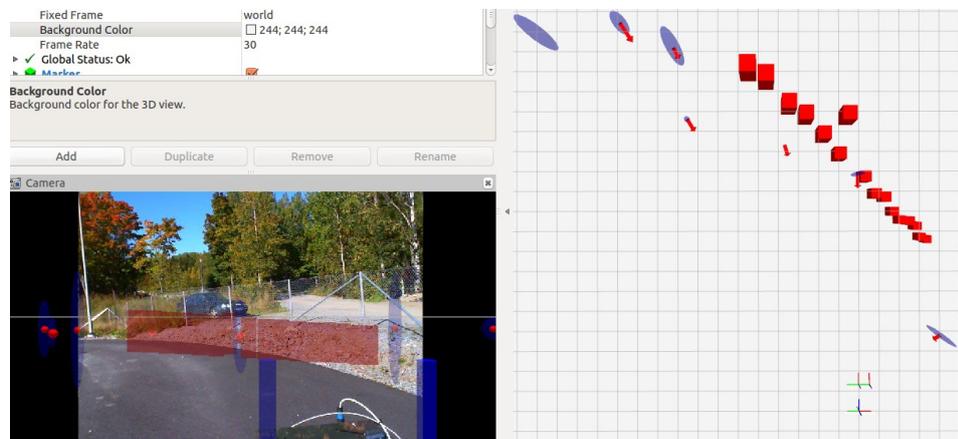
a



b



c



d

Figure 24: Test 11, Detection of different mobile machines, cars and fence via ADAS sensors

Figure 25 shows both the sensors (automotive radar and solid-state lidar sensor) detecting a building to a range of approx. 28 m. The maximum range for the automotive radar sensor to detect buildings is 30 m, and that of solid-state lidar sensor is 28 m.

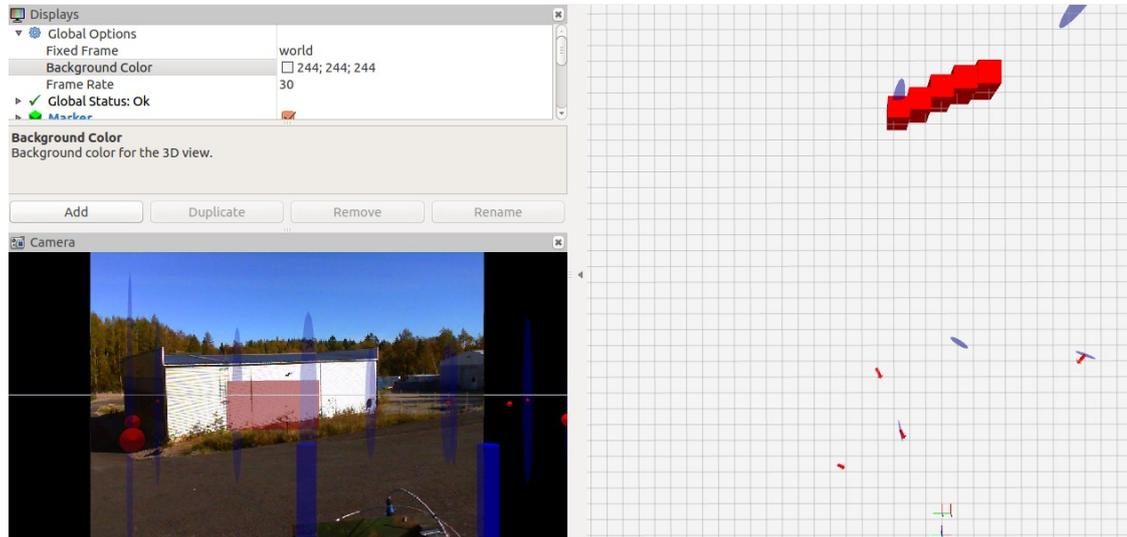


Figure 25: Test 12, Detection of a Building using ADAS sensors

Tests have also been performed in different environmental conditions. In Figure 26, the testing of the sensors is shown in winter season, after the snowfall. The sensors are performing optimally. As discussed before the automotive radar sensor is waterproof and can perform in a significantly wide range of temperature, whereas the solid-state lidar used for this purpose is the evaluation kit and a separate container was created for the sensor to make it waterproof. However, the sensors have not been tested during rain or snow fall.

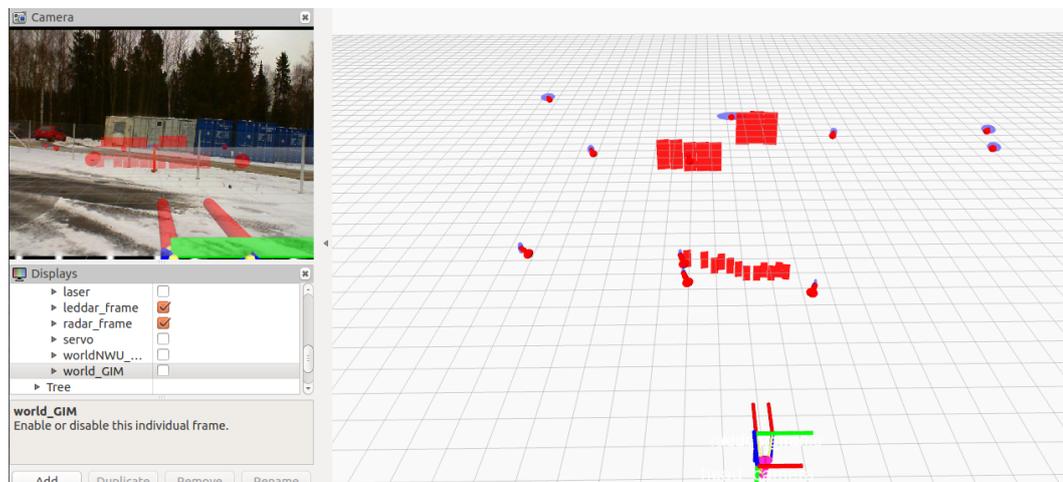


Figure 26: Test 13, Testing the sensors in snowy environment

4. RADAR ODOMETRY KINEMATICS

In the field of mechanics, Kinematics describes the motion of a body, point or particle and systems of bodies without using the masses of the bodies nor the external or internal forces acting on the bodies. Odometry or ego-motion is defined as the use of range /wheel-speed sensors to compute change in the position of a body over time. For the mobile work machine, complete odometry estimation requires computation of three components, linear speed, side slip and angular speed. In this chapter, kinematics equations will be used to model the GIM mobile machine's odometry using the automotive radar sensor. Automotive radar sensor provides information regarding the objects detected, which includes the distance of the object from the sensor, the radial speed of the object relative to the sensor, and the azimuth angle of the object relative to the sensor. Our main goal is to use this information to compute the odometry of the GIM mobile machine. These tests have been performed with GIM mobile machine, however it can be used for any vehicle.

Kinematics equations will be used under the assumption that the environment is static relative to the GIM mobile machine. We will compute 2 components of linear speed and a single component of angular speed relative to the body frame {B} (see Figure 27). GIM mobile machine with a body frame {B}(OXYZ) contains a radar sensor with a radar frame {R}(OXYZ) detecting an object ' p_i ' where $i=1 \dots n$, $n \leq 32$ is the object index. The complete motion of the machine is relative to the world frame {W}(OXYZ). For each object, an end frame is also defined {E}(OXYZ), with the same origin with the radar frame {R}. As shown in Figure 27, the x_{Ei} is directed towards the object p_i . The rigid motion is computed as the location of all the frames can be presented with the position of the origins and the rotation w.r.t world frame. Now for representing the spatial displacement, the simplest way is to decompose the rigid motion into a rotation and translation equation, which is as follow

$${}^W\mathbf{r}_p = {}^W\mathbf{r}_B + {}^W\mathbf{R}_B {}^B\mathbf{r}_R + {}^W\mathbf{R}_B {}^B\mathbf{R}_R {}^R\mathbf{r}_E + {}^W\mathbf{R}_B {}^B\mathbf{R}_R {}^R\mathbf{R}_E {}^E\mathbf{r}_p \quad (1)$$

In equation 1 ${}^W\mathbf{r}_p$ is the vector between the world frame origin and the point p_i with respect to the world frame {W}. Whereas ${}^W\mathbf{r}_B$ is the vector between the origin of world frame and that of the body frame with respect to the world frame {W}, ${}^W\mathbf{R}_B$ is the rotational matrix to transform the body frame to world frame, and ${}^B\mathbf{r}_R$ is the vector between the body frame and radar frame with respect to the body frame, ${}^B\mathbf{R}_R$ is the rotational matrix to transform the radar frame to the body frame, ${}^R\mathbf{r}_E$ is the vector between the Radar frame and the End frame which is zero in this case. ${}^R\mathbf{R}_E$ is the rotational matrix which transforms the radar frame to end frame E. ${}^E\mathbf{r}_p$ is the distance between the point and the end frame, it is the distance that the radar sensor detects.

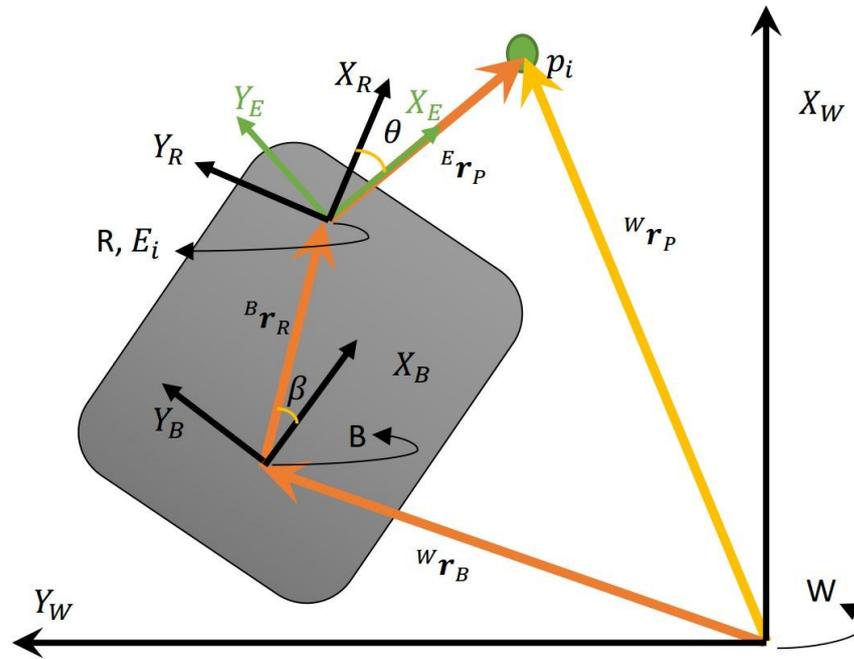


Figure 27: Point p_i represented in radar frame R , body frame R and world frame W

