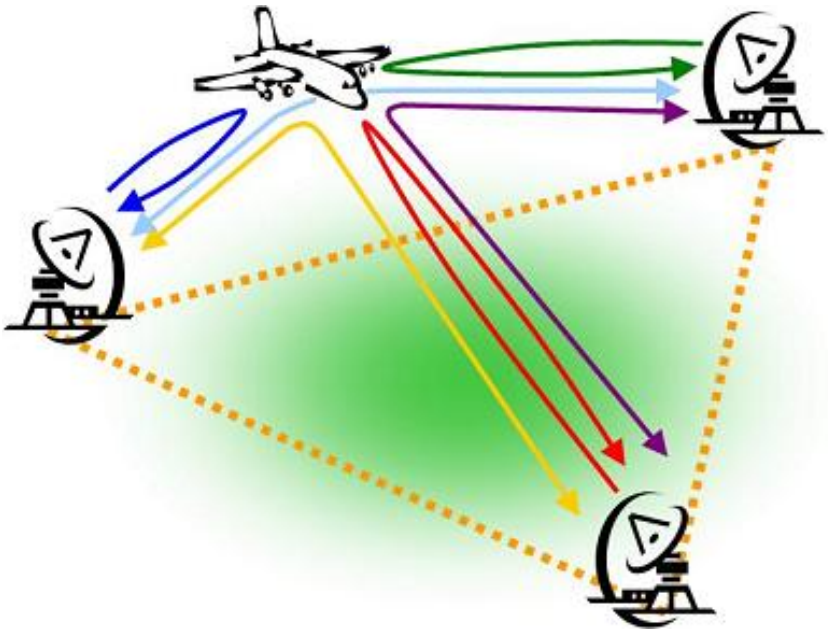
TNO innovation for life

TU Delft

MSc Thesis

Coherent Multistatic Radar Imaging



Author:
J.M.M. Verzeilberg
4044347

Supervisors:
ir. M.P.G. Otten *(Daily)*
Prof. ir. P. Hoogeboom

August 30, 2011

Delft University of Technology
Department of
Telecommunications

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled

Coherent Multistatic Radar Imaging

by

Johannes Markus Mark Verzeilberg

in partial fulfillment of the requirements for the degree of

Master of Science.

Dated: August 30, 2011

Chairman:	Prof. ir. P. Hoogeboom
Daily supervisor:	ir. M.P.G. Otten
NLDA supervisor:	KTZE dr. ir. F. Bolderheij
Reader:	Prof. F. le Chevalier

Abstract

After a MIMO revolution in the telecommunications world, this technology becomes increasingly popular in the radar world. The radar end-users are demanding more and more of their systems, while the targets are designed to have an as small as possible RCS, which makes them more difficult to detect. With multiple transmitters and receivers it may be possible meet the stringent demands of the users. Extending monostatic radar functions like target localization and Doppler velocity estimation to a multistatic configuration will show some significant improvements in resolution and overall situational awareness. With multistatic target localization, the cross-range resolution can be greatly improved and the overall system is much less susceptible to dips in the RCS, because it sees the target from different angles. With multistatic Doppler velocity estimations, a complete 2D or 3D target velocity vector can be composed to aid tracking and imaging techniques like ISAR. Multistatic ISAR is a perfect example of the strength of combining multiple radars. It can be used to achieve a higher resolution for the same integration time as a monostatic radar, or it allows a reduction of integration time, with a resolution equal to the monostatic case. A new concept in multistatic ISAR is the interpolation of the ISAR data. In case the geometry between the target and the radars does not meet the requirements for coherent combining the images, an interpolation of the data can be done. Not only the geometrical requirements can be eased, also the integration time can be artificially increased, allowing the radars to spend more time on other functions. Another new approach to multistatic ISAR is the incoherent combination of multiple ISAR images. This concept is called 2D ISAR and allows the creation of high resolution images with low resolution surveillance and MTI radars. The main idea of 2D ISAR is seeing the target under different angles, so the amplitude of the summation of scatters will be higher than that of the sidelobes, which allows to estimate the overall target dimensions.

Acknowledgement

I would like to thank the Netherlands Ministry of Defense for the opportunity they gave me to get my Masters degree, as well as Colonel Bolderheij for his support during my stay at the Netherlands Defense Academy and the Technical University. Furthermore I would like to express my gratitude to Professor Hoogeboom, Matern Otten and all the other colleagues I worked with at TNO Defense for their help and support during my stay there. Special thanks go to Philip van Dorp and Wim van Rossum for their help tackling some problems. And finally I want to thank my friends and family for their encouragement along the way.

Mark Verzeilberg
The Hague, The Netherlands
August 30, 2011

Contents

1. Introduction

After a revolution in the telecommunications world, where multiple antennas are currently widely used to increase coverage and reduce fading and dead spots, the technology becomes increasingly popular in the radar world. The radar end-users are demanding more and more of their systems, like increased range and coverage, while the targets are designed to have a small as possible RCS. Also an increase in down- and cross-range resolution is demanded by the users. With multiple transmitters and receivers it may be possible to meet the stringent demands of the clients. There are several different multi antenna scenarios. A scenario where several monostatic radars work together to create a MIMO [1] radar network, or a scenario where a single transmitter is combined with various receivers to create a Multistatic radar network(MSRN). It is possible to combine multistatic networks to create an even larger network. These MSRNs have increase radar coverage, are capable of detecting targets faster, because they see them under different angles, hereby increasing the chance of seeing RCS flashes and receiving more energy from the targets. One of the advantages is an increased power budget. Not only due to the usage of multiple antennas, but in the bistatic case also the need of having a duplexer and other receiver protection mechanics are gone. A second advantage is the possibility to image targets with a higher resolution, without increasing the bandwidth of the system.

The cross-range resolution for distant targets is almost always worse than the down-range resolution. If a second receiver sees the target under an angle, the intersection of the two range responses could increase the cross-range resolution. This will be discussed in chapter 2. Another possibility with a MSRN is to determine a 2D or 3D target velocity. This is accomplished by combining the observed Doppler shifts at the different receivers. This can help increasing the accuracy of the tracker, because no assumptions on the total target velocity vector are needed. Combining Doppler shifts is addressed in chapter 3. Besides the increase in resolution and the possibility to determine 3D velocity of the target, the jamming resistance of MSRN is much better than that of monostatic radars. Besides the anti-jamming methods that can be used in monostatic radars, MSRN have additional methods to arm themselves against jammers. For instance with smart jamming, the jamming signal is focused in the direction from which the received radar signal came, but with a MSRN the receiver(s) is (are) not in the same location as the transmitter(s), so this kind of jamming does not work effectively. Another kind of jamming is spot noise jamming. With this jamming technique not only in the direction of the received signal the jammer is active, but the jammer also focuses all its power in the frequency of the received signal. With multistatic radar, the different combinations of transmitters and receivers do not have to work with the same frequency, hence this type of jamming is also less effective against multistatic radar than against monostatic radar. Furthermore the survivability of the platforms can increase, because the opponent does not have the capability to detect the receiving platforms. Also there may be multiple transmitter platforms, so when one of the transmitters is disabled, other transmitters are still operational and the mission can continue. The latter effect is also called graceful degradation.

[1] Multiple Input Multiple Output

The main aspect that will be discussed in this thesis are methods to improve imaging techniques, by expanding the monostatic imaging techniques via bistatic to multistatic imaging. Non Cooperative Target Recognition (NCTR) is the basis of the need to expand from monostatic to multistatic imaging, because as stated before, targets are getting smaller, faster and more threatening. A possible way of creating a high resolution radar network with reasonably priced hardware, is the use of Inverse Synthetic Aperture Radar (ISAR). With ISAR a high cross-range resolution can be achieved by storing target responses over a certain period of time, and with the Doppler information in the signals the image can be compressed in azimuth direction. Combining ISAR images from multistatic radar configurations can even further improve the cross-range resolution for the same integration time, or the integration time can be reduced to achieve the same resolution as with a monostatic radar. The advantage of a reduced integration time is that there is a limited amount of time and energy available, which has to be shared by several different functions that can be performed with the radar. An introduction to ISAR will be given in chapter 4. After the introduction two different multistatic ISAR techniques will be discussed. In chapter 5 an incoherent approach will be shown and in chapter 6 a coherent approach. A downside of coherent multistatic ISAR is the stringent demands on the geometry of the situation. To overcome this problem a new technique is introduced to interpolate missing radar data. Another novel usage of ISAR is 2D ISAR. This is an incoherent merge of the ISAR data. When ISAR is performed from several different directions, a high resolution image of the target can be made, while using a small bandwidth. At the end of the ISAR discussion a simulation with a complex target with directive scatterers will show the possible real life improvements of multistatic ISAR. In the final chapter a larger summary of the research as well as conclusions and possible recommendations for future research are discussed. At the end of the thesis the Matlab code used to perform the different simulations is printed.

2. Range Imaging

2.1 Monostatic Range Resolution

For monostatic radar systems the range resolution is only dependent on the transmitted bandwidth. The return signal is matched filtered with the signal that is transmitted to determine the distance of the target. The range compressed signal has the following form:

$$s_{RC}(t) = \int_{-T/2}^{T/2} s_R(t) \cdot s_T^*(t-\tau)dt \tag{2.1}$$

Where $s_R(t)$ is the received signal and $s_T^*(t-\tau)$ the transmitted signal shifted over a certain amount of time τ. At a certain time shift the convolution will be maximal and this is the position of the target. If we write this expression in the frequency domain, the convolution becomes a multiplication:

$$S_{RC}(f) = S_R(f) \cdot S_T^*(f) \tag{2.2}$$

It is seen that the preferred signal $s_{RC}(t)$ can be obtained by applying the inverse Fourier Transform to $S_{RC}(f)$.

$$s_{RC}(t) = IFFT\left[S_{RC}(f)\right] \tag{2.3}$$

From this equation the relation between the transmitted bandwidth and the pulse length is shown. Assuming that a block pulse is transmitted, the relation between the width (duration) of the block and the width of its Fourier counterpart, a sinc-function, is known. For a signal with constant amplitude this relation is:

$$\Delta t = \frac{1}{B} \tag{2.4}$$

Where Δt is the duration of the pulse and B the bandwidth of the sinc-function. In a normal pulsed radar, where no pulse compression is applied, the range resolution is dependent on the duration of the transmitted pulse.

