# Sensor Introduction

# Contents

# 1 Welcome!

We are very happy that you have decided to have a closer look at what Acconeer's pulsed coherent radar technology can offer. We are working hard to give you the right tools to **"Explore the Next Sense"**, whether it is for remote monitoring of vital signs, smart city IoT, robot navigation or any other imaginable sensing application where precision, size, and power consumption are key parameters. This document serves as an introduction to Acconeer's sensor technology and product offer. The Acconeer radar system is described based on established radar theory to give you the right knowledge to integrate and configure the sensor in your product.

When starting to use the Acconeer sensor there are different alternatives for both hardware and software setup and we are adding more as we get to know your needs. Check out our website to see our current offer of sensors, modules, and evaluation kits. A typical development flow to get started is to setup one of our evaluation kits and:

- Use the Exploration Tool to get data from sensor into Python to start application development for your use case

- Use our Reference applications to get guidance on use case specific software solutions

- Use Acconeer SDK or Acconeer Module Software to start software development

To further support and guide you through the development process we also provide several user guides, data sheets, reference schematics, and reference layouts, which you can find at acconeer.com. Also check out our demo videos and application page to get inspiration on how you can solve different use cases and how the Acconeer sensor can be used in your application.



## 1.1 Radar basics and the Acconeer pulsed coherent radar

Radar is a well-established technology which has been used in many different applications where accurate and robust distance measurement is required. You can find radar in cars, in the process industry, in airplanes etc. However, most often these radar systems are big, power hungry and expensive, what Acconeer offer is a way to take radar into applications where size, cost and power consumption matter. Radar is an acronym for Radio Detection and Ranging and is a way of determining range to an object by transmitting and detecting radio waves. Acconeer's radar system is a time-of-flight system, which means that a radio wave is transmitted by a first antenna, reflected by an object, and then received by a second antenna. The time of flight between transmission and reception of the signal is measured, as illustrated in Fig. 1.1.
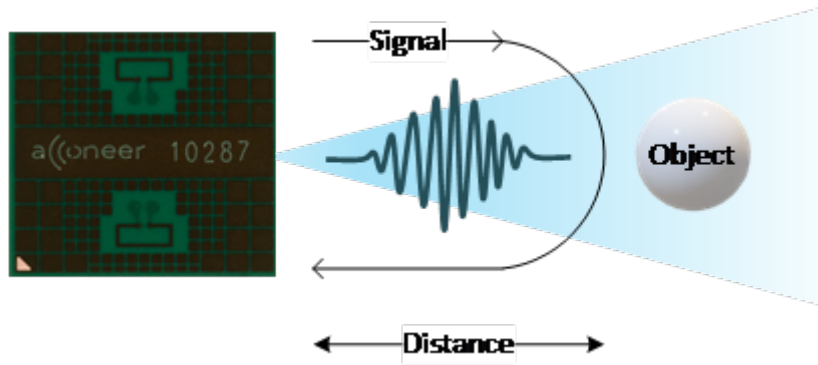
Fig. 1.1: Illustration of the pulsed coherent radar system where the time of flight is measured to determine distance to object

The distance to the object can then be calculated by multiplying the time-of-flight with the speed of the radio wave (same as speed of light) and then dividing by two as the distance the signal has traveled is equal to two times the distance to the object. More details about the radar and the Acconeer approach can be found in the chapter *System Overview*.

There are different approaches to building radar systems, each with its own pros and cons that typically results in a trade-off between accuracy and power consumption, see Fig. 1.2. At Acconeer we have solved this by combining two important features of a radar system, the first is that it is pulsed, which means that the radio part is shut down in between transmission of signals. In fact, the Acconeer radar is pulsed so fast that, typically, the radio is active less than 1 % of the time even when transmitting at maximum rate. This is how the power consumption can be kept low and optimized dependent on the update rate required by your application.
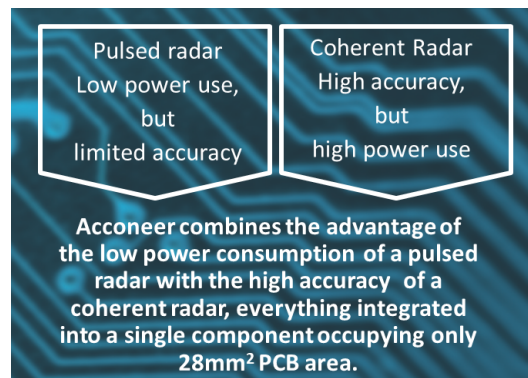


Fig. 1.2: Pulsed coherent radar.

The second feature is that it is coherent, which means that each transmitted signal has a stable time and phase reference on the pico second scale, which allows for high accuracy measurements. Coherent radar systems usually rely on a continuous generation of the radio signal, which consumes a lot of current independent on update rate, hence one of the innovations Acconeer has made is to combine the benefits of pulsed systems and the benefits of coherent systems into one product, the Pulsed Coherent Radar (PCR). The unique selling points of the PCR sensor are summarized in Fig. 1.3. The sensor makes it possible to perform high accuracy measurements while consuming very little power and the fast pulsing of the system makes it possible to track fast movements.
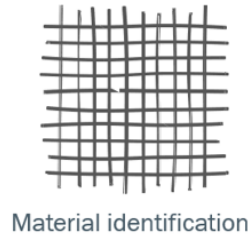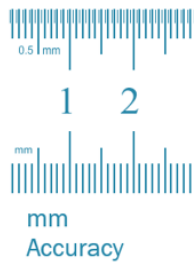
Fig. 1.3: Unique selling points of the Acconeer pulsed coherent radar.

Another benefit of the pulse coherent radar is that amplitude, time and phase of the received signal can be handled separately and allow for classification of different materials that the signal has been reflected on. These are all benefits when compared to sensors such as infra-red and ultrasonic. Additional benefits are that the Acconeer radar can be hidden behind colored plastic or glass and hence do not need an open or visible aperture, we call this optimized integration. The sensor is also robust as it is not sensitive to ambient light or sound and not sensitive to dust or even color of the object.

## 1.2 The Acconeer offer

The Acconeer offer consists of two parts, hardware and software, as illustrated in Fig. 1.4. In addition, Acconeer also provides various tools to aid the customer in the development process.

The A111 sensor is the core of the hardware offer and is available in modules and in evaluation kits. The purpose of the evaluation kit is to provide a platform to get acquainted with the pulsed coherent radar and to start use case evaluation. The sensor evaluation kits are based on Raspberry Pi, which is a well-known and available platform which also allows you to connected other types of sensors. The module is an integration of the A111 and a microcontroller unit (MCU) and has its own evaluation kit. Just as the sensor evaluation kit it can be used to get familiar with the pulsed coherent radar technology and get started with use case development. It can also be included as a single unit in your product to decrease your development cost and decrease time to market.

Fig. 1.5 outlines the software structure, platform for running it, and communication interfaces. The software for controlling the A111 sensor and retrieving data from it is called Radar System Software (RSS) and provides output at two levels:

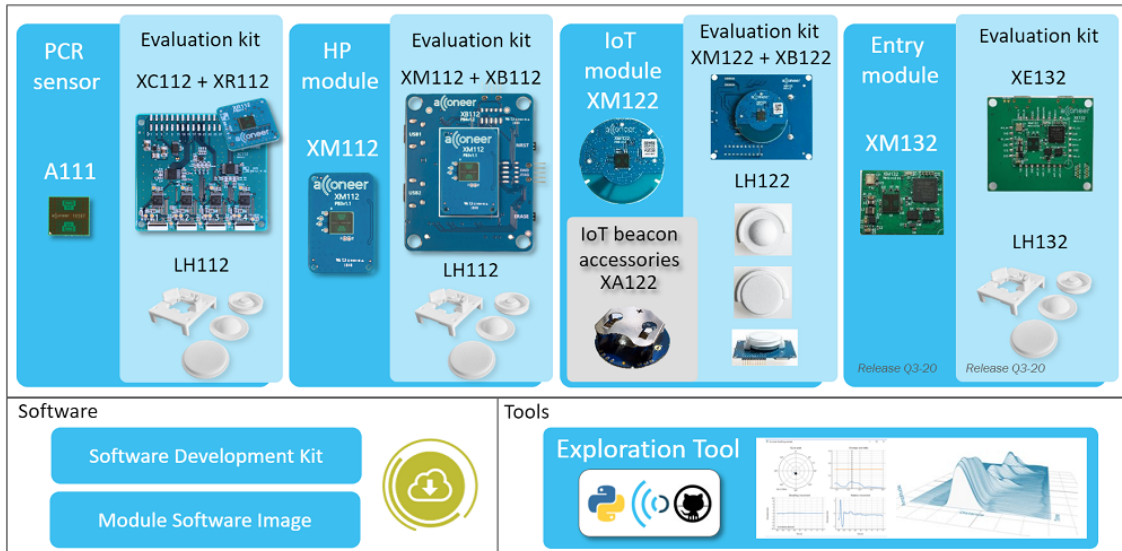- Service, provides pre-processed sensor data

Fig. 1.4: The Acconeer offer.