As we have to find the different components of speed for the GIM mobile machine, therefore we will derivating the equation 1, we get

$$\begin{aligned} {}^W \dot{\mathbf{r}}_p = & {}^W \dot{\mathbf{r}}_B + {}^W \dot{\mathbf{R}}_B {}^B \mathbf{r}_R + {}^W \mathbf{R}_B {}^B \dot{\mathbf{r}}_R + {}^W \dot{\mathbf{R}}_B {}^B \mathbf{R}_R {}^R \mathbf{R}_E {}^E \mathbf{r}_p + {}^W \mathbf{R}_B {}^B \dot{\mathbf{R}}_R {}^R \mathbf{R}_E {}^E \mathbf{r}_p + \\ & {}^W \mathbf{R}_B {}^B \mathbf{R}_R {}^R \dot{\mathbf{R}}_E {}^E \mathbf{r}_p + {}^W \mathbf{R}_B {}^B \mathbf{R}_R {}^R \mathbf{R}_E {}^E \dot{\mathbf{r}}_p \end{aligned} \quad (2)$$

Where ${}^W \mathbf{R}_B = \begin{bmatrix} \cos\Psi & -\sin\Psi \\ \sin\Psi & \cos\Psi \end{bmatrix}$, Ψ shows the body heading in world frame

$${}^W \dot{\mathbf{R}}_B = \omega \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\Psi & -\sin\Psi \\ \sin\Psi & \cos\Psi \end{bmatrix},$$

${}^B \mathbf{R}_R = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix}$, β is the mounting angle of the radar sensor in body frame

${}^B \dot{\mathbf{R}}_R = \dot{\beta} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix}$, $\dot{\beta} = 0$, because there is no change in the angle between the body frame and the radar sensor

${}^R \mathbf{R}_E = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}$ θ_i is the azimuth angle at which the i th object is detected, in general case, we will use θ only. Where ' i ' is the i th object detected by the radar sensor.

$${}^R \dot{\mathbf{R}}_E = \dot{\theta} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix},$$

${}^W\dot{\mathbf{r}}_p = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ the computation is under the observation that the world frame and the point are both in static position.

${}^W\dot{\mathbf{r}}_B = {}^W\mathbf{R}_B \begin{bmatrix} v_B \\ v_d \end{bmatrix}$ v_B and v_d are the two components of linear speed measured in body frame

${}^B\mathbf{r}_R = \begin{bmatrix} x_B \\ y_B \end{bmatrix}$, x_B and y_B are the longitudinal and lateral position of the automotive radar sensor in the body frame.

${}^B\dot{\mathbf{r}}_R = 0$ as the distance between the radar frame and base frame doesn't change

${}^E\mathbf{r}_p = \begin{bmatrix} d_i \\ 0 \end{bmatrix}$, where as d_i is, the radial distance measured by the automotive radar sensor for the i^{th} object, as shown in Figure 27, d_i is in the direction of X_{Ei}

${}^E\dot{\mathbf{r}}_p = \begin{bmatrix} -v_r \\ 0 \end{bmatrix}$, where v_r is the velocity of the object measured by the sensor and it is negative because it is measured in the opposite direction.

As discussed before the odometry problem will be studied in two cases. 1) complete radar odometry, with 2-components of linear speed (linear speed and side slip) and angular speed. 2) neglecting side slip and computing only a single component of linear speed and angular speed.

4.1 Complete Odometry including Side-slip

The main drawback of wheel odometry is: not detecting side slip or wheel slippage, because wheel odometry sensors only convert the rotational speed of the wheels to linear speed. On the other hand, radar odometry can be used to compute the side slip including linear speed and angular speed. Using equation 2 and putting in the values, we get

$$\begin{aligned} \begin{bmatrix} 0 \\ 0 \end{bmatrix} &= \begin{bmatrix} \cos\Psi & -\sin\Psi \\ \sin\Psi & \cos\Psi \end{bmatrix} \begin{bmatrix} v_B \\ v_d \end{bmatrix} + \omega_B \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\Psi & -\sin\Psi \\ \sin\Psi & \cos\Psi \end{bmatrix} \begin{bmatrix} x_B \\ y_B \end{bmatrix} + \\ \omega_B \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} d_i \\ 0 \end{bmatrix} + \dot{\theta} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\Psi & -\sin\Psi \\ \sin\Psi & \cos\Psi \end{bmatrix} \\ \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} d_i \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\Psi & -\sin\Psi \\ \sin\Psi & \cos\Psi \end{bmatrix} \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} -v_r \\ 0 \end{bmatrix} \end{aligned} \quad (3)$$

multiplying both sides of the equations with ${}^W\mathbf{R}_B^{-1}$ to remove Ψ from the equation

$$\begin{aligned} \begin{bmatrix} 0 \\ 0 \end{bmatrix} &= \begin{bmatrix} v_B \\ v_d \end{bmatrix} + \omega_B \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_B \\ y_B \end{bmatrix} + \omega_B \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} d_i \\ 0 \end{bmatrix} + \\ \dot{\theta} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} d_i \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} -v_r \\ 0 \end{bmatrix} \end{aligned} \quad (4)$$

Simplifying the equation 4 further we get

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v_B \\ v_d \end{bmatrix} + \omega_B \begin{bmatrix} -y_B \\ x_B \end{bmatrix} + \omega_B \begin{bmatrix} -d_i \sin(\theta + \beta) \\ d_i \cos(\theta + \beta) \end{bmatrix} + \dot{\theta} \begin{bmatrix} -d_i \sin(\theta + \beta) \\ d_i \cos(\theta + \beta) \end{bmatrix} - \begin{bmatrix} v_r \cos(\theta + \beta) \\ v_r \sin(\theta + \beta) \end{bmatrix} \quad (5)$$

Now we have three unknowns v_B , v_d , and ω_B

$$v_B - \omega_B y_B - \omega_B d_i \sin(\varphi) - \dot{\theta} d_i \sin(\varphi) - v_r \cos(\varphi) = 0 \quad (6)$$

$$v_d + \omega_B x_B + \omega_B d_i \cos(\varphi) + \dot{\theta} d_i \cos(\varphi) - v_r \sin(\varphi) = 0 \quad (7)$$

$\varphi = (\theta + \beta)$. Using equation 7 and solving it for $\dot{\theta}$, because $\dot{\theta}$ is not required

$$\dot{\theta} = -\frac{v_d}{d_i \cos(\varphi)} - \frac{\omega_B x_B}{d_i \cos(\varphi)} + \frac{v_r \sin(\varphi)}{d_i \cos(\varphi)} - \frac{\omega_B d_i \cos(\varphi)}{d_i \cos(\varphi)} \quad (8)$$

$$\dot{\theta} = -\frac{v_d}{d_i \cos(\varphi)} - \frac{\omega_B x_B}{d_i \cos(\varphi)} + \frac{v_r \tan(\varphi)}{d_i} - \omega_B \quad (9)$$

Now putting the values of $\dot{\theta}$ in equation 6

$$v_B - \omega_B y_B - \omega_B d_i \sin(\varphi) - \left(-\frac{v_d}{d_i \cos(\varphi)} - \frac{\omega_B x_B}{d_i \cos(\varphi)} + \frac{v_r \tan(\varphi)}{d_i} - \omega_B \right) d_i \sin(\varphi) - v_r \cos(\varphi) = 0 \quad (10)$$

Simplifying it further we get

$$v_B \cos(\varphi) + v_d \sin(\varphi) + \omega_B (x_B \sin(\varphi) - y_B \cos(\varphi)) - v_r = 0 \quad (11)$$

In general, after adding the indices we can write equation 11 as

$$(v_B - \omega_B y_B) \cos(\varphi_i) + (v_d + \omega_B x_B) \sin(\varphi_i) - v_{ri} = 0 \quad (12)$$

In equation (12) v_B , v_d and ω_B are the unknown components of linear and angular speed respectively. While using a single Radar with 3 detected points, the involved jacobian takes the following form

$$J = \begin{bmatrix} \cos(\varphi_1) & \sin(\varphi_1) & -y_B \cos(\varphi_1) + x_B \sin(\varphi_1) \\ \cos(\varphi_2) & \sin(\varphi_2) & -y_B \cos(\varphi_2) + x_B \sin(\varphi_2) \\ \cos(\varphi_3) & \sin(\varphi_3) & -y_B \cos(\varphi_3) + x_B \sin(\varphi_3) \end{bmatrix} \quad (13)$$

Equation (13) shows the jacobian for one radar and three detections. This matrix is rank deficient because the last column is the linear combination of first two, which shows that, the three components of odometry cannot be computed using a single radar sensor as concluded in [35]. Research thus suggests use of multiple radar sensors to overcome this problem or neglect the side slip and use a single radar sensor to estimate the odometry [32]. However, our approach suggests use of a vertical gyroscope to overcome this

singularity problem of the matrix, because MEMS gyroscopes are much cheaper than automotive radar sensors [24]

4.2 Odometry Excluding Side-slip

For the case where side slip is insignificant, we can neglect the linear speed component of side slip $v_d = 0$ and equation 12 can be written as

$$v_B \cos(\varphi_i) + \omega_B (x_B \sin(\varphi_i) - y_B \cos(\varphi_i)) - v_{ri} = 0 \quad (14)$$

Now we have only two unknowns in the equation therefore we do not need to use a gyroscope for angular velocity and both the linear speed and angular speed can be easily computed by a single automotive radar sensor. At least 2 radar object detections are needed for computing linear and angular velocity but to increase the performance more than 2-detections are used.

5. RADAR ODOMETRY EXPERIMENTATION

The automotive radar sensor is used for odometry estimation. Equation 12 is used for complete odometry estimation which includes 2 components of linear speed and a single component of angular speed, the angular speed is provided by a vertical gyro sensor mounted on the GIM mobile machine. For the second case equation 14 is used, where the side slip is neglected and the odometry can be estimated using a single automotive radar sensor.

The automotive radar sensor contains few built-in filters which can be used for the removing of invalid detection. As presented in table 3, the first is the probability filter, it provides the probability of the object detected, in our case we are using 20% probability. The second filter is the alive counter, the sensor counts the number of time a single object is detected, we are using value of 2 for our case. The third filter is the validity filter, the sensor is able to predict the validity of the object and if the value of this filter is set to 1 it means that it will only send the detection which is valid. Filters values are included in the ROS launch file as shown in Program 6, so that there is no need to change the source code again and again to check output with different filter settings.