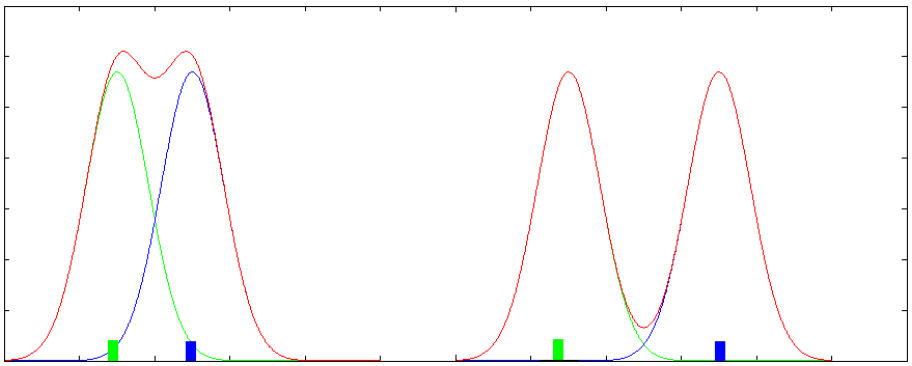


Figure 2.1: Range resolution

In figure 2.1 the relation between the pulse width and the resolution is shown. In this figure the absolute part of the two signals is shown, because it is possible that the phase of the signals allows a differentiation between the two targets. When two objects are close together, the receiver can't distinguish them anymore (first two pulses). If the targets are further apart from each other, two separate pulse are perceived by the radar and both targets can be detected (last two pulses). For radars that use pulse compression, the resolution is dependent on the frequency sweep that is made, instead of the length of the transmitted pulse. For both types of radars, with or without pulse compression, the following equation holds in a monostatic scenario:

$$\Delta R_r = \frac{cT}{2} = \frac{c}{2B} \tag{2.5}$$

The range resolution is defined as the resolution perpendicular to the surface of the sphere that is radiated with a monostatic isotropic radar. A 2D visualization of the monostatic range resolution is seen in figure 2.2 [1].

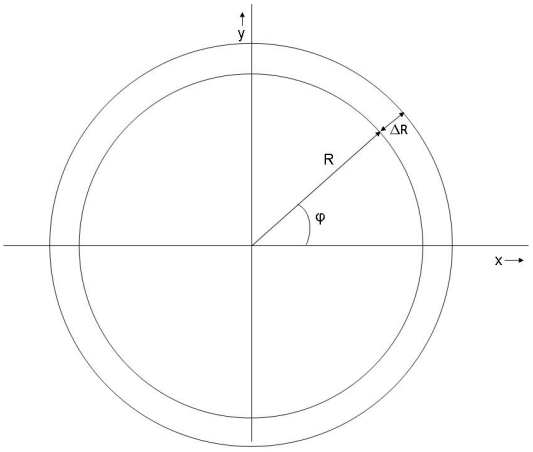


Figure 2.2: Monostatic range resolution, top view

[1] picture from [1]

2.2 Bistatic Range Resolution

For bistatic radars the same definition for the range resolution is used, but here the resolution is perpendicular to the ellipsoid radiated by the radar. In figure 2.3[2] the 2D visualization of the bistatic range resolution is shown. The 2D range response has now become an ellipse. For the bistatic range resolution a compensation term is needed for the fact that the radiation pattern is an ellipsoid and the range resolution is defined perpendicular to this surface, while the incoming and reflected waves are not perpendicular to the ellipse.

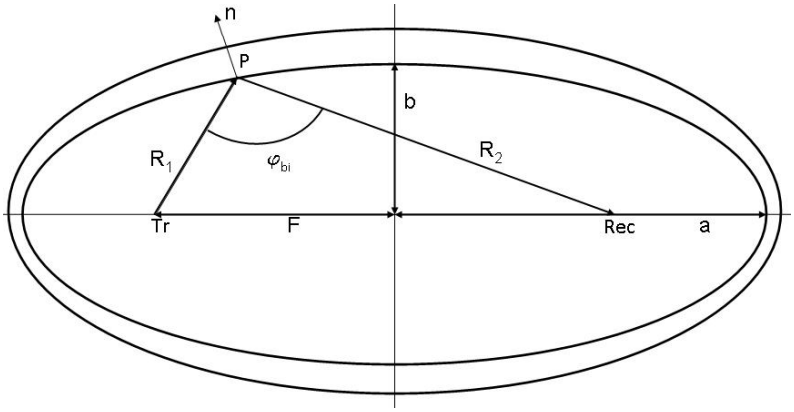


Figure 2.3: Bistatic range resolution, top view

The bistatic angle, φ_{bi} is the angle between the incident wave and the wave reflected towards the other focal point of the ellipse. With this bistatic angle, the range resolution can be calculated. As seen in figure 2.4 the resolution is dependent on the cosine of the bistatic angle and the monostatic range resolution. A zoom in on the target reflection can also be seen in figure 2.4.

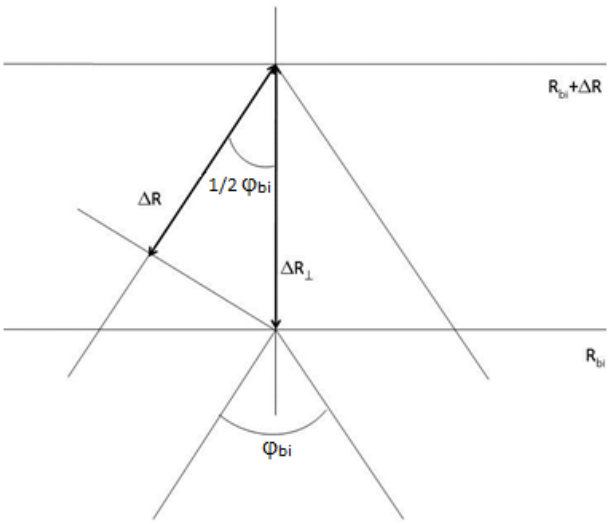


Figure 2.4: Bistatic Range resolution, zoom in on target reflection

[2]picture from [1]

Whereas in the monostatic configuration the range resolution is defined as:

$$\Delta R_{mono} = \frac{cT}{2} = \frac{c}{2B} \tag{2.6}$$

In the bistatic configuration an additional term is needed to come to the definition that the range resolution is perpendicular to the range response. With figure 2.4 the bistatic range resolution can be derived.

$$\cos\left(\frac{1}{2}\varphi_{bi}\right) = \frac{\Delta R_{mono}}{\Delta R_{\perp}} \qquad \Delta R_{\perp} = \Delta R_{mono} \cdot \frac{1}{\cos\left(\frac{1}{2}\varphi_{bi}\right)}$$

Filling in the definition of the monostatic range response leads to the equation for the bistatic range response, which can be seen in equation 2.7

$$\Delta R_{bi} = \frac{c}{2\cos\left(\frac{1}{2}\varphi\right) B} \tag{2.7}$$

To see if this holds, a simulation has been made with Matlab. In figure 2.5 the results are shown.

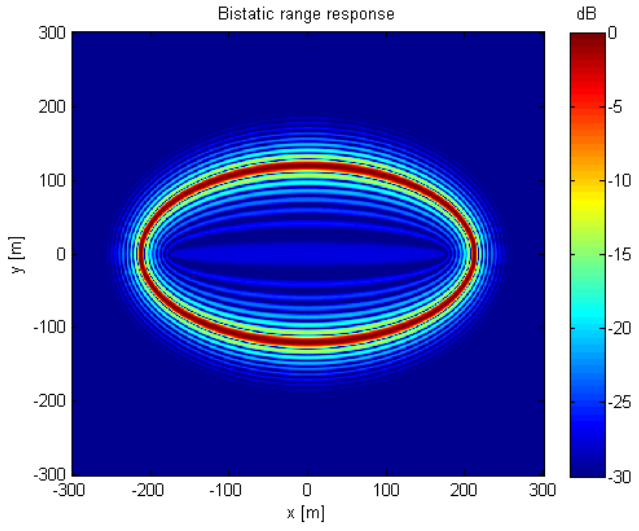
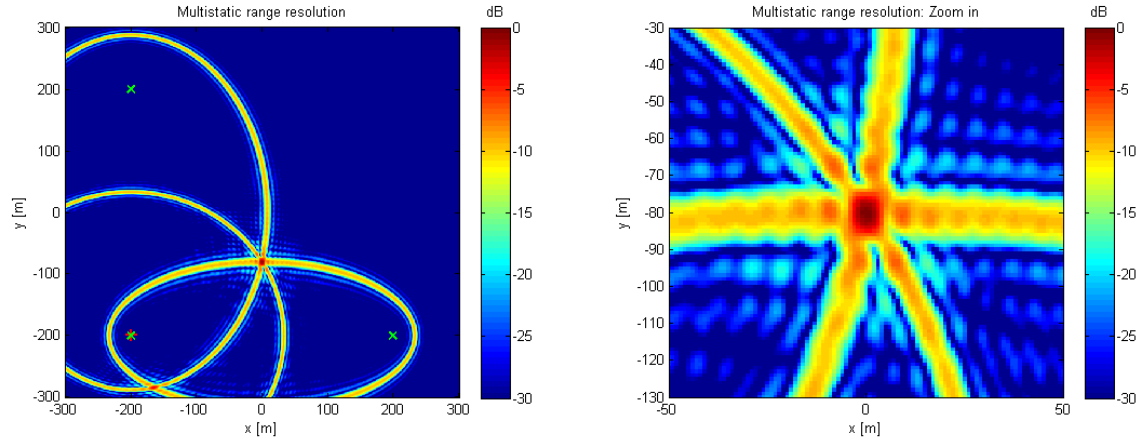


Figure 2.5: Bistatic Range Resolution simulation

It can be clearly seen that the resolution deteriorates when the bistatic angle increases. When the target is on the semimajor axis, the bistatic angle is zero and the range resolution becomes the monostatic resolution, whereas when the target is on the semiminor axis, the bistatic angle is maximum and the resolution is worst. This bistatic angle dependence of the range resolution is also the fundamental weakness of bistatic radar. If the target is right in between the transmitter and receiver the bistatic angle is π, and therefore the bistatic term that is related to the range resolution becomes zero. Because the resolution is zero the range can no longer be estimated.

