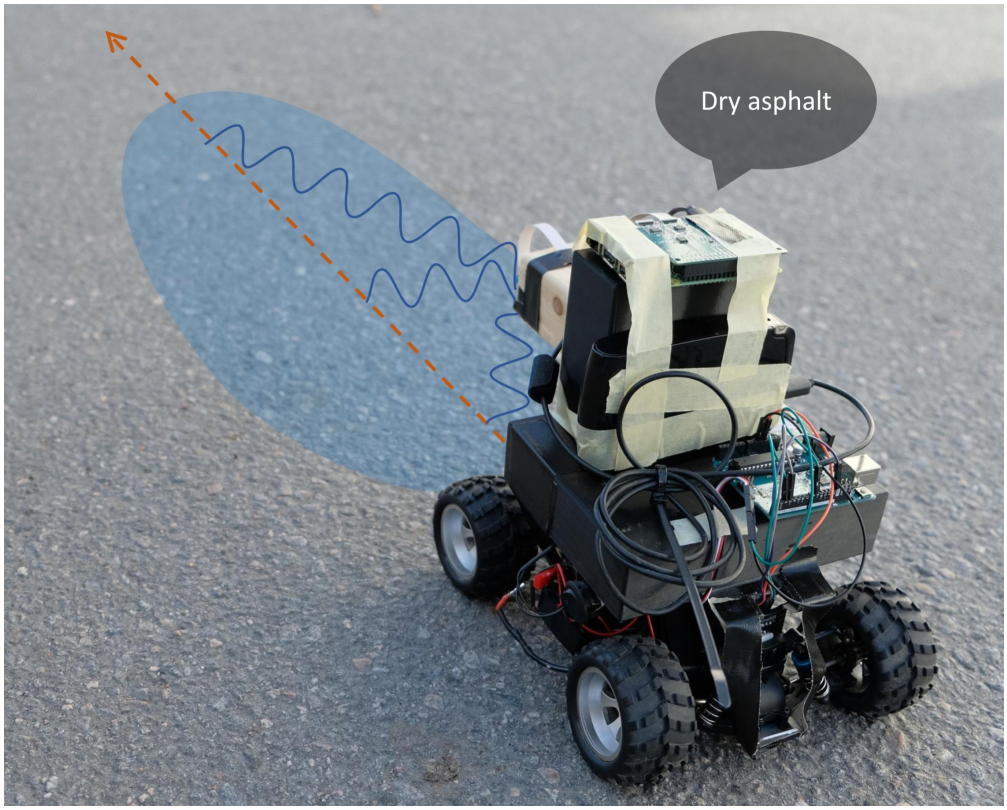
CHALMERS
UNIVERSITY OF TECHNOLOGY



Classifying Road Conditions with Radar and Supervised Learning

Bachelor's Thesis in Electrical Engineering

Anton Ingemansson
Carl Folkesson
Jonathan Jonsson
Kevin Wingård Olsson
Max Johansson
Rebecka Bergström

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Bachelor's Thesis: EENX15-20-03

Classifying Road Conditions with Radar and Supervised Learning

Anton Ingemansson
Carl Folkesson
Jonathan Jonsson
Kevin Wingård Olsson
Max Johansson
Rebecka Bergström

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2020

Classifying Road Conditions with Radar and Supervised Learning

Supervisor: Thomas Rylander, Department of Electrical Engineering
Examiner: Tomas McKelvey, Department of Electrical Engineering

Bachelor's Thesis EENX20-15-03
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The cover displays the data acquisition setup with an animation that projects the radar lobe on the surface.

Typeset in LaTeX
Gothenburg, Sweden 2020

Abstract

This thesis deals with the classification of road surfaces in motion at a constant velocity based on a 60 GHz pulse coherent radar and supervised machine learning. The surfaces chosen for classification are dry asphalt, wet asphalt, and snow. Data is acquired from different sites using the Acconeer A111 radar mounted on a radio-controlled car. Data sets are built through feature extraction methods based on averaging and autocovariance of amplitude. The machine learning algorithms evaluated are support vector machine, random forest, and neural network. These machine learning algorithms are tested and individually evaluated. The results suggest that it is possible to classify surfaces with great certainty.

Sammandrag

Den här uppsatsen handlar om klassificering av vägytor vid rörelse i konstant hastighet baserat på en 60 GHz puls-koherent radar och maskininlärning. Urvalet av ytor för klassificering består av torr asfalt, våt asfalt och snö. Datainsamling sker på olika platser med hjälp av Acconeers A111-radarenhet, monterad på en radiostyrd bil. Ett dataset är byggt med metoder baserade på medelvärdesbildning och autokovarians av amplitud. De utvärderade maskininlärningsalgoritmerna är *support vector machine, random forest* och *neural network*. Dessa maskininlärningsalgoritmer testas och utvärderas individuellt. Resultaten tyder på att det är möjligt att klassificera ytor med hög träffsäkerhet.

Keywords: Surface Classification, 60 GHz, Pulse Coherent Radar, Acconeer A111, Feature Extraction, Autocovariance, Supervised Machine Learning, Support Vector Machine (SVM), Random Forest (RF), Neural Network (NN)

Acknowledgements

This project would not be as elaborate without all the external support we have received. We especially want to thank our supervisor Thomas Rylander who has continuously contributed a lot of insightful knowledge and advice. His dedication and rational discussions have taught us a lot and greatly contributed to the project. We also want to share great gratitude to the people at UniqueSec AB who provided us with equipment and vision to fulfil this project. At last, we want to thank the CASE-Lab and the association at Chalmers for letting us use the lab, store equipment, and helping us with 3D printing components for the project.

List of Symbols

α	Attenuation constant
β	Phase constant
δ	Radar tilt angle
ϵ'	Real part of the relative permittivity
ϵ''	Imaginary part of the relative permittivity
ϵ_0	Vacuum permittivity
ϵ_r	Relative permittivity
η	Wave impedance
Γ	Reflection coefficient
γ	Propagation constant
$\hat{\sigma}^s$	Estimated average standard deviation of surface s
$\hat{\boldsymbol{\mu}}_y$	Mean vector of process y
$\hat{\boldsymbol{r}}_y$	The estimated autocovariance vector for slow-time frame n and time-lag k
λ	Wavelength
λ_c	Carrier wavelength
μ_0	Vacuum permeability
μ_x	Mean value of variable x
ω	Angular frequency
σ	Radar cross section
σ	Standard deviation
σ^o	Normalized backscatter coefficient
τ	Pulse duration
τ_0	Relaxation time
θ_1, θ_2	3 dB angles for E-plane
θ_i	Incident angle
θ_s	Reflection angle
φ_1, φ_2	3 dB angles for H-plane
$\boldsymbol{f}(n)$	Feature vector corresponding to slow-time frame n
$\boldsymbol{r}$	Vector from antenna to a point on surface
$\boldsymbol{y}(n)$	Slow-time frame of time-index n
A_e	Effective aperture
c_o	Speed of light
dS'	Differential area
f_c	Carrier frequency
f_d	Doppler frequency
f_r	Frequency of received signal
f_t	Frequency of transmitted signal

G	Antenna gain factor
h	Mounting height of the radar
L	Molecular radius
l	Thickness of dielectric
N	Number of slow-time frames per feature vector
n_{ds}	Number of fast-time samples per frame after downsampling
P_r	Received power
P_t	Transmitted power
Q_r	Backscattered power density
$Q_{t,iso}$	Isotropically radiated power density
r	Range to surface
R_x	Receiving antenna
r_{sr}	Distance from surface - receiver
r_{ts}	Distance from transmitter to surface
T	Temperature in Kelvin
T_x	Transmitting antenna
v	Viscosity

List of Acronyms

ACF Autocovariance Function.

ESC Electronic Speed Controller.

HPBW Half-Power Beam Width.
HWAAS Hardware Accelerated Average Samples.

NN Neural Network.

PCR Pulse Coherent Radar.
PRF Pulse Repetition Frequency.
PWM Pulse-Width Modulation.

RC Remote Control.
RCS Radar Cross Section.
ReLU Rectified Linear Unit.
RF Radio Frequency.
RF Random Forest.
RPM Revolutions Per Minute.

SNR Signal-to-Noise Ratio.
SVM Support Vector Machine.

WSS Wide-Sense Stationary.

Contents

List of Figures

List of Tables

1
Introduction

In this chapter, the relevant background information, the objective, and scope of the project are presented.

1.1 Background

Reliable identification of surfaces is applicable in numerous situations. An autonomous robot, such as a vacuum cleaner or industrial robot, can use such technology to remain on the desired surface. Another possible application is for quality inspection of surfaces, such as airport runways or paint-jobs. Characterization of surfaces can also be a useful component in cars, especially in self-driving cars, as the handling and therefore the necessary setup of the car greatly depends on the surface it is driving on. In slippery conditions, the wheel torque needs to be steady and the speed into a sharp corner has to be slower than while driving on dry asphalt.

Several methods can potentially be used for the identification of surfaces. Three common ones are optical, acoustic, and radar measurements.

Optical measurements can generate a high-resolution picture of a surface and the output is often intuitively interpretable by people. However, this requires a clean lens and a clear view of the surface. This technique is therefore sensitive to dirt, water, and fog [1, p. 4]. It also requires a source of light, which makes it difficult to use during the night [2, p. 8]. Due to these reasons, there are many situations where the optical method may be unsuitable.

Acoustic measurements can be conducted using different methods. One such method is the usage of ultrasound. A downside to this method is that it is sensitive to surrounding noises, of which there can be plenty in a home, in a factory, and especially on a road. The acoustic method can, therefore, be very complex in terms of filtering out unwanted noise. Another possible disadvantage may be in applications where latency is of importance. The speed of sound might in certain circumstances limit the response time.

Radar measurements, however, are not as sensitive to dirt or fog, and only sensitive to weather at long distances [1, p. 4][3, p. 56]. Vibrations add an uncertainty effect with radar, but this is common for all three methods. Using radar is also a good method when the latency is important. With these things in mind, radar is a

method worth examining for surface classification purposes and will be the subject of this project.

The term *radar* is an acronym for Radar Detection and Ranging [1, pp. 2-7]. Today, radars are so common that the word itself has become a standard English noun. Radar technology has been around since 1904 when Christian Hülsmeyer made the first successful experiment with radar [4]. Robert Watson-Watt proved in 1935 that radar can be used to detect airplanes [5]. The development of radar technology then took off during the 1930s, especially during World War II as it became an important tool for surveillance. Nowadays, the technology is still used widely in the military but also in many other applications, for instance as traffic radars, for measuring speed or as meteorological radars for weather monitoring [1, pp. 1-3].

Briefly, radar is an electromagnetic transmitter and receiver [6, pp. 1-2]. These are used for detecting and locating objects by measuring reflectively. The radar radiates electromagnetic waves through an antenna. The waves propagate in space and are reflected by some object at a certain distance. Some of the waves are reflected back in the direction of the transmitter as an echo and are received by the radar antenna. With signal processing, the received signal can be analyzed to decide whether an echo from a target is present. Then, desired information can be extracted such as the distance to the object and the speed of the object. The backscatter, and therefore the reflected signal, will be dependent on the material attributes of the scattering surface.

The raw data that the radar receives may be hard to interpret without the use of signal processing, but through different methods, meaningful information can be extracted. Depending on what the goal is, there are different ways to alter and filter out specific parts of the data.

A difficulty when using radar on surfaces is that data produced by radar is hard to interpret intuitively by humans. However, a computer might be able to distinguish surfaces given a model that can translate radar data to classifications.

One way of creating a classification model would be by studying the theory behind radar propagation and how it relates to surface attributes. This is explored by several studies [7]–[9], which analyze the backscatter at different polarizations to distinguish surface types. Another method would be a data-based approach. This would mean building such a classification model with a machine learning algorithm instead, using previously acquired data as its input. Montgomery and Holmén [10] demonstrate the feasibility of distinguishing grass from surfaces such as pavement and gravel using machine learning. The data-based approach to creating a classification model will also be the subject of this thesis.

1.2 Objective and Scope

The objective of this thesis is to classify road surface conditions, specifically dry asphalt, wet asphalt and snow, while in motion. A classification model will be developed using machine learning algorithms and acquired data. The data will be collected at a constant velocity by a radar attached to a remote-controlled (RC) car. Physical models related to radar propagation will also be examined.

The equipment used is the A111 radar sensor connected to the XC112 evaluation kit, both manufactured by Acconeer. The A111 sensor features 60 GHz carrier-frequency and pulse coherency. The equipment is provided for the project, so evaluating and selecting equipment is not necessary. Only surfaces mentioned in the objective are considered in this project. The RC car requires relatively flat surfaces to drive on, thereby only horizontal and even surfaces are the subject of data acquisition. The data will be acquired while traveling at a constant velocity. Measurements on snow are made in an indoor skiing facility with artificial snow.

2

Model of Experiment

In this chapter, the concepts of radar are described. It contains a description of the pulse coherent radar used in this thesis and of its signal characteristics. Furthermore, a model is derived that relates radar parameters to geometry and electromagnetic properties of materials.

2.1 Pulse Coherent Radar

The radar sensor that is used in this thesis is a pulse coherent radar (PCR). This type of radar transmits signals in form of short pulses [11], which is illustrated in Figure 2.1.

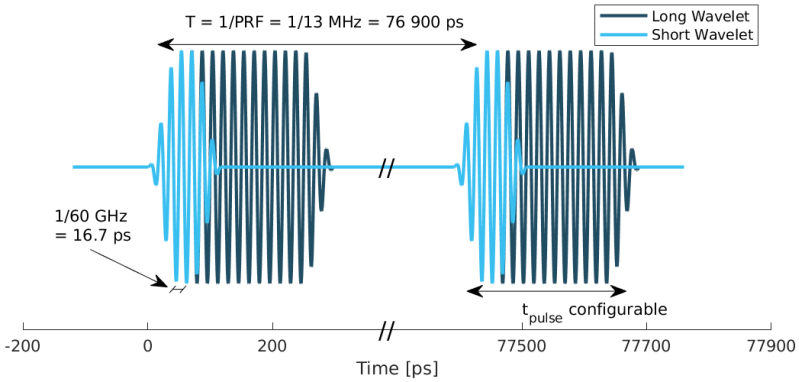


Figure 2.1: Illustration of transmitted pulse in the time domain for the Acconeer A111 radar sensor. Source: [11]

There are two characteristics in the figure worth noting; the pulse repetition frequency (PRF) and the pulse length. The PRF determines how often pulses are transmitted and the pulse length determines the transmission duration [11]. The choice of pulse length also determines the depth resolution. Short pulses provide high distance resolution but lower signal-to-noise ratio (SNR) in comparison to longer pulses [11]. Long pulses contain more energy which provides high SNR but lower distance resolution.

For a PCR, the radio is switched off between transmission of pulses which makes the radar energy efficient since the radio is active for only a small portion of time. The fact that the radar is coherent means that the transmitted and received signals have a constant phase relationship [1, p. 19]. Coherent radars allow for high accuracy measurements because small phase shifts, caused by the Doppler effect discussed later on, can be measured. Also, coherent radars have higher SNR than non-coherent radars [1, p. 19].

2.1.1 System Overview

Figure 2.2 below illustrates the subsystems usually found in a pulsed monostatic radar system, similar to the one used in this project. Monostatic means that the transmitter and receiver are collocated.