- Detector, provides results based on the sensor data - all Detectors are based on Services
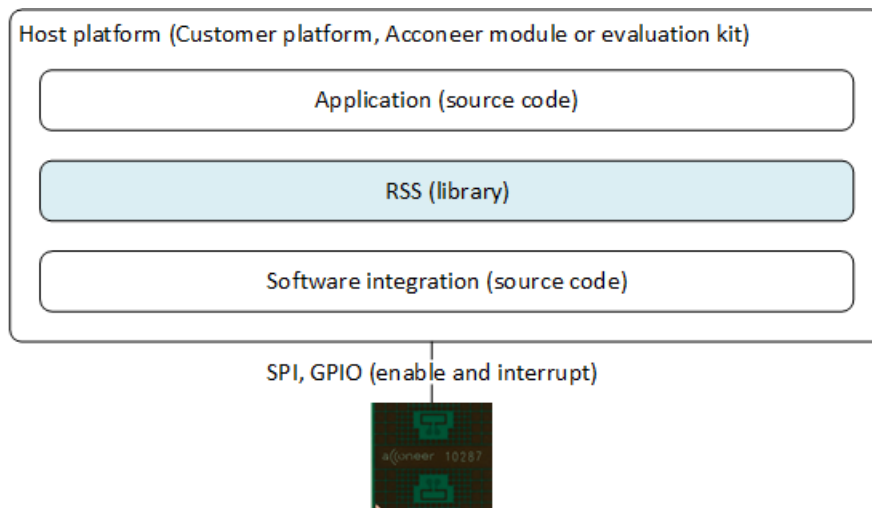


Fig. 1.5: System structure, the RSS software runs on a host that controls the sensor.

RSS is provided as library files and is written in C and designed to be portable between different platforms, a list of currently supported processor architectures and toolchains are available at the Acconeer developer site. Apart from RSS, Acconeer provides Example applications and stubbed software integration source code in the Software development kits (SDKs) as well as full reference integrations for selected platforms.

Acconeer provides four types of applications:

- Example applications: Example of how to use RSS, available in SDK at Acconeer developer site

- Reference applications: Use case specific reference application available in SDK at Acconeer developer site

- Streaming server: Application streaming data from sensor evaluation kit to PC,

available in SDK for Raspberry Pi at Acconeer developer site

- Module server: Application providing a register write based interface to Acconeer modules, available in Module software image at Acconeer developer site.

Both RSS and Applications run on a host platform and Acconeer provides a software integration reference with guidance on how to integrate to your host platform as well as specific integration for the modules and evaluation kits that Acconeer provides.

- For our EVK platforms we provide a software package and for

  – Raspberry Pi it includes hardware abstraction layer, device drivers, and build environment provided as source code

  – Modules it includes hardware abstraction layer and build environment provided as source code

- For STM32 platforms we provide example integration files and instructions for how to set up a project in STM32CubeIDE.

- Other ARM Cortex M0, M4 and M7 based platform can easily be used by writing a custom implementation of the HAL integration layer. A handful functions that use MCU specific driver functions for accessing timers, SPI and GPIO have to be implemented.

For more detailed information on how to implement the HAL integration layer used by RSS, there is a user guide available at acconeer.com.

Based on these deliveries it is possible for the customer to create their own integration layer for any platform that uses a supported processor architecture. The currently available products and corresponding software deliveries are listed in Fig. 1.6, refer to documentation for each specific product for further details.

| Product | | Software |
|---|---|---|
| Sensor | A111 | • SDK for supported targets |
| Evaluation kit | XC111 XR111 | • SDK for Rpi |
| Evaluation kit | XC112 XR112 | • SDK for Rpi |
| Module | HP XM112 IoT XM122 Entry Module XM132 | • SDK for Atmel SAME70, Nordic nRF-40 SoC, STM M0+ • Module software image for XM112, XM122, XM132 |
| Evaluation kit | XB112 XM112 XB122 XM122 XE132 | • SDK for Atmel SAME70, Nordic nRF-40 SoC, STM M0+ • Module software image for XM112, XM122, XM132 |

Fig. 1.6: Products and software deliverables.

At acconeer.com, there are modules and SDK variants and they all contain RSS, Software integration, and Example applications. The Module software image contains RSS, software integration, and Module server. The module can be used in two different setups:

- Stand-alone module: The module has got no dependency on external controllers. The application is customized to a specific use case by the customer and runs on the embedded MCU. The customers application is accessing the RSS API via a software interface.

- Controlled module: The module is connected to an external controller where the customer runs their application software. The customers are accessing the RSS API via a hardware interface through the module software, that provided register mapped protocol.

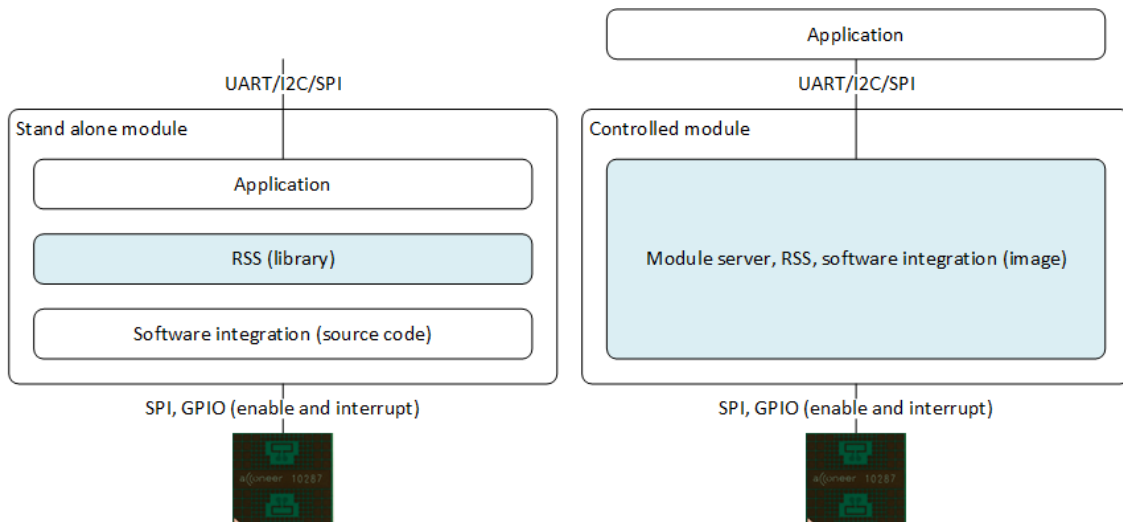The two setups listed above are also illustrated in Fig. 1.7.



Fig. 1.7: Setup.

For the Stand-alone module setup the customer should use the RSS library and Software integration source code provided in the corresponding SDK and build their own application on these deliveries. For the Controlled module regime, i.e. the modules designed by Acconeer, the complete software that runs on the module is delivered as an image. The customer can freely select between these two options, Acconeer supports both.