2.3 Multistatic Range Resolution

The extension to multistatic range resolution is a small one from the bistatic range resolution case. This due to the fact the multistatic configurations can be decomposed into sets of bistatic and monostatic radars. The range resolution is now dependent on the best available set of radars. The best available set is a very cryptic description, especially because the best set varies over time. For the multistatic case radar management starts to play a vital role in optimizing the detection probability and improving the resolution of the image. As explained in section 2.2, the bistatic range resolution is dependent on the bistatic angle φ_{bi}. This means that the best bistatic radar pairs probably have the smallest bistatic angle. The largest improvement over monostatic and bistatic configurations is the possibility of a complete 3D decomposition of the velocity of a target, whereas for mono- and bistatic scenarios only the radial speed of the target could be determined. This 3D decomposition of the target velocity is discussed in section 3.3. There is a limit to the number of radars that should be coherently combined to determine the position of the target. Otherwise, the responses of the different radar sets will interfere with each other. In figure 2.6 a combination of three receivers is shown. From this picture it can be deduced that using a lot of receivers eventually does not add any more information to the image. The responses will start to overlap and interfere with each other. Also not every radar pair will improve the range resolution, so it is more useful to select only the best pair(s). The selection of radar pairs can take place when the position of the target approximately known.



(a) Multistatic range resolution

(b) Zoom in at target

Figure 2.6: Multistatic Range Resolution with a zoom in at the target location

Another advantage of multistatic target localization is that the cross-range can be greatly improved, without the use of special imaging techniques like (I)SAR. This can be seen in figure 2.6, where one transmitter, red plus symbol at -200,-200, and three receivers, green exes symbols at -200,-200 200,-200 and -200,200, are used to determine the location of a target. The cross-range resolution is with a mono- and bistatic radar always dependent on the range and the beam width of the antenna. The beam width of an antenna is $\frac{\lambda}{d}[rad]$ which results in a cross-range response of $\frac{R \cdot \lambda}{d}[m]$. With a multistatic configuration it is possible to decouple the cross-range resolution from the range. When orthogonal spaced receivers are used the cross-range resolution is no longer dependent on the range of the target, but on the down-range resolution of the radars. Most of the time orthogonal sets are not available, but the cross-range is then a combination of the down-range resolution and the angle under which the radar sees the target.

3. Doppler Imaging

In this chapter about Doppler imaging all the simulations are made with one scenario in mind. The situation sketch is shown in figure 3.1

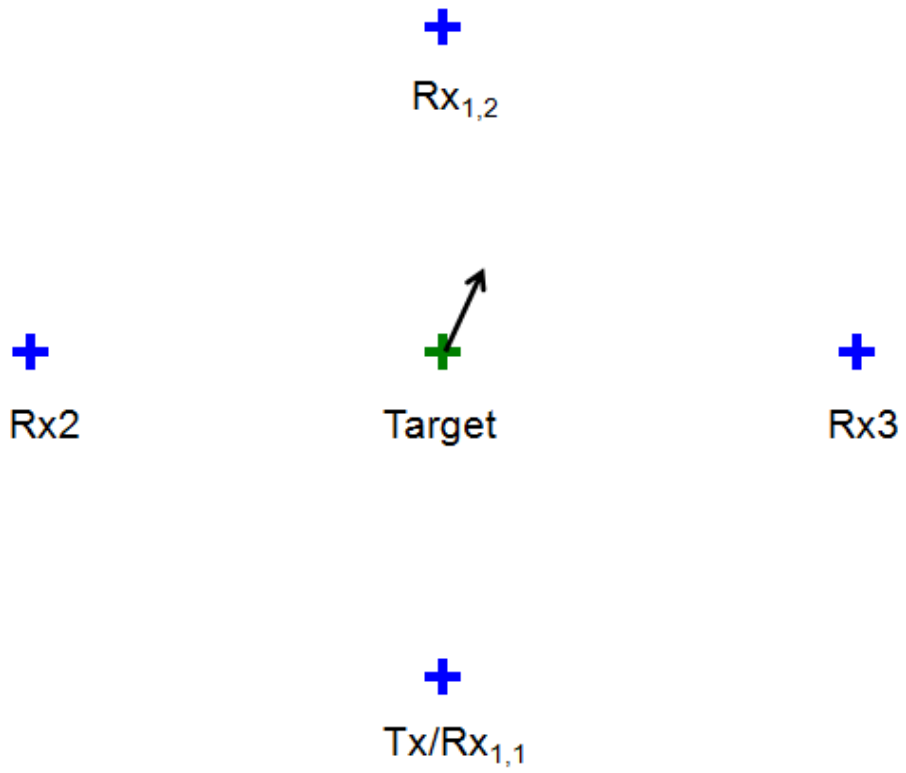


Figure 3.1: Situation sketch

For the simulations a monostatic radar is combined with two additional receivers, to create a multistatic radar network. The target is flying with a velocity vector of [20,50] m/s. In section 3.1, the monostatic radar $TX/RX_{1,1}$ is used. For the bistatic simulation in section 3.2 the transmitter and RX_2 are used. The multistatic simulation encompasses a scenario where the monostatic radar is used with receivers RX_2 and RX_3. A second multistatic simulation is made to explain a weakness of multistatic radar networks. The second multistatic scenario uses three bistatic radars, namely the transmitter and the receivers $RX_{1,2}$, RX_2 and RX_3. For the simulations a sufficiently high PRF[1] is chosen, so that the maximum received Doppler shift is well below the maximum detectable Doppler shift.

[1] Pulse Repetition Frequency

3.1 Monostatic Doppler Imaging

With a single monostatic radar it is possible to derive the speed of the target towards the radar, also called the radial speed. The speed is perceived as a shift in the frequency of the received signal. This shift is called the Doppler shift. If a target moves towards a radar, the frequency of the received signal will increase and if the target is moving away from the radar, the frequency decreases. If the target flies with a speed of $\frac{\lambda}{2}m/s$ towards the radar, the total path length will decrease with one λ or in phase 2π per second. The radar will see this shortened path length as an increase in the frequency of the signal by 1 *Hz*. With this example in mind the equation that describes the Doppler shift has the following form:

$$f_d = 2v_r/\lambda \tag{3.1}$$

where v_r is the speed of the target towards the radar. The resolution of the Doppler shift is determined by the integration time. The integration time is equal to the number of pulses, N, used in the Doppler processing multiplied with the PRI[2]. The speed resolution in Doppler processing for a monostatic radar is:

$$\Delta v_r = \frac{PRF \cdot \lambda}{2N} \tag{3.2}$$

In figure 3.2 a Doppler map is made of a target flying with a radial speed of 50 m/s and a transverse speed of 20 m/s.

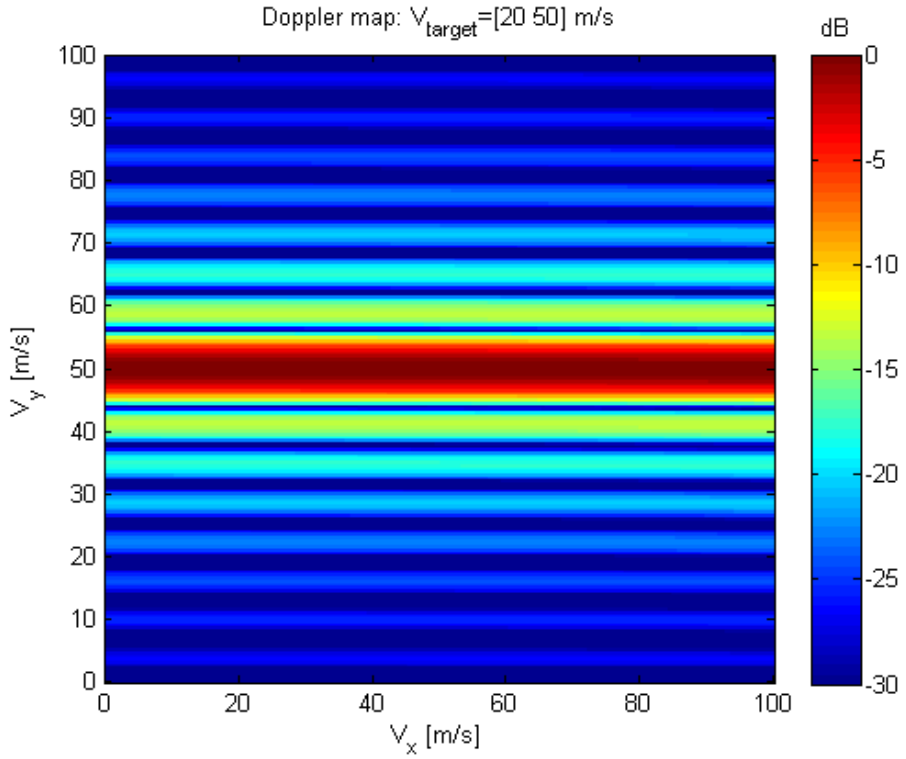


Figure 3.2: Doppler return of the target

As can be seen in figure 3.2, the monostatic radar is only capable of detecting radial velocities of the target. There is no speed detectable in the transversal plane, because when this speed is projected towards the radar, no Doppler shift will be present. i.e. If the target is flying on a circle around the radar, thus it flies at an equidistance, it will not add a Doppler shift to the received signals.

[2]Pulse Repetition Interval

3.2 Bistatic Doppler Imaging

Bistatic radar configurations have exactly the same weakness as the monostatic radars. They are not capable of a full decomposition in a 2D or 3D velocity only by measuring Doppler. In a bistatic radar set-up the target will not shift the received signal if it flies on an ellipse with the transmitter and receiver as the focal points.