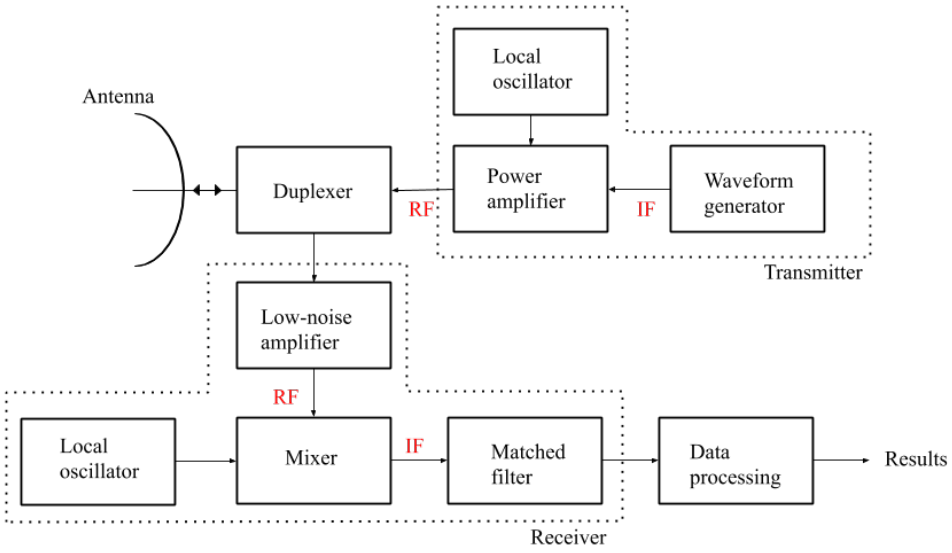


Figure 2.2: Simple block diagram of a pulsed monostatic radar

The main components of the PCR system are the transmitter, receiver, duplexer and antenna [1, pp. 2-7]. Furthermore, the transmitter and receiver consist of several subcomponents which are indicated by the dotted lines. In the transmitter, a signal is generated by the waveform generator. This signal is then amplified and modulated to a radio frequency (RF) by the power amplifier using a mixer combined with a local oscillator. The transmitter then feeds the signal to the duplexer. The function of the duplexer is to allow a single antenna to be used for both transmitting and receiving the radar signal [6, pp. 2-5]. The transmitted power is much higher than what the receiver can manage. Therefore, the duplexer toggles between the transmitter and receiver in order to protect the receiver from being damaged while the transmitter is activated.

The signal from the transmitter is routed through the duplexer to the antenna which radiates electromagnetic waves into space. The signal propagates and is reflected

by some object and parts of the signal propagates back to the antenna. The duplexer then routes the received signal to the receiver. Usually, the receiver is of a superheterodyne type whose goal is to separate out the desired signal [1, pp. 2-7]. The first stage in a superheterodyne receiver is usually a low-noise RF amplifier. This is followed by a mixer to lower the RF to intermediate frequencies, ultimately to baseband. Each modulation consists of a mixer combined with a local oscillator. Before the signal is ready for data processing, it goes through a matched filter which is used to maximize the SNR [6, pp. 2-5].

2.1.2 Signal Characteristics

The generated signal used for transmission in a PCR is often modelled as:

$$y_t(t) = A(t)\sin\left(2\pi f_c t + \phi(t)\right), \tag{2.1}$$

where $A(t)$ describes the modulated amplitude, f_c is the radar carrier frequency and $\phi(t)$ may be either a constant or represent a phase modulation [3, pp. 53-54]. The received signal will be altered due to delay, distance, noise and propagation loss among other things. Thus, a received signal from a single transmitted pulse scattered by an object at a distance R from the radar antenna can be expressed as

$$y_r(t) = B(t - t_0)\sin\left(2\pi f_c(t - t_0) + \theta(t)\right) + n(t) \tag{2.2}$$

where $B(t)$ is the amplitude of the scattered signal, $\theta(t)$ is the echo phase modulation and $n(t)$ describes some noise. The delay t_0 represents the delay of the scattered signal located at a distance $R = c_0 t_0/2$ from the radar antenna.

Usually, the received signal has a shifted frequency caused by the Doppler effect [3, pp. 92-94]. This phenomenon occurs when the transmitting and receiving sources move relative to each other. If the transmitted signal's frequency is referred as f_t, then the frequency for the received signal, f_r, is expressed as

$$f_r = \frac{1 + \frac{v}{c_0}}{1 - \frac{v}{c_0}} f_t = \frac{c_0 + v}{c_0 - v} f_t. \tag{2.3}$$

A positive radial velocity v corresponds to an object moving towards the radar, resulting in an increasing frequency for the received signal [3, pp. 92-94]. In the case of a monostatic radar, Equation (2.3) describes the effect of a moving object in the radar field. Usually, the velocity of the radar object is comparatively small to the speed of light c_0, which allows for an approximation of Equation (2.3):

$$f_r \approx (1 + \frac{2v}{c_0}) f_t. \tag{2.4}$$

Furthermore, the Doppler shift is the difference between the carrier frequency of the transmitted signal and the frequency of the received signal. The Doppler shift can be expressed through the estimate (2.4) as

$$f_d \approx \frac{2v}{c_0} f_t = \frac{2v}{\lambda_c},$$

where λ_c represents the wavelength of the transmitted signal, also known as the carrier wavelength.

2.2 Radar Wave Propagation

For radar wave propagation, it is of interest to predict how much power is reflected by a surface in relation to the transmitted power. This is interesting because the amount of reflected power is directly related to material properties [12]. For a point-scatterer, the common radar range equation [3, pp. 54-55] gives such a prediction. Figure 2.3 depicts a general description of the geometry in relation to the transmitting antenna T_x and receiving antenna R_x.

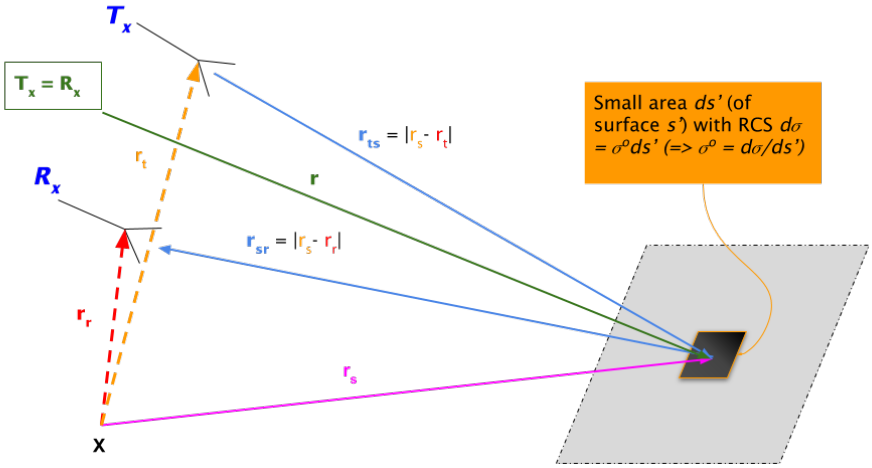


Figure 2.3: Geometry of the radar measurements and visualisation of vectors used for further derivation of the model

2.2.1 Radar Equation for Single Point-scatterer

The radar range equation is derived using conservation of energy. Let P_t denote the transmitted mean power of the time harmonic waves radiated by the transmitting antenna. The transmitted isotropic power density $Q_{t,iso}$ can be expressed as a function of distance r_{ts} as

$$Q_{t,iso}(r_{ts}) = \frac{P_t}{4\pi r_{ts}^2}. \tag{2.5}$$

However, the antenna of the A111 radar module is not an isotropic radiator [13] and therefore an antenna gain factor G is introduced. The A111 sensor is a monostatic radar, which means that the receiving and transmitting antenna are physically the same and a distinction between transmitting and receiving gain is not necessary. The length of r_{ts} is therefore also the same as r_{sr} for a monostatic radar. The distances r_{ts} and r_{sr} will therefore instead be denoted with r. Equation (2.5) is adopted by incorporation of the antenna gain factor as

$$Q_t = \frac{P_t G}{4\pi r^2} \tag{2.6}$$

The power continues to spread indefinitely until it gets scattered by a surface, or some other object. The scattering surface can also be regarded as a transmitter with

the difference being that it is completely passive. First, assume that the surface is lossless and that all incident power is backscattered isotropically. To calculate the incident power on a surface, the previously described power density in Equation (2.6) is multiplied by the electromagnetic cross-section of the illuminated object. The electromagnetic radar cross-section is RCS is denoted σ. The backscattered power density (Q_r) is, with the new additions, described as

$$Q_r = \frac{Q_t \sigma}{4\pi r^2} = \frac{P_t G \sigma}{(4\pi)^2 r^4}. \tag{2.7}$$

To derive the final equation for the received power, a few steps remain. The backscattered waves impinge on the radar antenna and the received power is $P_r = A_e Q_r$, where A_e is the effective aperture of the antenna. The effective aperture is related [14, p. 746] to the antenna gain and wavelength λ by $A_e = (\lambda^2 G)/(4\pi)$. Incorporation of effective aperture in Equation (2.7) gives the total received power as

$$P_r = \frac{P_t G \sigma}{(4\pi)^2 r^4} \frac{G\lambda^2}{4\pi} = \frac{P_t G^2 \sigma \lambda^2}{(4\pi)^3 r^4}. \tag{2.8}$$

Equation (2.8) is known as the radar range equation for a point scatterer.

2.2.2 Radar Equation for Surface Scatterer

Some alterations to Equation (2.8) must be made to fit the description of a surface scatterer. The model derived above assumes that the surface can be described as a point scatterer. Ground clutter is best described as a distributed surface scatterer [3, p. 57]. To model a surface scatterer, many parts in the previously described derivation are still relevant. Let the backscattered power density described by Equation (2.7) instead be described as a sum of differential areas $d's$. Mathematically, the RCS and antenna gain are described as $\sigma = \sigma^o(\hat{\boldsymbol{r}})$ and $G = G(\hat{\boldsymbol{r}})$. Where the normalized backscatter coefficient is defined as $\sigma^o = d\sigma/dS'$.

The A111 sensor uses a folded dipole antenna [13]. For simplicity we approximate the antenna gain to be constant G_0 between the half power beam width (HPBW) angles. The HPBW angles are in this case denoted by subscript 1 and 2 for the azimuth angle φ and elevation angle θ of a spherical coordinate system centered at the antenna. Here, φ_1 & φ_2 represents the 3 dB angles in the E-plane and θ_1 & θ_2 in the H-plane.

The backscattered power given a differential area ds' can, with the new alterations (as compared to Equation (2.8)) for a surface scatterer, be described as

$$dP_r(\boldsymbol{r}) = \frac{P_t G(\boldsymbol{r})^2 \lambda^2}{(4\pi)^3 \boldsymbol{r}^4} \frac{\mathrm{d}\sigma}{\mathrm{d}S} dS'. \tag{2.9}$$

[15]. Figure 2.4 depicts five incident rays and how they are or might be scattered depending on the roughness of the surface. To the left, 2.4 depicts specular reflection

(i.e mirror-like reflection) where the scattering angle θ_s is equal to the incident angle θ_i for all rays incident on the scattering surface. To the right diffuse reflection is depicted by scattered rays that spread in different directions due to the local surface variations of a rough surface.

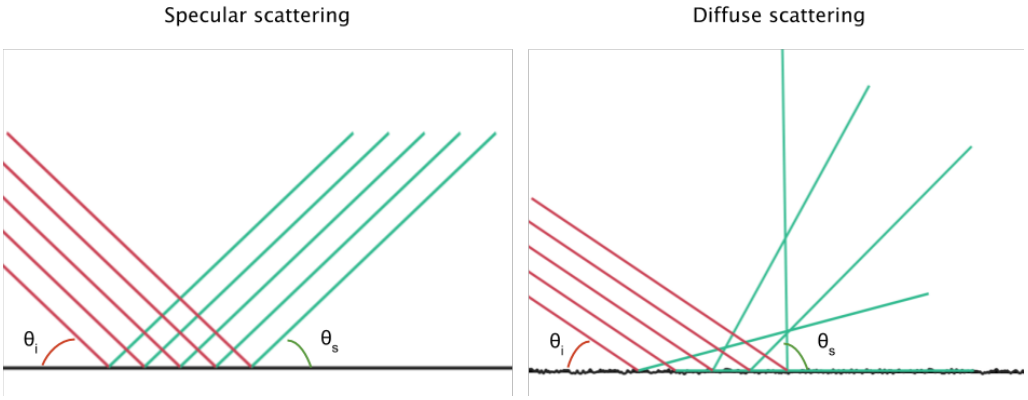


Figure 2.4: Example of specular and diffuse scattering

One way to model a perfectly diffuse surface is to use Lambert's cosine law [16]. Lambert's cosine law states that the normalized backscatter coefficient can be described as:

$$\sigma^o = \sigma_0^o \cos\theta_i \cos\theta_s = \{\text{Backscattering} : \theta_s = \theta_i\} = \sigma_0^o \cos^2\theta_i \tag{2.10}$$

To derive an estimate of the received power, the geometry must be known. Therefore, a simple geometry is derived based on the measuring platform. The radar sensor is mounted at an elevated position with the ground as a reference, let the radar's position be the origin. To use the sensor for surface classification, it is tilted with an angle δ towards the surface to illuminate an area, which is indicated by the yellow disk in Figure 2.5. The size of the disk is modeled using the HPBW of the radar. The measurement setup allows for manipulation of height h and angle δ which combined with the HPBW angles are enough to determine the remaining quantities given in the figure.

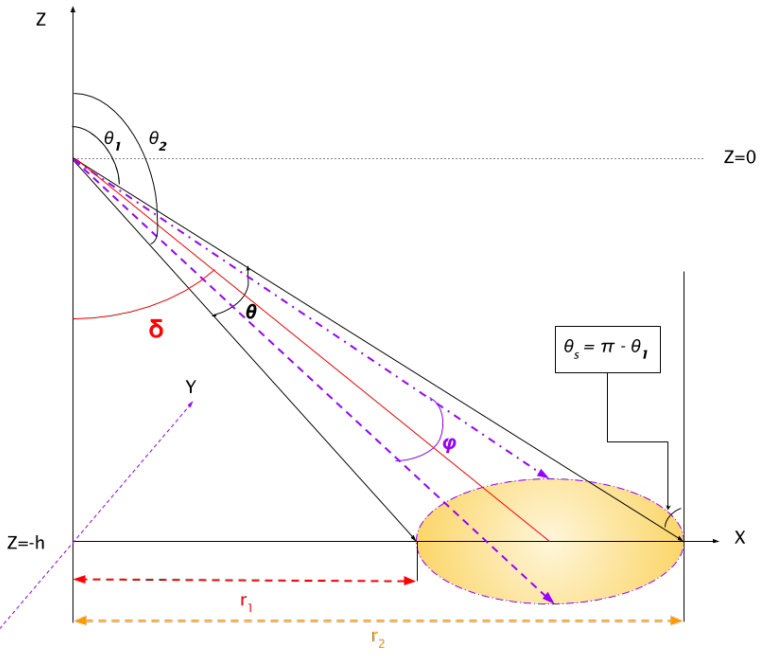


Figure 2.5: Fundamental geometry used to derive the scattered power from a surface

The full derivation for received power is given in appendix A and the result is

$$P_r = \frac{P_t \lambda^2 G_0^2 \sigma_0}{(4\pi)^3} \big(\varphi_2 - \varphi_1\big) h^2 \Big(\frac{1}{(r_1^2 + h^2)^2} - \frac{1}{(r_2^2 + h^2)^2} \Big). \tag{2.11}$$

2.2.3 Special Cases Based on Lambertian Scattering

Equation (2.11) can further be divided into two special cases.

Special case 1

For special case 1, let an omnidirectional radar illuminate an infinite plane, i.e. $r_1 = 0$, $r_2 \to \infty$, $\varphi_1 = 0$ and $\varphi_2 = 2\pi$.

By incorporation of the parameters in Equation (2.11), it can be derived that the received power is proportional to h^{-2};

$$P_r = \frac{P_t G_0^2 \lambda^2 \sigma_0^o}{8(4\pi)^3 h^2} \propto \frac{1}{h^2}. \tag{2.12}$$

Special case 2

For special case 2, consider a situation where only a finite portion of the plane that is illuminated, which subsequently means that r_2 is finite. The recieved power is

then given by

$$P_r = \frac{P_t \lambda^2 G_0^2 \sigma_o^o}{8(4\pi)^3 h^2} \Big(1 - \frac{1}{\Big[1 + (r_2/h)^2\Big]^2}\Big) \tag{2.13}$$

If $h \gg r_2 \implies P_r \propto 1/h^4$.

2.3 Modeling of Important Materials and Reflection from Surfaces

Reflection is, in the case of surface classification, a way to relate the incident radar waves to the reflected waves. Reflection is caused when the radar waves interact with dielectrics or other materials [14, Ch. 5]. A material can be characterised by the wave impedance [14, Sec 2.4] and it is related to electric permittivity.

From such material parameters it is possible to derive the ratio of reflected power in relation to incident power, where such derivations yield particularly simple expressions for an incident plane wave [17].

