Universität Rostock
Fakultät für Informatik und Elektrotechnik
Institut für Nachrichtentechnik
Albert-Einstein-Straße 26
D 18059 Rostock

Project Seminar

Lab Manual

Prof. Dr.-Ing. habil. Tobias Weber

January 13, 2023

Abstract

The present lab manual first gives an introduction to the hardware and the software being used in the lab. Furthermore, it describes four lab activities on radar signal processing being an essential part of the project seminar on direction of arrival estimation.

1 Hardware Setup

The objective of the lab activities is to gain some hands on experience in radar signal processing with a focus on direction of arrival estimation.

Two different radar sensors, the TinyRad from Analog Devices [ADI] and the V-MD3 from RFbeam [RFb] will be used. The latter one is based on the IWR6843 chip from Texas Instruments [TI]. With these radar sensors it is possible to stream the raw data from the analog to digital converters to a personal computer. The TinyRad radar sensor uses a USB connection whereas the V-MD3 radar sensor is connected via Ethernet. In the latter case change the IPV4 connection settings on the host computer to a static setting with the IP-Address 192.168.100.1 and the subnet mask 255.255.255.0. Make sure that the firewall does not block the UDP packets. The TinyRad radar sensor operates in the 24 GHz frequency band whereas the V-MD3 radar sensor operates in the 61 GHz frequency band. Both radar sensors use linear frequency modulated continuous wave signals and stretch processing [IR20]. Both radar sensors use fast chirping such that the Doppler effect within a single pulse is negligible. The IWR6843 chip from Texas Instruments being used in the V-MD3 radar sensor contains quadrature demodulators

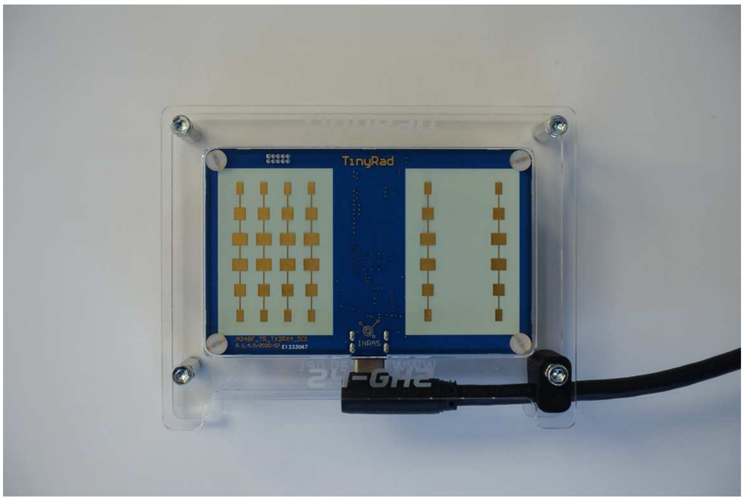


Figure 1: TinyRad radar sensor from Analog Devices. Front side with the antenna array.

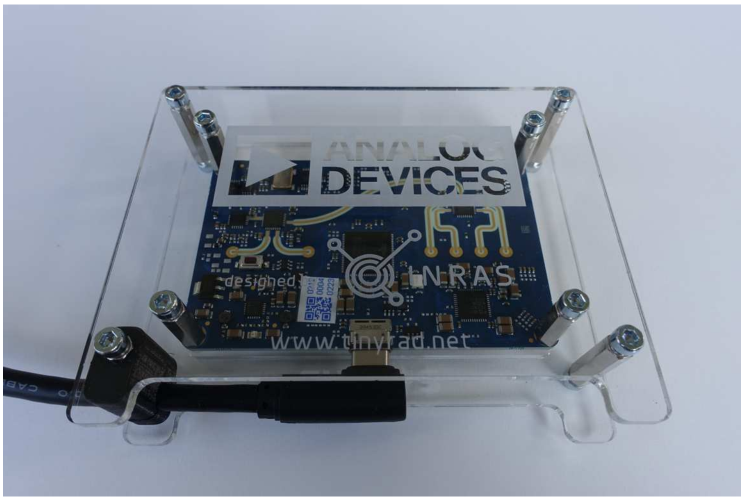
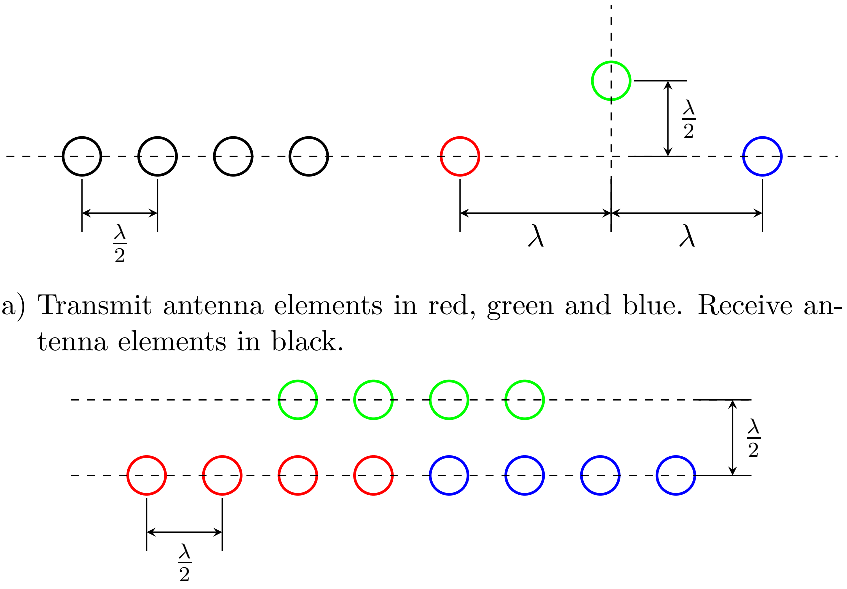