To estimate the radar odometry the GIM mobile machine was driven for 3-types of experimentation. 1) the GIM mobile machine was driven straight and then backwards. 2) the GIM mobile machine was driven in a circular path with high speed, as these tests were performed in winter on frozen terrain, therefore the slippage cannot be neglected. 3) the GIM mobile machine is driven in a circular path with low speed so that the slippage can be reduced.

The visualization of the automotive radar sensor is done in ROS, and to compute the radar odometry rosbag [43] files were utilized. GIM mobile machine contains sensors which include GNSS, gyroscopes, and Wheel speed sensors (Hall sensors) for navigation and odometry purposes. For experimentation, the data that has been recorded contains radar sensor data, Wheel speed sensors data, GNSS data, and gyroscope data. The bag files are further processed in MATLAB.

After reading the files in MATLAB, the next step is to use the data and apply equation 12 for complete odometry estimation, where the single radar sensor is aided by a vertical gyro, and equation 14 for odometry estimation without the side slip. In either case, we need to calculate two unknown variables. In the ideal case, two object points are enough for the estimation and any two points will lead to the same output. However, in reality due to noise and presence of moving objects, the results do not agree. After experimentation, it was concluded that using max 10 objects to estimate the unknowns yields better results. However, to remove moving objects and noise from the data

RANSAC [44] is applied. RANSAC is used to fit a model for experimental data. As discussed before, our assumption for radar odometry is that the objects in the environment are in stationary form. Therefore, any moving object will cause error in the computation. The radar sensor measures the radial velocity of the object detected, and a moving object used for calculation will always have a different linear and angular speed, and will be considered as an outlier and it will be removed by RANSAC. After applying RANSAC, mean is applied for both the unknown values of a single measurement cycle e.g. in a single measurement cycle 10 objects are detected, based on these object nine different values of linear and angular speed are computed, using equation 14, after applying RANSAC, mean is applied on these 9 values to obtain the linear and angular speed of the machine.

5.1 Odometry Estimation excluding side slip

As discussed earlier, two tests are conducted for odometry estimation where side slip can be neglected. 1) GIM mobile machine driven straight and then backwards and 2) GIM mobile machine driven in a circular path first clock wise and then anti clockwise with a slow speed. The results of the radar odometry are compared with the Kellner [33] implementation, and current wheel odometry of the GIM mobile machine. Kellner's approach is presented in Appendix B (pg. 64).

5.1.1 GIM mobile machine Driven Straight

Figure 28a shows the GIM mobile machine linear speed. The machine is driven straight till time step 170 and then driven backwards. The radar odometry graph is showing some noise but the trend is the same as that of the wheel odometry. Except few peaks the radar odometry results are quite accurate. Figure 28b shows the comparison of the angular speed of GIM mobile machine moving straight. The radar odometry is showing random peaks, it can be the reason that radar at some points does not have enough data to calculate the unknowns. Figure 28 also shows that the gyroscope provides excellent angular speed information.

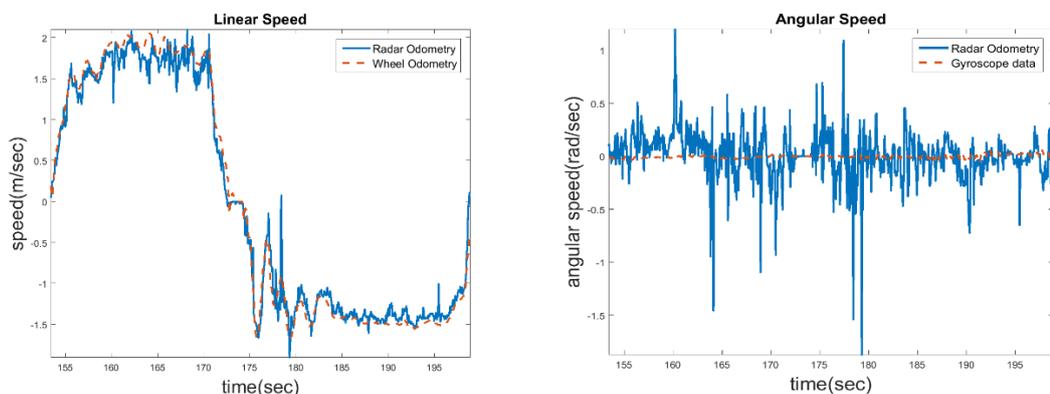


Figure 28: Comparison of Linear v_B and angular ω_B speed of Radar odometry with Wheel odometry and Gyroscope respectively

The comparison of Linear and angular speed of Radar odometry with Kellner [33] implementation is shown in Figure 29. Kellner [33] uses Ackermann conditions for computing linear and angular speed. As shown in Figure 29a the radar odometry computed in this research is much better than Kellner's method [33], because the GIM mobile machine is a non-Ackermann condition platform.

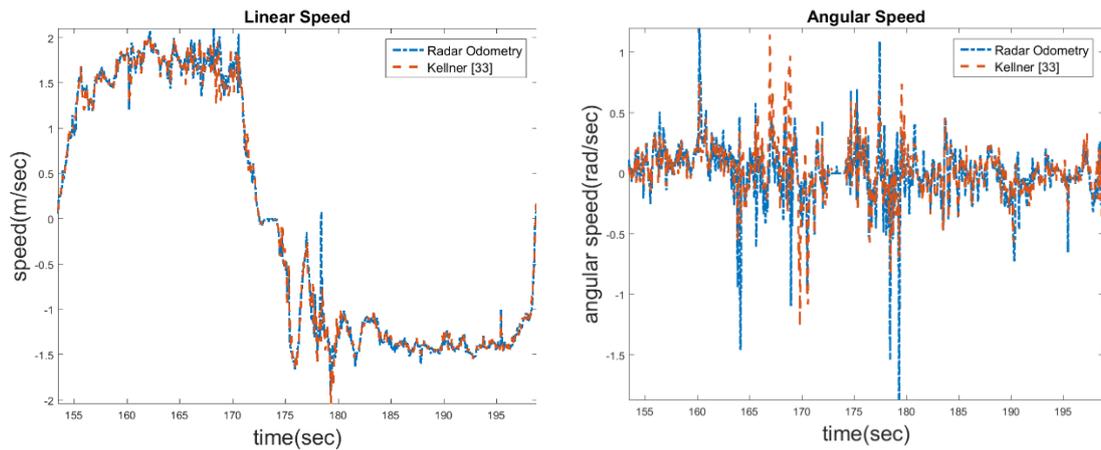


Figure 29: Comparison of Linear v_B and angular ω_B speed of Radar odometry with Kellner's Work [33]

5.1.2 GIM mobile machine Driven in circular path

The second test for which the side slip can be neglected, is the circular path test where the GIM mobile machine was driven a bit slow, so that the effect of the side slip could be minimum for the wheel odometry. The comparison of radar odometry with the wheel odometry is shown in Figure 30. The maximum linear speed detected by both the sensors is approx. 1.7 m/sec, which is slower than the maximum speed for the first test (Figure 28). As discussed before the GIM machine was first driven clockwise and then at time step of 704 approx. the GIM machine started rotating anticlockwise. It can be observed that the radar odometry contains noise, as it was with the first case. But in this case the noise in the angular speed is rather less than before. However, the linear speed contains almost the same amount of noise as before.

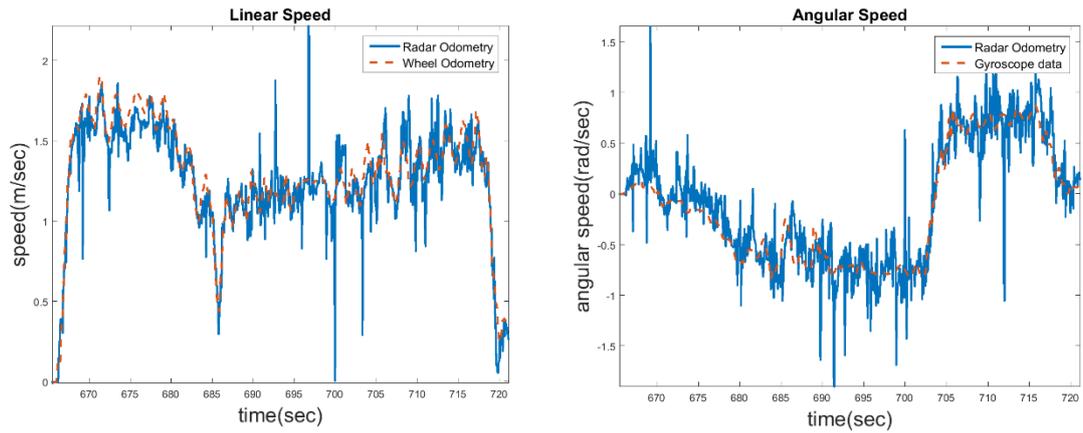


Figure 30: Comparison of Radar odometry with wheel odometry and gyroscope, for the GIM machine moving in a circular path

Coming towards the comparison of Radar odometry and Kellner's work [33], it can be observed from Figure 31 that both the algorithms exhibits almost same results. Noise can be detected in both the cases.