2.3.1 Dielectric Properties of Materials

The permittivity is the measure of the electric polarizability of a material [18]. A dielectric with high permittivity polarizes more easily in response to an electric field. The permittivity is denoted by ϵ [F/m]. The permittivity for a linear, homogeneous and isotropic material can be described by a scalar that is constant in space. For a dispersive material, the scalar, is a frequency-dependant complex number [14, pp. 4-25]. The real part $\epsilon'(\omega)$ of the permittivity corresponds to the ability to store electric energy. Furthermore, the imaginary part $\epsilon''(\omega)$, corresponds to the process of dissipating energy in the form of heat. The permittivity is usually expressed as two factors: the first is vacuum permittivity ϵ_0; and the second is the relative permittivity ϵ_r. Thus, we have the permittivity as

$$\epsilon = \epsilon_0 \epsilon_r = \epsilon'(\omega) - j\epsilon''(\omega). \tag{2.14}$$

From the dielectric properties, it is possible to define what is known as wave impedance. Wave impedance η is defined [14, Sec. 2.4] as the square root of the ratio between magnetic permeability μ and electric permittivity ϵ, i.e. $\eta = \sqrt{\mu/\epsilon}$. Here, we consider only non-magnetic mediums with $\mu = \mu_0 = 4\pi\ 10^{-7}$ H/m.

The relative permittivity can be used to distinguish one material from another and therefore only relative permittivity will be discussed in the following.

2.3.2 Propagation Constant

To describe a wave propagating in a medium, the propagation constant contributes features that describe attenuation and phase. Instead of describing attenuation and

phase as separate entities, they are merged into the propagation constant γ. This results in a complex number

$$\gamma = \alpha + j\beta, \tag{2.15}$$

as described in [14, p. 55]. Here, α is the attenuation constant and β is the phase constant. It was previously stated that permittivity is analogous to storing and dissipating electric energy. Therefore γ must be closely related to ϵ. The relation between γ and ϵ is given by (2.16) and (2.17) [19, Eq. 1.86].

$$\alpha = \frac{\omega}{\sqrt{2}}\sqrt{\epsilon'\mu_0\Big(\sqrt{1+(\epsilon''/\epsilon')^2}-1\Big)} \tag{2.16}$$

$$\beta = \frac{\omega}{\sqrt{2}}\sqrt{\epsilon'\mu_0\Big(\sqrt{1+(\epsilon''/\epsilon')^2}+1\Big)} \tag{2.17}$$

The wavelength of a time harmonic wave is, according to [14, Sec. 2.2], directly related the phase constant by

$$\lambda = \frac{2\pi}{\beta}$$

Another useful metric is the penetration depth. Penetration depth is defined as the distance l required for the amplitude of an electromagnetic wave to decay by a factor of e^{-1} or 37 % [14, p. 58]. Penetration depth l is calculated as $1/\alpha$.

2.3.3 Water and Its Reflective Characteristics

One substance of particular interest in the context of road surface classification is water in its solid and liquid phases. In this section, the reflective characteristics of water are modelled by permittivity and penetration depth. For liquid water, the Debye model can be used to model the permittivity [20].

Experiments carried out by P. Lunkenheimer et al. [20] show that, when the frequency is approximately 60 GHz and the temperatures 1-37 °C, the real part of the relative permittivity for water is approximately 10-12 and the imaginary part is roughly 20. The data presented by P. Lunkenheimer et al. is verified by the Debye model, as their data closely correlates with the predictions made by the model [20]. The Debye model is defined as

$$\epsilon(\omega) = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1 + j\omega\tau_0} = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1+(\omega\tau_0)^2} - j\frac{(\epsilon_s - \epsilon_\infty)\omega\tau_0}{1+(\omega\tau_0)^2}, \tag{2.18}$$

where ϵ_∞ denotes the optical limit, ϵ_s denotes the static permitivity and τ_0 the relaxation time for the molecule [21]. The relaxation time for liquid water is approximated as

$$\tau_0 = \frac{4\pi v L^3}{kT} \tag{2.19}$$

where v denotes the viscosity, L the molecular radius, k Boltzmann's constant and T the temperature in Kelvin [21, p. 92].

The Debye model has been plotted in Figure 2.6 using the following values [22]:

$$\epsilon_s = 87.9 \qquad k = 1.38 \cdot 10^{-23}\,\mathrm{J/K} \qquad \alpha = 2.75\ \mathring{\mathrm{A}}$$
$$\epsilon_\infty = 4.9 \qquad T = 290 \pm 15^\circ\,\mathrm{K} \qquad v = 1.5182 \cdot 10^{-3}\mathrm{Pa\,s}$$

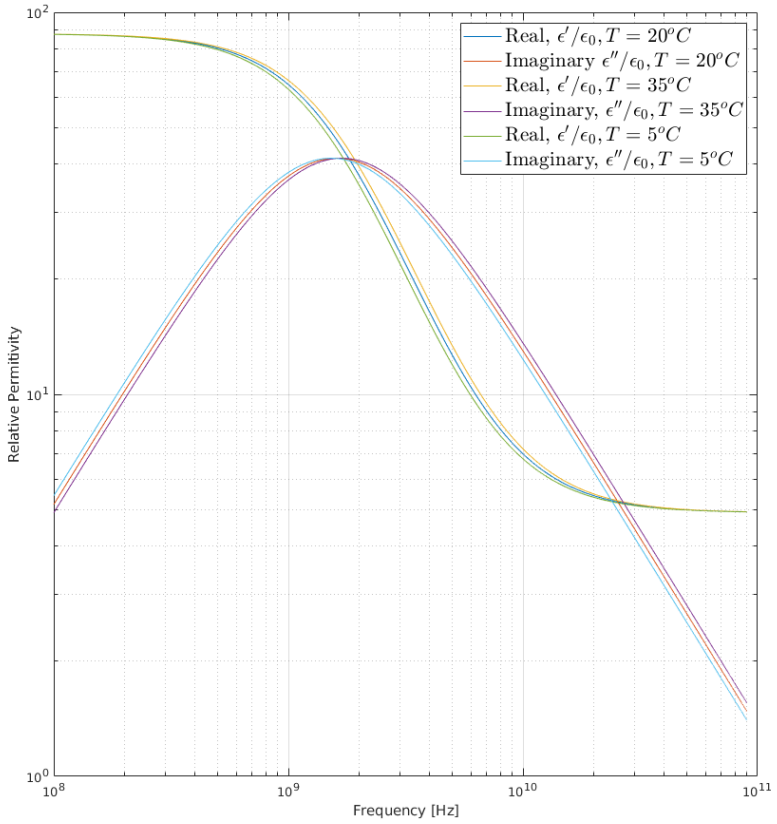


Figure 2.6: The real and imaginary part of the realtive permittivity as a function of frequency, derived using the Debye model

In Figure 2.6 the real part of the relative permittivity has a value of roughly 4 and the imaginary part a value of approximately 2. This differs from the values by Lunkenheimer et. al due to a slightly different value of τ_0. Since the Debye model is not linear, small alterations of τ_0 can cause large variations in the permittivity.

The real part of the relative permittivity acquired by [20] is further in good agreement with the measured relative dielectric constant of water presented by Acconeer [11], which is $\epsilon_r' = 11.1$ at 60 GHz.

Using $\epsilon_r' = 11.1$ and $\epsilon_r'' = 20$, the penetration depth of water is $l = 0.2\,\mathrm{mm}$ according to Equation (2.16).

2.3.4 Other Materials of Interest near 60 GHz

Table 2.1 presents a summary of the relative permittivity for the specific materials presented in this thesis. It is of interest to present an approximate relative permittivity based on values that are presented for nearby frequencies. The penetration depth l has been calculated as $1/\alpha$ where α is calculated using Equation (2.16).

Material	ϵ'	ϵ''	**Freq. [GHz]**	l **[m]**
Asphalt [9]	6	0	76	-
Ice [9]	3.05	0.004	76	0.35
Dry snow[21]	1-3	-	Above 1 MHz	-
Water [20]	10-12	20	60	$2 \cdot 10^{-4}$

Table 2.1: Summary of relative permittivities and penetration depths for relevant materials

The real part of the relative permittivity for snow has been shown to be largely frequency independent above 1 MHz [21]. Given that this is not the case for ice nor water, the reliability of this information can be discussed. However, the relative permittivity is dependent on the density of the snow. The values in Table 2.1 have been measured at 9 GHz and densities ranging from $0.1 - 1\,\mathrm{g\ cm^{-3}}$ for dry snow. Data for the imaginary part of the relative permittivity has not been found. Neither has the relative permittivity for wet snow, meaning snow close to 0°C with a percentage of liquid water present, been found for the relevant frequencies [21]. However, research done at 3-18 and 37 GHz show that the value of both the real and imaginary part of the relative permittivity increases with higher liquid water content. Data from the research done by Hallikainen et al.[23], indicates that wet snow closely follows the dispersion behavior of water.

2.3.5 Reflection from Surfaces

Rayleigh Criterion

Whether a surface is considered rough or smooth is a relative concept [24]. The surface's roughness is closely related to the wavelength λ of the transmitted wave and the incident angle θ. One criterion that can be used to determine whether the surface is rough or not is Rayleigh's criterion [24]. Rayleigh's criterion is formulated as

$$h' > \frac{\lambda}{8\cos\theta}, \tag{2.20}$$

where h' represents the mean height variation of the material. The surface is considered rough if the inequality in Equation (2.20) is evaluated to true. If the inequality is evaluated to false the surface is considered smooth.

Dielectric Half-Space with Planar Interface

Let the first assumption be that all materials are smooth, linear, homogeneous, and isotropic. This assumption allows for a deterministic derivation that describes electromagnetic wave interaction with materials. In the context of material classification, it is of interest to study how a material affects the incident electromagnetic wave. One way to describe this is by studying the reflected wave in relation to the incident wave. The reflection coefficient Γ is such a measure that relates the reflected wave in terms of the incident wave [14, Ch. 5]. Furthermore the reflected power p_s is directly related to Γ and the transmitted power p_t as [17].

$$p_s = |\Gamma|^2 p_t \tag{2.21}$$

For a non-magnetic medium, wave impedance reduces to

$$\eta = \frac{\eta_0}{\sqrt{\epsilon_r}}, \tag{2.22}$$

where η_0 represents the wave impedance of vacuum.

For the simple case of an infinitely thick dielectric, an incident wave can either be reflected or transmitted. The transmitted wave is expected to never be reflected since the material is infinitely thick. This special case is described by the Fresnel coefficients [14, Sec. 7.3] and for normal incidence the are given by

$$\begin{aligned} \Gamma_1 &= \frac{\eta_1 - \eta_a}{\eta_1 + \eta_a}, \\ \tau_1 = 1 + \Gamma_1 &= \frac{2\eta_1}{\eta_1 + \eta_a} \end{aligned} \tag{2.23}$$

where Γ_1 represents the reflection coefficient if the incident wave travels from left to right as portrayed by E_1^+ in figure 2.7.

Dielectric Slab with Planar Interfaces

According to [14, Sec. 5.4], the reflection coefficient Γ for a single dielectric slab is calculated as a function of the material's wave impedance, propagation constant, and thickness. The full derivation of Equation (2.24) is left out of the report but is fully derived in [14, Sec. 5.1-5.4], resulting in

$$\Gamma_1 = \frac{\rho_1 + \rho_2 e^{-j\gamma_1 l_1}}{1 + \rho_1 \rho_2 e^{-j\gamma_1 l_1}}. \tag{2.24}$$

To better understand Equation (2.24), Figure 2.7 depicts a single dielectric slab with elementary reflection coefficients ρ_2, ρ_1 and the corresponding transmission coefficients τ_2, τ_1. η_a, η_b and η_1 represents the wave impedance's of the mediums. Γ_1, Γ_2, Γ_1' and Γ_2' represent the reflection coefficients from different directions.

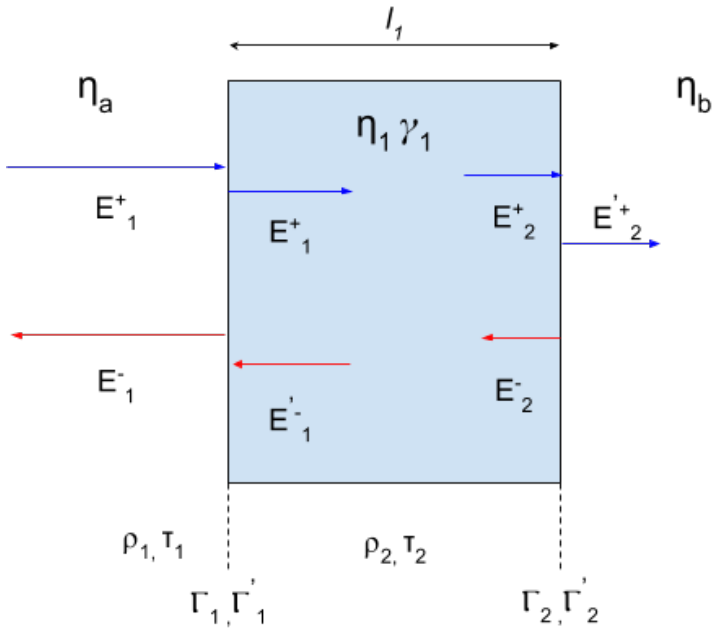


Figure 2.7: Incident field and reflecting electric fields

The aforementioned reflection and transmission parameters are calculated below [14, p. 164]. The exception is Γ since this is the factor that is of interest to predict.

$$\eta_a = \sqrt{\frac{\mu_0}{\epsilon_0 \epsilon_a}} \qquad \rho_1 = \frac{\eta_1 - \eta_a}{\eta_1 + \eta_a}$$
$$\eta_1 = \sqrt{\frac{\mu_0}{\epsilon_0 \epsilon_1}} \qquad \rho_2 = \frac{\eta_b - \eta_1}{\eta_b + \eta_1}$$
$$\eta_b = \sqrt{\frac{\mu_0}{\epsilon_0 \epsilon_b}} \qquad \tau_1 = 1 + \rho_1$$
$$\tau_2 = 1 + \rho_2 \tag{2.25}$$

The reason for introducing reflection as a part of the propagation section can now be better understood. Wave impedance for many materials has already been researched for a wide range of frequencies. The data presented in Section 2.3.4 allows for predictions regarding the amount of reflected power before any measurements have taken place. Previously, the model only considered geometry and material structure but not the electromagnetic properties of the material.

One possible way to simplify the reflection of the dielectric slab would be to investigate the penetration depth of the material [14, pp. 58]. If the penetration depth of the material $l \ll l_1$, then a good approximation will be to set $\rho_2 = 0$ because the magnitude of the incident electric field E_{2+} can be assumed to be close to zero. The special case for $\rho_2 = 0$ then reduces Equation (2.24) to

$$\Gamma_1 = \rho_1. \tag{2.26}$$

which is equivalent to that of the dielectric half space.

Rough Surfaces

As presented in Section 2.2.2, one way to model a rough surface is to do so using Lambert's cosine law. In reality, most surfaces can not be described as perfectly smooth or perfectly rough. A surface may be possible to model as locally smooth but when observed from a distance the surface may look rough. For instance, a water surface may look smooth when observed closely, but when observed from a distance may look wavy. Other imperfections found in real-world materials are inhomogeneities and pollution by other materials. For instance, in the case of snow, ice crystals are randomly distributed, air cavities may occur, liquid water may be present and other molecules may pollute. These impurities make the problem much harder to model. The presented methods for reflection from surfaces serve the purpose of being informative rather than a good model for the application considered in this report. The imperfections that are hard to model are instead left for the machine learning algorithm to manage.

3

Data Acquisition

This chapter describes the platform and method used for acquiring the data from the target surfaces. The surfaces, sites, and measurement campaigns are also presented.

3.1 Platform

Data from different surfaces are needed in quantity and quality, for both validating the radar equipment and the machine learning objectives. High-quality data is unbiased to the observed surfaces and sufficiently free from noise. For this, a data acquisition platform capable of collecting volumes of reproducible data in motion is constructed.

3.1.1 Remote-Controlled Car

To achieve constant velocity while collecting data, the car's throttle is controlled by a feedback loop. An encoder is placed on the drivetrain of the car to measure the speed. An Arduino Uno is used as a controller and is configured as a proportional-integral regulator. The Arduino communicates with the electronic speed controller (ESC) through pulse-width modulation (PWM). The ESC feeds the required voltage to the direct current motor concerning the received PWM signal. In total, the car is controlled by two inputs: throttle, and steering. The throttle is controlled by the feedback loop and steering is a manual input from the external remote.

