Digital Beamforming using DEMO BGT60TR13C

For range and angle estimation

About this document

Scope and purpose

The purpose of this document is to describe how the range and angle information can be obtained using Digital Beamforming (DBF). The document provides a fundamental understanding of how the DBF algorithm works and illustrates how the algorithm can be used to obtain an estimate for the range and angle of a target. The algorithm is provided as Python scripts based on the Python wrapper in the Radar Development Kit (RDK).

Intended audience

The intended audience for this document are design engineers, technicians, and developers of electronic systems who are interested in estimating the range and angle of targets using Infineon's XENSIV™ BGT60TR13C radar sensor with digital beamforming.

Revision 1.40
2024-01-15

Table of contents

About this document **1**
Table of contents **2**
1 Introduction **3**
2 Algorithm for Range-Angle Estimation **4**
2.1 Mean Removal 4
2.2 Moving Target Indicator 4
2.3 Range-Doppler 5
2.4 DBF 6
3 Running the Example **8**
Revision history **9**
Disclaimer **10**

1 Introduction

FMCW radar sensors allow measurement of a variety of different values. For example, it is possible to measure the separation between the sensor and a target, the speed of a moving object, or the angle between the radar sensor and a target. Using sophisticated algorithms, it is even possible to track persons, recognize gestures, or detect the heartbeat of a person. This versatility is a key advantage of radar sensors compared to other sensors like passive infrared (PIR) sensors.

But even simple algorithms like a presence detection algorithm open up a lot of use cases. In some cases, however, the information about the state of presence might not be sufficient while a full tracking algorithm might be too computationally expensive or too complex to implement. Digital Beamforming (DBF) can fill the gap, providing both the angle between sensor and target, and the range of the target. Potential applications might include surveillance cameras or air conditioners.

This document describes how the angle between a radar sensor and a target together with the range can be determined using the DBF algorithm. By combining Range-Doppler processing with DBF, the angle and the range of a target can be estimated. An example implementation is provided as Python scripts based on the Python wrapper of the Radar SDK. The example uses a BGT60TR13C radar sensor that is plugged into a Radar Baseboard MCU7, which in turn is connected to a PC via USB.

The implementation range-angle-map.py can be found in the directory radar_sdk\examples\py\BGT60TR13C within the Radar SDK. The Radar SDK is part of the Radar Development Kit which is available via the Infineon Developer Center.