Figure 2: TinyRad radar sensor from Analog Devices. Reverse side with the electronics.

resulting in complex baseband signals. Unfortunately, Analog Devices used only a single mixer per receive antenna element in the TinyRad radar sensor yielding only the real parts of the complex baseband signals. As the frequency shifts are always negative one can reconstruct the imaginary parts of the complex baseband signals using the Hilbert transform.

Both radar sensors have several receive antenna elements forming an antenna array. Furthermore, they also have several transmit antenna elements producing a virtual antenna array with an even larger number of antenna elements [Rao18]. The antenna array of the TinyRad radar sensor is a one dimensional linear array with four receive antenna elements and two transmit antenna elements, see Figure 1. The spacing of the receive antenna elements is $\lambda/2$ and the distance of the two transmit antenna elements is $3\lambda/2$, resulting in a virtual antenna array with a one antenna element overlap. The V-MD3 radar sensor has a third transmit antenna element being displaced in the vertical direction, see Figure 3. This allows for polar angle estimation in addition to the azimuthal angle estimation being possible with one dimensional antenna arrays.



(a) Transmit antenna elements in red, green and blue. Receive antenna elements in black.

(b) Virtual antenna array.

Figure 3: Antenna array of the V-MD3 radar sensor from RFbeam.

Usually great care is taken to connect the antenna elements with equal length wires resulting in no additional relative phase shifts, see Figure 2. Unfortunately, this was not done for the transmit antenna elements when designing the V-MD3 radar sensor. Consequently, a calibration is mandatory when using the V-MD3 radar sensor in MIMO-mode.

Most direction of arrival estimation techniques require several measurements with uncorrelated sources to compute the sample correlation matrix. However, in the radar scenario all targets are illuminated by the same transmitter resulting in correlated sources. One can mimic a spatial scenario with uncorrelated sources by taking measurements at different frequencies if the targets are at different ranges. The different ranges cause different relative phase shifts on the propagation paths so that the targets' received signals superimpose differently at the different frequencies. Consequently, besides averaging over measurements from different pulses which is only beneficial with respect to the noise one should also average over all the measurements from the individual pulses when computing the sample correlation matrix.

2 Software Environment

The radar signal processing will be implemented in Matlab [Mat]. A software environment for testing your algorithms will be provided to you. This software environment handles all the communication with the radar sensors. In the case of the TinyRad radar sensor some low level functions from the subdirectories `DemoRadUsb` and `Class` which are provided by Analog Devices will be used. The documentation of these low level functions is limited to a few examples on how to use them from Matlab. In the case of real time data streaming from a radar sensor a copy of the data will be recorded to a file. Alternatively, one can stream data from a previously recorded file. This simplifies software development and debugging as it is not necessary to repeatedly carry out real world measurements. A set of such recorded data files is provided to you to start your software development. A description of the underlying scenarios can be found in the file `scenarios.txt`. You shall test your algorithms with several of these data files.