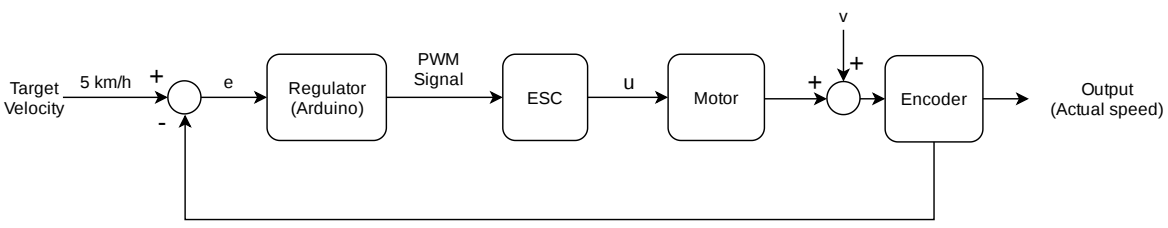


Figure 3.1: Feedback loop of car's throttle

In Figure 3.1, e is the error signal which is the difference between the target and the output velocity. The voltage supplied to the motor from the ESC is denoted u and v is the interference that arises from resisting forces on the car.

Large areas with uniform surfaces are difficult to come by, which entails that it is necessary to come up with a driving path to be able to measure surfaces in limited areas. One alternative is to drive the car in a circle. A drawback of driving in circles is that there will be a speed differential between the inside and outside of the illuminated area. This differential can be reduced simply by increasing the radius of the turn. However, then a larger area to measure is needed. Another alternative is to measure the surface driving straight and then turning back and forth, measuring the surface from both directions multiple times. The advantage of this strategy is that it avoids measuring while the car is turning and is more flexible to obstructions on different measurement sites. An essential part of this strategy is to drive the car at a low speed, so that the straight driving path does not become too long. Therefore, the constant speed of the car was set to 5 km/h.

Vibrations can arise both from the drivetrain and the motor, as well as from the surface structure itself. The car is equipped with four suspension springs, one for each wheel. Due to the additional weight from the added equipment on the car, the springs need more tension to not bottom out from the weight of the platform. This is solved by compressing the springs by putting spacers on the suspension body.

The suspension could be beneficial for measuring rougher surfaces. All measurements performed in this project are performed on relatively smooth surfaces. It can be assumed that the vibrations are of negligible amplitude and will not substantially affect the measurement.

3.1.2 Radar Installation

For the radar installation, a frame is 3D-printed and attached to the car, see Figure 3.2. On this platform, the radar is attached to an angled mount so that $\delta = 30°$ and $h = 12\,\text{cm}$ as illustrated in Figure 3.3.

Figure 3.2: The moving platform

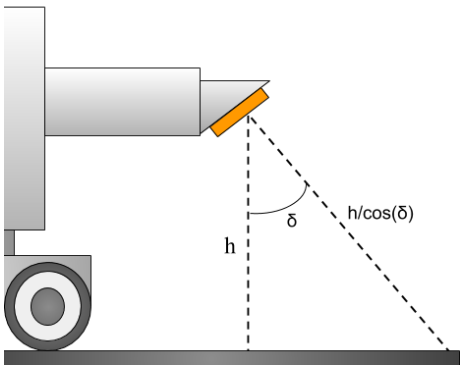


Figure 3.3: Geometry of radar setup

The radar is connected to a Raspberry Pi 3. The Raspberry Pi hosts a Wi-Fi network through which it is connected to a computer. The separate computer then starts and runs the program for collecting data. This allows the data to be collected by the RC car and simultaneously be stored on the computer.

3.1.3 Radar Settings

To configure the settings of the radar to suit the application of surface recognition, some parameters are of particular concern. Settings with greater importance include pulse duration, sampling frequency, signal averaging, distance interval, and which type of service to choose. Other settings are left at default values set by the manufacturer.

First of all, it is necessary to decide what type of service to use. There are four options for the Acconeer radar; power bins, envelope, IQ, and sparse [11]. The IQ service provides data in the closest form to raw data with both phase and amplitude information. When analyzing the stationary measurements, the IQ service is used to inspect the phase information as well as the amplitude information. Since the envelope service does not collect phase information, the envelope service can be used with a higher sampling frequency than the IQ service, due to the lower data bandwidth. Seeing that there is no use of phase information in the feature extraction process, and since a high update rate is desirable, the envelope service is selected.

The pulse duration, which is determined by the profile setting, is a trade-off between depth resolution and SNR which is described in Section 2.1. It is highly undesirable to have direct leakage affecting the measurement. Direct leakage is when a portion of the transmitted radar pulse travels directly to the receiving antenna without reflecting off of an object. In Acconeer's documentation [11], Table 4 displays direct leakages for different profile settings. Since the radar is located 12 cm above the ground, all profiles except profile 1 will affect the measurement data. Hence, profile 1 is selected.

The hardware accelerated average samples (HWAAS) controls the number of pulses used to build one data point [11]. For instance, if HWAAS is set to 10, it means that every data frame is built from 10 data frames averaged together.

The range interval for which the radar listens to the received echo is set to the interval from 10 cm to 42 cm. The upper limit is set as it was seen from previous experiments that the signal received after 42 cm is practically indistinguishable between surfaces. The lower limit was set due to the Acconeer-recommended lowest range start for profile 1 to avoid direct leakage [11].

The sampling frequency, also called the update rate, determines at which frequency the system produces new data frames. A greater sampling frequency is desirable since the system gathers more data points and thus more information about the surface. However, due to limitations of the system, the maximum sampling rate is determined by the signal averaging, profile, distance interval, and service settings. [11]. After testing several different frequencies, the sampling frequency was set to 320 Hz since this was the highest frequency of which the data collection resulted in an insignificant amount of missed frames.

Service	Update rate [Hz]	HWAAS	Profile	Range interval [cm]
Envelope	320	10	1	10-42

Table 3.1: Table of radar settings configuration for the moving platform

3.2 Campaigns

In this section, each campaign and stage of data acquisition is described through its site and surface conditions. All data from the different campaigns can be found on the project's repository on GitHub.[1]

3.2.1 Description of Measurement Sites

Pictures of each site can be found in appendix B.

Site nr.	Site type	Surface type	Estimated variations [mm]	Comment
1	Parking lot	Asphalt	1-3	Smooth asphalt with few variations.
2	Car road	Asphalt	1	The smoothest asphalt surface measured in this project.
3	Loading lot	Asphalt	2-6	Quite rough asphalt with relatively large variations.
4	Pedestrian road	Asphalt	1-3	Road with quite smooth surface.
5	Pedestrian road	Asphalt	2-6	Road with quite rough surface.
6	Pedestrian road	Asphalt	3-7	This is the roughest asphalt surface measured.
7	Indoor skiing facility	Snow/Ice	10	The snow/ice surface on this site is further explained in the next section.
8	Kitchen counter top	Metal	< 0.1	Smooth metal surface with few variations.

Table 3.2: Summary of measurement sites

[1] https://github.com/maxijohansson/kandidatarbete-eenx15-20-03

3.2.2 Surface Conditions

The three surface conditions examined in this thesis are described here.

Dry: The surface is in a dry state when measured on, with little dirt or other disturbances in the way.

Wet: The data is acquired from a surface with a thin water layer.

Snow: The data is acquired at an indoor skiing facility where the ground is covered in loose "snow". This snow is created by using a large machine to crush ice. This is therefore not natural snow, but rather ice of a similar structure.

3.2.3 Summary of Campaigns

During campaign 1, the goal is to collect stationary data which is used to test the radar parameters and the measurement setup. Noise and direct leakage data are examined by pointing the radar towards the sky and measuring unobstructed space. A stationary platform is then used to collect data for a metal surface, dry asphalt and wet asphalt for comparison. The metal surface data is also used to examine the influence of different angles and heights of the radar.

During campaign 2, data is acquired while moving for dry and wet asphalt from six different locations using the platform described in Section 3.1. Data is first collected at the site while the asphalt is dry and then with a thin layer of water on the same surface. The campaign aims to build a data set for the machine learning tasks

During campaign 3, data is collected on snow. This data is used to further expand the data set.

	Site	**Surface conditions**	**Drive path and velocity**
Campaign 1	1 & 8	dry, wet	Stationary measurments
Campaign 2	1-6	dry, wet	Straight line at constant velocity of 1.4 m/s
Campaign 3	7	snow	Straight line at constant velocity of 1.4 m/s

Table 3.3: Summary of campaigns

4

Data Pipeline

This chapter describes the process that transforms raw data from the radar equipment to machine learning-ready input. Preprocessing refers to the operations performed on the data in order to clean and reduce it. Feature extraction is the process of creating representations of the data that machine learning algorithms can train and predict with. Three methods for feature extraction will be presented.

4.1 Data Representation

The process of collecting data with the envelope service essentially involves measuring the reflected radar signal in two different time scales. The slow-time scale is what produces each frame, or raw data point. Each frame, in turn, consists of a number of samples in the fast-time scale at small time delays. Time delays in this sense are synonymous with distance steps and can be seen as each frame's observed variables.

In this chapter, a collected data matrix $\boldsymbol{D}$ with M frames can be described with the process $\boldsymbol{y}(n)$ where n is the slow-time index for each frame. Each frame consists of T fast-time samples.

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{y}(0)^\mathsf{T} \\ \boldsymbol{y}(1)^\mathsf{T} \\ \vdots \\ \boldsymbol{y}(M)^\mathsf{T} \end{bmatrix}, \boldsymbol{y}(n) = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_T \end{bmatrix} \tag{4.1}$$

After preprocessing and feature extraction, feature vectors $\boldsymbol{f}(n)$ are produced corresponding to slow-time frames n. The frequency at which these feature vectors can be produced constitutes the classification rate and depends on parameters during feature extraction. The classification rate is the rate at which new predictions can be produced by the machine learning model.

4.2 Analysis

Before transforming data and creating features, analysis can help determine if the data is of high quality and what information can be extracted from the data. This can involve visual tools such as graphing the acquired data or more quantitative statistical methods.

A simple way to analyze a dataset is to calculate the mean of each variable over the observations. If $\boldsymbol{y}^s(n)$ describes each data point when measuring M slow-time frames from a surface s, then the mean vector is

$$\hat{\boldsymbol{\mu}}^s = \frac{1}{M}\sum_{n=1}^{M}\boldsymbol{y}^s(n). \tag{4.2}$$

A set of observations can also be characterized with the covariance matrix:

$$\widehat{\boldsymbol{C}}^s = \frac{1}{M-1}\sum_{n=1}^{M}(\boldsymbol{y}^s(n)-\hat{\boldsymbol{\mu}}^s)(\boldsymbol{y}^s(n)-\hat{\boldsymbol{\mu}}^s)^{\mathsf{T}} \tag{4.3}$$

The distance between two surfaces' mean vectors along with each class's average standard deviation can indicate how linearly separable they are with the observed variables. The distance between classes s_i and s_j is the Euclidean distance

$$\hat{D}_{ij} = ||\hat{\boldsymbol{\mu}}^{s_i} - \hat{\boldsymbol{\mu}}^{s_j}||. \tag{4.4}$$

The diagonal of the covariance matrix contains each variable's individual variance, and the standard deviation of a variable is the square root of its variance. Therefore, the average standard deviation can be derived as

$$\hat{\sigma}^s = \frac{1}{M}\sum_{i=1}^{M}\sqrt{\widehat{C}^s_{ii}}. \tag{4.5}$$

4.3 Preprocessing

Preprocessing is an important step before feature extraction. The raw data retrieved from the radar unit often contains redundant information along with noise. Preprocessing is therefore about cleaning and reducing the amount of data needed to represent the desired information.

The radar by default has a distance resolution in fast-time of approximately 0.48 mm. This resolution may not be necessary with certain methods of feature extraction and can potentially produce a large number of features, thereby increasing the need for computation and memory resources. Downsampling by a factor D can be expressed as

$$y'_t(n) = y_{tD}(n) \tag{4.6}$$

Where $y_t(n)$ is the value of index t in frame $\boldsymbol{y}(n)$. The amount of fast-time samples after downsampling is therefore $n_{ds} = T/D$. Choosing the amount of downsampling should be done as to not lose the information needed for classifying the surfaces. If the signal to be downsampled contains periodic characteristics, such as peaks at even intervals, then one could consider the Nyquist frequency, which in this case is half of the downsampling frequency. Any periodic information above the Nyquist frequency would be subject to aliasing. If the signal does not contain periodic components, then different downsampling factors can be tested to see how the separation or classification accuracy is affected.

4.4 Feature Extraction

When a data point is used as input to a machine learning algorithm, it is represented by its feature vector. Creating a feature vector $\boldsymbol{f}(n)$ involves translating the observed variables to features through feature extraction. The variables in $\boldsymbol{f}(n)$ are known as features.

4.4.1 Moving Average of Amplitude

This feature extraction method involves characterizing data by its amplitude at fast-time intervals. By averaging a number of slow-time frames, noise and variance can be suppressed at the cost of latency in classification. The moving average feature vector over N frames before slow-time index n is

$$\boldsymbol{f}(n) = \frac{1}{N} \sum_{i=0}^{N-1} \boldsymbol{y}(n-i). \tag{4.7}$$

The number of features is therefore equal to the number of input variables.

4.4.2 Periodic Average of Amplitude

The moving average feature extraction method results in one feature vector for every slow-time frame, with exception for the first $N-1$ frames. Each feature vector contains information about the most recent feature vectors in slow-time. Another method of feature extraction would be downsampling the moving average by a factor N. This would mean that every feature vector contains information about frames not present in any other feature vector.

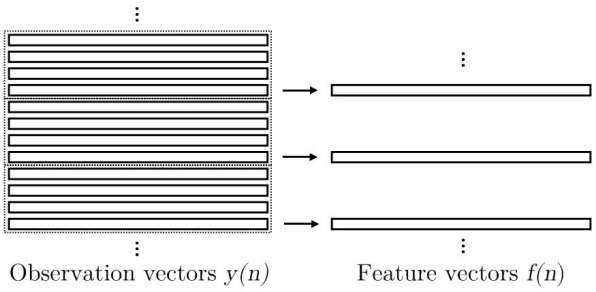


Figure 4.1: The periodic average feature extraction method functions like the moving average, but one feature vector is created every N frames. The figure illustrates this with $N = 3$.

4.4.3 Autocovariance

The autocovariance function ACF is a special case of the covariance function [25] used for analyzing time series. In probability, covariance is a way to measure how two variables change together. The covariance function is defined for two stochastic variables, X and Y, as:

$$\text{cov}(X, Y) = E[(X - E[X])(Y - E[Y])], \tag{4.8}$$

where E denotes the expected value of a random variable.

The ACF is a special case of the covariance function, where $X = Y$, that gives the covariance of the process itself but at different pairs of time points [26]. If the data from the radar measurements are treated as stochastic, then a method for investigating time-domain dependencies is the ACF. A stochastic process is said to be wide-sense stationary (WSS) if the auto-covariance does not vary with respect to time, the mean of the process is constant, and that the variance of the process is finite. Usually, for a WSS stochastic process x_t, the auto-covariance of x_t is defined as

$$r_y(k) = \text{cov}[y_t, y_{t-k}] = E[(y_t - \mu_y)(y_{t-k} - \mu_y)^\mathsf{T}], \tag{4.9}$$

where $(.)^\mathsf{T}$ denotes the transpose and μ_y is the mean value of y. The variable k denotes the time-lag in discrete time indices, which defines how much the process is shifted with respect to itself.