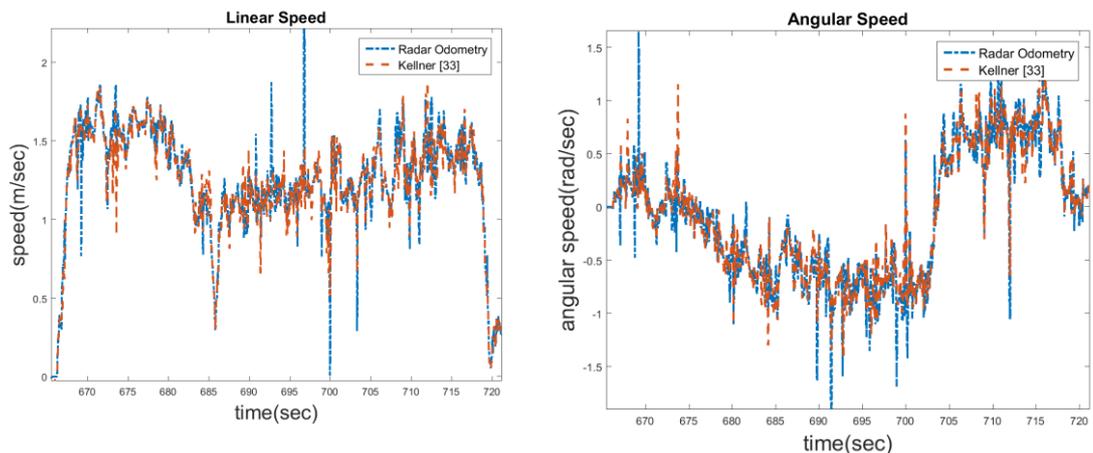


Figure 31: Comparison of Radar odometry with Kellner's Work [33], for the GIM machine moving in a circular path

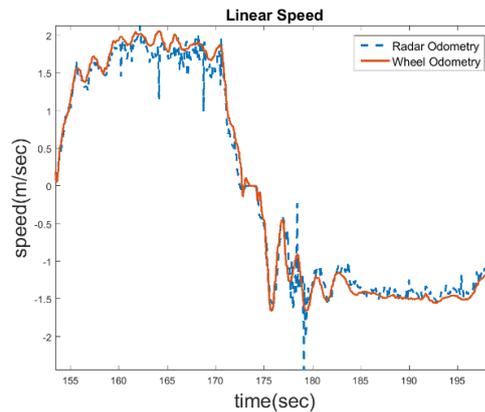
5.2 Complete Odometry Estimation Including side slip

Complete odometry estimation of a moving vehicle is not possible using a single radar sensor [35]. Therefore, a vertical gyroscope is fused with the Automotive radar sensor for complete estimation of the machine, which includes 2 components of linear speed, and 1 single component of angular speed. The performance evaluation of the radar odometry is also done in this section. The results are compared with the complete navigation solution, formed with the fusion of IMUs, wheel odometry and GNSS.

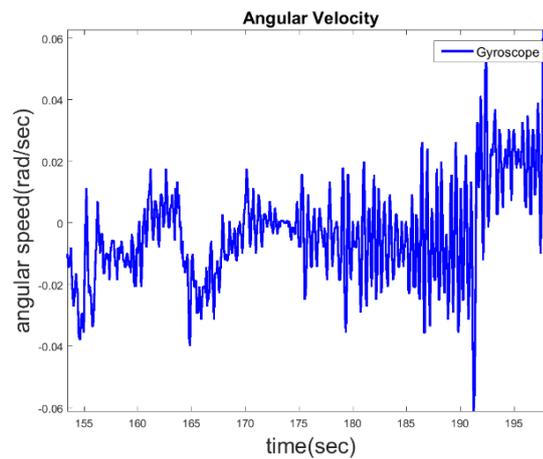
For complete odometry estimation, two tests will be examined. First will be the machine driven in a straight path and then backwards. Second is the circular path test with much higher speed, so that the side slip can be calculated.

5.2.1 GIM mobile machine Driven Straight

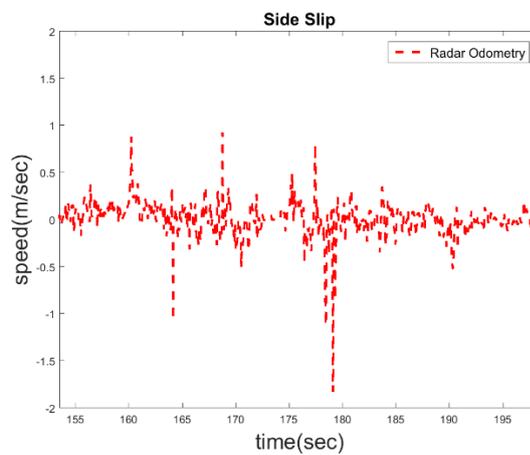
This is same test which was previously used for odometry estimation without side slip. In Figure 32a the linear speed graph is shown. If the results are compared with the Figure 28, it can be observed that the noise is significantly reduced. It is due to the use of gyroscope angular speed which contains very less noise as shown in Figure 32b. The side slip can also be detected in Figure 32c. However, the amplitude of side slip is not much.



a



b



c

Figure 32: a) Linear speed v_B of the GIM machine compared with wheel odometry, b) angular speed ω_B of the GIM mobile machine measured with gyroscope, c) Side Slip v_d of the GIM machine measured with radar sensor

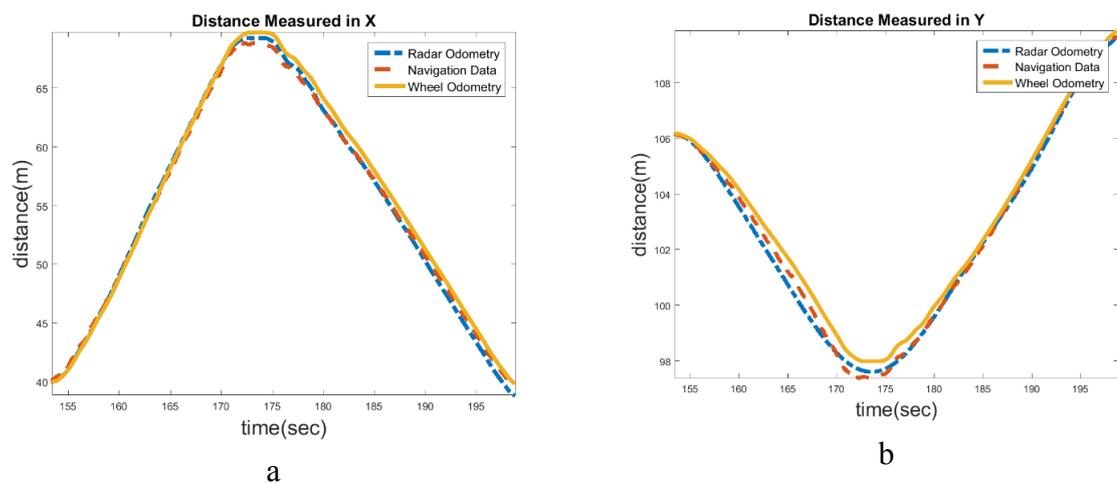
To compute the performance of the radar odometry, the results have been compared with the complete navigation solution, which is a fusion of Wheel odometry, IMU and GNSS. Dead reckoning trajectories are integration of kinematic equation [15,16], which are as follow.

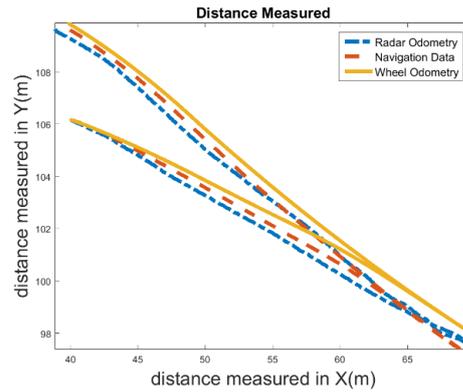
$${}^w\dot{\mathbf{r}}_B = {}^w\mathbf{R}_B \begin{bmatrix} v_B \\ v_d \end{bmatrix} \quad (15)$$

$$\dot{\Psi} = \omega_B \quad (16)$$

where ${}^w\mathbf{R}_B = \begin{bmatrix} \cos\Psi & -\sin\Psi \\ \sin\Psi & \cos\Psi \end{bmatrix}$ and Ψ , is the heading angle of the GIM mobile machine.

The initial values are obtained using the navigation solutions. Figure 33 shows the comparison of the results, of radar odometry with wheel odometry and navigation data. Figure 33a shows the comparison of result for the distance measured in X direction, Figure 33b shows the comparison of results for the distance measured in Y direction and Figure 33c shows the comparison of results of X and Y. Table 4 shows the mean errors of radar odometry and wheel odometry, for X direction and Y direction, relative to the complete Navigation system. It can be observed that the radar odometry yields better results with lesser RMS, max and min error. Other than that, this test doesn't involve any non-systematic error which could have been used to conclude better performance of radar odometry. Figure 34 a and b, shows the complete RMS error of radar odometry and wheel odometry over time for X and Y direction relative to the complete navigation system. Odometry estimated with any type of sensor always accumulates error with time therefore GNSS and IMU sensors are fused with these sensors to counter the error.



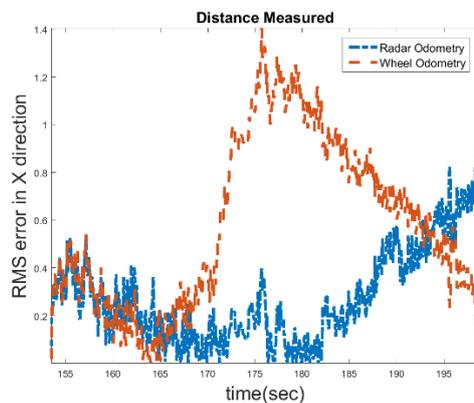


c

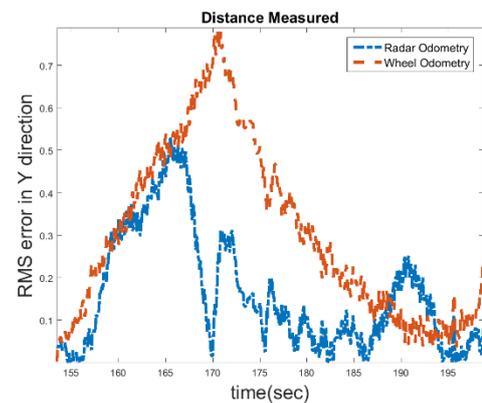
Figure 33: Comparison of Radar odometry with navigation data and wheel odometry for X and Y direction.

Table 4: Error estimation of Radar Odometry and Wheel Odometry relative to complete navigation system for Straight path test

Straight Path Test	RMS error in X direction	RMS error in Y direction
Radar Odometry RMS error	0.3381	0.2128
Wheel Odometry RMS error	0.6851	0.3674
Radar Odometry Max error	0.9966	0.5297
Wheel Odometry Max error	1.4039	0.7882
Radar Odometry Min error	0.0004	0
Wheel Odometry Min error	0.0016	0.0030



a



b

Figure 34: RMS Error estimation of Radar odometry and Wheel odometry relative to Complete Navigation system for straight line test a) in X direction, b) in Y direction

5.2.2 GIM mobile machine Driven in circular path

In this test, the machine is driven in a circular path with high speed which is approx. 2 m/sec as shown in Figure 35a. As discussed before, by using the gyroscope for the angular speed, the noise in the linear speed has also decreased. Figure 35b shows the

angular speed measured with a vertical gyroscope mounted on the GIM mobile machine. The terrain on which the GIM mobile machine was driven for this test was frozen, therefore the side slip should be more than regular and it can easily be observed in Figure 35c.