2 Algorithm for Range-Angle Estimation

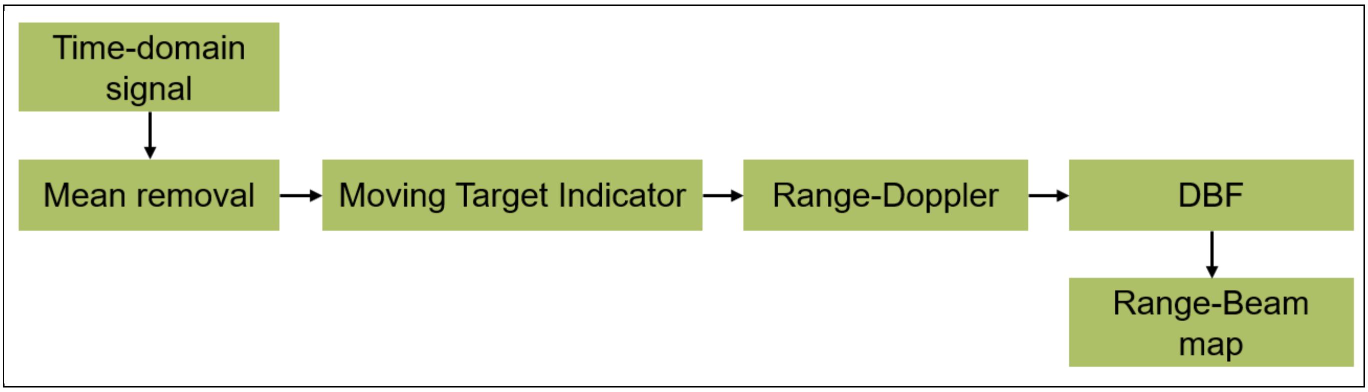

Figure 1 Overview of the algorithmic processing chain

Figure 1 summarizes the algorithmic processing chain. The input is the time-domain signal obtained from the BGT60TR13C radar sensor. In the first step, the mean-value of the input signal is computed and subtracted. A Moving Target Indicator (MTI) is used to eliminate static targets. From the output of the MTI, the Range-Doppler map is computed, which in turn is the input of the DBF algorithm. The output of the DBF is a Range-Beam map, a three-dimensional object which gives the magnitude for all combinations of range and beam number. The range and angle of the target can be obtained from the Range-Beam map through a simple peak search or adaptive peak searching algorithms like CAPON.

In the following, we briefly describe the individual algorithms.

2.1 Mean Removal

Each frame consists of the time-domain data for all receiving antennas, all chirps, and all samples for each chirp. The input signal is normalized to values between 0 and 1. To remove the DC offset, the average over all samples is computed and the average is removed.

2.2 Moving Target Indicator

The Moving Target Indicator (MTI) is a radar targeting method that helps discriminating moving targets from static targets and clutter. The MTI filter is basically a first order Infinite Impulse Response (IIR) filter. It is typically used on a range spectrum to filter out stationary targets. Thus, the name "Moving Target Indication" filtering.

The MTI filtering is realized according to the following recurrence formulae (n=0,1,2,…)

$$F_n = I_n - H_{n-1},$$

$$H_n = \alpha I_n + (1 - \alpha) H_{n-1},$$

where I_n denotes the input, F_n the filtered result, and H_n the history. In each step the input data is filtered, and the history is updated. α is the filter-coefficient of the IIR filter and must be between 0 and 1. Small values of α mean that the filtered output strongly depends on the history and the filter will behave laggy as a result. Vice versa, a value of α close to 1 means that the history hardly has any influence and as a consequence the filter reacts quickly to changes but might also miss to filter out some static targets.

2.3 Range-Doppler

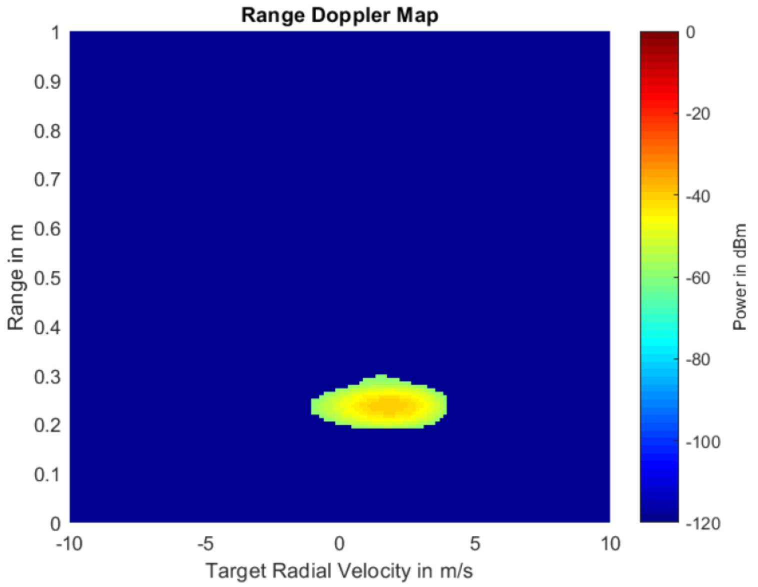

Figure 2 Example of Range-Doppler map

The Range-Doppler map is a 2D representation showing the ranges and velocities of detected objects relative to the radar system. The map's structure is a 2D grid with the range being represented on the ordinate and the velocity being represented on the abscissa as shown. The map's origin is usually located on the bottom side in the center, as there can be no negative values for the range but equally positive and negative values for the velocity. Positive velocity values indicate, that the object is heading away from the radar system, while negative velocity values indicate that the object is approaching the radar system. The detected objects are represented by different colors of a colormap according to their strengths. An example of a Range-Doppler map is displayed in Figure 2.