In detail the following functions will be used for data streaming:

- The data streaming is initialized by the function `startstreaming.m`. Here you have to specify the `mode` and the file name for the data. There is set of modes `'adisimo'`, `'adimimo'`, `'rfbsimo'` and `'rfbmimo'` for streaming data in real time from different radar sensors using different antenna configurations. Furthermore, there are corresponding modes `'adisimofile'`, `'adimimofile'`, `'rfbsimofile'` and `'rfbmimofile'` for streaming data from previously recorded files. Take care to use the mode matching to the format of the file. Also be aware that files in the `.h5`-format which is used by the TinyRad radar sensor can not simply be overwritten when repeating measurements, i.e., you have to manually delete the file first.

- The function `getframe.m` retrieves the next data chunk containing measurements from one frame of pulses. The function `getframe.m` returns one data cube of samples from the complex baseband signals, i.e., in the case of the TinyRad radar sensor it also reconstructs the imaginary parts. Furthermore, the function `getframe.m` applies the calibration. In case of the

TinyRad radar sensor the calibration data provided by Analog Devices is retrieved from the radar sensor whereas in the case of the V-MD3 radar sensor it is stored in the file `caldata.mat`. Due to the high data rates it may easily happen that some data gets lost during real time data streaming. In this case the software will try to resynchronize the communication with the radar sensor.

- The data streaming is stopped by the function `stopstreaming.m`.

3 Range-Doppler Processing

The objective of this first lab activity is to recapitulate conventional radar signal processing in the fast time domain and in the slow time domain to determine a target's range and velocity. Later on this could be combined with spatial domain signal processing being the focus of the following lab activities. The places in the program code in `rangedoppler.m` where you have to make your changes are marked by comments in capital letters like this `% ADD YOUR CODE HERE`.

The range determines the frequency shift in the fast time domain and the velocity determines the Doppler frequency in the slow time domain. Consequently, the Fourier transform implemented in the Matlab function `fft` has to be applied twice in the fast time domain and in the slow time domain. Use a twofold oversampling for both Fourier transforms. You may use the Matlab function `fftshift` to move the zero frequency to the center of spectrum. The range Doppler map shown in Figure 4 shows the magnitude square of the spectrum averaged over all antenna elements. It can be plotted using the Matlab function `imagesc`.

- Why is it beneficial to use oversampling?
- Why is it reasonable to average the magnitude squared spectra obtained from the different antenna elements?
- To which ranges and velocities do the individual samples of the spectra being computed with the `fft`-function belong?

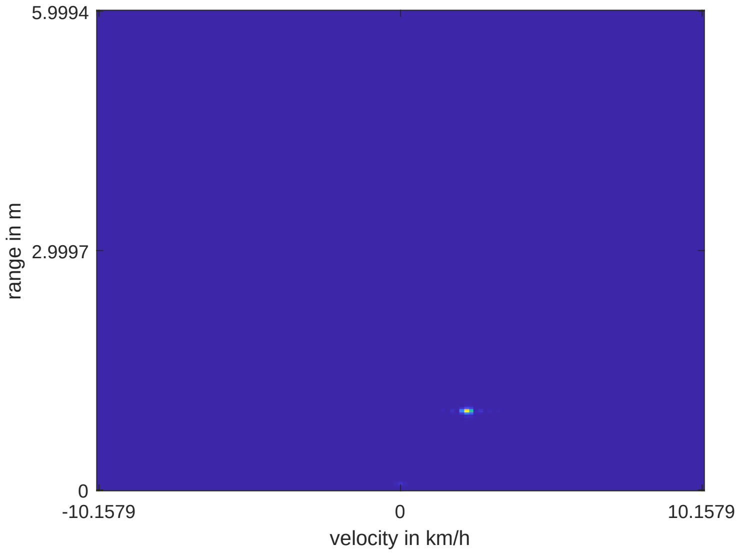


Figure 4: Exemplary range Doppler map with a single target.

4 Beamforming

The objective of this lab activity is to reinforce the understanding of classical spatial signal processing techniques based on beamforming. Both conventional beamforming and Capon's beamformer shall be implemented. The places in the program code in `conventional.m` and `capon.m` where you have to make your changes are marked by comments in capital letters like this `% ADD YOUR CODE HERE`.

First you need to compute the sample covariance matrix. Use all the available signal samples in the fast time domain and in the slow time domain. Next you have to compute the spectrum for the conventional beamformer. For uniform linear arrays this could be done using a Fourier transform. However, within this lab you shall not make any assumptions about the antenna configurations. Instead you shall compute the spectrum using the array manifold vectors which can be computed using the `manifold.m` function being provided to you. This function takes the antenna configurations of the individual radar sensors into consideration. Figure 5 shows an exemplary spectrum.

