EITN90 Radar and Remote Sensing
Lab 3

February 23, 2018

1 Learning outcomes

In this lab we examine how to make an image using synthetic aperture radar (SAR). Most of the lab deals with using the Doppler Beam Sharpening (DBS) algorithm on synthetic data. The more realistic data are mainly collected in a circular path, with the image being calculated using backprojection.

2 Synthetic data

2.1 Simple point targets

2.1.1 Preparations

Mark A. Richards has authored an instructive matlab script for SAR imaging of simple point targets. Download the script from

- http://radarsp.com (look for "matlab supplements")

Unpack the zip-file on your computer. Start matlab. Inside matlab, navigate to the folder `FRSP_Demos\FRSP common\`, and add this folder to the matlab path (either by right-clicking it in the matlab GUI and using the menus, or use the command line and type `addpath('{start folder}\FRSP_Demos\FRSP common\')`). The scripts we will use are in the folder

- `FRSP_Demos\FRSP Non-GUI Demos\DBS\`

There should be two files, `makeSARdata.m`, and `procSARdata_DBS.m`.

2.1.2 Basic operation of the scripts

Data is generated using `makeSARdata.m`, where a number of parameters can be set to adjust the targets and data acquisition. These include target locations, carrier frequency, pulse length, antenna azimuth width, etc. The data generated will be saved in a .mat file (you name it while running `makeSARdata.m`), the name of which has to be entered in the beginning of `procSARdata_DBS.m`.

The scripts simulate a measurement using chirped pulses of length τ, with bandwidth B (variable `W` is used in the script) chosen to give the desired resolution after pulse compression. The data is collected in a stripmap configuration, that is, the antenna platform is moving in a straight line and illuminating the target scene at broadside.

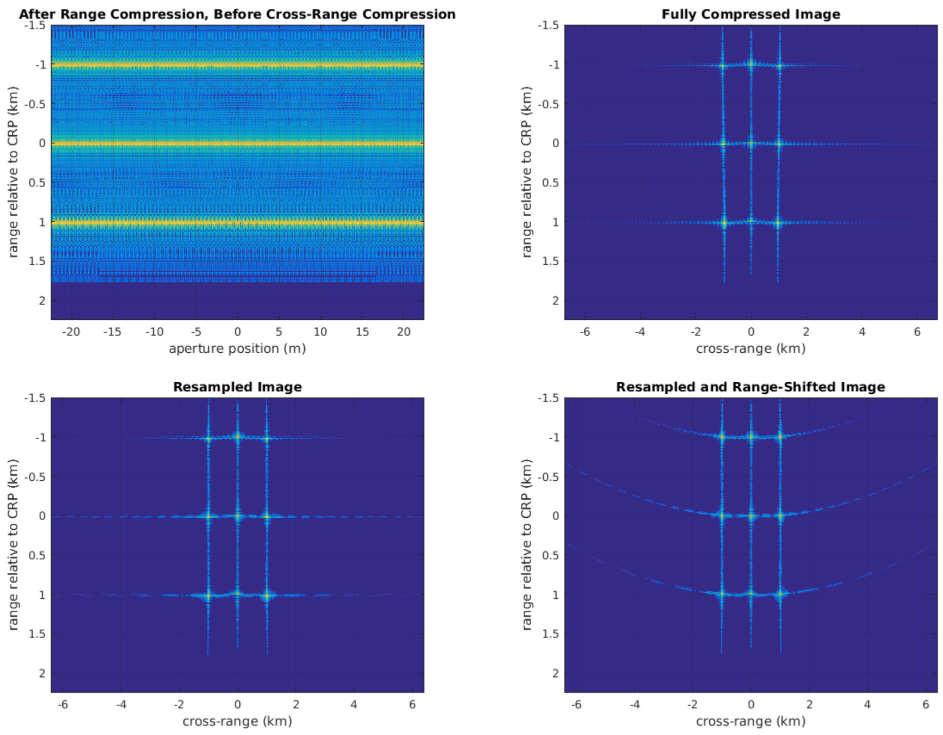

Figure 1: Typical outputs from the DBS processing script. The last two figures are the results of geometry corrections (the data are obtained in a cylindrical coordinate system of range and delay, which is projected back to the rectangular coordinate system of the image).

2.1.3 Experiments

Generate data using the original settings of the script. You will see a few plots with data generated, look particularly at Figure 1 where the chirped pulse is displayed.

Run the processing script. The result should be a number of figures as in Figure 1.

- Plot the received signals for one of the pulses before and after compression (you can save the original pulse shape by inserting a line like "`y_org = y;`" early in the `procSARdata.m` script, the variable "`y`" will later be overwritten by the compressed pulse). Can you verify that the pulse compression ratio is close to τB? The bandwidth B is stored as the variable "`W`" in the script. The compression ratio can be defined as τ/τ_1, where τ is the full pulsewidth (start-to-end) of the original pulse, and τ_1 is the peak-to-null width of the compressed pulse (or $2\tau_1$ is the null-to-null width).