## 1.3 The Acconeer tools

To help you to get to know the Acconeer products and get started quickly with application development we provide a Python based tool which consists of several scripts that gives you access to real time data and sensor configuration to easily start developing signal processing for specific use cases. The scripts can also be used to graphically display the radar output and to investigate the reflective properties of different objects. The Exploration Tool requires that the Streaming server or Module server is installed on your sensor evaluation kit or module evaluation kit, respectively. The Streaming Server and Module server reflects the RSS API, which helps to understand how to manage the RSS API in your application. The Exploration Tool is provided for all our evaluation kits and is available at Acconeer GitHub. An overview of how Exploration Tool interface software and hardware for the evaluation kits is presented in Fig. 1.8.
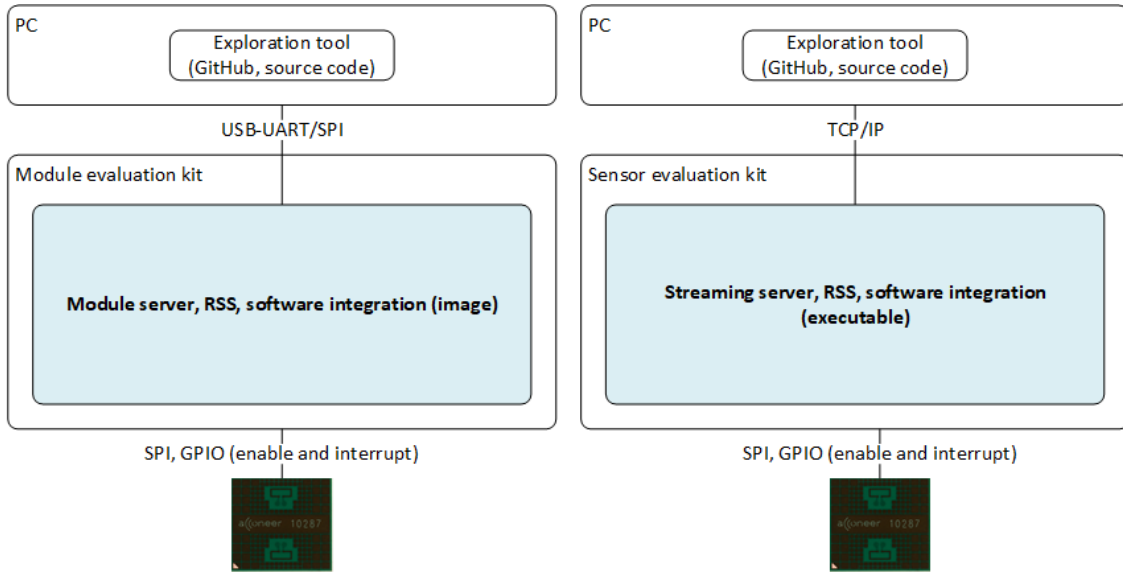
8

Fig. 1.8: Overview of software and hardware interfaces to Acconeer tools.

# 2 Services and Detectors

The RSS provides output at two different levels, Service and Detector. The Service output is pre-processed sensor data as a function of distance. Detectors are built with this Service data as the input and the output is a result, in the form of e.g. distance, presence, angle etc. Services and Detectors currently available are listed in Fig. 2.1.

Each Detector is built on top of a Service, i.e. you have the possibility to use our out-of-the-box Detectors or develop your own. To select the Service or Detector applicable for your use case it is recommended to use the Exploration tool (see Section *Acconeer tools*) to observe the different outputs and understand what they represent. Each Service and Detector also comes with its own user guide, which can be found at acconeer.com.

At developer.acconeer.com, we have several movies showing demos where the Acconeer sensor is used in different use cases. Together with the demo movies, corresponding reference applications are available in our different SDKs at Acconeer developer site. These reference applications are written in C code and use our Services and Detectors, check them out to get inspiration on how to build your product with the Acconeer sensor.

## 2.1 Services

### Envelope and Power Bins services

Fig. 2.2 and Fig. 2.3 show outputs from the Power Bins and Envelope Services obtained with one of the scripts in Exploration Tool, the setup and environment are identical for the two data sets. Here it can be seen that Power Bins and Envelope Services provide output of the same type, i.e. amplitude of received signal as a function of distance. The difference lies in the signal processing done and the Power Bins output has lower SNR, lower resolution in range, but requires less processing and memory allocation than Envelope.

9
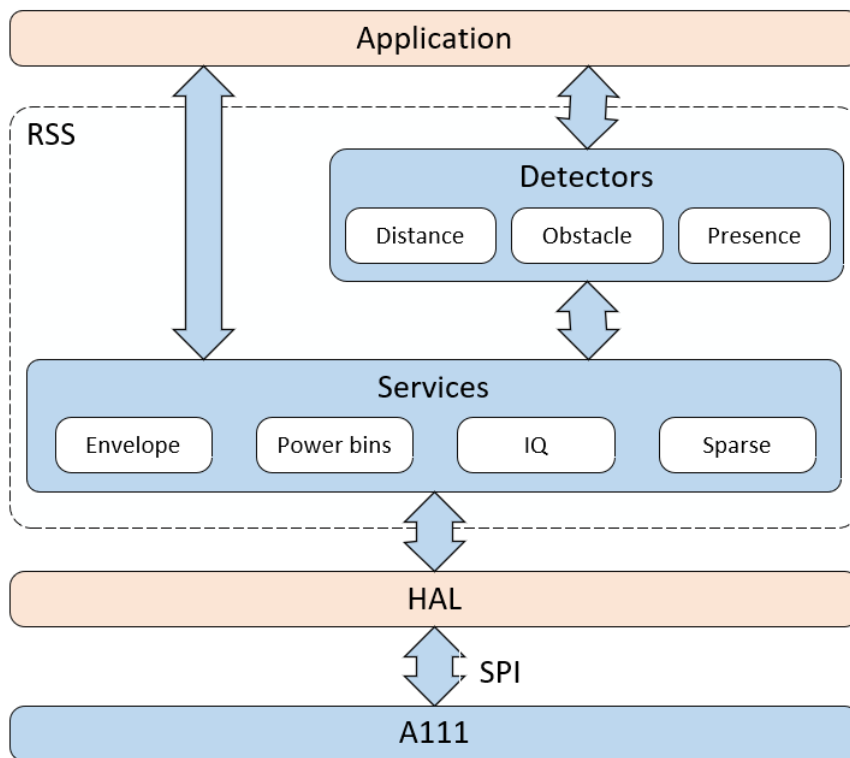
Fig. 2.1: Available Detectors and Services.
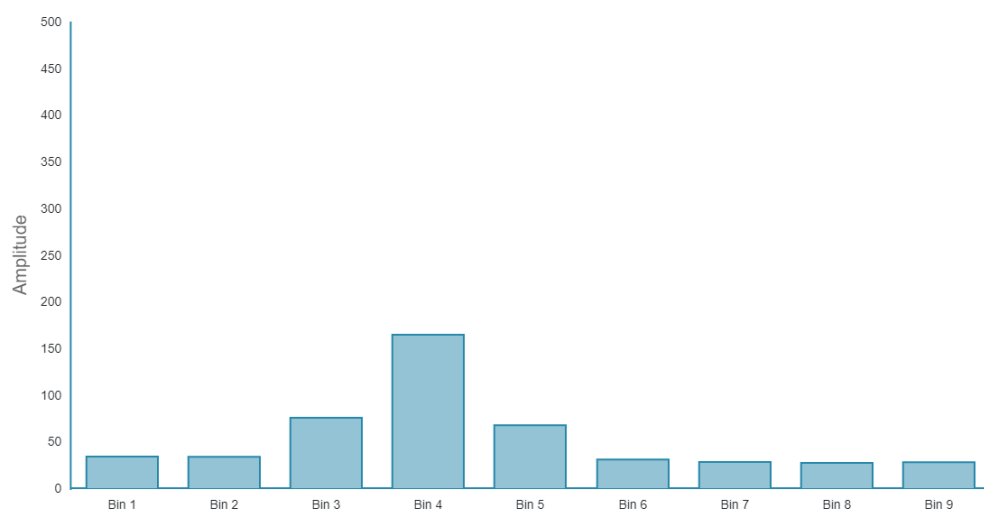