In the case of only a single realization of a process y_t, an estimation of the ACF is usually used [26]. One reason for calculating an estimation of the ACF is that the population mean is usually unknown and therefore an estimation of the mean for the samples is used. Normally, there are two ways to estimate the ACF, a biased and an unbiased estimate [26]. Consider the unbiased ACF estimate

$$\hat{r}_y(k) = \frac{1}{N-k} \sum_{n=k+1}^{N} (y_t - \hat{\mu}_y)(y_{t-k} - \hat{\mu}_y)^\mathsf{T}, \tag{4.10}$$

where $\hat{\mu}_y$ is the estimated mean value of process y_t and N denotes the number of included data points in the calculation. This estimation of autocovariance is quite

accurate for lower orders of time-lags; a rule of thumb is to limit the calculations to $N/4$ time-lags [26]. Equation (4.10) is then slightly modified to better describe the use of the data representation introduced earlier in the chapter and the format of features. For a slow-time frame n, an autocovariance vector for lag k over the $N-k$ most recent frames can be formed as

$$\hat{\boldsymbol{r}}_{\boldsymbol{y}}(n,k) = \frac{1}{N-k} \sum_{i=0}^{N-k-1} (\boldsymbol{y}(n-i) - \hat{\boldsymbol{\mu}}_y)(\boldsymbol{y}(n-k-i) - \hat{\boldsymbol{\mu}}_y)^{\mathsf{T}}. \tag{4.11}$$

Where $\hat{\boldsymbol{\mu}}_y$ is the estimated mean of the N most recent frames of the process, namely:

$$\hat{\boldsymbol{\mu}}_y(n) = \frac{1}{N} \sum_{i=0}^{N-1} \boldsymbol{y}(n-i) \tag{4.12}$$

To construct the feature vector corresponding to a slow-time frame n, (4.11) is applied to the frame for each lag $k : 0 \leq k \leq q$. This is done for every N:th frame, much like as in Figure 4.1, so that the feature vectors are all based on different frames.

$$\boldsymbol{f}(n) = [\hat{\boldsymbol{r}}_{\boldsymbol{y}}(n,0), \hat{\boldsymbol{r}}_{\boldsymbol{y}}(n,1), ..., \hat{\boldsymbol{r}}_{\boldsymbol{y}}(n,q)]^{\mathsf{T}} \tag{4.13}$$

Each of the vectors $\hat{\boldsymbol{r}}_{\boldsymbol{y}}$ in $\boldsymbol{f}$ are concatenated so that $\boldsymbol{f}$ is a vector of length Tq where T is the number of fast-time samples and q is the number of lags used.

5

Machine Learning Algorithms

In this chapter, the machine learning algorithms used in this project are presented. Also, the process for developing and evaluating a machine learning model is described.

5.1 Overview of Algorithms

This project focuses on three common classification algorithms: support vector machine (SVM), random forest (RF) and neural network (NN). Emphasis is placed on the Scikit-learn library for Python [27], since it is used for implementation.

5.1.1 Support Vector Machine

Support vector machines are a commonly used classifier that can be either linear or nonlinear [28, pp. 153-162]. The goal of a SVM is to find a separating hyperplane that maximizes the distance between itself and the nearest training instances, where training instances in different classes are located on different sides of the hyperplane. This concept is illustrated in Figure 5.1.

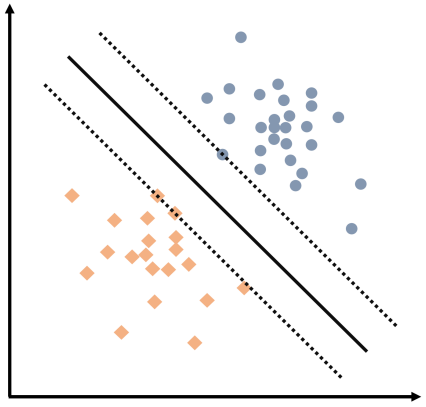


Figure 5.1: A support vector machine algorithm aims to find a decision boundary that maximizes the distance between itself and a number of observations in each class.

In many cases, data is not linearly separable such as in Figure 5.1. For such situations, the model can benefit from a kernel trick [28, pp. 157-158]. A kernel trick achieves the effect of mapping the data to a higher-dimensional feature space, without actually having to calculate transformations on the input vectors [29]. A common kernel besides the linear one, which produces linear decision boundaries, is the radial basis function (RBF). The math behind the SVM algorithm and its kernels are beyond the scope of this project, but some important parameters can be explained intuitively. The `gamma` parameter in Scikit-learn's implementation is part of the kernel function and sets the influence that each data point has on the algorithm. A high `gamma` will result in decision boundaries highly affected by variance in the data set, in other words increasing the risk of overfitting. A low `gamma` may not be able to capture the complexity in the data. The `C` parameter can be used to find a trade-off between high classification accuracy during training and a simple, high-margin decision boundary [30]. The margin can be thought of as the width of the separating zone in Figure 5.1.

5.1.2 Random Forest

Random forests are built up by relatively simple and intuitive decision trees. A decision tree is formed by asking a series of questions, typically whether a class is more likely to be on one side or another of a one-dimensional decision boundary. These questions and answers are structured in a tree, forming paths from the root node (first question) down to the the leaves (the predictions). In Scikit-learn, the classification and regression trees (CART) algorithm is used for this task, which only results in binary trees with true or false questions [28, pp. 175-179]. One of the most important hyperparameters for a decision tree is its maximum depth, which relates to the maximum number of allowed nodes on a path from the root node to a leaf. Decreasing the `max_depth` parameter restrains the decision tree, reducing the risk for overfitting [28, pp. 175-179].

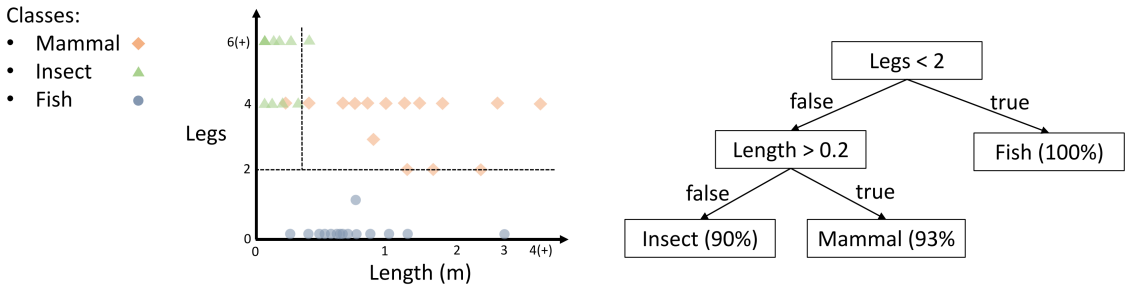


Figure 5.2: A simplified example of a decision tree for classifying classes of animals

A random forest uses multiple decision tree classifiers applied to samples of the training data [31]. This improves accuracy and reduces overfitting from a single decision tree [28]. The Scikit-learn library implements random forests by ensembling its decision tree classifiers [31]. The number of single decision trees is set by the hyperparameter `n_estimators`.

The main hyperparameters to be optimized for random forests are `n_estimators` and `max_depth`. Other parameters can be found on Scikit-learn's reference page [31].

5.1.3 Neural Network

Neural networks are inspired and built upon the basic understanding and principles of the human brain. The nodes in the network roughly compare to neurons and the connections between nodes correspond to synapses in the brain [32]. A feed-forward neural network contains one input, one output layer and a number of hidden layers. In Figure 5.3, a neural network with one hidden layer is demonstrated.

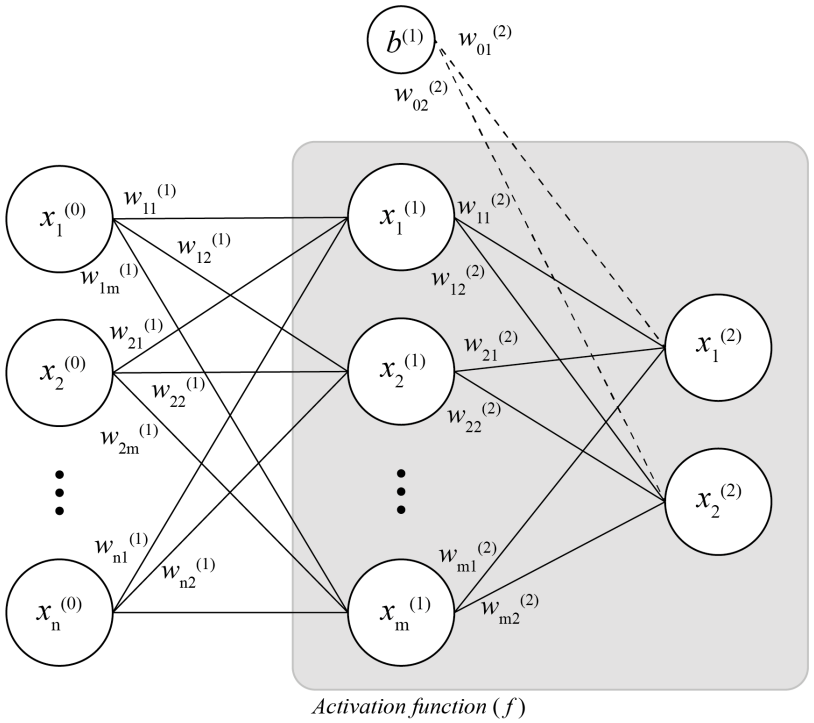


Figure 5.3: A feed-forward neural network consisting of one hidden layer and m hidden nodes.

Equation (5.1) illustrates how the first layer of the neural network can be derived by multiplying the input layer and its weights.

$$\boldsymbol{x}^{(1)} = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \vdots \\ x_m^{(1)} \end{bmatrix} = \begin{bmatrix} w_{11}^{(1)} & w_{21}^{(1)} & \dots & w_{n1}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & \dots & w_{n2}^{(1)} \\ \vdots & \vdots & \vdots & \dots \\ w_{1m}^{(1)} & w_{2m}^{(1)} & \dots & w_{nm}^{(1)} \end{bmatrix} \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \vdots \\ x_n^{(0)} \end{bmatrix} \tag{5.1}$$

In Equation (5.1), a simple matrix-vector product is carried out, which is a linear operation. For the neural network to be able to solve non-linear problems, it is necessary to use an activation function, called f, that adds a non-linear property to

the model [33]. Arguably the most commonly used activation function is the rectified linear unit (ReLU). Mathematically, it is expressed as $f(x_i^{(j)}) = \max(0, x_i^{(j)})$. Due to its simplicity, it is highly efficient to compute which is the reason why it is used in this project. Also, a bias term is added. The purpose of the bias term is to shift the output of the activation function to a higher or lower value.

The output of the model can now be expressed as

$$\boldsymbol{x}^{(2)} = \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \end{bmatrix} = \begin{bmatrix} w_{11}^{(2)} & w_{21}^{(2)} & \dots & w_{m1}^{(2)} \\ w_{12}^{(2)} & w_{22}^{(2)} & \dots & w_{m2}^{(2)} \end{bmatrix} \begin{bmatrix} f(x_1^{(1)}) \\ f(x_2^{(1)}) \\ \vdots \\ f(x_m^{(1)}) \end{bmatrix} + \begin{bmatrix} w_{01}^{(2)} \\ w_{02}^{(2)} \end{bmatrix} b. \tag{5.2}$$

Neural networks are commonly used for both supervised learning as well as unsupervised learning. In this project supervised learning will be used, meaning that the neural network model will learn to classify surfaces from labeled datasets through backpropagation. Backpropagation is an algorithm that is regularly used for training neural networks. The backpropagation algorithm is used to minimize the cost function with respect to weights and biases of the connections and nodes. This is accomplished by finding a local minimum by following the negative direction of the gradient of the cost function, hence this procedure is called gradient descent [34]. A more detailed description of backpropagation is provided in Ref. [35, Ch. 7].

A neural network is tremendously customizable and the output of the network will vary depending on its hyperparameters. Hyperparameters such as the number of hidden layers, hidden nodes, the value of the regularization parameter, and also the value of the learning rate are considered and tested. More information about the hyperparameters can be found in Ref. [27], [36], [37].

5.2 Model Development Process

The development of a model often involves training and evaluating the data on several different models in order to decide which variants are the most suitable. This can be achieved by varying the parameters and using the data in various ways [28, p. 75-80]. This can be a tedious task, but fortunately the Scikit-learn library for Python contains many tools for handling data and building and evaluating machine learning models [27], [28].

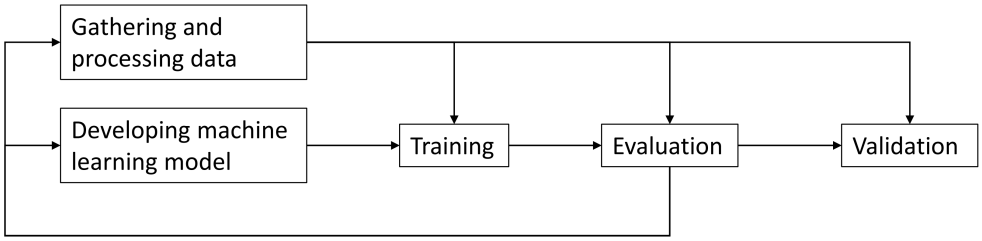


Figure 5.4: A general machine learning process

5.2.1 Partitioning of Data

For machine learning tasks, data is commonly partitioned for training, validation and testing. Validation data is used to find the optimal hyperparameters for training the model. Testing data is left unseen to the model until its final evaluation. By setting aside testing data that never has an impact on the creation or tuning of the model, final evaluation can be performed objectively and overfitting can be detected.

The first step of partitioning data is to set aside data for the testing set. For this study, since two of the surfaces (dry asphalt and wet asphalt) are acquired in equal amounts from different sites, an equal amount of testing data is taken from each site. The snow data is partitioned arbitrarily since it is all collected from the same location. The training set, consisting of 80% of the data, is then in turn partitioned to produce validation data used for tuning the hyperparameters.

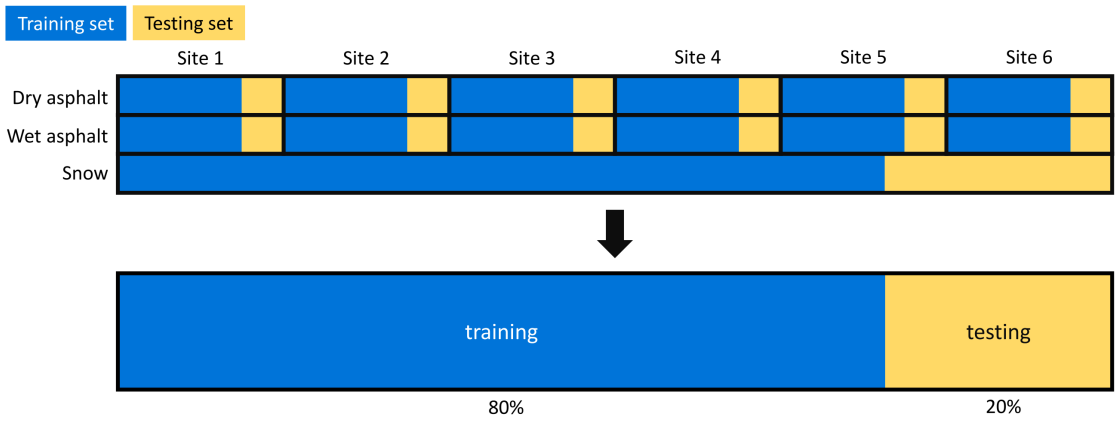


Figure 5.5: 20% of the data is set aside for the testing set. These 20% are chosen so that the sites for the site-specific classes are represented equally in both training and testing.

Many algorithms assume that the data is centered around zero and that the values are near-zero [27]. Therefore, before training and testing, the data is scaled to the interval $[-1, 1]$. This is done with Scikit-learn's `MinMaxScaler`.

5.2.2 Hyperparameter Tuning

The hyperparameters in machine learning algorithms are parameters that remain constant during a training session but can be varied to improve the performance of the trained model. To find an optimized combination of these parameters, grid search with cross-validation is commonly used. Grid search involves training with all the possible relevant combinations of parameters and evaluating the model after each parameter combination. The combination that yields the highest accuracy upon evaluation is then chosen for further usage. The evaluation during a grid search can be performed by k-fold cross-validation, where the data is iteratively trained and evaluated on different partitions of validation data, as described in Figure 5.6.

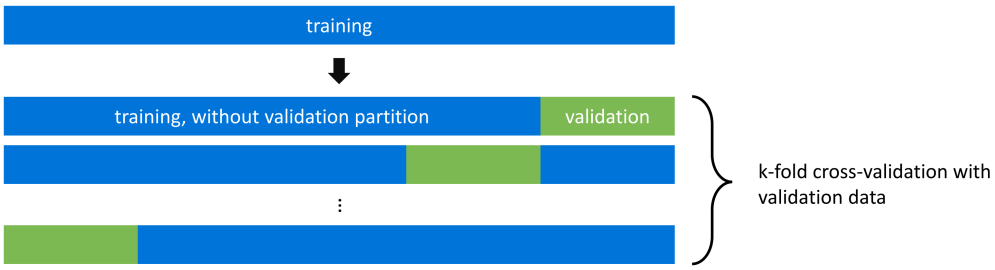


Figure 5.6: k-fold cross-validation trains with $\frac{k-1}{k}$ of the training data and validates on the remaining $\frac{1}{k}$. An average of the performance of each of the k train-validate cycles is produced as the total performance for each parameter combination in the grid search.