For detecting not only the range of objects but also their velocity relative to a radar system, a frame of consecutive chirps needs to be sent out towards the object. As the range of a moving object changes linearly over time, the small range differences ΔR from one chirp to the next during the time period T_I can be used to determine the object's velocity v by:

$$v = \frac{\Delta R}{T_I}$$

As the range differences ΔR are usually very small, detecting the change of the IF signal's frequency f_{IF} needs to be accurate in a way that is beyond the signal processing's precision in practical radar systems which are usually based on FFT methods. The frequency Δf resulting from the range differences ΔR from chirp to chirp usually fall within one FFT bin and thus can't be detected as frequency changes directly.

Nevertheless, as the phase φ_{IF} of the IF signal is also proportional to the range R and much more sensitive for small changes, the velocity v can be derived from the IF signal's phase $\Delta\varphi_{IF}$ change from one chirp to the next over time period T_I

$$v = \frac{\lambda}{4\pi T_I} \Delta\varphi_{IF}$$

The underlying data structure used for representing a range-doppler map is a 2D matrix which is spanning a grid. The first dimension represents certain range values, while the second dimension represents certain velocity values. Thus, a range-angle matrix is always a discrete representation, only evaluating predefined range/velocity pairs. The strength of the radar signal, i.e. the presence of objects at the specific point, is represented by the different matrix entries themselves.

2.4 DBF

DBF is a method to focus a radar transmitter or receiver in a certain direction in 3D space. In this context, the left to right direction is commonly referred to as azimuth and the high to low direction as elevation. The antennas of DBF systems are set up as a Uniform Linear Array (ULA) while each direction needs a ULA of at least two antennas positioned in its plane to perform beamforming. Thus, the minimum requirement for detecting azimuth and elevation is a ULA with three antennas in an L shape.

Figure 3 depicts a planar wave reaching a ULA with an Angle of Arrival (AoA) Θ. The antennas are spaced by the distance d. It can be seen that the upper receivers are reached first, resulting in a phase shift between the different antenna signals. In beamforming, the different antennas of the ULA are weighted individually in their phase and amplitude shown in Figure 3 by W_x and summed up at the end. In this active mathematic weighting process, beamforming differs from other analytic algorithms for AoA detection, as it can be compared to applying a windowing function in 3D space.

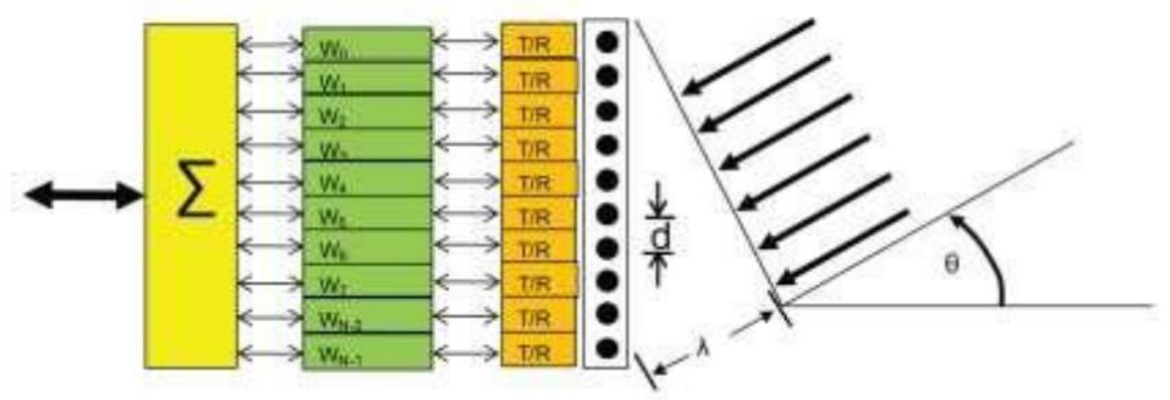

Figure 3 A planar wave arriving at a ULA with subsequent beamforming

For each specific direction of the beam, the applied weights and phase shifts differ so that all other directions are actively set close to zero. As a result, only when the signals of all ULA antennas are shifted in-phase and eventually weighted by 100% the maximum sum over all signals can be generated. While Figure 3 shows a receiver ULA, beamforming can be applied with the same theory for both receive and transmit signals.