Fig. 2.2: Output from the Power Bins service in Exploration Tool. Each bin correspond to a region of the scanned range, where Bin 1 is closest to the sensor.
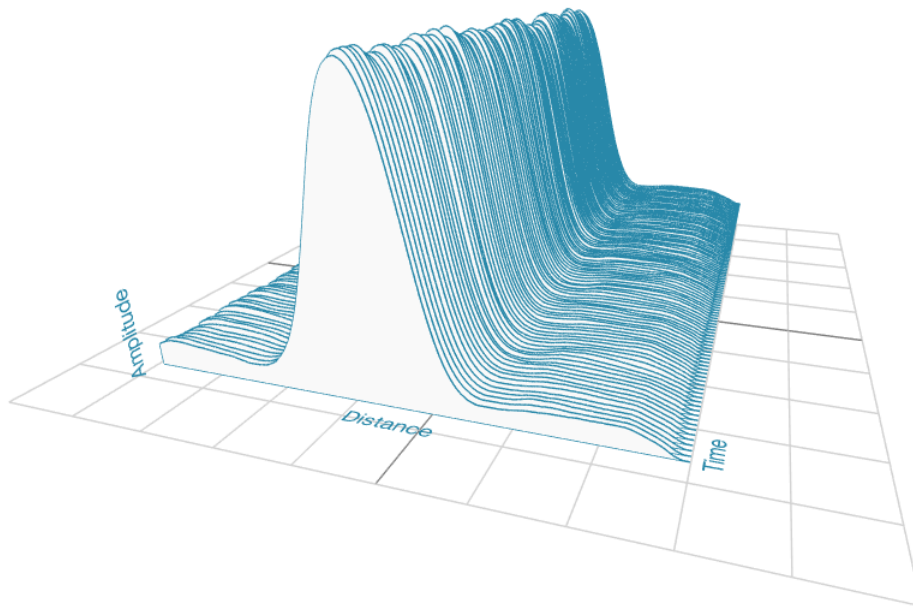
Fig. 2.3: Output from the Envelope service in Exploration Tool.

## IQ service

The IQ Service provides complex data in cartesian form, which is shown in Fig. 2.4 with distance on the third axis and data taken with the same setup as for Envelope and Power bins in Fig. 2.2 and Fig. 2.3.

The cartesian data can be transformed to polar data providing phase and amplitude of the signal. Having the phase of the signal available makes it possible to perform more accurate measurements as compared to the Power bins and Envelope Services where only the amplitude is available. This is illustrated in Fig. 2.5 where an object is moving towards the radar. The envelope of the signal only varies slightly when the object is moving, while the value of the coherent signal at a fixed time delay varies substantially. This change will be present in the phase of the data from the IQ Service.

The IQ Service is the choice when high accuracy is required, and higher processing power and memory allocation can be tolerated.
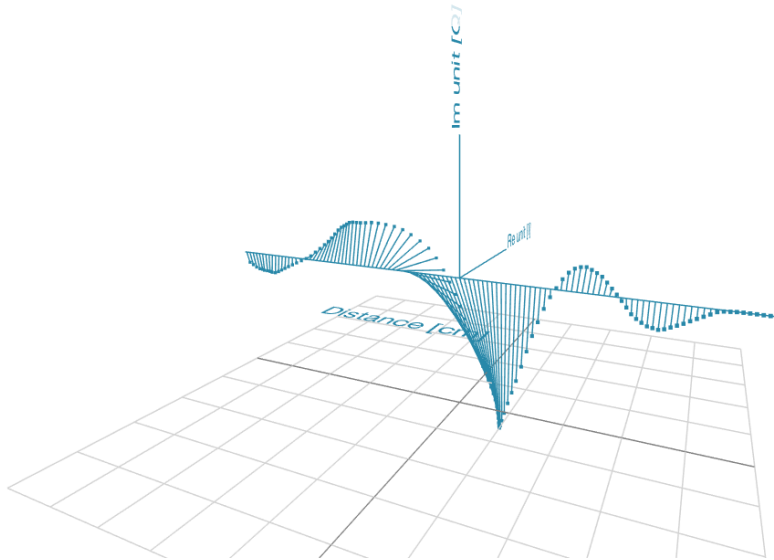
Fig. 2.4: Output from the IQ Service in Exploration Tool.



Fig. 2.5: Illustration of envelope and phase change of a received pulse for a reflection from a moving object, what is returned from the IQ Service is in cartesian form.

### Sparse service

The other services, envelope-service, iq-service, and pb-service, are all based on sampling the incoming waves several times per wavelength (effectively ~2.5 mm). In the Sparse service, the incoming waves are instead sampled approximately every 6 cm and the amount of processing is minimal, which makes Sparse data fundamentally different from data generated by the other services.
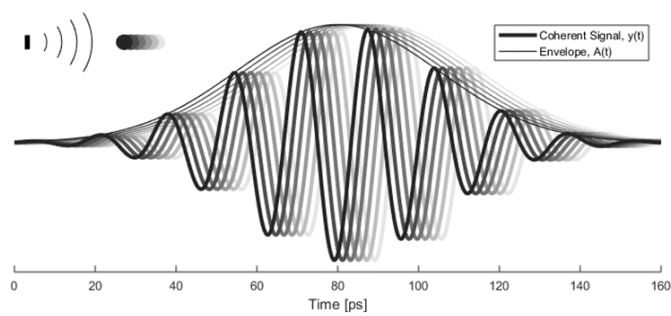
Due to the highly undersampled signal from the sparse service, it should not be used to measure the reflections of static objects. Instead, the sparse service should be used for situations, where detecting moving objects is desired. Sparse is optimal for this, as it produces sequences of very time accurate measurements at these sparsely located sampling points. More details here.

## 2.2 Detectors

Detectors take Service data as input and produce a result as the output that can be used by the application. Currently we have four Detectors available that produce different types of results and that are based on different Services. User guides for the different Detectors are available at acconeer.com and the Detectors are also available in the Exploration Tool.

In addition, we provide several Reference applications which use Services or Detectors to demonstrate how to develop applications based on our technology, you can find these in the various SDKs at Acconeer developer site.

### Distance detector

This is a distance detector algorithm built on top of the envelope-service service – based on comparing the envelope sweep to a threshold and identifying one or more peaks in the envelope sweep, corresponding to objects in front of the radar. The algorithm both detects the presence of objects and estimates their distance to the radar. More details about the detector is found here.

### Presence detector

Detects changes in the environment over time based on data from the Sparse service. More details about the detector is found here.

### Obstacle detector

Assumes that the Acconeer sensor is placed on a moving object with a known velocity, such as a robotic vacuum cleaner or lawn mower. The detector creates a virtual antenna array and uses synthetic aperture radar (SAR) signal processing to localize objects. This detector is used in the Obstacle localization demo movie. More details about the detector is found here.

# 3 System overview

The Acconeer sensor is a mm wavelength pulsed coherent radar, which means that it transmits radio signals in short pulses where the starting phase is well known, as illustrated in Fig. 3.1.
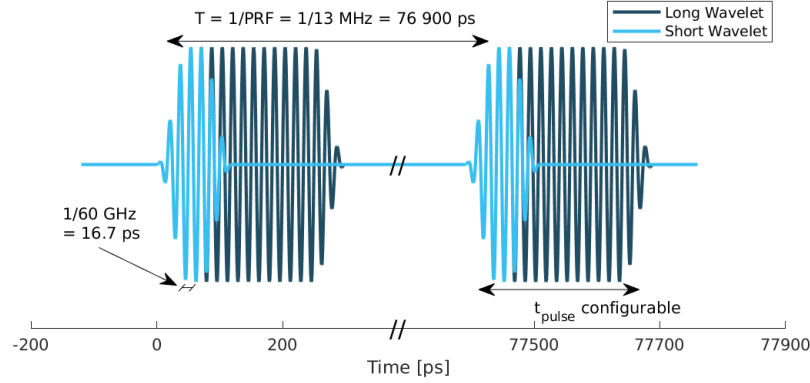


Fig. 3.1: Illustration of the time domain transmitted signal from the Acconeer A111 sensor, a radar sweep typically consists of thousands of pulses. The length of the pulses can be controlled by setting Profile.

These transmitted signals are reflected by an object and the time elapsed between transmission and reception of the reflected signal ($t_{delay}$) is used to calculate the distance to the object by using

$$d = \frac{t_{delay}v}{2} \tag{3.1}$$

$$v = \frac{c_0}{\sqrt{\varepsilon_r}} \tag{3.2}$$

where $\varepsilon_r$ is the relative permittivity of the medium. The '2' in the denominator of (3.1) is due to the fact that $t_{delay}$ is the time for the signal to travel to the object and back, hence to get the distance to the object a division by 2 is needed, as illustrated in Fig. 1.1. As $f_{RF}$ is 60.5 GHz one wavelength ($\lambda$) is roughly 5 mm, which then corresponds to a distance to the object of 2.5 mm.