- Try moving two targets closer together (their positions are controlled by the array `coords` in `makeSARdata.m`). At what distance can you no longer distinguish between them? Does it correspond to the theoretical resolution $c/(2B)$?

- Try reducing the range (variable `Rcrp`) by a factor of 10. How does it distort the image? Can you decide from which direction the illumination is (from the top or from the bottom of the figure)?

- Try changing the carrier frequency with a fixed requirement on resolution. Does the number of required pulses increase or decrease as the frequency increases? Hint: simply observe how the variable `Npulses` changes after running `makeSARdata.m`.
- What causes the star shaped patterns in the final image? Can you suppress them? Hint 1: in `procSARdata.m`, try an amplitude taper in slow time (between different pulses), and another amplitude taper in the matched filter. Hint 2: for a given pulse stored in a matlab vector "`pulse`", a simple amplitude taper starting and ending with amplitude zero and having maximum in the center can be achieved for instance by "`pulse.*sin(linspace(0, pi, length(pulse)))`".

2.2 Civilian vehicles data dome

The matlab scripts used in the previous part of the lab dealt with completely synthetic point targets. More realistic targets can be obtained from numerical simulations of complex geometries like cars. A dataset based on a number of civilian vehicles has been made available for download (after registration) at the web site https://www.sdms.afrl.af.mil/, the "Civilian Vehicle SampleSet". To facilitate this lab, the files are available at the location `T:\Protected\eitn90\Civilian_Vehicle_SampleSet`. Please observe the dataset is more than 3 GB, and does *not* need to be copied to your own folder, only a few matlab scripts need to be copied.

The document `CVDomes.pdf` explains the simulations in detail. In particular, the bandwidth is 5.35 GHz, giving a down-range resolution of $c/(2B) = 2.8\,\text{cm}$.

2.2.1 Preparations

Copy the folder `T:\Protected\eitn90\Civilian_Vehicle_SampleSet\code` to your own folder, typically created in `T:\Unprotected`. In the folder "code", there is a script `formImageCVDomes.m` which can be used to plot the SAR image.

Insert the path `T:\Protected\eitn90\Civilian_Vehicle_SampleSet` as "basePath" on line 24. Insert the path to your own folder at the very last line, where the figure may be saved. Or simply delete the line to not save the figure.

The data consists of backscattering data at 9.6 GHz for a large number of possible angles of incidence from a half-sphere above the vehicle. You can change elevation angle, and what region of azimuth angle you want to use in the image formation. In this script, the image is obtained by the backprojection method, that is, the amplitude at each image pixel is the superposition of the backpropagated signal from each measurement position. This is a straight-forward, but relatively costly, method of computing the image. The core of the algorithm can be found in lines 74–105 in `bpBasicFarField.m`, but does not have to be fully understood to interprete the images.

You find the different datasets in the folder `Domes`. The selected dataset is input in the beginning of the script, together with the elevation angle.

2.2.2 Experiments

Run the script with the standard settings to start with. The result should be similar to Figure 2. A particularly interesting target can be the ToyotaTacoma, that has a loading platform with several corner reflectors. Its geometry is also shown in Figure 1 of the `CVDome.pdf` document.

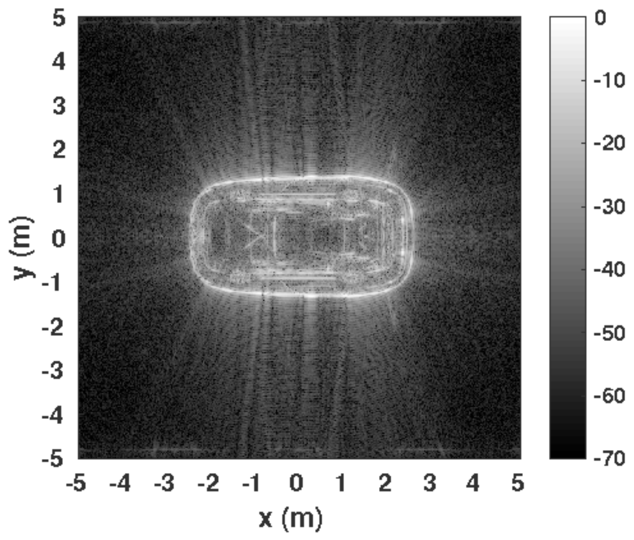

Figure 2: Example output from the Civilian Vehicle SampleSet, the Sentra at 30° elevation.