Next you shall implement Capon's beamformer. Figure 6 shows an exemplary pseudospectrum. Discuss the results. In particular you shall

- compare the results being obtained with the different beamformers,
- consider both single target and multiple target scenarios with different target separations and
- try out different radar sensors with different antenna configurations.

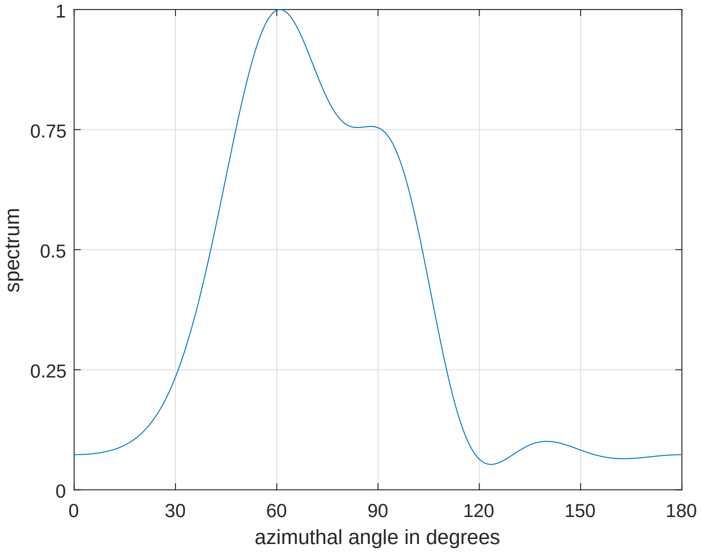


Figure 5: Conventional beamforming. Two targets at 60° and 90°.

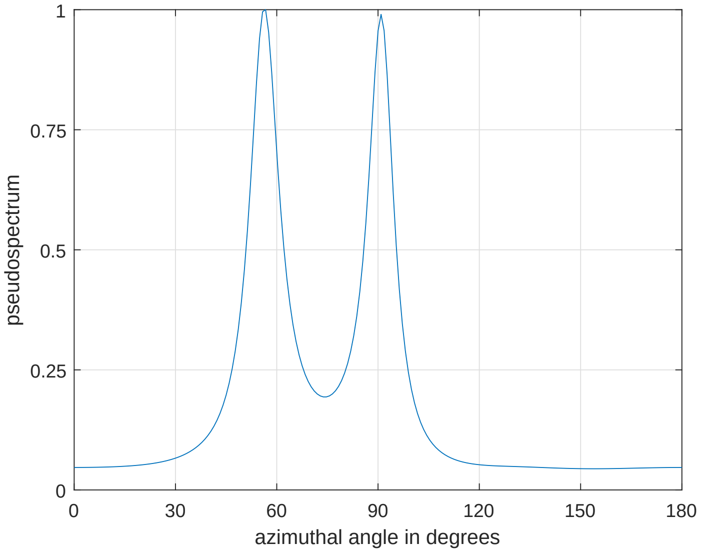


Figure 6: Capon's beamformer. Two targets at 60° and 90°.

5 MUltiple SIgnal Classification (MUSIC)

The objective of this lab activity is to reinforce the understanding of subspace based direction of arrival estimation. Specifically you will implement the MUSIC algorithm. The places in the program code in `music.m` where you have to make your changes are marked by comments in capital letters like this `% ADD YOUR CODE HERE`.

The first major step in subspace based direction of arrival estimation is the computation of the eigendecomposition of the correlation matrix. Make sure that the eigenvalues are in non increasing order. Next you have to compute the null spectrum. Here again you may use the Matlab function `manifold.m` to compute the array manifold vectors. Figure 7 shows an exemplary null spectrum.

- How do the results of the MUSIC algorithm compare to the results from beamforming? Consider both single target and multiple target scenarios with different target separations.
- What happens if there are several targets at the same range?
- What happens if an incorrect number of targets is given to the MUSIC algorithm?