Fig. 3.2 shows a block diagram of the A111 sensor. The signal is transmitted from the Tx antenna and received by the Rx antenna, both integrated in the top layer of the A111 package substrate. In addition to the mmWave radio the sensor consists of power management and digital control, signal quantization, memory and a timing circuit.

Fig. 3.3 shows a typical radar sweep obtained with the Envelope Service, with one object present. The range resolution of the measurement is ~0.5 mm and each data point correspond to transmission of at least one pulse (depending on averaging), hence, to sweep 30 cm, e.g. from 20 cm to 50 cm as in Fig. 3.3, requires that 600 pulses are transmitted. The system relies on the fact that the pulses are transmitted phase coherent, which makes it possible to send multiple pulses and then combine the received signal from these pulses to improve signal-to-noise ratio (SNR) to enhance the object visibility.
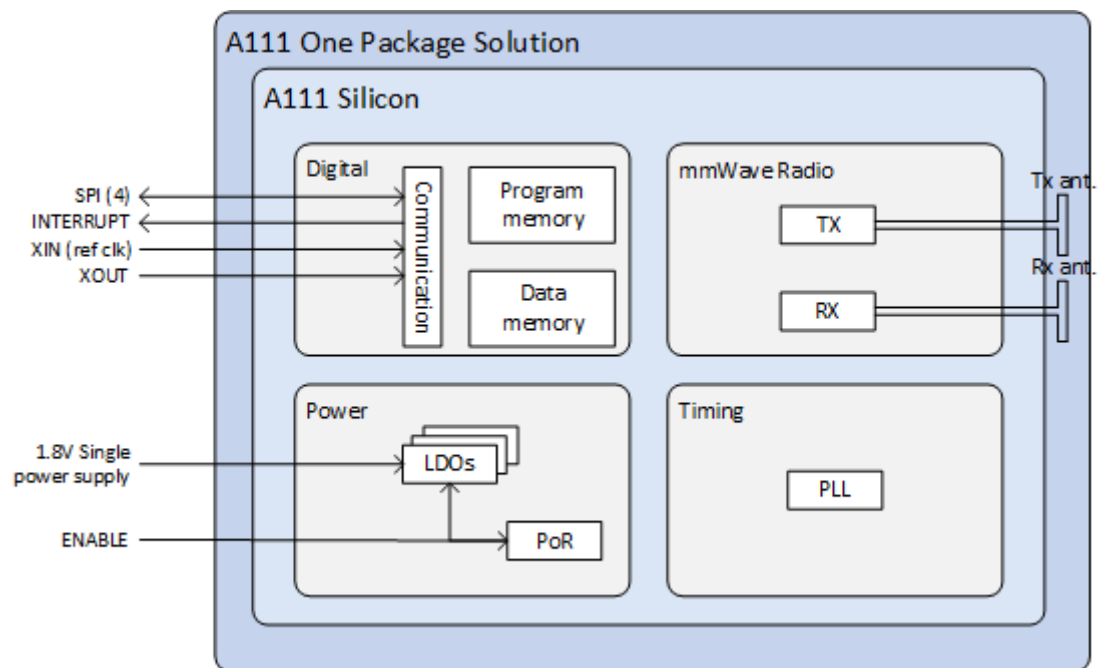
Fig. 3.2: Block diagram of the A111 sensor package, further details about interfaces can be found in the A111 data sheet.
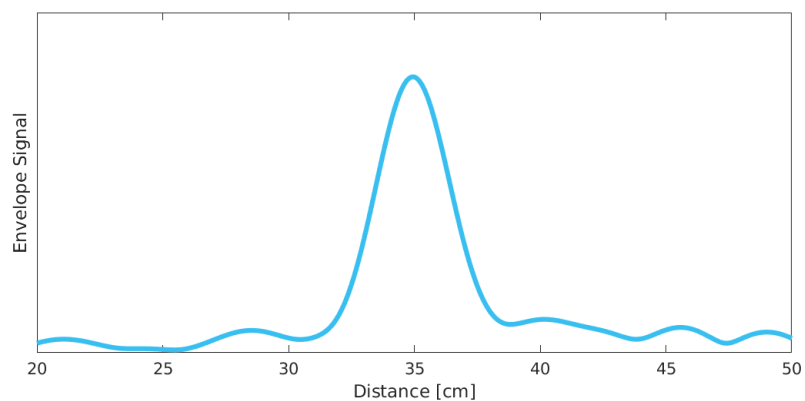


Fig. 3.3: Output from Envelope service for a typical radar sweep with one object present.

## 3.1 Reflectivity

The amount of energy received back to the Rx antenna depends on the reflectivity of the object ($\gamma$), the radar cross section (RCS) of the object ($\sigma$), and the distance to the object ($R$). A reflection occurs when there is a difference in relative permittivity between two media that the signal is propagating through. $\gamma$ is then given as

$$\gamma = \left( \frac{\sqrt{\varepsilon_1} - \sqrt{\varepsilon_2}}{\sqrt{\varepsilon_1} + \sqrt{\varepsilon_2}} \right)^2 \tag{3.3}$$

where $\varepsilon_1$ and $\varepsilon_2$ is the relative permittivity, at 60 GHz, on either side of the boundary. The relative permittivity for common materials can be found in various data bases, but keep in mind that it is frequency dependent. As an example, Table 3.1 lists approximate values for the real part of the relative permittivity for some common materials.

Table 3.1: Relative permittivity of common materials

| Material | Real($\varepsilon$) at 60 GHz | $\gamma$ with air boundary |
|---|---|---|
| ABS plastic | 2.48 | 0.049 |
| Mobile phone glass | 6.9 | 0.02 |
| Plaster | 2.7 | 0.059 |
| Concrete | 4 | 0.11 |
| Wood | 2.4 | 0.046 |
| Textile | 2 | 0.029 |
| Metal | – | 1 |
| Human skin | 8 | 0.22 |
| Water | 11.1 | 0.28 |
| Air | 1 | 0 |

Table 3.1 shows that some materials are semi-transparent to 60 GHz signals and it is hence possible to detect reflecting objects behind a surface of these materials, each boundary with a change in permittivity gives a reflection. This is a useful property in applications where the use case requires that the sensor measures through, e.g., a wall, clothing or plastic housing.

## 3.2 Radar cross section

The radar cross section is the effective area of the object that the signal is reflected against, for simple geometrical shapes, where the size is larger than the wavelength of the signal (~5 mm) and is in the far-field distance, it can be expressed analytically as in Fig. 3.4. The far-field distance depends on the object size and its distance to the radar source. Generally speaking, far-field applies when the waves reflected by the object can be considered plane-waves. Representative back scattering pattern of a sphere, flat plate and trihedral corner reflector are shown in the polar plots. It is seen that the objects can have different maximum RCS, but also different radiation patterns, a flat plate for instance is very directive and if tilted away from the radar, the received energy will be decreased, whereas the corner has less angular dependence and is a more robust reflector in terms of angle with respect to the radar.

For most objects it is not possible to analytically calculate $\sigma$, instead it needs to be measured or modelled.

$$\sigma_{max} = \pi R^2 \qquad \sigma_{max} = \frac{4\pi w^2 h^2}{\lambda^2} \qquad \sigma_{max} = \frac{4\pi L^4}{3\lambda^2}$$
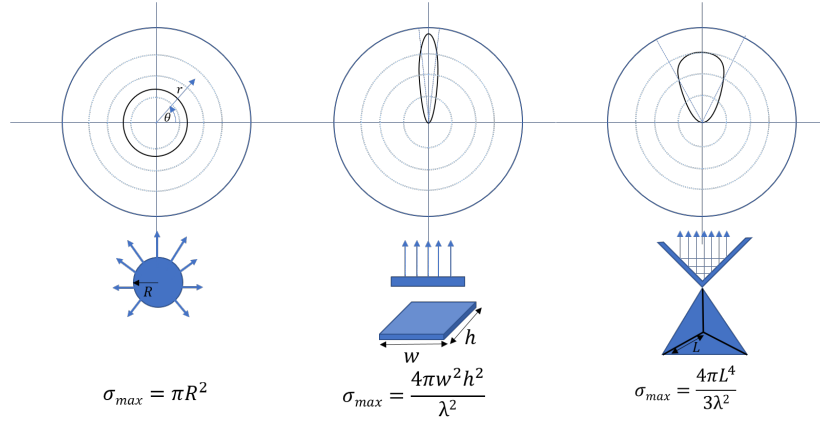
Fig. 3.4: Radiation pattern and analytical expressions for simple geometrical shapes.