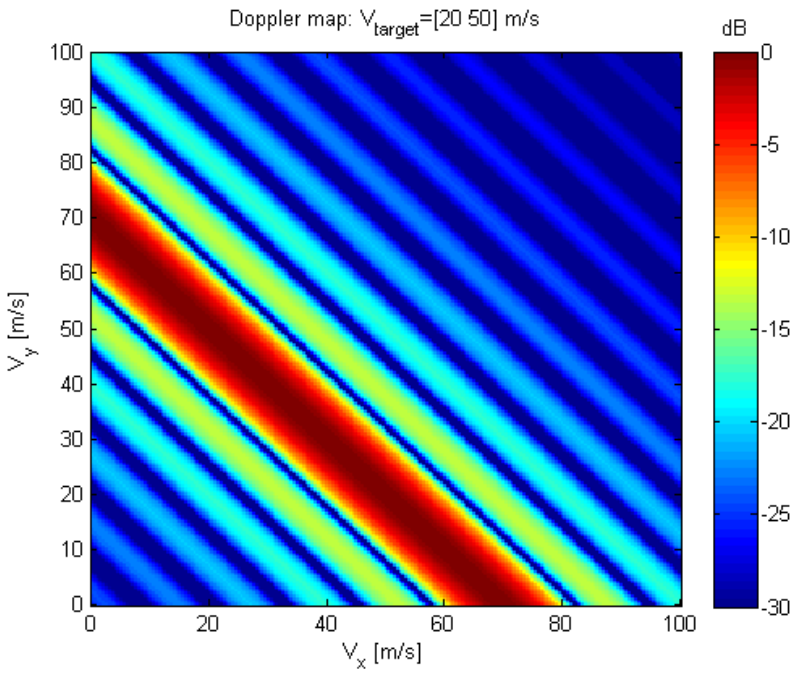


Figure 3.3: Doppler return of the target

In figure 3.3 the ambiguities in speed are clearly visible. Only here the ambiguity is even worse than in the monostatic case. In the monostatic set-up one could determine the speed towards the radar, but with a bistatic configuration the Doppler shift is a combination of the speed towards the transmitter and receiver. The Doppler shift in bistatic configurations is calculated as follows:

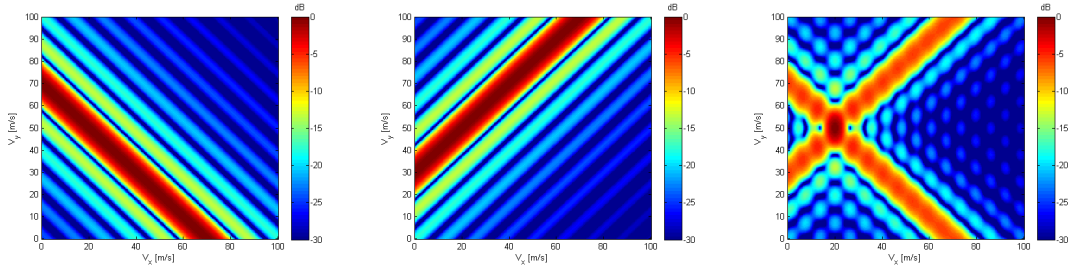
$$f_d = -\left(\frac{R_t \bullet V}{|R_r|} + \frac{R_r \bullet V}{|R_r|}\right)/\lambda \tag{3.3}$$

Where V is the velocity vector, R_t the distance vector from the transmitter to the target and R_r the distance vector from the target to the receiver. To normalize the dot product between the distance and velocity vectors, the individual components are divided by the 2-norm of the distance vector. Equation 3.3 can also be used to calculate the monostatic Doppler shift, but in that case the two components (to and from the target) are equal and the equation can be simplified to 3.1. To calculate the speed resolution of the bistatic Doppler response, the same compensation term as with the bistatic range resolution can be used. The bistatic Doppler resolution is not only dependent on the integration time, but also on the bistatic angle. Applying the bistatic compensation to equation 3.2 gives the following expression for the speed resolution:

$$\Delta v_r = \frac{PRF \cdot \lambda}{2\cos(\frac{1}{2}\varphi_{bi})N} \tag{3.4}$$

3.3 Multistatic Doppler Imaging

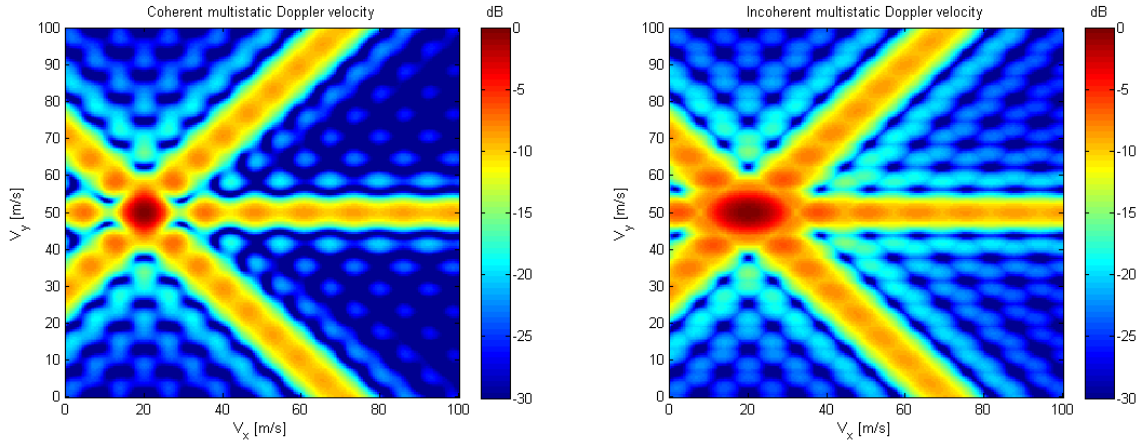
With multistatic Doppler imaging it is possible to determine a 3D velocity vector of the target, due to the factorization of the target's Doppler shift towards different receivers. When the Doppler shifts are combined, a 2D or 3D velocity vector of the target is constructed. This is shown in figure 3.4.



(a) Receiver 1 (b) Receiver 2 (c) Combination

Figure 3.4: Coherent Multistatic Doppler

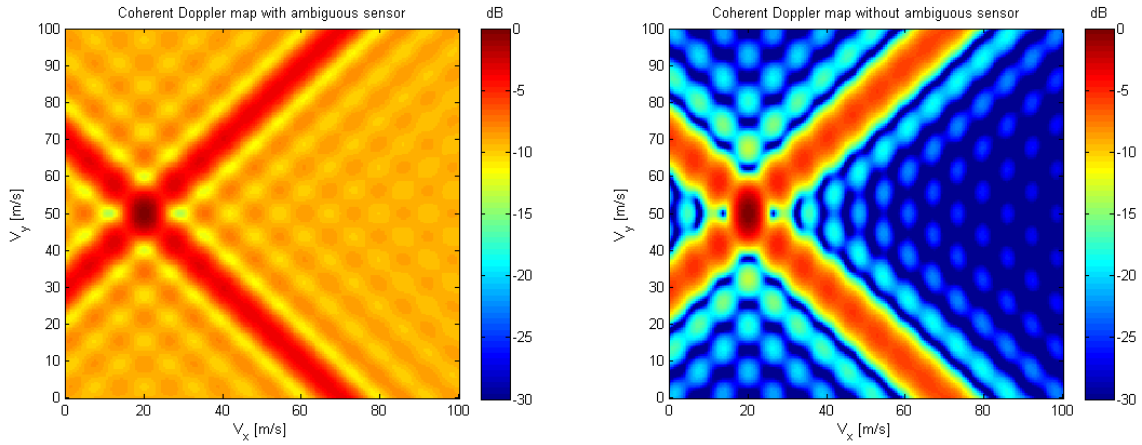
A coherent combination of the different images received by the stations gives a better resolution in the actual Doppler speed than the incoherent combination. The main peak of the Doppler velocity is smaller and the side lobes of the coherent combination are approximately 3 dB lower than the side lobes of the incoherent combination, in the case of a combination of 3 receivers. A comparison between a coherent and an incoherent combination is shown in figure 3.5.



(a) Coherent combination (b) Incoherent combination

Figure 3.5: Difference between coherent and incoherent summation

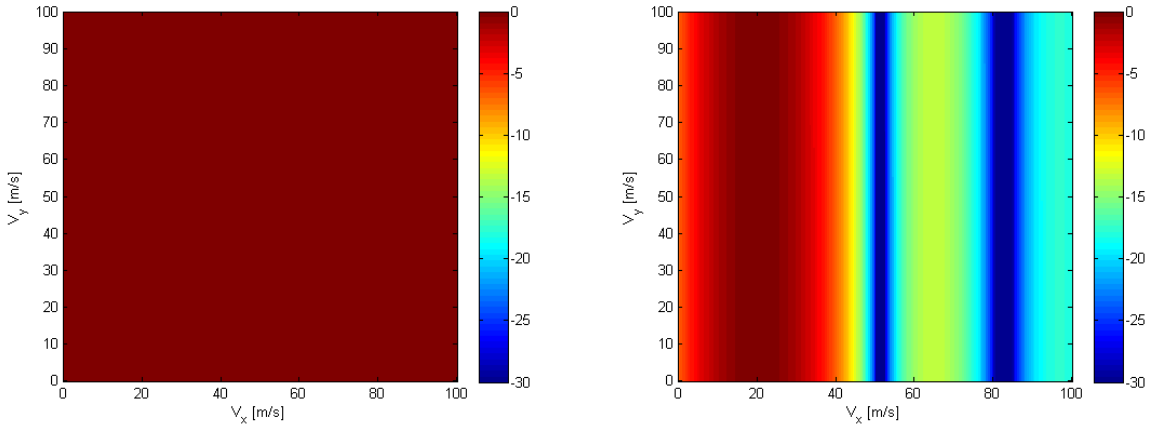
One of the main problems with a multistatic environment is when a target is right in between a transmitter and a receiver. As already noted in section 2.2 the ability to estimate the range of the target is lost, because the bistatic angle is π. The same holds for the Doppler shift. Here the loss of the Doppler shift can be explained with the fact that when the target is right in between the transmitter and receiver, the path length does not change. When the length of the path does not change no shift in frequency is detected. This measurement is severely deteriorating the end result of the coherent or incoherent combination of the different images. For the coherent case the difference in a measurement with and without an ambiguous sensor is shown in figure 3.6.



(a) (b)

Figure 3.6: Difference with or without ambiguous sensor

To zoom into the problem of the ambiguous sensor, the output of the ambiguous sensor is shown in figure 3.7. In figure 3.7a a target is right in between the transmitter and receiver. Nothing can be seen in the image, because the total variation in amplitude of the simulation was around 0.005 dB. In figure 3.7b a target is 100 meters from the 1 km long baseline of the corresponding bistatic pair. This measurement is quite distorted and some information can be retrieved, but it will still deteriorate the output of the combination of multiple sensors considerably.