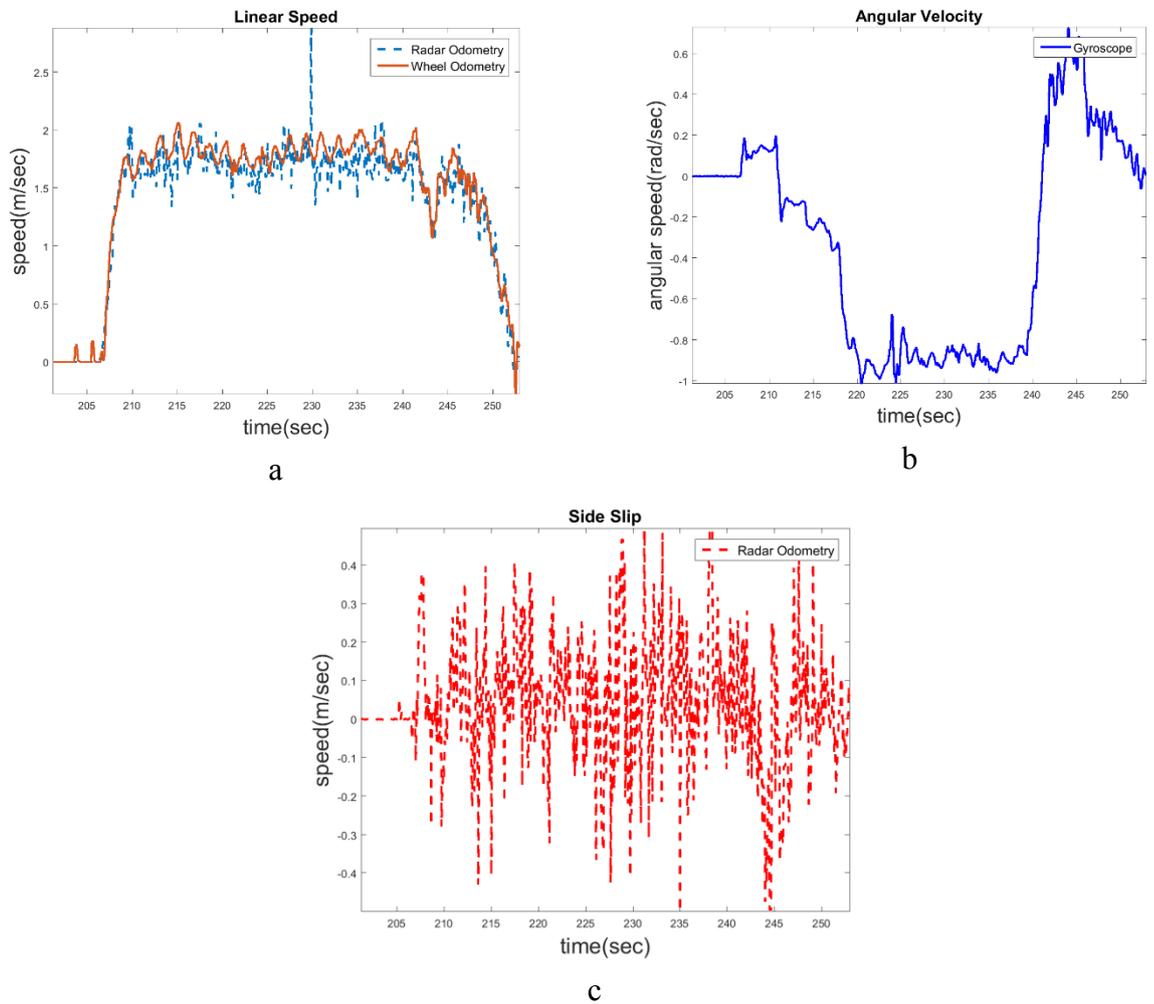


Figure 35: a) Linear speed \mathbf{v}_B of the GIM machine compared with wheel odometry for circular path, b) angular speed $\boldsymbol{\omega}_B$ of the GIM mobile machine measured with gyroscope for circular path, c) Side Slip \mathbf{v}_a of the GIM machine measured with radar sensor for circular path

As done before the performance evaluation of the radar odometry for the GIM machine moving in a circular path is shown in Figure 36. The radar odometry measured in X and Y directions are compared with the navigation solution and the Wheel odometry. It can be observed that the radar odometry graphs follows the same trend as that of the navigation solution, although by the end there is difference between the two graphs but still the performance of the radar odometry is much better than the wheel odometry. As shown in the Figure 36c, where the graph is plotted for X and Y directions. Error comparison of radar odometry and wheel odometry for X and Y direction relative to complete navigation system is shown in Table 5. Same as before the error for radar odometry is much less than wheel odometry. Figure 37 shows the RMS error for X and

Y direction over time relative to the complete navigation system. It can be observed that there is no significant difference between the two odometry's.

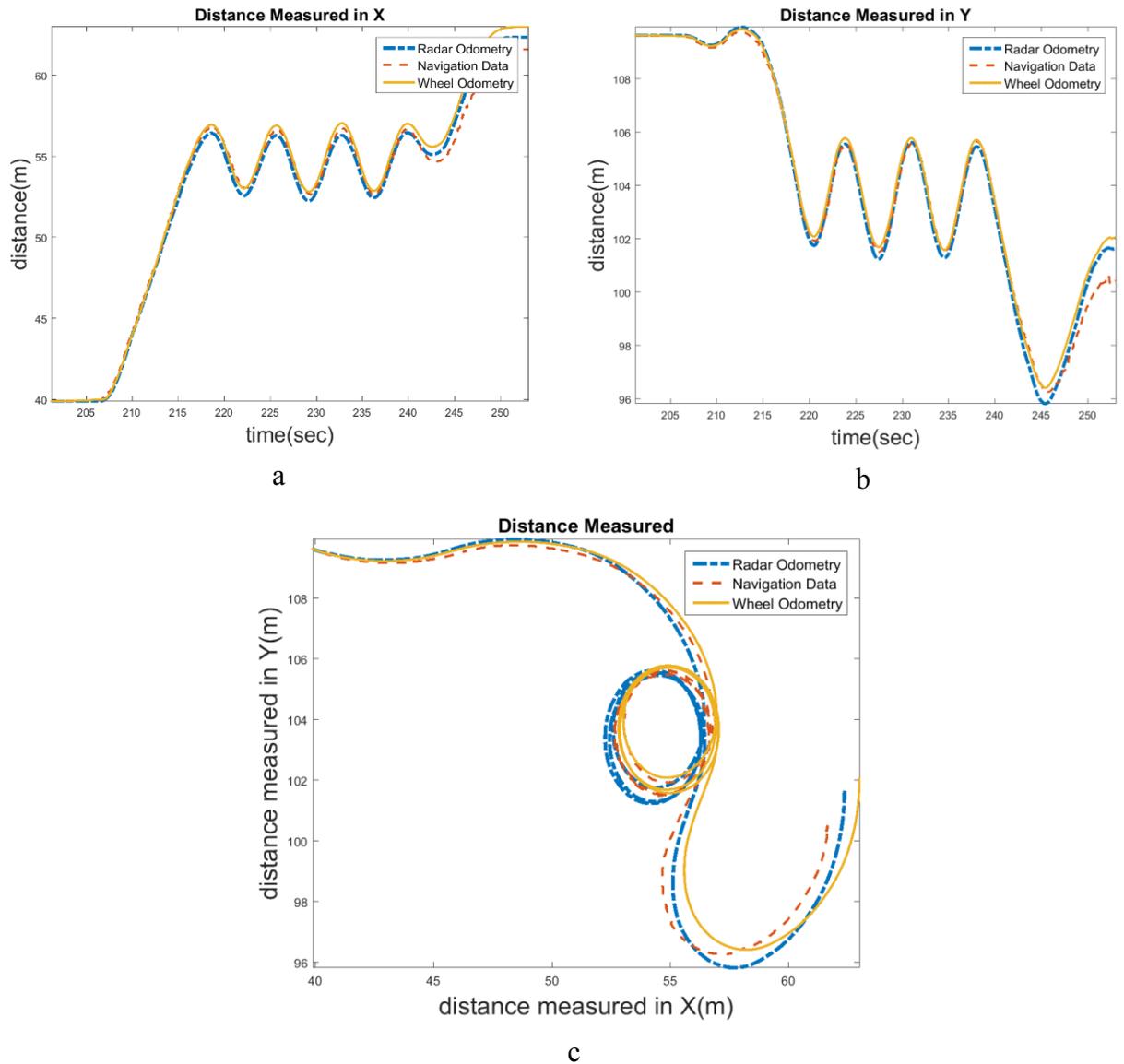


Figure 36: Comparison of Radar odometry with navigation data and wheel odometry for X and Y direction for the GIM mobile machine moving in a circular path

Table 5: Error estimation of Radar Odometry and Wheel Odometry relative to complete navigation system for Circular path test

For Circular path	RMS error in X direction	RMS error in Y direction
Radar Odometry RMS error	0.4861	0.3601
Wheel Odometry RMS error	0.6765	0.4379
Radar Odometry Max error	1.2087	1.3098
Wheel Odometry Max error	1.7954	1.7145
Radar Odometry Min error	0.0004	0.0001
Wheel Odometry Min error	0.0009	0.0006

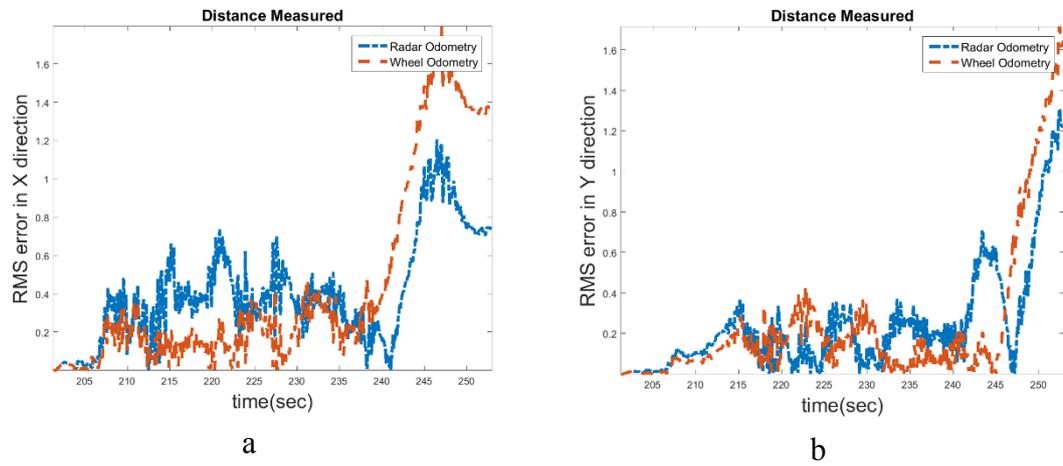


Figure 37: RMS Error estimation of Radar odometry and Wheel odometry relative to Complete Navigation system for Circular path test a) in X direction, b) in Y direction

6. CONCLUSION

6.1 *Human detection using ADAS sensors*

With the experimentation done in chapter 3, following results have been concluded.