## 3.3 Typical ranges for different objects

In Table 3.2 and Table 3.3 the visibility for a range of objects with common shapes (cylinder, plate, etc.) and of varying reflectivity, i.e. materials, is shown. Objects are at normal incidence and the governing system parameters are $\sigma$, $\gamma$, and C, as shown in Eq. (3.4). The envelope service was used to collect the data with Profile 2. The object counts as distinguishable from the noise with a SNR > 10 dB (Y), barely visible between 5 dB and 10 dB (-) and not visible with a SNR < 5 dB (N). The range can be further increased based on the configuration of the sensor, as described in Section *Configuring the Acconeer sensor* and by optimizing the physical integration, as will be described in Section *Physical integration aspects*. As an example for such an optimization Table 3.2 shows results with an added radar Fresnel lens.

Table 3.2: Typical ranges using the envelope service and Profile 2, **without radar lens**.

| Object | 0.5 m | 1 m | 2 m | 5 m | 10 m |
|---|---|---|---|---|---|
| Corner reflector ($a$ = 4 cm) | Y | Y | Y | Y | N |
| Planar water surface | Y | Y | Y | Y | Y |
| Disc ($r$ = 4 cm) | Y | Y | Y | Y | Y |
| Cu Plate (10x10 cm) | Y | Y | Y | Y | Y |
| PET plastic Plate (10x10 cm) | Y | Y | Y | Y | – |
| Wood Plate (10x10 cm) | Y | Y | – | N | N |
| Cardboard Plate (10x10 cm) | Y | Y | Y | N | N |
| Al Cylinder ($h$ = 30, $r$ = 2 cm) | Y | Y | – | N | N |
| Cu Cylinder ($h$ = 12, $r$ = 1.6 cm) | Y | Y | Y | N | N |
| PP plastic Cylinder ($h$ = 12, $r$ = 1.6 cm) | Y | N | N | N | N |
| Leg | Y | Y | – | N | N |
| Hand (front) | Y | Y | N | N | N |
| Torso (front) | Y | Y | Y | N | N |
| Head | Y | Y | N | N | N |
| Glass with water ($h$ = 8.5, $r$ = 2.7 cm) | Y | Y | N | N | N |
| PET Bottle with water ($h$ = 14, $r$ = 4.2 cm) | Y | Y | N | N | N |
| Football | Y | Y | N | N | N |

Table 3.3: Typical ranges using the envelope service and Profile 2, **with 7 dB radar lens**.

| Object | 0.5 m | 1 m | 2 m | 5 m | 10 m |
|---|---|---|---|---|---|
| Corner reflector ($a$ = 4 cm) | Y | Y | Y | Y | Y |
| Planar water surface | Y | Y | Y | Y | Y |
| Disc ($r$ = 4 cm) | Y | Y | Y | Y | Y |
| Cu Plate (10x10 cm) | Y | Y | Y | Y | Y |
| PET plastic Plate (10x10 cm) | Y | Y | Y | Y | Y |
| Wood Plate (10x10 cm) | Y | Y | Y | Y | N |
| Cardboard Plate (10x10 cm) | Y | Y | Y | Y | – |
| Al Cylinder ($h$ = 30, $r$ = 2 cm) | Y | Y | Y | Y | – |
| Cu Cylinder ($h$ = 12, $r$ = 1.6 cm) | Y | Y | Y | Y | – |
| PP plastic Cylinder ($h$ = 12, $r$ = 1.6 cm) | Y | Y | Y | N | N |
| Leg | Y | Y | Y | Y | N |
| Hand (front) | Y | Y | Y | N | N |
| Torso (front) | Y | Y | Y | Y | N |
| Head | Y | Y | Y | – | N |
| Glass with water ($h$ = 8.5, $r$ = 2.7 cm) | Y | Y | Y | – | N |
| PET Bottle with water ($h$ = 14, $r$ = 4.2 cm) | Y | Y | Y | N | N |
| Football | Y | Y | Y | N | N |

## 3.4 Radar sensor performance metrics

Radar sensor performance metrics (RSPMs) for the Acconeer radar system provides useful information on the performance of the system: sensor, RSS and reference integration. The list contains the RSPMs that are applicable to services that produce radar data. However, not all RSPMs are applicable to all radar services. The RSPMs is used in our Radar Datasheet.

### Radar loop gain

The SNR can be modelled as a function of a limited number of parameters: the RCS of the object ($\sigma$), the distance to the object ($R$), the reflectivity of the object ($\gamma$), and a radar sensor dependent constant referred to as radar loop gain ($C$). The SNR (in dB) is then given by

$$\mathrm{SNR}_{dB} = 10\log_{10}\frac{S}{N} = C_{dB} + \sigma_{dB} + \gamma_{dB} - k10\log_{10} R \qquad (3.4)$$

Fig. 3.5 shows how the received energy drops with increasing $R$ for objects where the exponent $k$ is equal to 4, which applies for objects which are smaller than the area which is illuminated coherently by the radar. For objects that are larger than this area the $k$ is smaller than 4, with a lower limit of $k = 2$ when the object is a large flat surface.
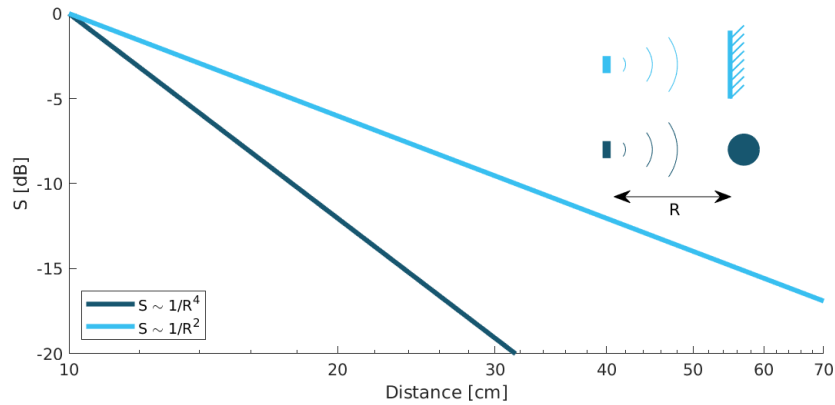
Fig. 3.5: Received signal power versus distance. Note: signal, S, is plotted in dB.

## Depth resolution

The depth resolution determines the minimum distance of two different objects in order to be distinguished from each other.

## Distance resolution

The Acconeer radar systems are based on a time diluted measurement that splits up as a vector of energy in several time bins it is important to know the bin separation. This is the delay resolution of the system and in A111 radar sensor the target is ~3 ps on average, which corresponds to a distance resolution of ~0.5 mm between distance samples.

## Half-power beamwidth

The half-power beamwidth (HPBW) radiation pattern determines the angle between the half-power (-3 dB) points of the main lobe of the radiation pattern. The radiation pattern of the sensor depends on both the antenna-in-package design and the hardware integration of the sensor, such as surrounding components, ground plane size, and added di-electric lenses for directivity optimizations, valid for both vertical and horizontal plane.

## Distance jitter

The distance jitter determines the timing accuracy and stability of the radar system between sweep updates. The jitter is estimated by calculating the standard deviation of the phase, for the same distance bin, over many IQ sweeps.

**Distance linearity**

The distance linearity deterministic the deterministic error from the ideal delay transfer function. Linearity of the service data is estimated by measuring the phase change of the IQ data vs distance.

**Update rate accuracy**

The update rate accuracy determines the accuracy of the time between sweep updates or similarly the accuracy of the update rate, typically important when the radar data is used for estimating velocity of an object.

**Close-in range**

The close-in range determines the radar system limits on how close to the radar sensor objects can be measured.

**Power consumption**

The power consumption determines the radar sensor power usage for different configurations as service depends, the power save mode, the update rate, downsampling, sweep length, etc.

# 4 Configuring the Acconeer sensor

The Acconeer sensor is highly configurable and can operate in many different modes where parameters are tuned to optimize the sensor performance for specific use cases.

## 4.1 Profiles

The first step is to select pulse length profile to optimize on either depth resolution or radar loop gain, or in terms of use cases, optimized for multiple objects/close range or for weak reflections/long range, respectively.