In DBF all ULA's antenna signals are sampled and therefore present for processing at the same time. Thus, DBF allows for a precise and simultaneous processing of different angles and frequencies, only limited by the computational effort, but theoretically with arbitrary precision. The theory and implementation of DBF algorithms as described here can not only be used to analyze and weight signals in 3D space, but also for FFT processed spectral representations or derived properties like Range-Doppler maps.

The weighting and shifting of each antenna signal grouped in matrix A as column vectors is usually provided and implemented in matrix form, with w being a row vector containing the individual weights for each antenna signal by:

$$r = Aw.$$

This results in a sum vector r which is referred to as conventional beamforming and is visualized in Figure 4. More sophisticated methods for DBF like Capon and MUSIC are also available, but despite the higher computational effort, they do not always perform better than conventional beamforming.

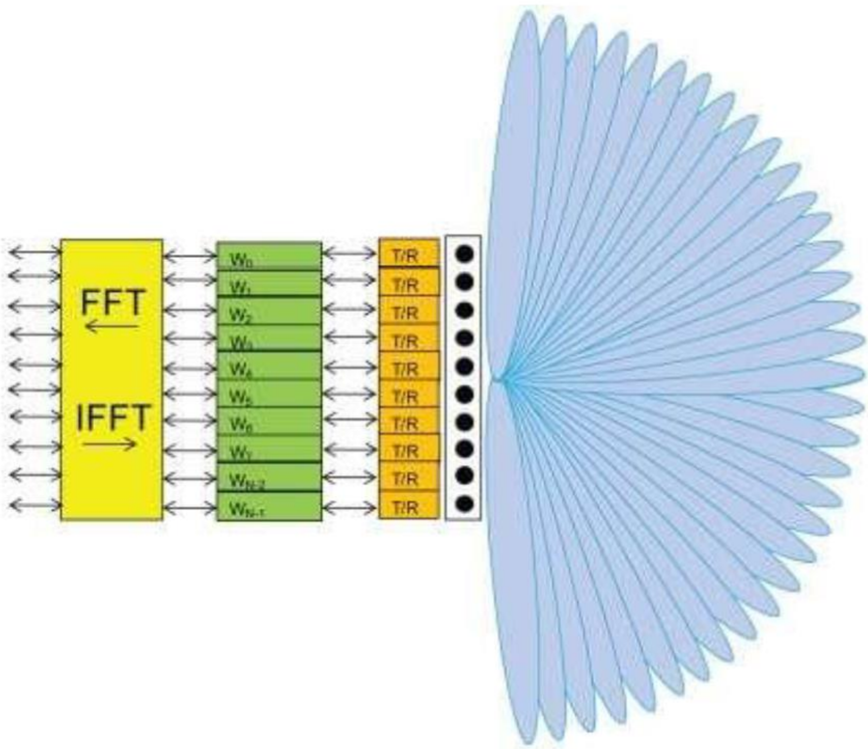

Figure 4 The theory of beamforming visualized with a ULA for elevation

3 Running the Example

An example implementation of the processing chain explained in the previous section is available in Python. The scripts are based on the Radar SDK Python wrapper, which is part of the Radar Development Kit. To run the script, make sure you have a Windows 10 PC with a 64-bit Python installation. Python version 3.7 or later is required. The packages scipy, numpy, and matplotlib need to be installed.