1. Solid-state lidar sensor is tested to detect humans to a distance of approx. 20 m and the maximum distance for automotive radar sensor is 18 m
2. The sensors are currently used for detection and not for the recognition of the objects
3. Automotive Radar sensor and solid-state lidar sensor are tested to detect cars and mobile machines moving with a speed of approx. 20 km/hr. However, the sensors can detect much faster moving cars and mobile machines.
4. Automotive Radar sensor can detect 32 objects in a single measurement cycle, as the tests are performed in a close fence, therefore, the sensor is detecting the fence in addition to object which is to be detected.
5. Objects/Humans moving in different directions in front of the sensors do not cause any problem in detection.
6. Automotive Radar sensor gives a noisy data when used indoors.
7. Solid-state lidar sensor can detect maximum 96 objects, each segment is capable of detecting 6 objects.
8. Flag Information of Solid-state lidar sensor can be utilized to check the validity of the detection.
9. Leddar sensor sends the information of total number of detections every 50 ms
10. Self-provided filters of the automotive radar sensor can be used to refine the data of the sensor, the values of the parameters can be changed depending on the application it is used for
11. Radar sensor provides the radial velocity of the objects. For the purpose of visualization, we chose to point the speed vectors towards the vehicle, although actual direction of the speed vectors is unknown immediately from sensor output.
12. Figure 38a (pg.51) and 38b (pg.51) shows the field of view of Solid-state Lidar sensor and automotive Radar sensor respectively for Human Detection.
13. Table 6 (pg.51), shows the maximum distance for detection of objects for both the sensors.
14. Table 7 (pg.51), shows the maximum objects detected by the sensors.
15. Table 8 (pg.51), shows the different testing conditions for the sensors.

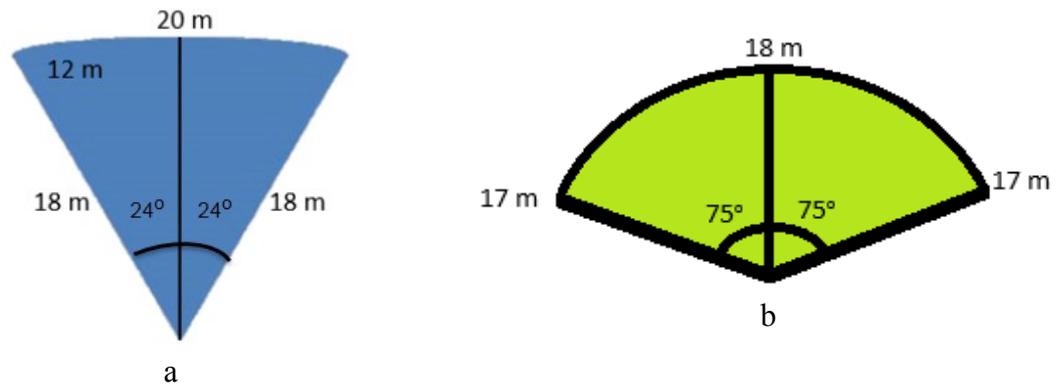


Figure 38: a) Field of view of Solid-state Lidar sensor for Human Detection, b) Field of view of Automotive Radar sensor for Human detection

Table 6: Maximum distance for detection of objects for Solid state Lidar and Automotive Radar sensor

	SOLID-STATE LIDAR	RADAR
Car	22 m	22 m
Human	20 m	18 m
GIM mobile Machine	22 m	21 m
Fence	20 m	24 m
Building	28 m	30 m
Bicycle	20 m	20 m

Table 7: Maximum objects detected by the sensors

	MAXIMUM DETECTIONS BY SENSORS
SOLID-STATE LIDAR (SEGMENTS)	96
AUTOMOTIVE RADAR (OBJECTS)	32

Table 8: Different testing conditions for both the sensors

	CONDITIONS	SOLID-STATE LIDAR	RADAR
OUTDOOR	Sensor stationary, object stationary	Stable Results	Stable Results
	Sensor Stationary, Object moving (20km/hr. max.)	Stable Results	Stable Results
	Moving Sensor, Stationary Object	Stable Results	Stable Results
	Moving Sensor, Moving Objects	Stable Results	Stable Results
INDOOR	Stationary Sensor, Stationary Objects	Stable Results	Unstable Results
	Stationary Sensor, Moving Objects	Stable Results	Unstable Results

6.2 Radar Odometry for Mobile Work Machine

Research suggests use of multiple automotive radar sensors for estimating a complete odometry of a vehicle or mobile machine. However, the method proposed in this research suggests use of a single automotive radar with a vertical gyroscope, to compute the complete odometry of the vehicle, i.e. to compute 2-components of the linear speed and a single component of the angular speed. It can be concluded that any radar (with no moving parts) or range sensor which provides the information of radial velocity and the azimuth angle of the object detected can be used to estimate the odometry of a vehicle/mobile machine. This approach is not limited to Ackermann platforms, it can also be applied to non-Ackermann platforms as well. Furthermore, the proposed algorithm computes the odometry in a single measurement cycle and does not depend on the previous values. The performance of the algorithm for dead reckoning trajectories is computed with reference of the complete navigation system. The next step is to integrate the proposed solution to complete navigation system of the GIM mobile machine and for Simultaneous Localization and Mapping (SLAM)

REFERENCES

- [1] Postolache O, Girão PS, Madeira RN, Postolache G. Microwave FMCW Doppler radar implementation for in-house pervasive health care system. In Medical Measurements and Applications Proceedings (MeMeA), 2010 IEEE International Workshop on 2010 Apr 30 (pp. 47-52). IEEE.
- [2] Pienaar M, Odendaal JW, Joubert J, Cilliers JE, Smit JC. Active calibration target for bistatic radar cross-section measurements. *Radio Science*. 2016 May 1;51(5):515-23.
- [3] Klugmann D, Wang H. Solid-state 94 GHz Doppler radar for meteorology.
- [4] Fleming B. Recent advancement in automotive radar systems [Automotive Electronics]. *IEEE Vehicular Technology Magazine*. 2012 Mar;7(1):4-9.
- [5] Xu J, Liao G, Zhang Y, Ji H, Huang L. An Adaptive Range-Angle-Doppler Processing Approach for FDA-MIMO Radar Using Three-Dimensional Localization. *IEEE Journal of Selected Topics in Signal Processing*. 2016 Oct 5.
- [6] Vivekanandan J, Loew E, Tsai PS, Lee WC, Moore J. Airborne polarimetric Doppler weather radar: Trade-offs between various engineering specifications. In *Phased Array Systems and Technology (PAST)*, 2016 IEEE International Symposium on 2016 Oct 18 (pp. 1-6). IEEE.
- [7] Sokolov GY. A security Doppler radar experiment. In *Radar Methods and Systems Workshop (RMSW)*, IEEE 2016 Sep 27 (pp. 79-80). IEEE.
- [8] Ryde J, Hillier N. Performance of laser and radar ranging devices in adverse environmental conditions. *Journal of Field Robotics*. 2009 Sep 1;26(9):712-27.
- [9] Gu X, Xian B, Guo F, Zhang Y, Zhang X. Pose estimation based on laser range finder for a quadrotor unmanned aerial vehicle in GPS-denied environment. In *InControl Conference (CCC)*, 2013 32nd Chinese 2013 Jul 26 (pp. 667-672). IEEE.
- [10] Nara S, Miyamoto D, Takahashi S. Position measurement of crane hook by vision and laser. In *IEEE, Industrial Electronics, IECON 2006-32nd Annual Conference on* 2006 Nov 6 (pp. 184-189). IEEE.
- [11] Chun TW, Kim KM, Lee HH, Kim HG, Nho EC. Fast scanning method for container stacking profile with one laser sensor. In *Industrial Electronics Society*,

2003. IECON'03. The 29th Annual Conference of the IEEE 2003 Nov 2 (Vol. 2, pp. 1092-1096). IEEE.
- [12] Charniya NN, Dudul SV. Simple low-cost system for thickness measurement of metallic plates using laser mouse navigation sensor. *IEEE Transactions on Instrumentation and Measurement*. 2010 Oct;59(10):2700-5.
- [13] Lindner P, Wanielik G. 3D LIDAR processing for vehicle safety and environment recognition. In *Computational Intelligence in Vehicles and Vehicular Systems, 2009. CIVVS'09. IEEE Workshop on 2009 Mar 30* (pp. 66-71). IEEE.
- [14] Matsui S, Goktug G. Slit laser sensor guided real-time seam tracking arc welding robot system for non-uniform joint gaps. In *Industrial Technology, 2002. IEEE ICIT'02. 2002 IEEE International Conference on 2002* (Vol. 1, pp. 159-162). IEEE.
- [15] Wu F, Wen C, Guo Y, Wang J, Yu Y, Wang C, Li J. Rapid Localization and Extraction of Street Light Poles in Mobile LiDAR Point Clouds: A Supervoxel-Based Approach. *IEEE Transactions on Intelligent Transportation Systems*. 2016 Jun 22.
- [16] Safdarinezhad A, Mokhtarzade M, Zoj MJ. Coregistration of Satellite Images and Airborne LiDAR Data Through the Automatic Bias Reduction of RPCs. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 2016 Jul 14.
- [17] Leddar Evaluation Kit | Solid-State LiDAR Module. In: Leddar tech. <http://leddartech.com/leddar-evaluation-kit/>. Accessed 16 Feb 2017
- [18] Péter G, Zsolt S, Szilárd A. *Highly Automated Vehicle Systems*. 1st ed. BME MOGI; 2014.
- [19] Motorsport B. Collision Avoidance System CAS-M light [Internet]. BOSCH; 2017. Available from: http://www.bosch-motorsport.com/media/catalog_resources/Collision_Avoidance_System_CAS-M_lightpdf.pdf
- [20] Borenstein J, Everett H, Feng L, Wehe D. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*. 1997;14(4):231-249.
- [21] GIM - Finnish Centre of Excellence in Generic Intelligent Machines Research [Internet]. [Gim.aalto.fi](http://gim.aalto.fi). 2013 [cited 9 January 2017]. Available from: <http://gim.aalto.fi/>