Depth resolution, $d_{res}$, is the ability to resolve reflections which are closely spaced, and hence depends on $t_{pulse}$ according to

$$d_{res} \approx \frac{t_{pulse} v}{2} \tag{4.1}$$

Fig. 4.1 illustrates how the ability to resolve closely spaced reflections can be improved by decreasing $t_{pulse}$. On the other hand, decreasing $t_{pulse}$ means that the total energy in the pulse is decreased and hence decrease the SNR in the receiver, this is the trade-off that is made by selecting between the five profiles. Each service can be configured with five different pulse length profiles (see Table 4.1), where

- shorter pulses provides higher distance resolution at the cost of a reduced SNR

- longer pulses provides higher SNR at a cost of reduced depth resolution

**20**
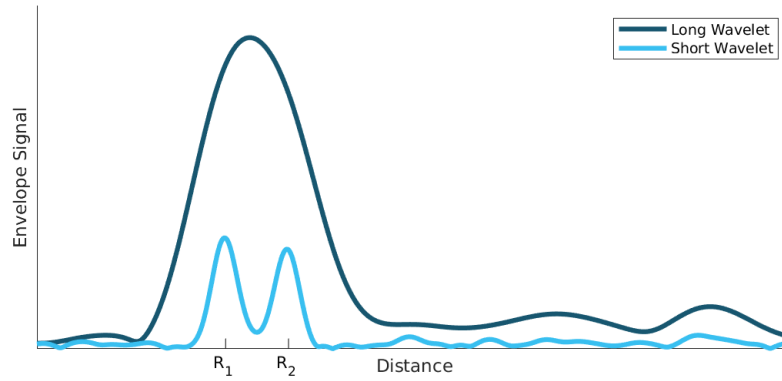
Fig. 4.1: Illustration of received signal containing 2 echoes. A longer pulse increases the radar loop gain, but also limits the depth resolution. The displayed data corresponds to the two setups in Fig. 4.2.



Fig. 4.2: Illustration of scenarios that can produce the data in Fig. 4.1. A strong reflector, such as a flat metallic surface, can give a moderate radar signal if the angle to the radar is high. $R_1$ is identical in the two illustrations as well as $R_2$.

Optimizing on depth resolution also means that close-in range performance is improved. The A111 sensor has both the Tx and Rx antenna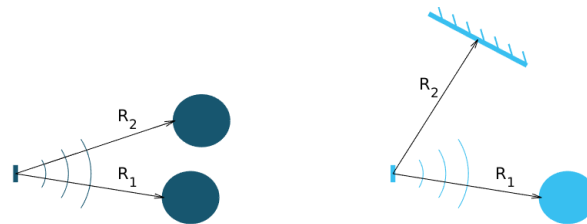 integrated and since they are so closely spaced, there will be leakage between the two antennas. This means that any object close to the sensor will have to be filtered from this static leakage. The ability to do this is improved if a short $t_{pulse}$ is used, as illustrated in Fig. 4.3.

If angular information is needed one possibility is to mechanically move the sensor to scan an area and produce a synthetic aperture radar (SAR). One such case is for autonomous robots using sensor input for navigation. Another option is to use multiple A111 sensors and merge data from them to calculate the position of the object by trilateration. This can be achieved by running the sensors sequentially and merge the data in the application.
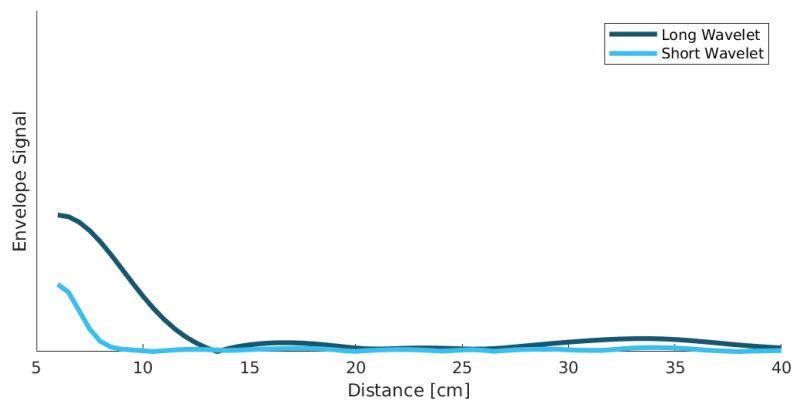


Fig. 4.3: Illustration of how the leakage between the Tx and Rx antenna will appear in the Envelope Service data for Profile 1 and Profile 2 pulse lengths.

Table 4.1: **Rough** comparison of the envelope service behavior for different profiles.

| Profile | Relative SNR improvement [dB] | Direct leakage [m] |
|---|---|---|
| Profile 1 | 0 | ~0.06 |
| Profile 2 | ~7 | ~0.10 |
| Profile 3 | ~11 | ~0.18 |
| Profile 4 | ~13 | ~0.36 |
| Profile 5 | ~16 | ~0.60 |

## 4.2 Signal averaging and gain

In addition to the Profile configuration parameter, two main configuration parameters are available in all Services to optimize the signal quality:

- Hardware Accelerated Average Samples (HWAAS) is related to the number of pulses averaged in the radar to produce one data point. A high number will increase the radar loop gain but each sweep will take longer to acquire and therefore limit the maximum update rate.

- The gain of the amplifiers in the sensor. Adjusting this parameter so the ADC isn't saturated and at the same time the signal is above the quantization noise is necessary. A gain figure of 0.5 is often a good start.

## 4.3 Sweep and update rate

A sweep is defined as a distance measurement range, starting at the distance *start range* and continues for *sweep length*. Hence, every sweep consists of one or several distance sampling points.

A number of sweeps $N_s$ are sampled after each other and the time between each sweep is $\Delta t_s$, which is configurable. We usually refer to this as the *update rate* $f_s = 1/\Delta t_s$.

In addition, the sparse service introduces a concept of frames defined here.

## 4.4 Repetition modes

RSS supports two different *repetition modes*. They determine how and when data acquisition occurs. They are:

- **On demand**: The sensor produces data when requested by the application. Hence, the application is responsible for timing the data acquisition. This is the default mode, and may be used with all power save modes.

- **Streaming**: The sensor produces data at a fixed rate, given by a configurable accurate hardware timer. This mode is recommended if exact timing between updates is required.

Note, Exploration Tool is capable of setting the update rate also in *on demand* mode. Thus, the difference between the modes becomes subtle. This is why *on demand* and *streaming* are called *host driven* and *sensor driven* respectively in Exploration Tool.

## 4.5 Power save modes

The power save mode configuration sets what state the sensor waits in between measurements in an active service. There are five power save modes, see Table 4.2. The different states differentiate in current dissipation and response latency, where the most current consuming mode *Active* gives fastest response and the least current consuming mode *Off* gives the slowest response. The absolute response time and also maximum update rate is determined by several factors besides the power save mode configuration. These are profile, length, and hardware accelerated average samples. In addition, the host capabilities in terms of SPI communication speed and processing speed also impact on the absolute response time. Nonetheless, the relation between the power save modes are always kept such that *Active* is fastest and *Off* is slowest.

Another important aspect of the power save mode is when using the service in repetition mode Streaming. In streaming mode the service is also configured with an update rate at which the sensor produces new data. The update rate is maintained by the sensor itself using either internally generated clock or using the externally applied clock on XIN/XOUT pins. Besides the fact that power save mode *Active* gives the highest possible update rate, it also gives the best update rate accuracy. Likewise, the power save mode *Sleep* gives a lower possible update rate than *Active* and also a lower update rate accuracy. Bare in mind that also in streaming mode the maximum update rate is not only determined by the power save mode but also profile, length, and hardware accelerated average samples. Power save mode *Off* and *Hibernate* is not supported in streaming mode since the sensor is turned off between its measurements and thus cannot keep an

update rate. In addition, the power save mode *Hibernate* is only supported when using Sparse service.

Table 4.2 concludes the power save mode configurations.

Table 4.2: Power save modes.

| Power save mode | Current consumption | Response time | Update rate accuracy |
|---|---|---|---|
| Off | Lowest | Longest | Not applicable |
| Hibernate | . . . | . . . | Not applicable |
| Sleep | . . . | . . . | Worst |
| Ready | . . . | . . . | . . . |
| Active | Highest | Shortest | Best |

As part of the deactivation process of the service the sensor is disabled, which is the same state as power save mode *Off*.