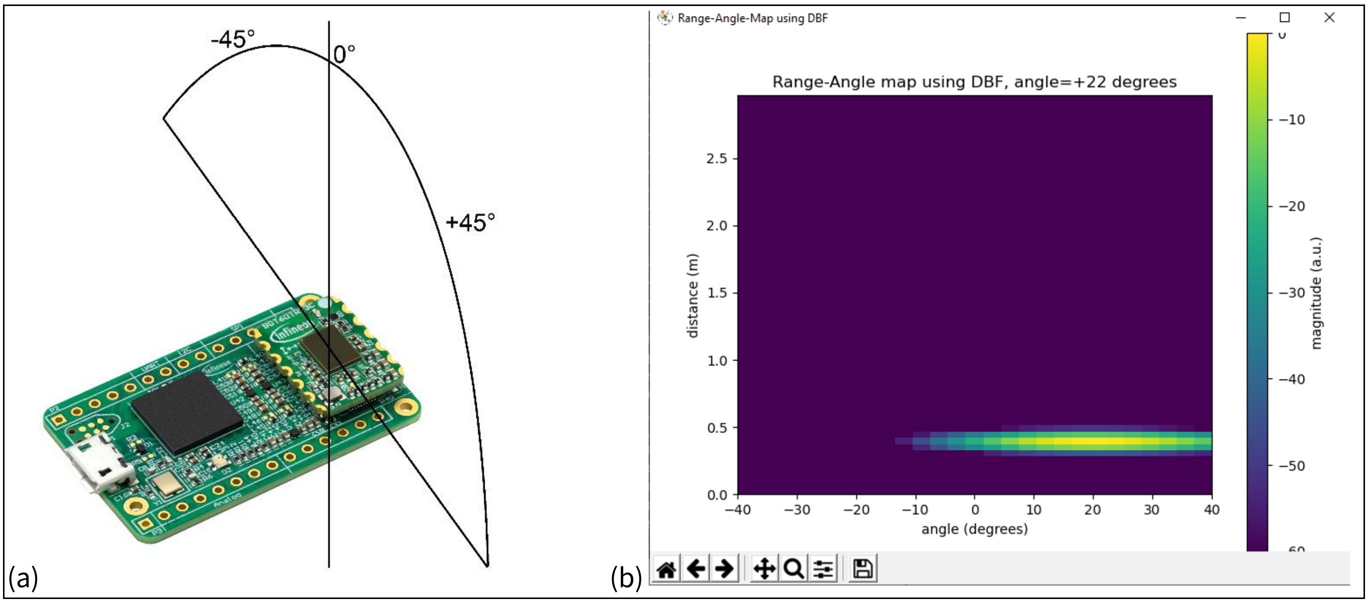

Figure 5 Measured angle (a), and output of range-angle-map.py script (b)

The content of the scripts is listed in Table 1. The main script is range-angle-map.py. Make sure that a Radar Baseboard MCU7 board with a BGT60TR13C radar sensor is connected to your computer and run the script range-angle-map.py. The script creates a live-plot which indicates the range and angle of the target as shown in Figure 5. The plot depicts the magnitude (in arbitrary units) as a function of the angle (in degrees) and the distance (in meters). Brighter colors indicate a stronger signal. In the example depicted in Figure 5, a target is located at a range of about 40 cm with an angle of about 22°.

Table 1 Python scripts and modules used

File	Description
range-angle-map.py	This script connects to a BGT60TR13C radar sensor, sets the radar configuration, fetches the time-domain data, computes the Range-Beam map using DBF, and plots the output in real-time.
DBF.py	Implementation of DBF algorithm described in section 2.4.
doppler.py	Module to compute the Range-Doppler map.
fft_spectrum.py	Computation of FFT spectrum, used by doppler.py.

In the example, we use 27 beams to measure the angle between -40° and +40°. The output of each plot is normalized to its maximum value, visually pronounce the peak. For the sake of brevity, a more sophisticated peak searching algorithm like CAPON or a classifier of the quality of the estimate has not been implemented.

Even though the output only displays the range and the angle of a target, the information about the speed of the target is present as well and can be obtained from the Range-Doppler map.

Revision history

Document revision	Date	Description of changes
1.00	2022-01-31	Initial revision
1.10	2022-02-11	Slight rephrasing, changed some wordings, use abbreviation DBF consistently after the introduction
1.20	2022-04-27	Updated for Radar Development Kit 3.2 release
1.30	2022-08-25	Registered to AN652 Updated for Radar Development Kit 3.3 release
1.40	2024-01-15	Miscellaneous document cleanup updates

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2024-01-15

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AN155322