(a) (b)

Figure 3.7: Difference with or without ambiguous sensor

4. Inverse Synthetic Aperture Radar

A monostatic imaging technique to improve radar cross-range resolution is Inverse Synthetic Aperture Radar (ISAR). ISAR makes use of the rotational motion of a target, while (in this case) the radar itself is stationary. In this chapter the introduction to ISAR will be given as well as a time domain processing technique to perform ISAR imaging. Also the expansion to bistatic ISAR will be discussed. In the following chapters an incoherent multistatic ISAR approach (5) and a coherent multistatic ISAR (6) approach will be described.

4.1 Monostatic ISAR

The resolution [2] that can be achieved with monostatic ISAR is dependent on the rotation of the target. The larger the observed rotation from the target is, the higher the cross-range resolution is. In figure 4.1 a scatterer (S) rotates around a certain point. The cross-range and velocity of the scatterer are both denoted in the figure and will be used to derive an equation for the monostatic ISAR cross-range resolution.

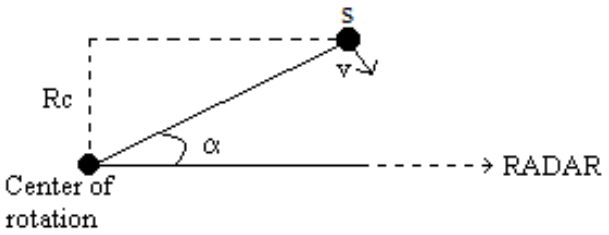


Figure 4.1: ISAR concept

The speed of the scatterer (target) in figure 4.1 can be calculated using the rotational speed, ω, and the cross-range,R_c, to the center of rotation.

$$v = \frac{\omega R_c}{\sin(\alpha)} \tag{4.1}$$

The Doppler shift can be calculated as follows:

$$f_d = \frac{2v\sin(\alpha)}{\lambda} \tag{4.2}$$

where $v\sin(\alpha)$ is the radial speed of the target. Combining 4.1 and 4.2 gives

$$f_d = \frac{2\omega R_c}{\lambda} \qquad \text{or} \qquad R_c = \frac{\lambda f_d}{2\omega} \tag{4.3}$$

In equation 4.3 it can be seen that the Doppler frequency and the cross-range are linearly related to each other. By differentiating 4.3 a relation between an increment in Doppler and cross-range can be found.

$$\Delta f_d = \frac{2\Delta R_c \omega}{\lambda} \tag{4.4}$$

The resolution in Doppler is directly related to the observation time, Δf_d is equal to $\frac{1}{T}$. Substituting this in equation 4.4 gives:

$$\frac{1}{T} = \frac{2\Delta R_c \omega}{\lambda} \qquad \text{or} \qquad \Delta R_c = \frac{\lambda}{2\omega T} \tag{4.5}$$

The angular rotation rate times the integration time gives the total rotation of the target. This leads to the following equation:

$$\Delta R_c = \frac{\lambda}{2\phi} \tag{4.6}$$

where ϕ is the total observed rotation of the target in radians. In equation 4.6 the cross-range resolution achievable for mono-static radar is seen. If we change this scenario into a scene where the scatterer is not rotating around a fixed center, but moving along a fixed radar, the same equation can be used and the observed rotation is equal to angular displacement of the target. Hence, the faster the target moves, the larger the observed angle can be. If the target flies a perfect circle around the radar, there is no Doppler shift and the cross-range resolution cannot be improved. Also if the target flies in a straight line, without acceleration away from the radar, no increase in cross-range resolution can be achieved, because only one Doppler shift is seen by the radar. A rule of thumb for the expected cross-range resolution is

$$\Delta R_c = \frac{v}{B_{Doppler}}$$

which relates the observed Doppler bandwidth (is rotation) to the cross-range resolution.

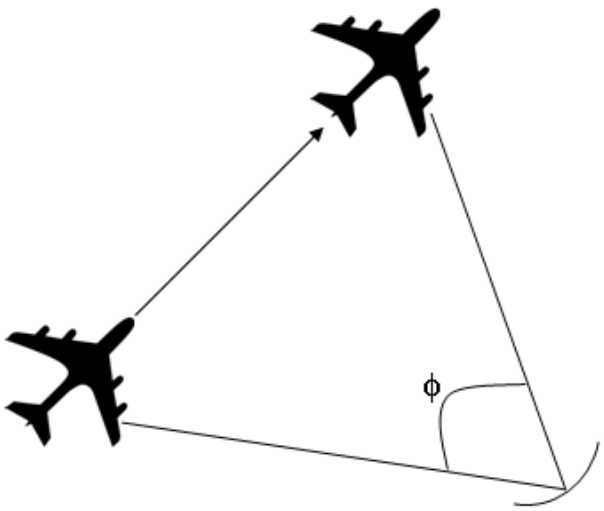


Figure 4.2: ISAR scenario

From figure 4.2 can be deduced that the rotation of the target is perceived by the radar as a Doppler frequency. Targets can be distinguished by a different Doppler history. A different range will cause a different slope in the Doppler history, while an angular displacement will cause a time shift in the Doppler responses. So two targets that are at the same distance will

have the same Doppler response, only shifted in time. With this information a high resolution cross-range estimate can be made. To convert the Doppler frequencies into a cross-range, the equation for the Doppler frequencies is rewritten. The equation to calculate the Doppler shift is:

$$f_d = -\frac{2v_r}{\lambda} = -\frac{2v\sin(\alpha)}{\lambda} \tag{4.7}$$

The sine can be written in terms of down- and cross-range as follows:

$$f_d = -\frac{2v\frac{R_c}{R_s}}{\lambda} \tag{4.8}$$

where R_c and R_s are the cross-range and slant-range respectively. And if we rewrite equation 4.8 the following expression for cross-range is found:

$$R_c = -\frac{f_d \cdot \lambda \cdot R_s}{2v} \tag{4.9}$$

Now that we have an expression for the cross-range, the axis can be scaled correctly and a figure of the monostatic ISAR simulation can be made. On the vertical axis the cross-range is plotted and on the horizontal axis the slant range. In the simulation shown in figure 4.3 the slant range is equal to the down-range.

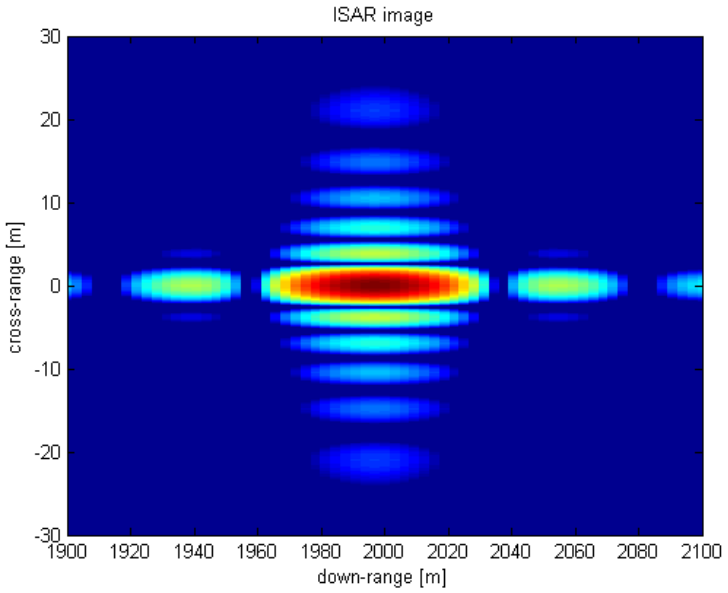


Figure 4.3: ISAR simulation

The ISAR image seen in figure 4.3 is made by storing received signal in a buffer. After a certain amount of time, a fixed number of pulse responses are present in the buffer. The down-range compression is done by convolving the received signals with the transmitted signal. As can be seen in figure 4.4 these signals are stored in a buffer. All the signals have a slightly different Doppler, which will be used for the azimuth compression. For azimuth compression the signals are convolved with a Doppler history.

This history is dependent on the range and angular displacement as discussed before. As we saw in chapter 2, a larger transmit bandwidth is directly related to a high range resolution, here a large Doppler bandwidth is related to a high cross-range resolution as explained in this chapter. A visualization of the buffer that contains range compressed signals and the corresponding cross-range FFT is seen in figure 4.4. The FFT is denoted due to the fact that a convolution takes place much faster in the frequency domain. The compression does not take place by taking a FFT over the received signals.

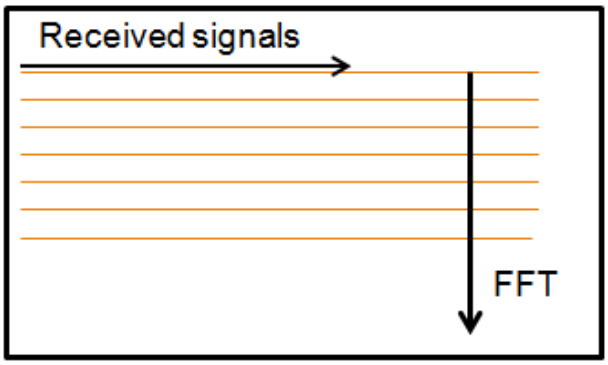


Figure 4.4: Frequency domain processing

4.2 Bistatic ISAR

When the expansion to a bistatic scenario is made, a compensation for the geometry has to be included. This is almost the same compensation as with the range resolution, except that also the observed rotation differs from the monostatic case.

$$\Delta R_c = \frac{\lambda}{\left(2\cos(\frac{1}{2}\varphi_{bi})\right)\theta} \tag{4.10}$$

with φ_{bi} is the bistatic angle and θ the average observed angle of the transmitter and receiver.

$$\theta = \frac{\alpha + \beta}{2}$$

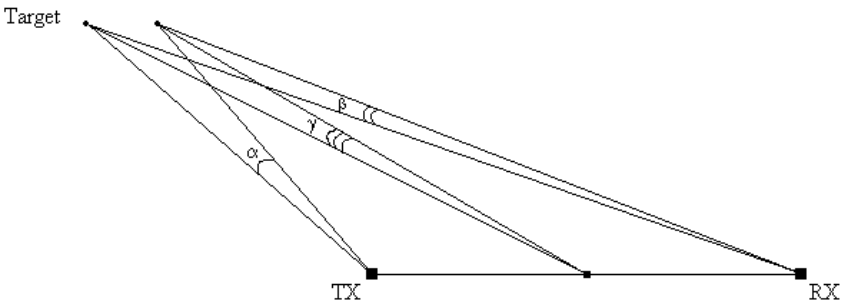


Figure 4.5: Bistatic observed angles

As with the monostatic configuration a conversion from Doppler frequencies to cross-ranges has to be made. With bistatic configurations the Doppler is calculated in a somewhat more complex way, because it is a combination of frequency shifts towards different positions. As stated before, the equation for the Doppler frequency in bistatic configurations is:

$$f_d = -\left(\frac{R_t \bullet V}{|R_r|} + \frac{R_r \bullet V}{|R_r|}\right)/\lambda \tag{4.11}$$