## 4.6 Configuration summary

Table 4.3 shows a list of important parameters that are available through our API and that can be used to optimize the performance for a specific use case, refer to product documentation and user guides for a complete list of all parameters and how to use them.

Table 4.3: List of sensor parameters

| Parameter | Comment |
|---|---|
| Profile | Selects between the pulse length profiles. Trade off between SNR and depth resolution. |
| Start | Start of sweep [m]. |
| Length | Length of sweep, independently of Start range [m]. |
| HWAAS | Amount of radar pulse averaging in the sensor. |
| Receiver gain | Adjust to accommodate received signal level. |
| Repetition mode | On demand or Streaming. |
| Update rate | Desired rate at which sweeps are generated [Hz] (in repetition mode Streaming). |
| Power save mode | Tradeoff between power consumption and rate and accuracy at which sweeps are generated. |

## 5 Physical integration aspects

The A111 sensor contains the mmWave front-end, digital control logic, digitization of received signal and memory, all in one package. To integrate it in your application it is required to have a reference frequency or XTAL (20-80 MHz), 1.8 V supply, and a host processor, as illustrated in Fig. 5.1, supported platforms and reference schematics are available at developer.acconeer.com.

In addition to the above it is also important for optimized integration to consider the electromagnetic (EM) environment, both in terms of what is placed on top of the sensor as
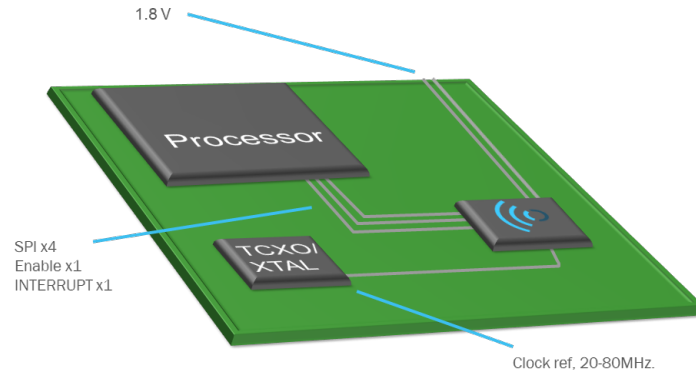
Fig. 5.1: Illustration of integration into host platform, the A111 is marked with the Acconeer logo.

well as to the side of the sensor. To evaluate the EM integration a Radar loop measurement can be conducted by placing an object in front of the sensor and rotating the sensor around its own axis, as illustrated in Fig. 5.2. The received energy from e.g. the Envelope Service can then be used to plot the amplitude versus rotation angle ($\theta$).
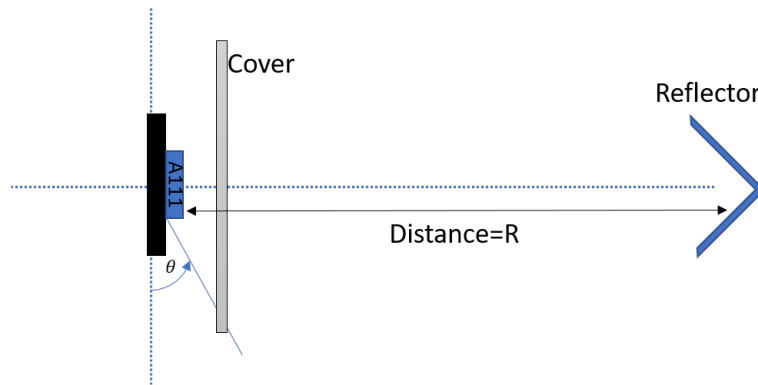


Fig. 5.2: Setup configuration for radar loop pattern measurements.

The radiation pattern of the integrated antennas will be affected by anything that is put on top of the sensor as a cover. The transmission through a material is given by 1-$\gamma$, where $\gamma$ is the reflectivity calculated in Equation 3. Hence, materials with low reflectivity are good materials to use as a cover on top of the sensor, plastic is a good choice and the sensor is not sensitive to the color of the material. Figure 21 shows the measured Radar loop pattern for 3 different scenarios, plastic (ABS), gorilla glass (GorillaGlass) and free space (FS). To further optimize the cover integration the thickness of the material should be considered. One can also use a layered cover which uses materials of different $\varepsilon$ for optimum matching to the medium in which the signal is going to propagate or even to increase the directivity, as shown in Figure 21, where the beam width has been decreased by adding material on top of the sensor. More information on the EM integration aspects can be found in "Electromagnetic Integration - Basic Guidelines" document available at developer.acconeer.com.
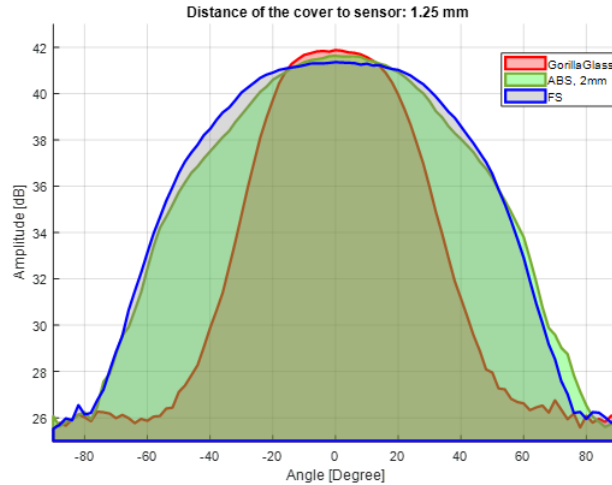
Fig. 5.3: Integration of sensor cover and how different materials impact the radiation pattern on the H-plane. The object used is a trihedral corner of radius 5 cm.

# 6 Summary

Acconeer's Pulsed coherent radar technology is unique as it combines high precision and low power consumption into a tiny package and for the first time enables radar in products where size, cost and power consumption matters. We are committed to making the technology available to everyone and we are working hard to make it easy for you to take your product to the market, whether you need pre-integrated hardware or new Detectors we will help you to get the product to your customers. Sign up for our newsletter or check out our website and Github for updates on new cool features that we have released, we are constantly innovating, **"Explore the next sense!"**.

# 7 Disclaimer

The information herein is believed to be correct as of the date issued. Acconeer AB (**"Acconeer"**) will not be responsible for damages of any nature resulting from the use or reliance upon the information contained herein. Acconeer makes no warranties, expressed or implied, of merchantability or fitness for a particular purpose or course of performance or usage of trade. Therefore, it is the user's responsibility to thoroughly test the product in their particular application to determine its performance, efficacy and safety. Users should obtain the latest relevant information before placing orders.

Unless Acconeer has explicitly designated an individual Acconeer product as meeting the requirement of a particular industry standard, Acconeer is not responsible for any failure to meet such industry standard requirements.

Unless explicitly stated herein this document Acconeer has not performed any regulatory conformity test. It is the user's responsibility to assure that necessary regulatory conditions are met and approvals have been obtained when using the product. Regardless of whether the product has passed any conformity test, this document does not constitute any regulatory approval of the user's product or application using Acconeer's product.

Nothing contained herein is to be considered as permission or a recommendation to infringe any patent or any other intellectual property right. No license, express or implied,

to any intellectual property right is granted by Acconeer herein.

Acconeer reserves the right to at any time correct, change, amend, enhance, modify, and improve this document and/or Acconeer products without notice.

This document supersedes and replaces all information supplied prior to the publication hereof.

# 8 Document history

| Author | Comments | Version | Date |
|---|---|---|---|
| Acconeer AB | Product sw offer figure updated. | 2.7 | 2020-09-29 |
| Acconeer AB | Product offer figure updated. | 2.6 | 2020-09-28 |
| Acconeer AB | Updated with new distance detector. | 2.5 | 2020-08-14 |
| Acconeer AB | Minor fixes. | 2.4 | 2020-05-27 |
| Acconeer AB | Minor fixes. | 2.3 | 2020-03-13 |
| Acconeer AB | Minor fixes. | 2.2 | 2020-02-27 |
| Acconeer AB | Added power save mode Hibernate. | 2.1 | 2020-01-17 |
| Acconeer AB | Initial version for API 2.0. | 2.0 | 2019-12-01 |