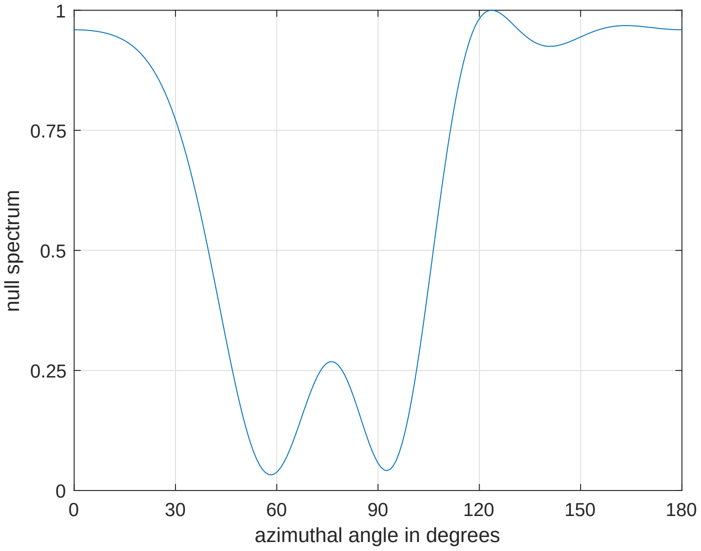


Figure 7: Null spectrum obtained with the MUSIC algorithm. Two targets at 60° and 90°.

6 Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT)

The objective of this lab activity is to deepen the understanding of subspace based direction of arrival estimation. Specifically you will implement the ESPRIT algorithm. The places in the program code in `esprit.m` where you have to make your changes are marked by comments in capital letters like this `% ADD YOUR CODE HERE`.

The ESPRIT algorithm can only be applied to antenna arrays where two identical subarrays can be formed. In SIMO mode both the TinyRad radar sensor and the V-MD3 radar sensor use a uniform linear array with four antenna elements. In this case two maximally overlapping subarrays with three antenna elements can be formed. But also in MIMO mode it is possible to identify suitable subarrays for both radar sensors. You may use the Matlab function `subarrays.m` which is aware of the antenna configurations to compute the selection matrices of the subarrays and the displacement. Implement the ESPRIT algorithm to compute the directions of arrival. Plot the eigenvalues, see Figure 8. Discuss the distribution of the eigenvalues of the correlation matrix. Consider scenarios with different numbers of targets and different antenna configurations. Which additional steps not being considered so far would be necessary to compute the directions of arrival using beamforming techniques or the MUSIC algorithm?

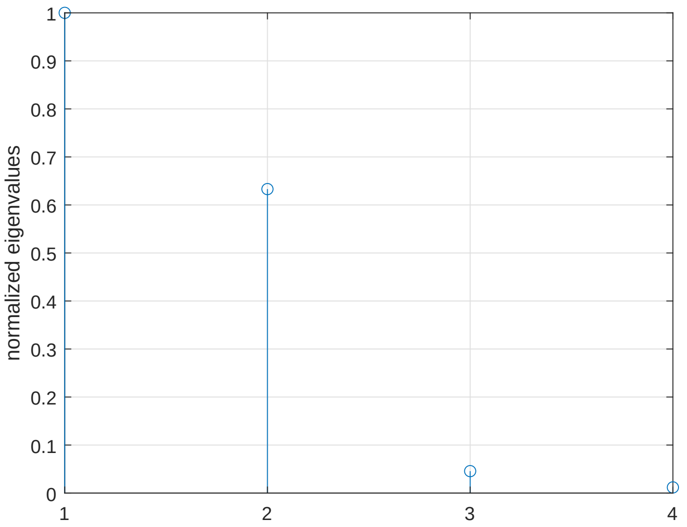


Figure 8: Eigenvalues of a correlation matrix. Two targets at 60° and 90°.

References

[ADI] *EV-TINYRAD24G User Guide. Evaluating the TinyRad* 24 GHz *Demonstration Platform.* UG-1709. Analog Devices. Norwood, MA, 2020.

[IR20] C. Iovescu and S. Rao. *The Fundamentals of Millimeter Wave Sensors.* SPYY005A. Texas Instruments. Dallas, TX, 2020.

[Mat] *Matlab Primer.* MathWorks. Natick, MA, 2022.

[Rao18] S. Rao. *MIMO Radar.* SWRA554A. Application Report. Texas Instruments. Dallas, TX, 2018.

[RFb] *User Manual V-MD3 Digital 3D Radar Transceiver.* RFbeam Microwave. St. Gallen, 2020.

[TI] *IWR6843, IWR6443 Single-Chip* 60- *to* 64 GHz *mmWave Sensor.* SWRS219D. Texas Instruments. Dallas, TX, 2020.