Scikit-learn provides the class `GridSearchCV`, which performs grid search and finds the optimal parameter combination with a given space of parameters. `GridSearchCV` also implements stratified k-fold cross-validation as default. Stratified means that the algorithm tries to represent each class equally when partitioning the validation data [27].

5.2.3 Model Evaluation

Evaluating the model may seem straight-forward, but can be done in many ways. The most obvious is perhaps to evaluate the model on the testing set that has been set aside earlier. Since both training and hyperparameter optimization is done with the training set, the testing set has had no influence on the model. A poor performance on this unseen testing set would indicate overfitting, meaning that the model or its hyperparameters are specifically tailored for the training set. The drawback of this method is that it only trains and tests on portions of the data.

Figure 5.7: Evaluating the model on the data that was set aside before the construction of the model will give a result that reveals eventual overfitting, but is limiting due to the relatively small size of the testing set.

Another evaluation method is k-fold cross-validation with all available data. This would allow training and testing using all of the data in separate iterations of partitioning. Unlike evaluating with the virgin testing set, this method is likely to be oblivious to overfitting since it tests with data that has influenced the model.

Figure 5.8: k-fold cross-validation will allow testing on, in total, a large number of data points, but it would be somewhat oblivious to overfitting. The order of the data points is randomized.

Since this experiment collects data from different sites, it may also be interesting to evaluate how well the model generalizes between sites. The two methods described so far include data from each site in both the training and testing partitions. It is therefore likely that data points in the testing set have very similar counterparts in the training set. Maybe the data point taken directly before or after a given test point is present in the training set, meaning that the surface structure is probably very similar. Therefore, evaluating with a testing set that is sure to not be too similar to any training data is important. To achieve this, the testing data can be partitioned based on site, as in Figure 5.9. The class snow, where all data is collected at the same location, would be distributed evenly among the partitions. In effect, this is similar to k-fold cross-validation, where k is equal to the number of sites.

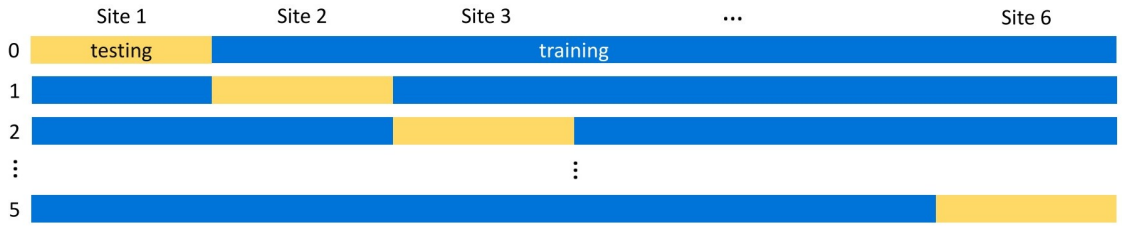


Figure 5.9: Evaluating iteratively with each site-based partition as the testing set may reveal how well the model generalizes to classifying surfaces at unseen locations.

6

Results

This chapter will present results from data acquisition, feature extraction and machine learning.

6.1 Initial Tests of the Radar

To evaluate and examine the behavior of Acconeer's A111 radar sensor, tests from a stationary platform are conducted. This data is collected in campaign 1 as described in Chapter 3.2. The amplitude on the y-axis in the following graphs is unitless, likely because the radar performs normalization before producing the raw data. Also, the standard deviation curves above and below the mean curves correspond to the variability with respect to frames rather than features, where the variability of features can be significantly reduced by increasing N.

6.1.1 Noise and Leakage Measurements

To measure the noise and leakage signal of the radar, data is acquired while the radar is directed towards the sky with no nearby objects.

Theoretically, there should be no reflected signal when pointing the radar at the sky. In Figure 6.1, it can be seen that there is a measured signal, which can be attributed to leakage and noise. However, this signal is relatively low when compared to typical measurements on asphalt.

6.1.2 Stationary Measurements

To further test the radar system, different stationary platform setups are examined and compared.

Metal Surface

While gathering data on the metal surface at site 8, the radar sensor is mounted at various heights h and directed at various angles δ. The purpose of these measurements is to evaluate the effect the height h and angle δ has on the resulting data, and compare this to the model formulated in Chapter 2.

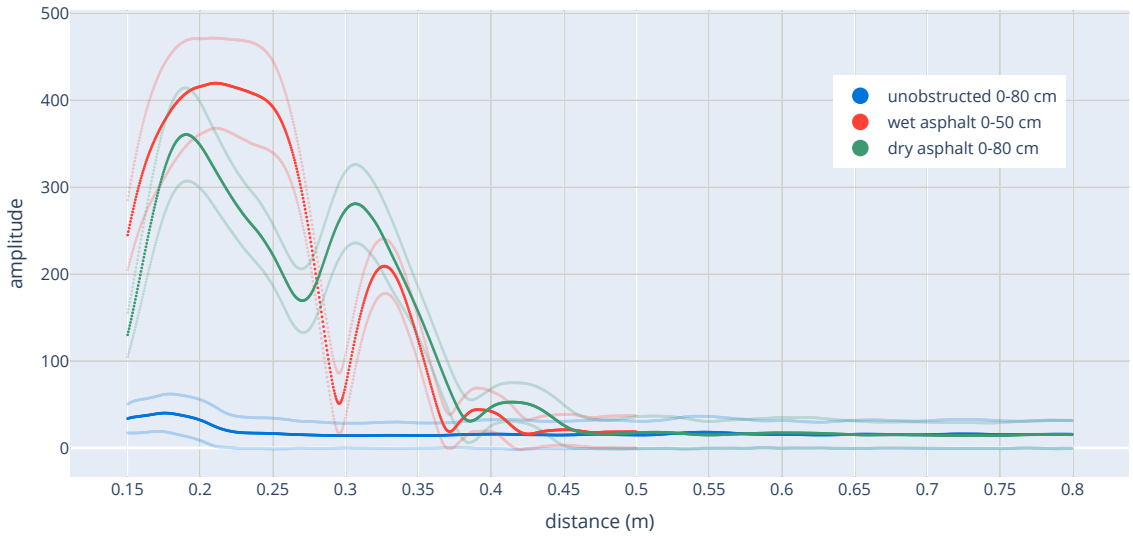


Figure 6.1: Result from a measurement examining direct leakage and noise for the A111 radar. Each curve represents 500 frames as a mean and an interval of two standard deviations.

The radar waves exhibit specular reflection when reflected from a smooth metal surface. The maximum amplitude for the measurements all occur at a distance corresponding to the height underneath the radar. This is the distance to the area of the surface closest to the radar and also where the angle of incidence is 0°. Thus it is expected to receive the maximum reflected power from this area according to Section (2.12).

The amplitude peaks for $h = 17.8\,\text{cm}$ and $33.5\,\text{cm}$ are located near their respective distances, but the peak for $h = 26.5\,\text{cm}$ is not located at the expected distance. This may be due to a fault in the measurement process.

For the following statements it is assumed that the amplitude presented in Figure 6.2 corresponds to power. In Figure 6.2, the maximum amplitude at a height of $17.8\,\text{cm}$ is approximately 1500. The maximum power at a height of $33.5\,\text{cm}$ is at approximately 300 and the maximum amplitude for $26.5\,\text{cm}$ is at approximately 600. This result is expected since the received power from the surface reflection decreases with the square of the distance h.

Figure 6.3 shows that the amplitude is greater when the radar is directed at a lower angle. At both angles, the maximum received power responds to the power reflected from the surface directly under the radar. At a higher radar tilt angle δ, the transmitted power from the radar to the surface directly underneath will be lower, as seen in Figure 2.5. This in turn results in a lower amplitude for the received signal, which is confirmed by the results depicted in Figure 6.3.

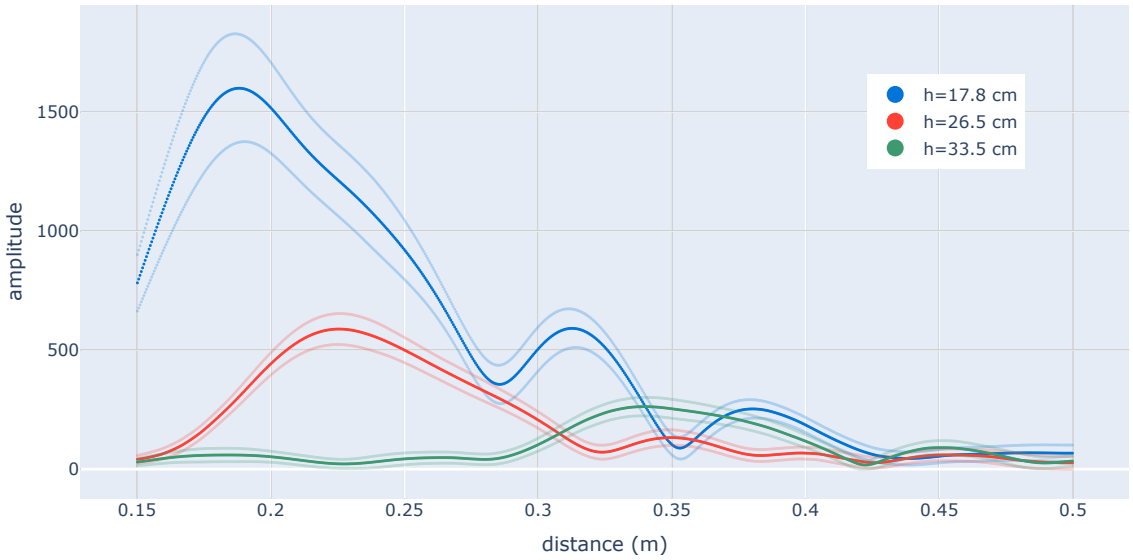


Figure 6.2: The height h of the radar is varied with the tilt angle δ fixed at 45°. Each curve represents 500 frames as a mean and an interval of two standard deviations. This data is from site 8.

Target Surfaces

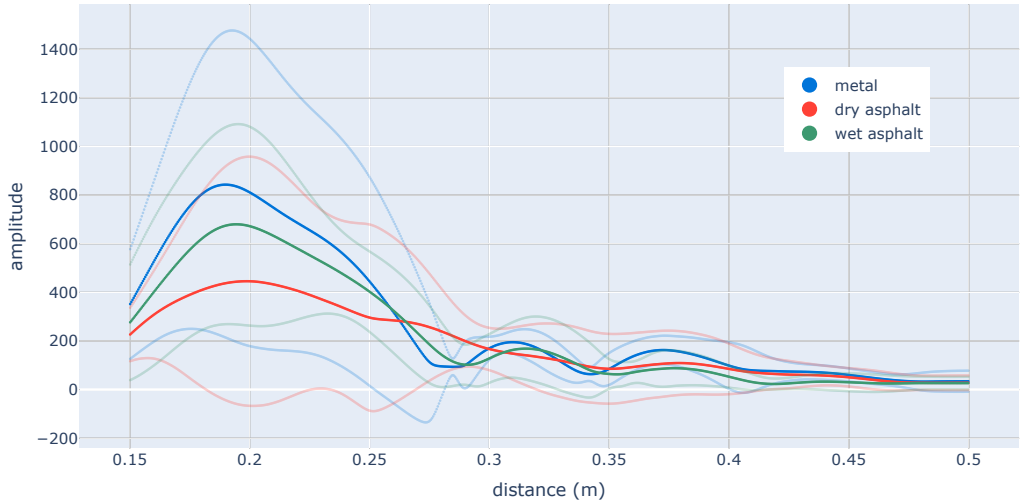


Figure 6.4: The angle of the radar is 30° and the height is 17.5 cm. The data for asphalt is from site 1, while the metal data is from site 8. The curves are an average of 1500 frames each.

The maximum power for the three measurements occurs at approximately the same height, which as previously explained corresponds to the effect from the surface directly beneath the radar. The highest amplitude occurs from the metal surface. This is expected due to the surface being more reflective than the other two, as discussed in Section 2.3.

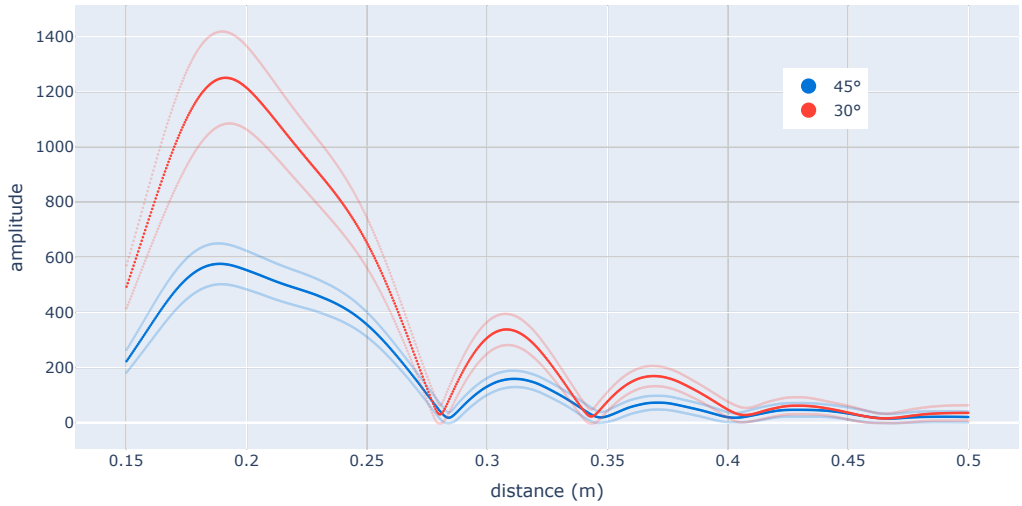


Figure 6.3: The tilt angle δ of the radar is varied with the height fixed at 17.8 cm. Each curve represents 500 frames as a mean and an interval of two standard deviations. This data is from site 8.

6.2 Data-acquisition with Moving Platform

Measurement campaigns 2 and 3 are executed with the remote-controlled car as described in Section 3.1. These campaigns contain 30 000 slow-time frames for each type of surface; dry asphalt, wet asphalt and snow. Note that the envelope service is used for the data collection, which is why there is a difference compared to the stationary measurements, where the IQ service was used.

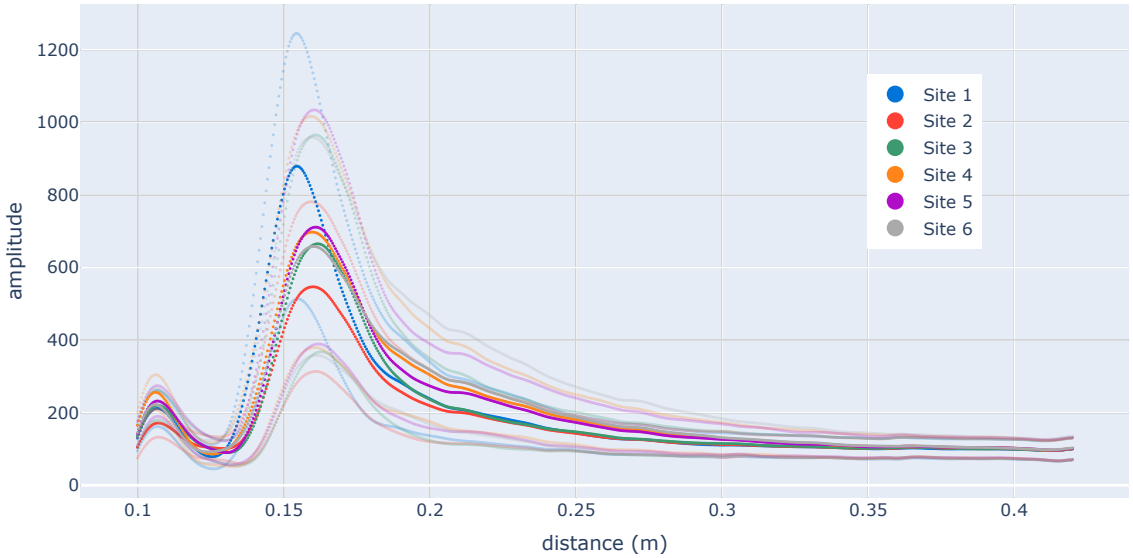


Figure 6.5: Each curve is an average of the 5000 frames collected on dry asphalt at each location.