As can be seen, there are too many unknowns, so we cannot solve the equation directly. If we assume that we know the target position, and couple the cross-range seen by transmitter and receiver by using the known distances between them, equation 4.11 can be rewritten as follows:

$$\begin{aligned} f_d &= -\frac{V \cdot R_c}{\lambda \cdot R_{d1}} + \frac{V(D - R_c)}{\lambda \cdot R_{d2}} = \frac{V}{\lambda}\left[\frac{(D - R_c)}{R_{d2}} - \frac{R_c}{R_{d1}}\right] \\ \frac{f_d \cdot \lambda}{V} &= \frac{D \cdot R_{d1}}{R_{d1} \cdot R_{d2}} - \frac{(R_{d1} + R_{d2})}{R_{d1} \cdot R_{d2}} R_c \\ R_c &= -\left[\frac{f_d \cdot \lambda}{V} - \frac{D \cdot R_{d1}}{R_{d1} \cdot R_{d2}}\right] \frac{(R_{d1} + R_{d2})}{R_{d1} \cdot R_{d2}} \end{aligned} \tag{4.12}$$

With D is the distance between the transmitter and receiver in meters and R_{d1} and R_{d2} are the distances from the target to the transmitter and receiver respectively. The cross-range is defined in the plane parallel to the baseline of the bistatic set.

5. Incoherent multistatic ISAR

Two different multistatic approaches of ISAR will be discussed in this thesis. In chapter 6 coherent multistatic ISAR will be described and why it is useful if the observed target does not rotate enough to increase the cross-range resolution. Also some geometrical constraints will be discussed. In this chapter incoherent multistatic ISAR will be discussed and another technique to calculate ISAR images, called brute force processing. The processing technique is coherent, but the summation of the images takes place incoherent.

In the scenario that will be used throughout this chapter, the transmitters and receivers are placed far apart from each other, so the target can be observed from different non overlapping angles. Multistatic ISAR may then be used to create relatively high resolution 2D images of a target, without the use of radars with a large transmit bandwidth. Typical surveillance and MTI radars may benefit from a low resolution to perform their functions. For the MTI radar it is important that the target stays in one range-bin to perform clutter rejection. The speed is unknown in advance, so a low resolution is desired to cover a large range of possible speeds. After the target is detected imaging can take place with the same low resolution radars. An advantage of a smaller bandwidth is a possible exploitation of multiple sub-bands. To distinguish between the different transmitters they can be identified by their transmit frequency. If a smaller bandwidth is used by the radars, more transmitters can operate in limited bandwidth and thus the hardware can stay reasonably priced. Also synchronization of transmitters and receivers is easier to accomplish, because the allowable synchronization errors are related to the transmitted bandwidth [4]. To create a 2D ISAR image the configuration does not have to be coherent,

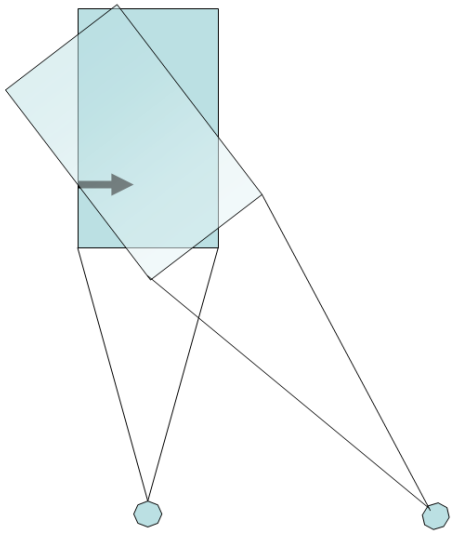


Figure 5.1: 2D ISAR situation

because resolution improvements are also achieved by incoherent summation of the images. It is for example possible to make use of multiple data sets from independent monostatic or bistatic

radars, as long as they see the target under different angles. The observation angles should preferably not be orthogonal, because then there will be a chance that the second sensor will not provide any extra information. The combination of ISAR images may create a usable 2D image of the target to perform a better and/or faster classification, because the overall dimensions of the target may be better estimated. Also scatterer patterns can be used to determine what kind of target has been imaged. In figure 5.1 a situation sketch for 2D ISAR is shown. Two different radar systems image the target under a different angle. In the traditional approach discussed in the following section the imaged areas (here in light blue) have a different orientation, which will have some consequences on the merging of the data.

5.1 Traditional approach

When there are several measurements available, they may be combined. To solve the problem that the measurements are made from different positions, an appropriate method has to be found to combine them. At first glance (fig 5.1) it looks like one of the images has to be rotated and translated to fit the other image. When this approach is followed, many difficulties will arise. For example, when rotating a non-uniform grid, i.e. axes have a different step size, the new axes are a rotated version of this non-uniform step size and the step size becomes position dependent. A rotation of a matrix is done by multiplying a matrix with another matrix, called a rotation matrix. A 2D-rotation matrix has the following form:

$$\begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

The size of the rotated ISAR image is exactly the same of the original, in terms of samples or grid points. As stated before, some non-uniform scaling has to be applied, because the axes are now a combination of two different step size axes. A standard linear scaling matrix has the following form:

$$\begin{bmatrix} v_x & 0 & 0 \\ 0 & v_y & 0 \\ 0 & 0 & v_z \end{bmatrix}$$

A non-uniform scaling has functions for the parameters v_x, v_y and v_z, instead of just a number. A problem with this non-uniform scaling that arises when the combination with the first image takes place, is that it has a different step size than the rotated and scaled image. The baseline image has to be interpolated to match the new grid of the second image. After some consideration this method of combining the two ISAR images is rejected. A lot of post processing is needed to combine the two ISAR images, without knowing if the end result will be in any way an improvement. The non-uniform scaling and the rotations are feasible, but the coherent combination of the two images is disputable. The outcome of this method is more likely an incoherent combination.

5.2 Time Domain approach

With a Brute Force approach, or Time Domain processing, most of the calculations are done in the time domain. The second Fourier transform that is normally used for the cross-range compression is now substituted with an interpolation and a summation in the time domain. One of the advantages is that with time domain processing the ISAR images can directly be calculated on any arbitrary grid, thus eliminating the need for the rotation and scaling as with the traditional approach. For the range compression the FFT is still used, because it is a good and fast way to perform the range compression. The amount of CPU power and the available time are a bottleneck for brute force processing, because it requires a great amount of computation. The advantage is that is very accurate, because it does not work with assumptions, that are required to get fast results with the traditional FFT methods. Due to the linearity of time domain processing it can be easily programmed and with the right CPU it is possible to calculate the images in real time. The processing described here is a derivation of the Brute Force SAR approach by [5].

The first step in brute force ISAR processing is making a range compressed signal, using an FFT and the transmitted signal. This range compressed signal can then be interpolated to an arbitrary grid. When imaging takes place from different directions, this grid can be located at exactly the same point in space for all the radars, which makes the fusion of the ISAR images very easy. The grid as defined here is chosen in such a way that the main scatterer is always at the center of the grid. This means that with every new radar observation a new geometry between the target and radar is present and a new grid has to be calculated around the target. Two values are given to all the grid points, a two-way range towards that specific grid point and a phase corresponding to the transmitted signal. With the range information the interpolation over the grid is performed and the phase information allows for the cross-range compression, which is essentially the crux of the time domain processing. The distance towards each point in the grid is calculated using the Pythagorean Theorem. The one-way phase shift can be calculated using:

$$\phi_k = 2\pi \frac{r_k}{\lambda} \tag{5.1}$$

Here ϕ_k is the phase of the k-th pulse at range position r_k, which is the one-way distance between the radar and the target. The two-way phasor corresponding to this phase shift is

$$\Theta_k = e^{j2\pi\phi_k} \tag{5.2}$$

Θ_k is the phase grid that is needed for the processing. After interpolation of the range compressed signals over a predefined grid, all the grid points are multiplied with this phase grid. Where the phasor and the transmitted signal have the same phase, they will coherently add up. The received signal is then correctly shifted back along the expected phase

$$v_k = s_k \cdot \Theta_k^* \tag{5.3}$$

In equation 5.3 the representation of the k-th pulse, v_k, is shown. The range compressed and interpolated signal s_k is multiplied with the phase grid Θ_k. Assuming that the reflections of the target only add up coherently for the spots of interest, the image can be reconstructed by just adding up all the images v_k. The final ISAR image v is thus the summation of all k representations:

$$v = \sum_{k=1}^{N} v_k \tag{5.4}$$

To visualize the equations 5.1 to 5.4 a block diagram and a simulation of a target is made. In figure 5.2 the block diagram of the necessary steps for Time domain ISAR processing is shown.