- Try different polarization settings, HH, VV, or HV. Does any of them provide a clearly better or worse image? Can you explain why?
- Try different elevation angles and different vehicles. Do you see any common features depending on elevation? Compare with the theory for the "layover" phenomenon in Figure 21-36 in the book.
- Try using a smaller part of the available azimuth data, for instance from 0 to 90° with some variations. How does it affect the image?
- Enable the taper for sidelobe control (see beginning of `formimageCVDomes.m`). How does it affect the image?

3 Real data

We finally turn to some real data, obtained from measurements.

3.1 Target discrimination

3.1.1 Preparations

A dataset from a measurement campaign where an airborne platform, operating in X-band (around 10 GHz), was flown around a parking lot with a number of cars can be downloaded (after registration) from the site https://www.sdms.afrl.af.mil/. To facilitate the lab, the files have been made available at `T:\Protected\eitn90\Target-Discrimination-CP`. You do *not* have to copy all the data to your own folder, a few matlab scripts are sufficient. The data is collected for a spotlight scene, with a carrier frequency of around 10 GHz, and comprises more than 3 GB.

Copy the folder `Utilities` to your own folder. Open the script `bpGUI.m`. This is a GUI to access the different datasets. Choose from the datasets found in the folder `T:\Protected\eitn90\Target-Discrimination-CP\Data`.

There is a slight flaw in the program. It plots a sequence of plots in one window, and if you happen to change which window is active by clicking with your mouse somewhere, the plot may end up in the wrong window. Either you keep your cool and sit still during the plotting phase, or do the following change. Go to the for-loop starting at line 239, and insert `figure(2)` as the first line of code inside the for-loop. This enforces figure number 2 to be active at each plot, which is sometimes not the case on all platforms. If this is not done, sometimes the plot might be made in figure number 1.

The image in this script is computed using the backprojection method, which is suitable for a wide-angle dataset. There are not that many parameters to change except for the size of the image, but many different targets to explore. Inside each target folder, you will find data for each of the 31 orbits of the aircraft, numbered from 214 to 244. You can typically just pick the first orbit, 214.

The dataset is explained in detail in the document `Dungan_SPIE12_final.pdf`, in the folder `T:\Protected\eitn90\Target-Discrimination-CP\Documentation`. In particular, the sensor bandwidth is specified as $B \approx 600\,\mathrm{MHz}$, indicating a down-range resolution of $c/(2B) = 2.5\,\mathrm{dm}$. This will be seen as the typical width of the specular reflections against the vehicle sides.

3.1.2 Experiments

- Run the script `bpGUI.m` for one of the vehicle targets, for instance the one in `Data\van1\PH_van1_0214.mat`. Figure number 1 is the coherent image, making full use of the phase information. Figure number 2 is an intermediate figure, displaying all sub-images from the backprojection procedure. Figure number 3 is the non-coherent (average of the amplitudes of the sub-images) image of the scene. Observe the difference between the coherent (figure 1) and non-coherent (figure 3). Try to observe similarities and differences with the synthetic data used in the previous assignment.
- Make sure to try the dataset `open`. This corresponds to a flat piece of asphalt, and is a typical example of clutter return. Note the remarkable difference between the coherent and non-coherent images.

3.2 SAR focusing tutorial

This dataset is provided from a satellite measurement using a stripmap setup. It can be downloaded from `https://saredu.dlr.de/unit/focussing_matlab` after registration. The main purpose of this dataset is to demonstrate focusing and multilook despeckling on some real data.

3.2.1 Preparations

To facilitate the lab, the necessary files have been made available as a compressed zip-file at `T:\Protected\eitn90\ID_1111_SAR-EDU_Tutorial_Focusing_Matlab.zip`. This is a very reduced dataset, so you can copy the whole thing to your own folder, and unzip it. Start matlab. Navigate to the folder `{start folder}\Data&MFiles\Data\`, and add this folder to the matlab path.

While inside matlab, navigate to `{start folder}\Data&MFiles\Executables\`, and open the file `SAR_focusing_tutorial_solution.m`.

Some documentation is available in `{start folder}\Readme\`. In principle, the script takes raw SAR data and correlates it with one LFM chirped pulse in down-range (for pulse compression) and another in cross-range. This is a different approach than the FFT used in the DBS algorithm with regard to cross-range (azimuth) data, but fills the same purpose of focusing.

3.2.2 Experiments

- Run the script. Compare the raw data in figure number 1, with the processed data in figure number 4. Note the remarkable enhancement in detail after doing the SAR processing. In particular, try to spot the various corner reflectors in the image, which should be visible as bright crosses.
- There is a parameter "multilook" inside the script. This is a simple way to reduce speckle effects by averaging the image, which of course reduces resolution. Try this out.

Once you completed all the tasks, discuss with the lab leader to be approved for the lab.