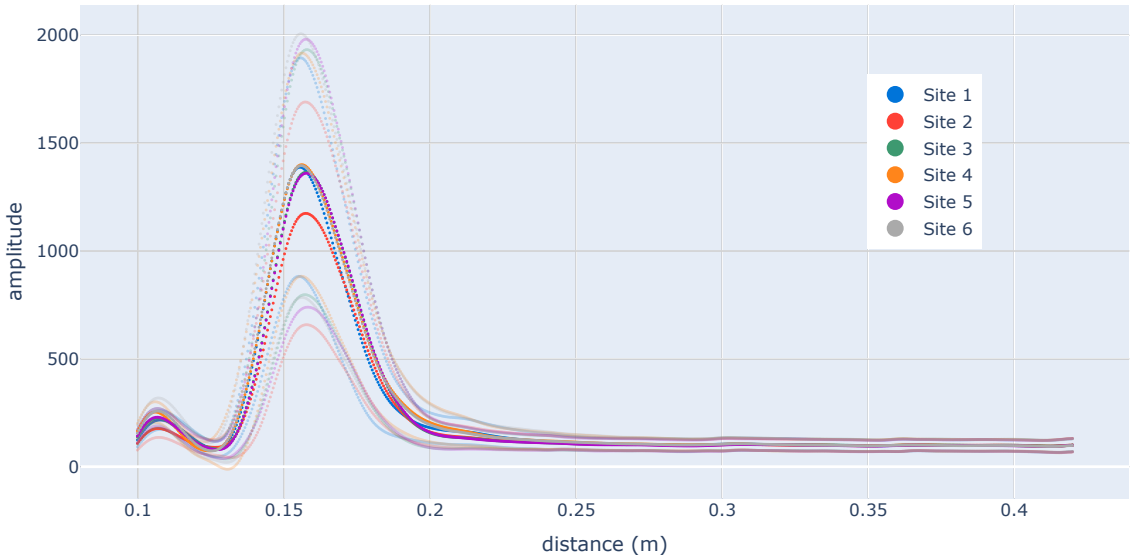


Figure 6.6: Each curve is an average of the 5000 frames collected on wet asphalt at each location.

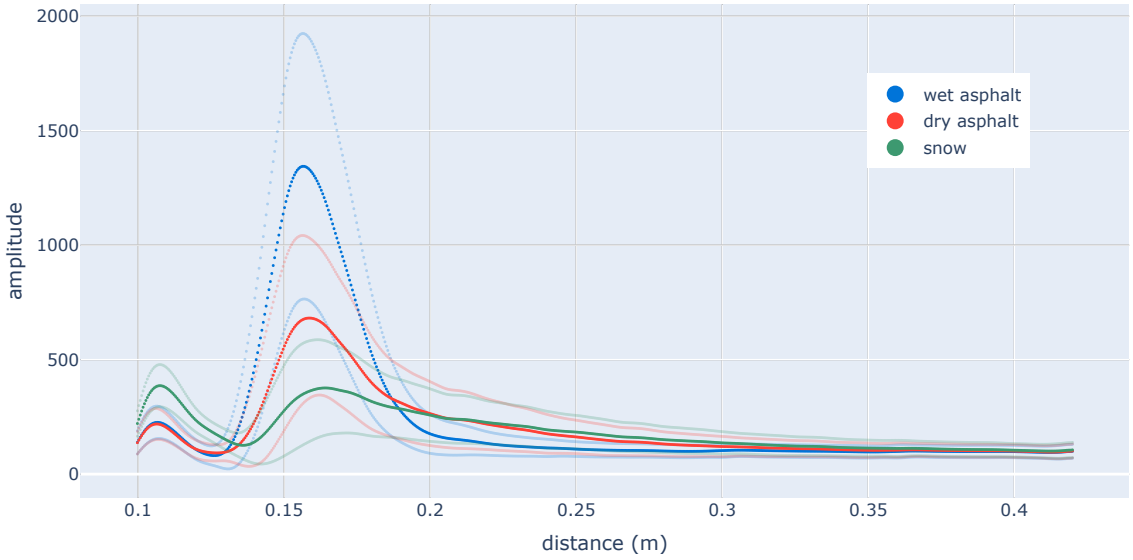


Figure 6.7: Each curve is an average of the 30 000 frames collected from each surface type.

As described in Section 3.2, the snow surface exhibits the greatest structural variations in the surface. The variations in asphalt are smaller, where the thin layer of water reduces the variations further. This correlates with the curves for each surface in Figure 6.7. The smoother the surface, the taller the amplitude peaks are. Curves with the taller peaks also reach a lower amplitude at distances beyond their peaks. At 25 cm, the smoother surfaces seem to backscatter less of the transmitted radar signal.

Autocovariance Comparison

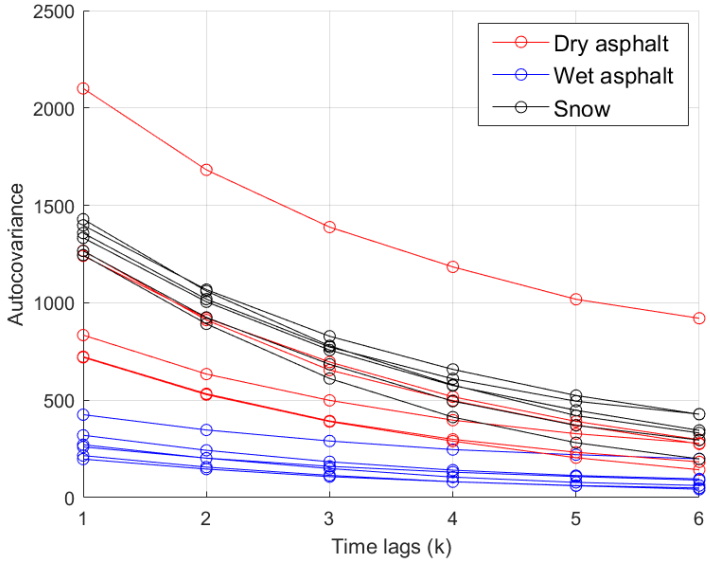


Figure 6.8: Resulting autocovariance curves for the first 6 time lags at fast-time index 280, corresponding to approximately 25 cm. Each asphalt curve is calculated from 5000 frames at a unique location. The snow curves are all from one location.

Figure 6.8 depicts the computed autocovariance for a single distance from the radar where each line in the graph represents the result of 5000 frames from different sites. The ACF produces different results for each fixed distance from the radar (fast-time index) included in the computation which means that similar graphs can be depicted for each instance of distance but for simplicity only one is displayed.

In Figure 6.8, the same type of surfaces exhibit similar time-domain dependencies, except the top curve which can be an outlier. The graph indicates that the different surfaces can be separated in the case where the ACF is calculated for 500 slow-time frames.

Mean-value Analysis

	Distance to dry	**Distance to wet**	**Distance to snow**	σ^s
Dry	0.00	795.80	420.33	156.27
Wet	795.80	0.00	1183.40	317.87
Snow	420.33	1183.40	0.00	87.72

Table 6.1: The distance between each surfaces' amplitude mean-vectors and their respective average standard deviations are shown as described in 4.2. The data is downsampled to 20 fast-time samples.

The distances between the mean vectors of the surfaces are all greater than their average standard deviations. This indicates that the collected data is separable at least to some extent.

6.3 Assessment of Feature Extraction

Grid search is used to find the best combination of n_{ds} (number of fast-time samples after downsampling) and N (number of slow-time frames used to calculate each feature vector) for each feature extraction method. Each combination is evaluated with location-based cross-validation on a support vector machine with Scikit-learns default parameters and the results are presented in Tables 6.2, 6.3 and 6.4. Here, 80% of the data is used since 20% is reserved as testing data for final evaluation. The performance measure is the mean percentage of correct classifications ± two standard deviations during cross-validation. The standard deviation gives an indication to how varied the results are. Since the data is not normally distributed, this interval can add up to over 100%.

	$N = 10$	$N = 20$	$N = 50$
$n_{ds} = 662$	98.58 ± 2.82	99.36 ± 1.03	100.00 ± 0.00
$n_{ds} = 330$	98.68 ± 2.45	99.36 ± 1.02	100.00 ± 0.00
$n_{ds} = 165$	98.78 ± 2.36	99.37 ± 1.02	100.00 ± 0.00
$n_{ds} = 50$	98.75 ± 2.59	99.50 ± 0.74	100.00 ± 0.00
$n_{ds} = 20$	98.85 ± 2.55	99.69 ± 0.48	100.00 ± 0.00

Table 6.2: Performance of moving average feature extraction for each parameter combination

	$N = 10$	$N = 20$	$N = 50$
$n_{ds} = 662$	98.42 ± 3.10	99.50 ± 1.14	100.00 ± 0.00
$n_{ds} = 330$	98.25 ± 3.57	99.50 ± 1.14	100.00 ± 0.00
$n_{ds} = 165$	98.47 ± 3.53	99.50 ± 1.14	100.00 ± 0.00
$n_{ds} = 50$	98.89 ± 2.42	99.58 ± 0.66	100.00 ± 0.00
$n_{ds} = 20$	98.89 ± 2.42	99.83 ± 0.27	100.00 ± 0.00

Table 6.3: Performance of periodic average feature extraction for each parameter combination

	$N = 10$	$N = 20$	$N = 50$
$n_{ds} = 40$	79.47 ± 5.60	85.94 ± 4.2	90.07 ± 5.06
$n_{ds} = 20$	76.65 ± 5.28	84.39 ± 5.12	90.49 ± 5.65

Table 6.4: Performance of autocovariance feature extraction for each parameter combination

The averaging feature extraction methods in Tables 6.2 and 6.3 perform much better than autocovariance. With $N = 50$, they classify every instance in the validation sets correctly. The chosen feature extraction method is periodic average, since it generally performs better than moving average. Also, $N = 20, n_{ds} = 20$ is chosen as

the parameter combination to use for final evaluation. This is because the accuracy is near-perfect, yet the classification rate is over twice as high as $N = 50$.

6.4 Machine Learning Models and Parameters

The parameters are optimized for each machine learning model with grid search as explained in Section 5.2.2. The training set consisting of 80% of the total data is used. The combination of parameters that results in the highest classification accuracy with cross-validation is chosen as the best.

6.4.1 Support Vector Machine

The following parameter intervals were tested for the SVM algorithm:[1]

```
C = [1, 10, 100, 1000]
gamma = [1,0.1,0.001,0.0001, 'scale', 'auto']
kernel = ['linear', 'rbf']
```

After performing a grid search evaluation of all parameter combinations above, the highest accuracy was attained with the following parameters:

```
C = 10
gamma = 0.001
kernel = 'rbf'
```

6.4.2 Random Forest

The following parameter intervals were tested for the random forest algorithm. The `min_samples_leaf` parameter sets the minimum number of samples allowed in a leaf node after splitting, the default is 1. The `min_samples_split` sets the minimum number of samples for a node to be split, the default is 2.[2]

```
n_estimators = [100, 300, 500, 700, 1100, 1300]
max_depth = [10, 20, 30, 40, 50, None]
min_samples_split = [2, 5, 10]
min_samples_leaf = [1, 2, 4]
```

After performing a grid search evaluation of all parameter combinations above, the highest accuracy was attained with the following parameters:

```
n_estimators = 700
max_depth = 20
min_samples_split = 5
```

[1] Reference for parameters and implementation with Scikit-learn: `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

[2] Reference for parameters and implementation with Scikit-learn: `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`

```
min_samples_leaf = 2
```

6.4.3 Neural Network

These are the parameters that have been tested for the neural network algorithm. For setting `hidden_layer_sizes`, `(x,)` denotes a single hidden layer with `x` nodes.[3]

```
solver = ['lbfgs', 'adam']
alpha = [1e-3, 1e-4, 1e-5]
hidden_layer_sizes = [(50,), (20,), (50,20), (100,), (20, 10)]
learning_rate = ['constant', 'adaptive']
learning_rate_init = [1e-2, 1e-3, 1e-4]
```

After performing a grid search evaluation of all parameter combinations above, the highest accuracy was attained with the following parameters:

```
solver = 'adam'
alpha = 1e-4
hidden_layer_sizes = (50,20)
learning_rate = 'constant'
learning_rate_init = 1e-3
```

6.4.4 Evaluation of Models

The models are tested using the evaluation methods described in Section 5.2.3. The tuned hyperparameters are used for each algorithm.

[%]	**SVM**	**RF**	**NN**
Unseen testing set	100.00	100.00	100.00
K-fold cross-validation, k=6	99.93 ± 0.19	99.90 ± 0.45	99.98 ± 0.10
Site-based cross-validation	99.80 ± 0.46	99.73 ± 0.85	99.80 ± 0.37

Table 6.5: The performance of each classification algorithm evaluated with three methods.

All of the models perform close to 100%. The site-based classification does not perform much worse than the other methods, indicating that the models can generalize well between different locations. The neural network performs with a slightly higher accuracy and exhibits the lowest standard deviation of the cross-validation results.

[3]Reference for parameters and implementation with Scikit-learn: `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html`

7

Discussion

This chapter will primarily reflect on the main findings of research in terms of the methodology in relation to the performance of the machine learning algorithms.

The final classification results are much higher than expected. They are so high, that at first doubts were raised as to if there was a fault in the code handling the data. The code was rewritten and different approaches to structuring the data were tried. The partitioning and implementation of the algorithms were done in large part independently by two people and yet the same results were attained. When the procedure was found to be faultless, different evaluation methods were tried. The evaluation methods, each designed to detect different flaws, all gave a similar picture. Perhaps the classification of these surfaces is a relatively easy task. After all, simple methods such as mean vector analysis and observing the graphs show that the data exhibits differentiating characteristics between the surfaces. This indicates that accurate classification should, in fact, be possible with the acquired data.

Since there are noticeable differences in the asphalt at different sites, it would be plausible that the trained algorithm may have a hard time generalizing between them when evaluating with site-based cross-validation. While this evaluation method performed with the least accuracy, it was still very near perfect, indicating that the differences between surface types are far greater than the differences between locations. Surprisingly, the unseen testing set outperformed the others with perfect classification accuracy. The reason behind this could be that the unseen testing set represents the sites equally, which gives the trained model an equal chance of learning to classify the asphalt at each site. On the other hand, the k-fold method contains random amounts of each site in each fold, meaning that certain training folds may underrepresent certain sites.

The random forest algorithm is slightly less accurate and exhibited a slightly higher standard deviation than the other two algorithms, even if it was only by a small margin. Random forests are very powerful and can be applied to complex problems. They also need delicate tuning to avoid under- or overfitting. Perhaps finer steps in the grid search intervals for the parameters would allow a more finely tuned random forest model. This would, however, increase the run-time of the grid search considerably. A sensible approach, looking back, would be to estimate the run-time of the grid search and accommodate the grid search intervals to maximize the number of testable combinations.

A surprising outcome is that feature sets with a lower number of fast-time samples per feature vector n_{ds} generally perform better for the averaging features. For autocovariance features, downsampling twice as much does not decrease the accuracy substantially. This can mean that only a general shape of the amplitude is needed for distinguishing the surfaces. More resolution in fast-time in this case merely adds variance to the data points. The autocovariance features performed worse in comparison to the averaging features. The performance of the autocovariance features increased significantly when the number of slow-time frames N increased. Thus, with a sufficient number of frames, the ACF could also result in high performance. However, a high classification rate is often desirable which makes the ACF unpractical on its own if the sampling frequency is not sufficiently high. However, in combination with averaging features, the ACF could contribute valuable temporal information about the surfaces that the averaging features can not provide.

The unexpectedly accurate classification results naturally lead to questioning of the data. While poor results could have been attributed to poor data, unreasonably easy classification could very well be the result of flaws in data acquisition. If the collection of data is treated differently between surfaces, differences not stemming from surface attributes could arise. This could happen due to a simple mistake such as accidentally tilting the radar when changing surfaces. The utilized data-acquisition method, however, mitigates these risks. Data for dry and wet asphalt is collected in an alternating manner during the campaigns and the procedures for using the equipment are under control.