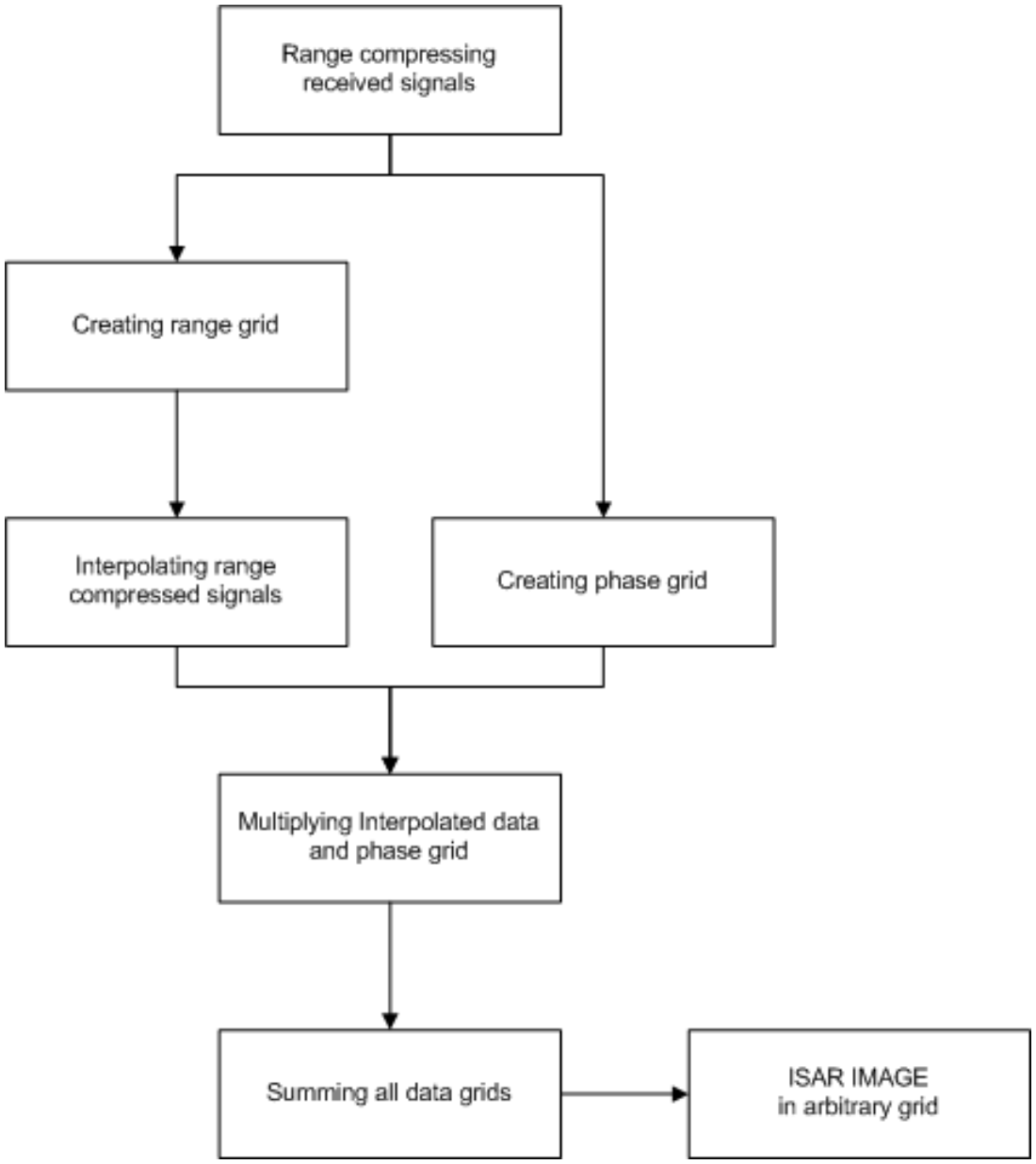


Figure 5.2: Block diagram for time domain ISAR processing

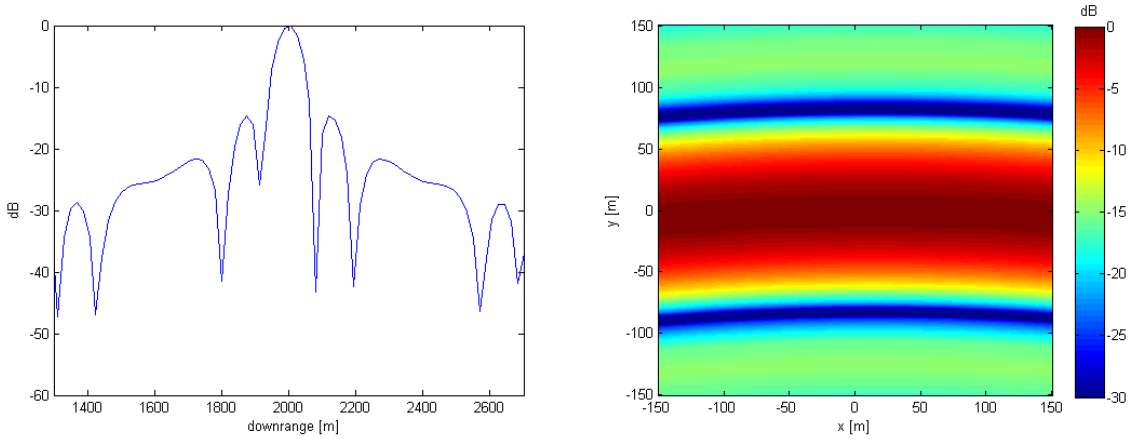


Figure 5.3: Range compressed signal and the interpolation of the signal on a grid

In figure 5.3a a range compressed response of a target can be seen. After interpolation over a predefined grid the image $v_k = s_k[r_k]$ is seen. Figure 5.3b represents the interpolated image that belongs to the range compressed signal. The curvature of the response is clearly visible and comparable to what has been seen in chapter 2, i.e. the monostatic range response.

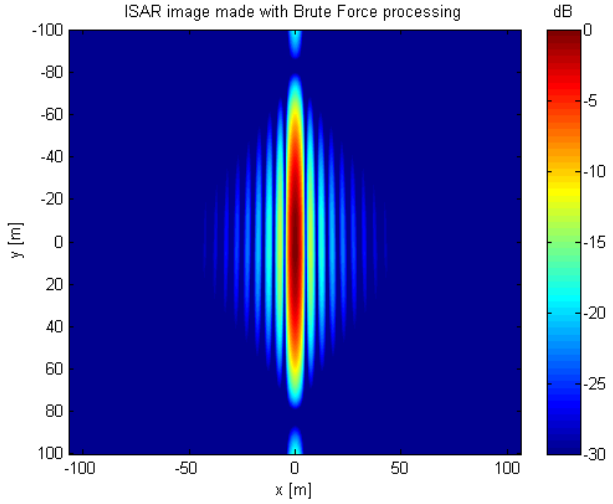


Figure 5.4: Result of equation 5.4

When this is multiplied with the appropriate phasor and then summed with the responses at later time instances, the result will look like figure 5.4. In every grid that has been made, the target was exactly in the middle and the phase of the target is correctly shifted with the phasor grid. After summation only the part where the phases are constructively interfering so that the response is amplified are visible in the picture.
To check whether this approach gives results that are correct a comparison with the theoretical outcome has to be made. First lets sketch a monostatic scenario where the target is flying at a distance of 3 km with a speed of 90 m/s in a straight line past a radar, which has an integration time of one second and a wavelength of 0.15 m. In equation 4.6 the cross-range resolution of a monostatic ISAR configuration is shown. The total observed angle is:

$$\theta = 2 * \arctan\left(\frac{45}{3000}\right) = 0.03rad$$

which leads to a cross-range resolution of:

$$\Delta R_c = \frac{0.15}{2 * 0.03} = 2.5m$$

An intersection of figure 5.4 is shown below, with the top and the first null (both) selected, which indicates the same distance as the -3 dB points. As can be seen, the Brute force ISAR gives the cross-range resolution as calculated before.

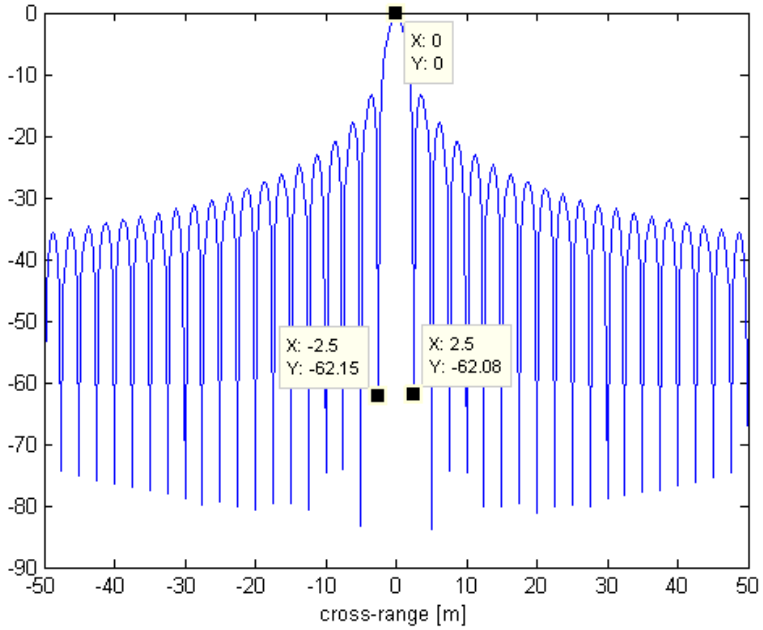
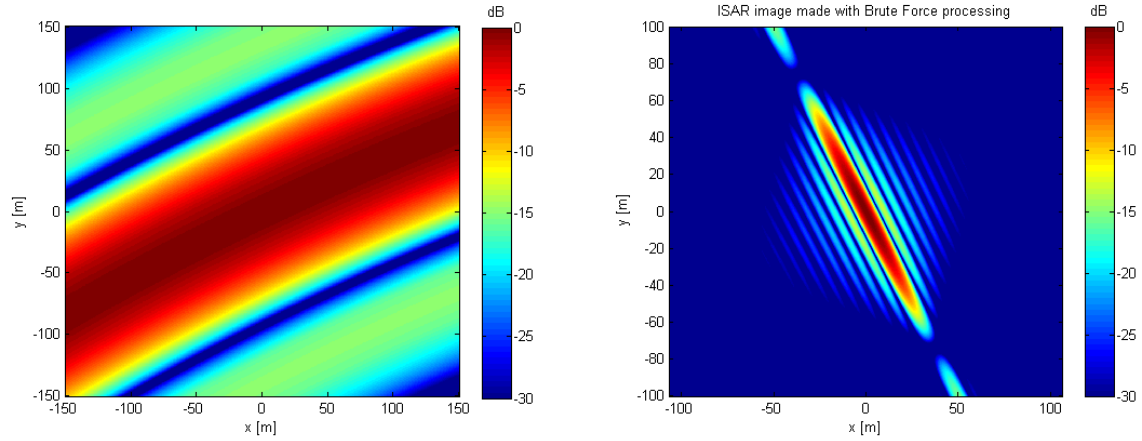


Figure 5.5: Cross-range resolution

If we repeat this approach for a second monostatic radar that looks at the target from an angle, like in figure 5.1, we can again create an image as in figure 5.3b and 5.4.