- [22] Mohammadi Aref M, Ghabcheloo R, Kolu A, Mattila J. Position-based visual servoing for pallet picking by an articulated-frame-steering hydraulic mobile machine. 6th International Conference on Robotics, Automation and Mechatronics (RAM). 2013.
- [23] Ghabcheloo R, Hyvönen M. Modeling and motion control of an articulated-frame-steering hydraulic mobile machine. 17th Mediterranean Conference on Control & Automation. 2009.
- [24] Ghabcheloo R, Hyvönen M, Uusisalo J, Karhu O, Järä J, Huhtala K. Automotive Motion Control of a Wheel Loader. ASME 2009 Dynamic Systems and Control Conference, Volume 2. 2009. pp. 427-434.
- [25] Dudek G, Jenkin M. Inertial Sensors, GPS, and Odometry. Springer Handbook of Robotics. Springer; 2008. p. 477-490.
- [26] Shen J, Tick D, Gans N. Localization through fusion of discrete and continuous epipolar geometry with wheel and IMU odometry. American Control Conference (ACC). 2011.
- [27] Merriaux P, Dupuis Y, Vasseur P, Savatier X. Wheel odometry-based car localization and tracking on vectorial map. Intelligent Transportation Systems (ITSC), IEEE 17th International Conference. 2014.
- [28] Jha A, Kumar M. Two wheels differential type odometry for mobile robots. Reliability, In focom Technologies and Optimization (ICRITO) (Trends and Future Directions), 3rd International Conference. 2014.
- [29] Häne C, Sattler T, Pollefeys M. Obstacle detection for self-driving cars using only monocular cameras and wheel odometry. Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference. 2015.
- [30] Vivet D, Checchin P, Chapuis R. Radar-only localization and mapping for ground vehicle at high speed and for riverside boat. Robotics and Automation (ICRA), IEEE International Conference. 2012.
- [31] Checchin P, Gérossier F, Blanc C, Chapuis R, Trassoudaine L. Radar Scan Matching SLAM Using the Fourier-Mellin Transform. Field and Service Robotics. 62nd ed. Springer Tracts in Advanced Robotics; 2010. p. 151-161.
- [32] Kellner D, Barjenbruch M, Klappstein J, Dickmann J, Dietmayer K. Instantaneous ego-motion estimation using Doppler radar. Intelligent Transportation Systems - (ITSC), 16th International IEEE Conference. 2013.

- [33] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann and K. Dietmayer, "Instantaneous Ego-Motion Estimation using Multiple Doppler Radars", in 2014 IEEE International Conference on Robotics & Automation (ICRA), Hong Kong, 2014
- [34] M. Rapp, M. Barjenbruch, K. Dietmayer, M. Hahn and J. Dickmann, "A fast probabilistic ego-motion estimation framework for radar", in Mobile Robots (ECMR), 2015 European Conference, 2015.
- [35] A. Corominas-Murtra, J. Vallvé, J. Solà, I. Flores and J. Andrade-Cetto, "Observability analysis and optimal sensor placement in stereo radar odometry", in Robotics and Automation (ICRA), 2016 IEEE International Conference, 2016.
- [36] Garcia F, de la Escalera A, Armingol J. Enhanced obstacle detection based on Data Fusion for ADAS applications. Intelligent Transportation Systems - (ITSC), 16th International IEEE Conference. 2013.
- [37] Simulink - Simulation and Model-Based Design [Internet]. Se.mathworks.com. 2015 [cited 1 January 2017]. Available from: <https://se.mathworks.com/products/simulink.html>
- [38] Simulink Real-Time - Simulink - MATLAB & Simulink [Internet]. Se.mathworks.com. 2015 [cited 1 January 2017]. Available from: <https://se.mathworks.com/products/simulink-real-time.html>
- [39] The MathWorks. xPC Target, For Use with Real-Time Workshop [Internet]. The MathWorks; 2007. Available from: https://www.mathworks.com/tagteam/37937_xpc_target_selecting_hardware_guide.pdf
- [40] ROS/Introduction - ROS Wiki [Internet]. Wiki.ros.org. 2014 [cited 4 May 2016]. Available from: <http://wiki.ros.org/ROS/Introduction>
- [41] Fernandez E. Learning ROS for robotics programming. 1st ed. PACKT Publishing; 2013.
- [42] roscore - ROS Wiki [Internet]. Wiki.ros.org. 2017 [cited 3 April 2016]. Available from: <http://wiki.ros.org/roscore>
- [43] ROS/Tutorials/Recording and playing back data - ROS Wiki [Internet]. Wiki.ros.org. 2017 [cited 14 January 2017]. Available from: <http://wiki.ros.org/ROS/Tutorials/Recording%20and%20playing%20back%20data>

- [44] Fischler M, Bolles R. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*. 1981;24(6):381-395.

APPENDIX A: THERMAL SENSOR

The non-contact temperature sensor measures the surface temperature of an object. The silicon lens collects radiated heat (far-infrared ray) emitted from an object onto the thermopile sensor in the module. The radiated heat (far-infrared ray) produces an electromotive force on the thermopile sensor. The analog circuit calculates the temperature of an object by using the electromotive force value and a measured temperature value inside the module. The measured value is outputted through an I2C bus. The module can also be used for detecting the presence of human beings. Omron's non-contact temperature sensor can solve the shortcomings of a conventional pyroelectric sensor, which cannot catch the signal of a stationary person because the sensor detects the change of signal. In cases where a D6T sensor is used for detecting human beings, the application will be limited to close range.

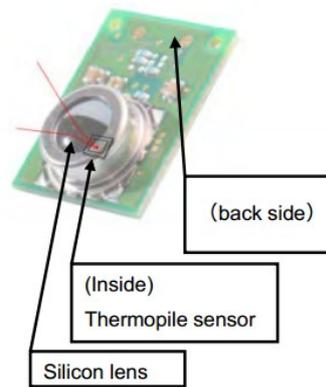


Figure 40: Omron thermal Sensor

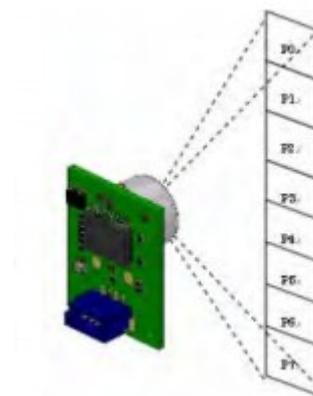


Figure 39: 8 segment Thermal Sensor

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller. Thermal Sensor sends data via I2C, so to read I2C 'wire library' of Arduino is utilized. After reading the data in microcontroller it is sent to computer via serial port. Code for Arduino Uno is shown below.

```

1  #include <Wire.h>
2  #define D6T_addr 0x0A
3  #define D6T_cmd 0x4C
4  byte rbuf[19];
5  float temp[9]; // amb + 8 temp_readings
6  void setup()
7  {
8      Wire.begin();
9      Serial.begin(9600); //
10 }
11 void loop()
12 {
13     int i;
14     Wire.beginTransmission(D6T_addr);
15     Wire.write(D6T_cmd);
16     Wire.endTransmission();
17     delay(70);
18     Wire.requestFrom(D6T_addr,19);
19     delay(100); // < ----- 100MS DELAY
20     for (i = 0; i < 19; i++)
21     {
22
23         while(!Wire.available()); // <----- ADDED
24         rbuf[i] = Wire.read();
25     }
26     for (i=0; i<9; i++)
27     {
28         temp[i]=(rbuf[(i*2)]+(rbuf[(i*2+1)]<<8))*0.1;
29     }
30     if (temp[0]>0) // <----- ADDED
31     {
32         for (i=0; i<9; i++)
33         {
34             Serial.print (temp[i]) ;
35             if ( i < 8 ) {
36                 Serial .print ( "," ) ;
37             }
38             else {
39                 Serial . println ( ) ;
40             }
41         }
42     }
43     delay(100);
44 }

```

Program 7. Arduino code for Omron thermal sensor

The data is received via serial port in MATLAB real-time Simulink. It contains 9 bytes of data, 1st of which is the reference temperature value and rest of the 8 are each for the 8 segments. These 8 segments are further sent to ROS for visualization. The change in the temperature value changes the color of the cubic marker.