Since the RC car is relatively small and light, and because it has very simple suspension, it is susceptible to vibrations and disturbances. Uneven surface structures can cause the car to jump and shake while moving, leading to small variations in the radar's height and angle relative to the ground. If the effect of this were to differ between surface types, then this could be reflected by bias in the data. The assessment is, however, that the car's handling should not be too different when comparing dry and wet asphalt. There should be a greater difference when comparing sites since the structure of the asphalt differs between sites. Differences between sites could also arise because changing sites involves activities that can introduce disturbances. The equipment is moved and reset, which can lead to accidental changes in the setup. Also, the battery is notoriously unreliable, so after long sessions of use, the voltage may drop and the speed may be less steady.

The differences between sites, however, seem to be minimal since the curves in Figure 6.5 and 6.6 are very similar, with exceptions in sites 1 and 2. Also, the site-based cross-validation evaluation shows that the data-based model generalizes very well between sites.

One could reason that the snow data, which is collected under different circumstances and all at the same location, could be differentiated by external factors rather than by surface attributes. This risk exists, but since snow only represents one third of the data, it can not solely cause misleading results. Also, by observing the differences in the surfaces' data in Figure 6.7, the differentiating characteristics

seem to be correlated with physical attributes. The snow surface, which has the greatest topographical variations, is also the surface that seems to observe the most diffuse scattering.

The platform for data acquisition has several areas of potential improvement. The first limitation is that the remote-controlled car is originally a toy. One of the drawbacks of it being a toy is that the motor and control electronics do not allow for slow speeds and fine adjustments. The feedback control system is particularly sensitive to inclines when the revolutions per minute (RPM) of the motor is low. This sometimes results in the car losing the constant speed by slowing down or outright stopping. Such problems are dealt with by removing the troublesome measurement and by repeating the measurement. Another small drawback is the battery capacity only allowing for a limited amount of measurements to be done.

A better data-acquisition platform would be built in order to be reliable and high-performing. Also, it would benefit from being better tailored to the applications in mind. For classifying road-surface conditions, one could even mount the radar on a car, which would allow collection of more realistic data.

The radar setup also has its quirks, being based on an evaluation kit and not specialized hardware. The most apparent disadvantage is the transfer rate between the radar module and the data acquisition equipment. The maximum allowable sampling rate in slow-time, before saturating the data bus, is approximately 320 Hz. To achieve a high classification rate, this update rate must be set as high as possible. The classification rate is directly related to the latency of classification. The latency of classification is one of the most important metrics for a system that relies on fast feedback, such as cars in motion. To improve this aspect, more powerful and specialized hardware must be used. The desirable data rate depends on the velocity and the latency tolerance which is therefore something that has to be evaluated for each application.

Another possible drawback of the A111 module is its range if it is to be used in, for example, a fast-moving vehicle. The desirable range is directly related to the SNR and to the time it takes for a pulse to travel back and forth to an object. The processing time for classifying a data point must be within the latency requirements of the application. Otherwise, once the system has classified a data point, the surface corresponding to that data point may already been passed, which renders the system useless. In order to support long processing time, the range of the system must increase, thereby being able to classify surfaces at a distance. The sensor inherently has a noise level that must be overcome by the received signal to be detected. The power level of the received signal is directly related to the transmitted power as discussed in Section 2.2.1. The solution to increasing the range is, therefore, either to increase the transmitted power or to reduce the noise level of the receiving component.

8

Conclusion and Further Work

This thesis explores whether it is possible to classify different types of road surfaces using radar while in motion. This is examined by collecting data from several types of surfaces using a dynamic platform moving at a constant velocity. Features are extracted using various mathematical methods such as autocovariance and amplitude averaging to develop a machine learning algorithm. The performance of the classification model indicates that it is possible to confidently distinguish between dry asphalt, wet asphalt, and snow with radar and machine learning.

The hardware solution used in this thesis is not optimal for measuring road conditions in motion. To ensure a high classification rate, a high sampling frequency is required which could be achieved with more powerful and specialized hardware. Also, to ensure the quality and repeatability of the acquired data, a more robust platform that is not sensitive to inclines and has better speed and steering control should be used. If, for example, the goal is to classify road conditions while driving a car, the next steps could include studying the impact of varying velocity or using more realistic data.

The work presented offers a starting point for the use of radar and machine learning in the context of surface classification. The scope is relatively narrow, so the results merely give an indication that the task should be possible. There are several ways to expand on the experiment in order to be more applicable to real-world scenarios. This thesis only examines three types of surfaces, one of which is artificial snow which is not present outdoors. A starting point could be to study surfaces that are better represented in reality. Introducing the complications and variations that more realistic data entails, would surely make the classification task more difficult. Therefore, additional combinations of feature extraction methods and machine learning algorithms could be examined to investigate if the performance could be improved. The performance also needs to be evaluated in more ways than simply classification accuracy. Other important factors include classification rate, latency and detection of transitions between surfaces.

Bibliography

[1] M. Richards, "Introduction to radar systems", in *Fundamentals of Radar Signal Processing*. McGraw-Hill, 2005, ch. 1.

[2] F. Mayinger, *Optical measurements: techniques and applications*. Springer Science & Business Media, 2013.

[3] M. Richards, "Signal models", in *Fundamentals of Radar Signal Processing*. McGraw-Hill, 2005, ch. 2.

[4] H. Griffiths, P. Knott, and W. Koch, "Christian hülsmeyer: Invention and demonstration of radar, 1904", *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 9, pp. 56–60, 2019. [Online]. Available: `https://ieeexplore.ieee.org/abstract/document/8826668`.

[5] J. Cannon and R. Crowcroft, *The Oxford Companion to British History*. Oxford University Press, USA, 2015.

[6] M. I. Skolnik, "An introduction and overview of radar", in *Radar Handbook*, 3rd ed. McGraw-Hill, 2009, ch. 1.

[7] V. V. Viikari, T. Varpula, and M. Kantanen, "Road-condition recognition using 24-ghz automotive radar", *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, Dec. 2009.

[8] N. Kees and J. Detlefsen, "Road surface classification by using a polarimetric coherent radar module at millimeter waves", *IEEE MTT-S International Microwave Symposium Digest(Cat. No.94CH3389-4)*, vol. 3, pp. 1675-1678, 1994.

[9] R. Finkele, "Detection of ice layers on road surfaces using a polarimetric millimetre wave sensor at 76 ghz", *Electronics Letters*, vol. 33, pp. 1153–1154, Jun. 1997.

[10] D. Montgomery and G. Holmén, "Surface classification with millimeter-wave radar for constant velocity devices using temporal features and machine learning", Student Paper, Lund University, Lund, Sweden, 2019. [Online]. Available: `http://lup.lub.lu.se/student-papers/record/8974567` (visited on 05/05/2020).

[11] *Radar sensor introduction*, Lund, Sweden: Acconeer, 2020. [Online]. Available: `https://acconeer-python-exploration.readthedocs.io/en/latest/sensor_introduction.html` (visited on 03/06/2020).

[12] V. Bulovic, R. Ram, S. Leeb, J. H. Lang, and Y. Gu, "Reflection & transmission of em waves", 2011. [Online]. Available: `https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-007-`

electromagnetic-energy-from-motors-to-lasers-spring-2011/lecture-notes/MIT6_007S11_lec29.pdf (visited on 05/07/2020).

[13] *A111 – pulsed coherent radar (pcr)*, Lund, Sweden: Acconeer AB, 2020. [Online]. Available: https://developer.acconeer.com/download/a111-datasheet-pdf/ (visited on 02/07/2020).

[14] S. J. Orfanidis, *Electromagnetic Waves and Antennas*. Rutgers University, 2016. [Online]. Available: http://eceweb1.rutgers.edu/~orfanidi/ewa/ewa-1up.pdf (visited on 03/29/2020).

[15] T. Henderson, *Specular vs. diffuse reflection*, 1998. [Online]. Available: https://www.physicsclassroom.com/class/refln/Lesson-1/Specular-vs-Diffuse-Reflection (visited on 05/07/2020).

[16] "Lambert's cosine law", 2019. [Online]. Available: https://en.wikipedia.org/wiki/Lambert%27s_cosine_law (visited on 04/27/2020).

[17] D. Mittleman, "Fresnel's equations for reflection and transmission". [Online]. Available: https://www.brown.edu/research/labs/mittleman/sites/brown.edu.research.labs.mittleman/files/uploads/lecture13_0.pdf (visited on 05/01/2020).

[18] *Permittivity*, in, The Editors of Encyclopaedia Britannica, Encyclopædia Britannica, inc., Dec. 10, 2014. [Online]. Available: https://www.britannica.com/science/permittivity (visited on 05/07/2020).

[19] J. Tang and F. Resurreccion, "1 - electromagnetic basis of microwave heating", in *Development of Packaging and Products for Use in Microwave Ovens*, M. W. Lorence and P. S. Pesheck, Eds., Woodhead Publishing, 2009, 3–38e. DOI: https://doi.org/10.1533/9781845696573.1.3.

[20] P. Lunkenheimer *et al.*, "Electromagnetic-radiation absorption by water", *Phys. Rev. E*, vol. 96, 6 Dec. 2017. DOI: 10.1103/PhysRevE.96.062607.

[21] W. Stiles and F. Ulaby, "Dielectic properties of snow", *U.S. Army ColdRegions Research and Engineering Laboratory, Special Report Vols. 82-18,"Properties of snow"*, pp.91–103, 1982.

[22] D. Nandedkar, "Analysis of dipole relaxation time for water molecules at temperature of 293 0 k", *Physics Journal*, vol. 2, pp. 15–22, Dec. 2015.

[23] M. Hallikainen, F. Ulaby, and M. Abdelrazik, "Dielectric properties of snow in the 3 to 37 ghz range", *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 11, pp. 1329–1340, 1986.

[24] esa, "Radar course 2". [Online]. Available: https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/ers/instruments/sar/applications/radar-courses/content-2/-/asset_publisher/qIBc6NYRXfnG/content/radar-course-2-parameters-affecting-radar-backscatter (visited on 03/29/2020).

[25] J. Brownlee, "Introduction to multivariate statistics", in *Basics of Linear Algebra for Machine Learning: Discover the Mathematical Language of Data in Python*. Machine Learning Mastery, 2018, ch. 17, pp. 138–141.

[26] A. Jacobsson, "Stochastic processes", in *An Introduction to Time Series Modeling*, 2nd ed. Studentlitteratur AB, 2015, pp. 11–14.

[27] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[28] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Inc., 2019.

[29] A. Ng and T. Ma, "Cs229 lecture notes, part v kernel methods", 2000. [Online]. Available: http://cs229.stanford.edu/syllabus.html.

[30] S.-l. Developers, *Sklearn.svm.svc*, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

[31] S.-l. Developers, *3.2.4.3.1. sklearn.ensemble.randomforestclassifier*, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[32] L. O. Hall, "Computational intelligence", *Access Science*, Feb. 2019. DOI: 10.1036/1097-8542.801910.

[33] A. Oppermann, *Activation functions in neural networks*, Oct. 2019. [Online]. Available: https://www.deeplearning-academy.com/p/ai-wiki-activation-functions (visited on 05/01/2020).

[34] M. Nielsen, *Neural Networks and Deep Learning*. Determination press, 2015.

[35] R. Raúl, *Neural networks: a systematic introduction*. Springer, 2000.

[36] *1.17. neural network models (supervised) — scikit-learn 0.22.2 documentation*. [Online]. Available: https://scikit-learn.org/stable/modules/neural_networks_supervised.html (visited on 04/30/2020).

[37] S.-l. Developers, *Sklearn.neural_network.mlpclassifier*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier (visited on 05/06/2020).

A

Derivation of Received Power

To derive total received power the assumption of constant gain between the Half-Power Beam Width (HPBW) angles can be defined in terms of scattering. Consider the surface to only scatter between the HPBW angles as (A.1).

$$\sigma(\theta, \varphi) = \begin{cases} \sigma^o & \text{for } \theta_2 < \theta < \theta_1 \text{ and } \varphi_1 < \varphi < \varphi_2 \\ 0 & \text{others,} \end{cases} \tag{A.1}$$

For lambertian scattering it was previously concluded that the scattering angle θ_s and incident angle θ_i are identical. The monostatic scattering coefficient at scattering point $\boldsymbol{r}$ and angle θ is (A.2).

$$\begin{aligned} &\sigma_0^o \cos\theta_i \cos\theta_s = \\ &= \sigma_0^o \cos\left(\theta_i = \pi - \theta\right) \cos\left(\theta_s = \pi - \theta\right) \\ &= \sigma_0^o \cos^2\left(\pi - \theta\right) = \sigma_0^o [-\cos\theta]^2 \\ &= \sigma_0^o \cos^2\theta \end{aligned} \tag{A.2}$$

Distance from antenna at position $\boldsymbol{r'} = \boldsymbol{0}$ and scattering point $\boldsymbol{r} = r\hat{r} - h\hat{z}$ is (A.3).

$$R = |\boldsymbol{r} - \boldsymbol{r'}| = \sqrt{r^2 + h^2} \text{ (cylindrical coordinates)} \tag{A.3}$$

Recieved power is then (A.4)

$$
\begin{aligned}
P_r &= \frac{P_t \lambda^2 G_0^2}{(4\pi)^3} \int_{\varphi=\varphi_1}^{\varphi_2} \int_{r=r_1}^{r_2} \frac{\sigma_0^o \cos^2\theta}{(r^2+h^2)^2} \underbrace{r dr d\varphi}_{dS'} \\
&= \frac{P_t \lambda^2 G_0^2 \sigma_0^o}{(4\pi)^3} \big(\varphi_2 - \varphi_1\big) h^2 \int_{r=r_1}^{r_2} \frac{r}{(r^2+h^2)^3} dr \\
&= \left\{ \text{Subs. } \xi = r^2 + h^2, d\xi = 2r\ dr,\ \xi_{1,2} = r_{1,2}^2 + h^2 \right\} \\
&= \frac{P_t \lambda^2 G_0^2 \sigma_0^o}{(4\pi)^3} \big(\varphi_2 - \varphi_1\big) h^2 \int_{r_1^2+h^2}^{r_2^2+h^2} \frac{d\xi}{\xi^3} \\
&= \frac{P_t \lambda^2 G_0^2 \sigma_0^o}{(4\pi)^3} \big(\varphi_2 - \varphi_1\big) h^2 \Big[\frac{1}{-2\xi^2} \Big]_{r_1^2+h^2}^{r_2^2+h^2} \\
&= \frac{P_t \lambda^2 G_0^2 \sigma_0}{(4\pi)^3} \big(\varphi_2 - \varphi_1\big) h^2 \Big(\frac{1}{(r_1^2+h_1^2)^2} - \frac{1}{(r_2^2+h_1^2)^2} \Big)
\end{aligned}
\tag{A.4}
$$

B

Pictures of Sites

Figure B.1: Site 1 Parking lot
57.689259 N, 11.979689 E

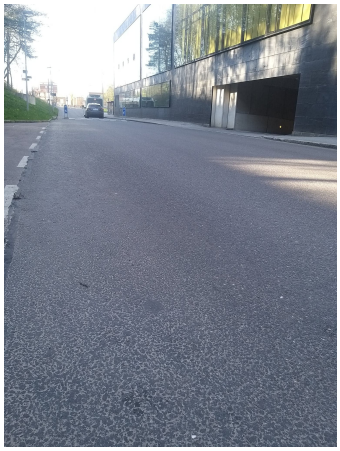


Figure B.2: Site 2 Car road
57.689068 N, 11.973658 E

Figure B.3: Site 3 Loading lot
57.689739 N, 11.977875 E

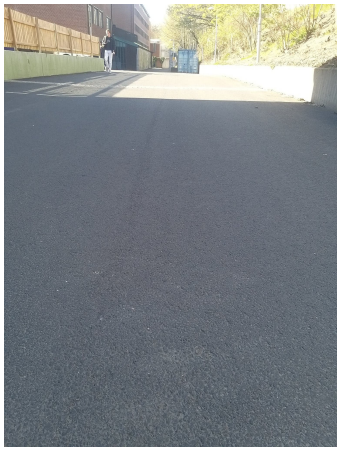


Figure B.4: Site 4 Pedestrian road
57.687780 N, 11.977237 E

Figure B.5: Site 5 Pedestrian road
57.687979 N, 11.977886 E

Figure B.6: Site 6 Pedestrian road
57.685060 N, 11.980295 E

Figure B.7: Site 7 Indoor skiing facility
57.738329 N, 12.035443 E

Figure B.8: Site 8 Kithen counter top
57.689055 N, 11.979197 E