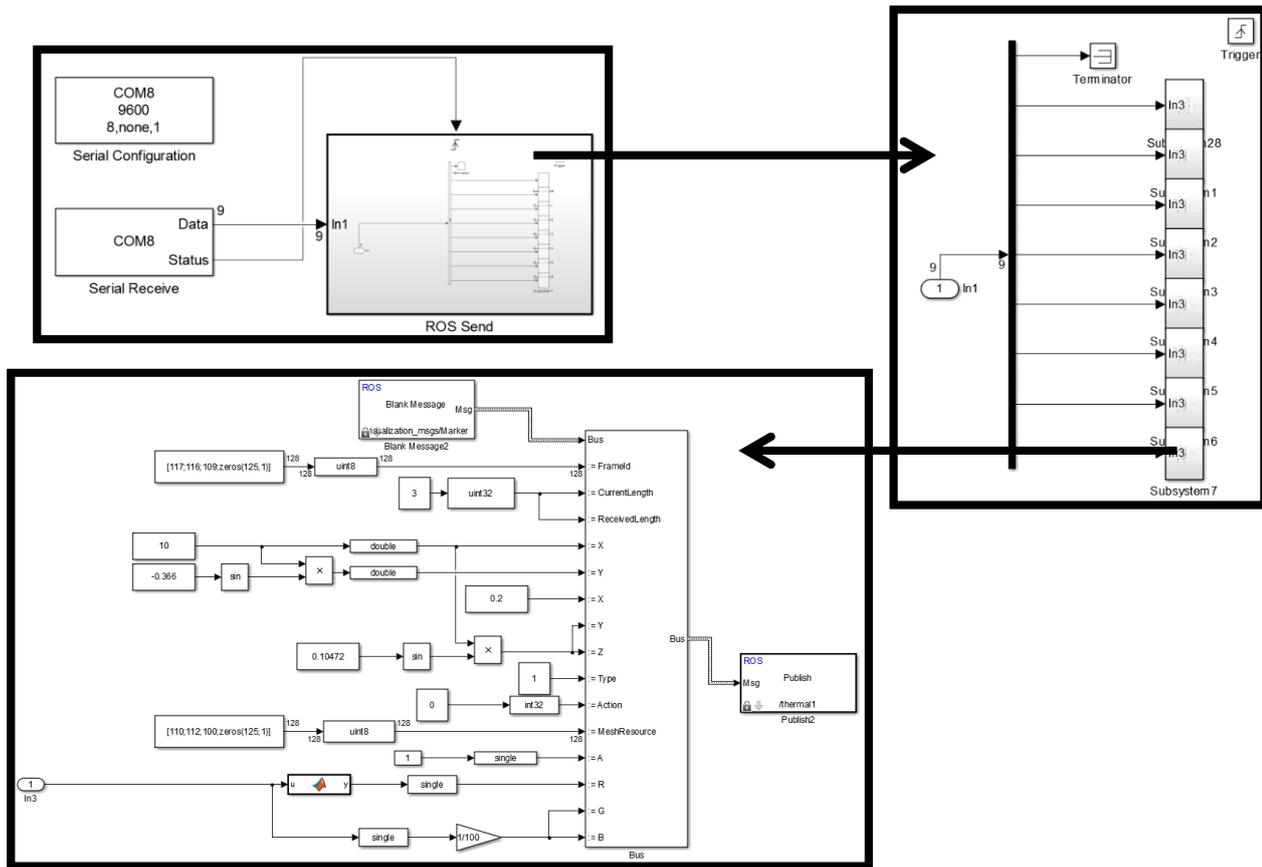
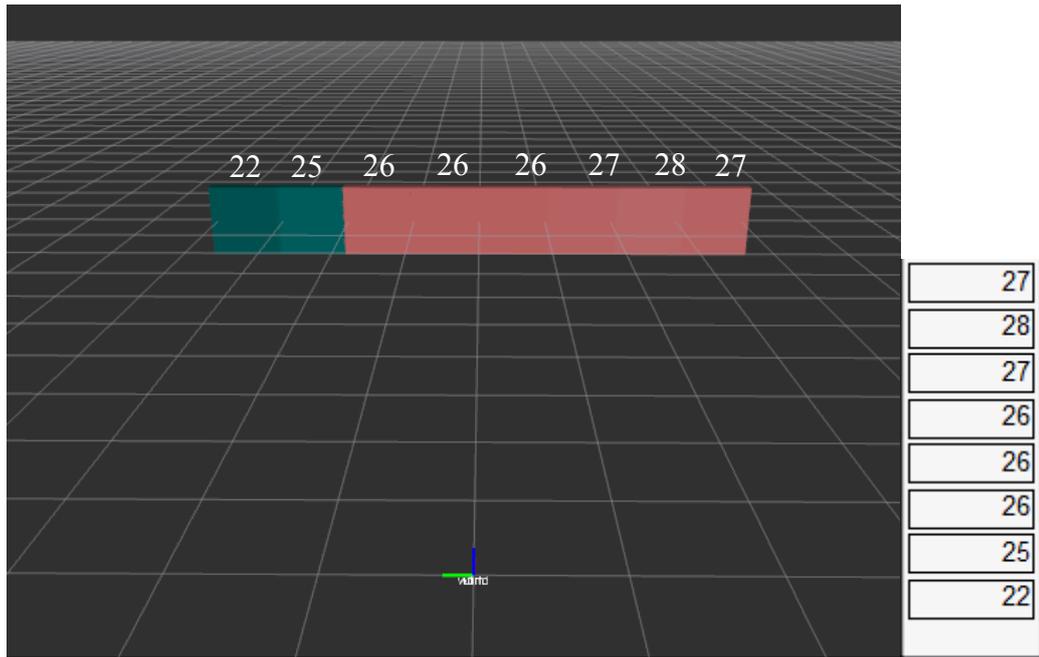


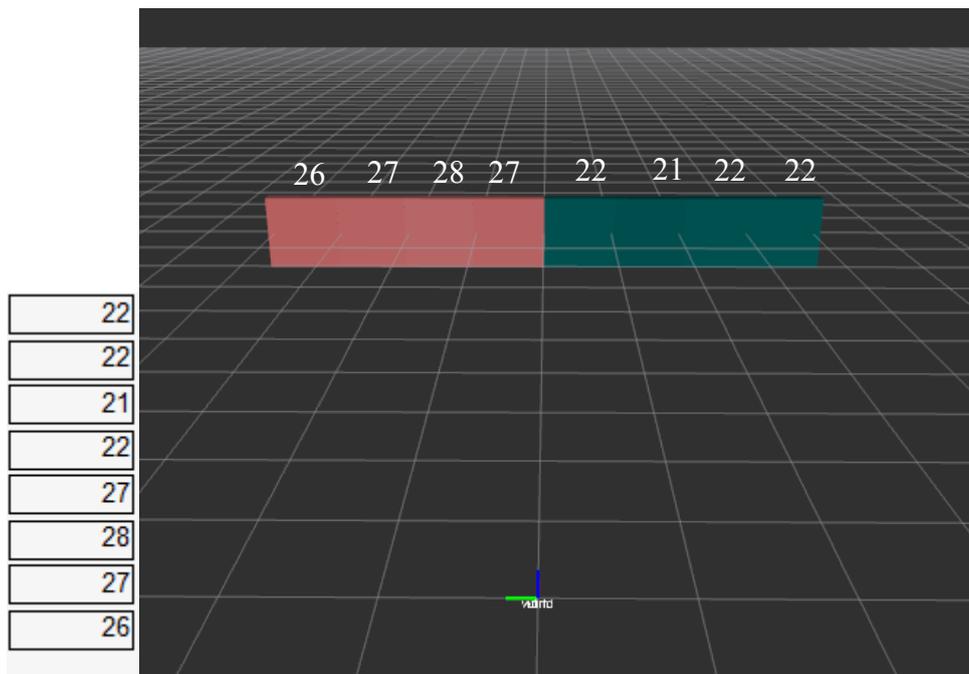
Figure 41: Receiving Data from Arduino and sending Data to ROS

Indoor testing of Thermal Sensor

Indoor tests have been performed with Omron D6T-8L thermal sensor. Arduino is utilized to read the sensor and send the data through serial port to computer where the data is further processed and sent to ROS for visualization. Below is the image of visualization of ROS for the thermal sensor.

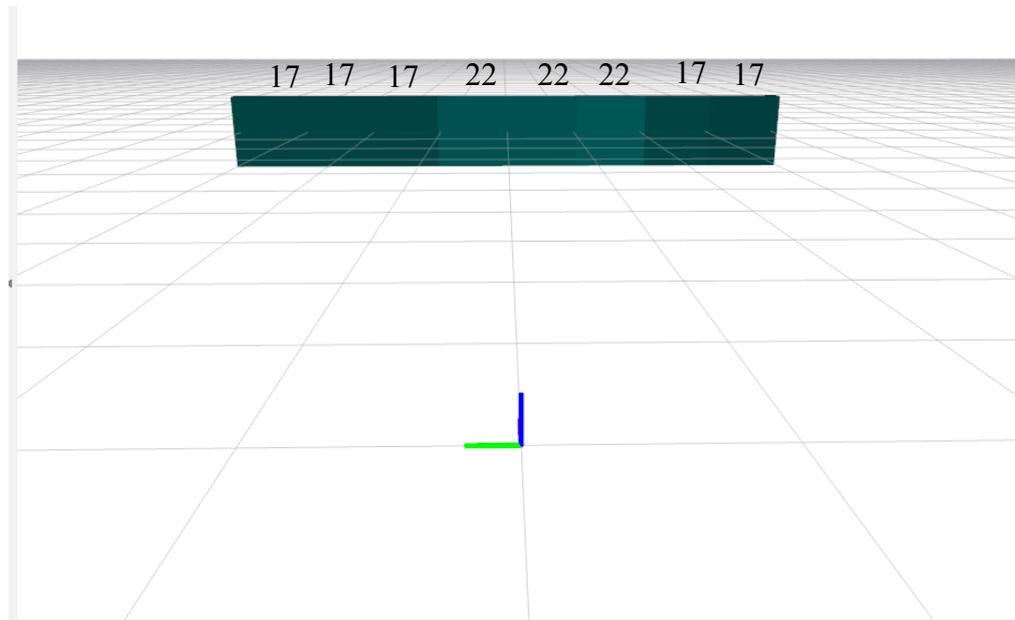


In the image above the 8 segments can be seen visualized in ROS. each segment's value in degree is presented over the segments. red shade segments are those in which the sensor is detecting human and green are the ones in which there is no presence of human.



In this figure the sensor is detecting human presence in four of its eight segments and the rest of the four segments to the right are not detecting any human presence. this should be kept in mind that these tests have been performed for indoor conditions. for outdoor conditions the ranges may differ from these and sensor can discriminate on basis of temperature only so any object having temperature equal to that human will also be classified as human by the sensor.

Outdoor testing by thermal sensor



Outdoor testing of the thermal sensor has been performed in the testing track with both static and dynamic conditions. the maximum range of the sensor to detect human is 3 m, but it should be kept in mind that it depends on the environment weather if it is hot than the range might decrease because after 3 m range the detection temperature of human is 22° which is not correct because if a human is near to the sensor till 1.5 m range the temperature of human is shown up to 28°. In this image three segments in between are able to detect human the rest of the segments are detecting the environment. this test is performed with static sensor condition.

APPENDIX B: KELLNER'S APPROACH FOR ODOMETRY ESTIMATION

In [32] Kellner et al. calculated the linear and angular velocity of the vehicle using automotive radar. The technique involves using heading angle Ψ of the machine and sensor velocity ${}^w\dot{\mathbf{r}}_B$

A least-square approach is used by Kellner et al. [32] to calculate the odometry based on the radial velocity v_{ri} and azimuthal angle θ_i of all the N objects.

$$\begin{bmatrix} v_{r1} \\ \vdots \\ v_{rN} \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \vdots & \vdots \\ \cos(\theta_N) & \sin(\theta_N) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad 17$$

$$\text{With } v_x \cos\theta_i + v_y \sin\theta_i = v_{ri} \quad 18$$

After calculating v_x and v_y heading angle Ψ and sensor velocity ${}^w\dot{\mathbf{r}}_B$ are calculated using the equations

$$v_x = -\cos(\Psi) {}^w\dot{\mathbf{r}}_B \text{ and } v_y = -\sin(\Psi) {}^w\dot{\mathbf{r}}_B \quad 19$$

Later these both variables are used to calculate the linear and angular velocity, based on the Ackermann conditions of the machine on which the sensor is mounted.

$$v_B = \left(\cos(\Psi) - \frac{y_B}{x_B} \sin(\Psi) \right) {}^w\dot{\mathbf{r}}_B \quad 20$$

$$\omega_B = \frac{\sin(\Psi)}{x_B} {}^w\dot{\mathbf{r}}_B \quad 21$$

In equation 20, v_B is the linear velocity of the machine and, y_B and x_B are the position of the sensor from the base. In equation 21, ω_B is the angular velocity of the machine.