(a) One image

(b) After summation

Figure 5.6: Single image and summation of images from a target seen under a certain angle.

Both simulations are made on the same grid, so the summation of the two ISAR images is no longer a problem, as it was with the frequency domain calculations, where a rotation and scaling was needed to come to an equivalent grid.

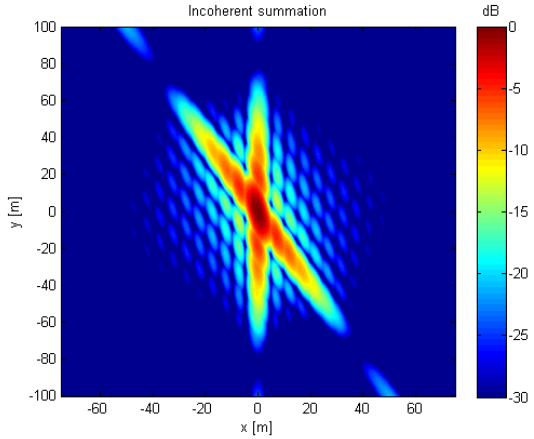
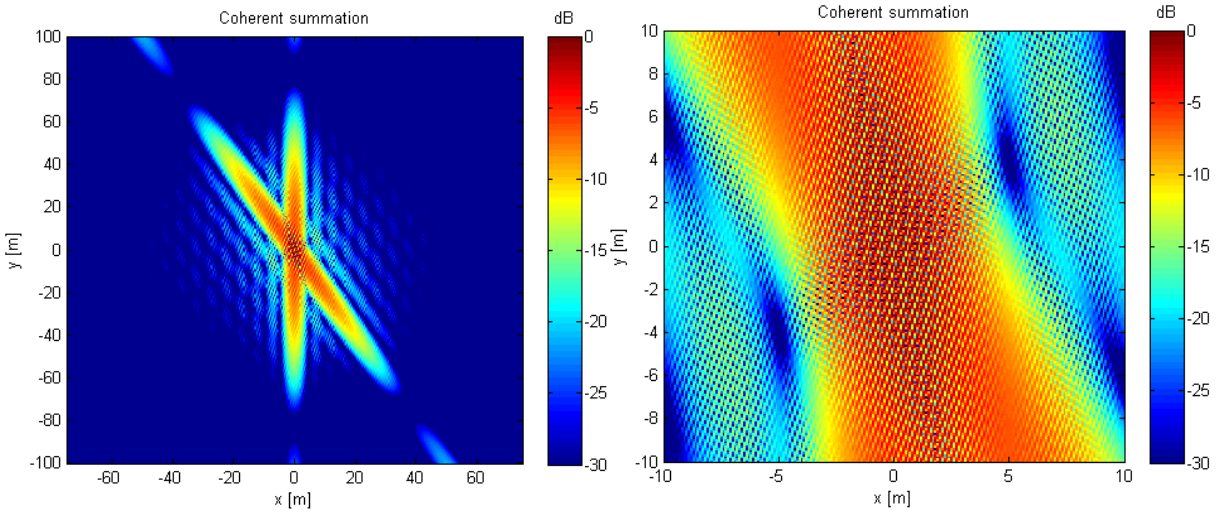


Figure 5.7: Incoherent combination of two voxels

Because all the responses are perfectly aligned, as a result of the multiplication with the phasor grid, the "incoherent" summation of the two signals already gives the best result possible, due to the fact that the main peak of the scatterers are coherent. If a coherent summation is performed, a lot of grating lobes will be visible in the end result. This is caused by the fact that the phase responses of both signals have a different angle, i.e. the waves have propagated in different directions, so the direction of the wave front is different. If a compensation is made for the direction of the phase, by multiplying the second signal by a phase corresponding to a path length difference, the coherent summation starts to look like the incoherent summation. This seems correct, because the two responses are already completely coherent at the scatter positions, as a result of the multiplication with the phasor grid.



(a) Coherent summation

(b) Zoom-in of figure 5.8a

Figure 5.8: Coherent combination of two voxels

The main idea of this 2D ISAR approach was to improve the down-range resolution by adding another receiver to the configuration. This scenario can only detect scatters that are not too close to each other, otherwise after summation the relatively large down-range resolution will create additional scatters, while they are not present. The limitation of this approach is shown in figure 5.9. This can only be solved by increasing the bandwidth, which results in a better down-range resolution. Increasing the transmission bandwidth is not desirable. Although not the correct amount of scatterers is shown, an estimate of the size of the target can be made. It can be clearly seen that the lower left scatterer and the upper right scatterer are from the target.

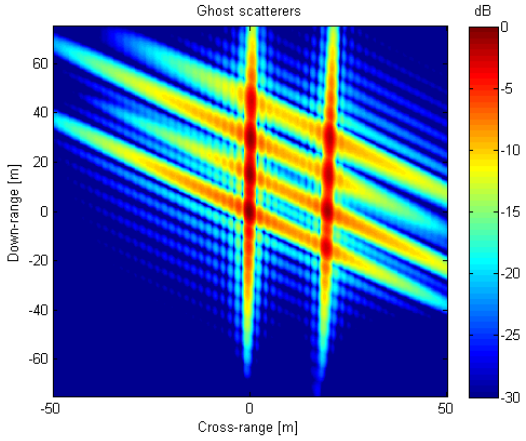


Figure 5.9: Multiple scatters relatively close to each other

Another disadvantage of this approach of combining two or more images is that a brute force approach requires a lot of computational power, either CPU or GPGPU[1] power. The great advantages are the easy (in)coherent summation that is possible, the high accuracy, because no assumptions had to be made and the straightforward implementation.

5.3 Incoherent bistatic ISAR

Just as before, the geometry in the bistatic case is even more important than in the monostatic case, because the geometry has some consequences for the creation of the phasor grid and the interpolation on the predefined range grid. For the creation of the phasor grid, the one-way distance to the target was calculated in the monostatic configuration, by knowing that the two-way path length is just double the one-way length. In the bistatic case the length of the path from transmitter towards the target is almost always different from the target to receiver path. This means that both path lengths have to be calculated in order to create the range grid and the phasor grid.

$$r_k = |TX - Target| + |RX - Target|$$

With this newly calculated range towards the phase-grid, equation 5.2 becomes

$$\Theta_k = e^{j\pi\phi_k} \tag{5.5}$$

It looks like equation 5.2, but because the factor 2 in the exponent is already incorporated into r_k and in ϕ_k it is a little bit different. To check whether these adjustments to the equations are sufficient another simulation will be made. An extension on the monostatic scenario as described

[1]General-Purpose computation on Graphics Processing Units

before will be made. The transmit and receive antennas are separated 3 km from each other and the target is 3 km away from the center of the base line. Again the target is flying in a straight line with a speed of 90 m/s and the wavelength is 0.15 m. Now equation 4.10 will be used to calculated the theoretical cross-range. The average observed- and bistatic angle are:

$$\theta = \arctan\left(\frac{3000}{1455}\right) - \arctan\left(\frac{3000}{1545}\right) = 0.024rad$$

$$\varphi = 2 * \arctan\left(\frac{1500}{3000}\right) = 0.9273rad$$

which leads to a cross-range resolution of:

$$\Delta R_c = \frac{0.15}{\left(2\cos(\frac{1}{2}\cdot 0.9273)\right)0.024} = 3.5m$$

An intersection of the bistatic ISAR image is shown below, with the top and the first null (both) selected, which indicates the same distance as the -3 dB points. As can be seen, the Brute force ISAR gives the same cross-range resolution as calculated before.

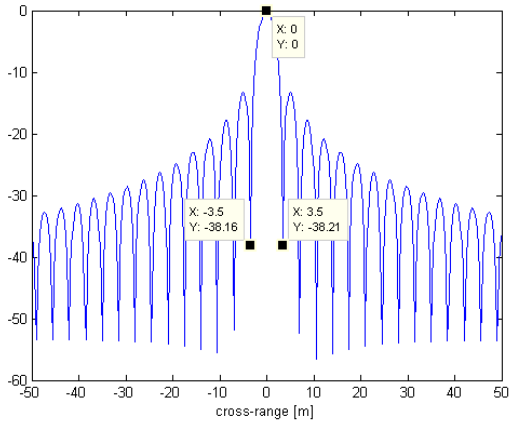
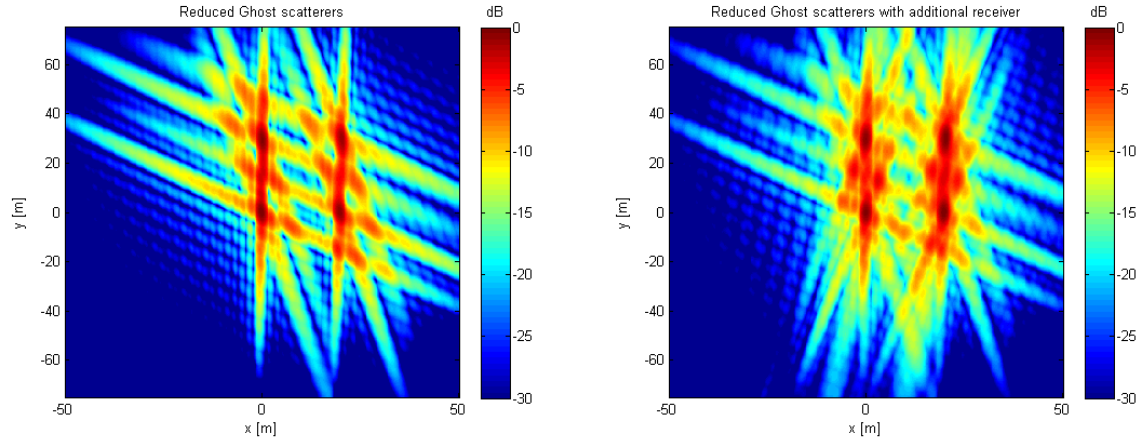


Figure 5.10: Bistatic cross-range resolution

5.4 Incoherent multistatic ISAR

Just as with the Doppler velocity vectors and range resolution, the Multistatic 2D ISAR scenario can be decomposed into mono- and bistatic sets. As already shown in figure 5.9 when there are several scatterers close to each other, ghost scatterers begin to appear. To mitigate this problem, the bistatic contribution of the two monostatic radars can be used as well. Because the bistatic response has the maxima on the same positions as the two monostatic responses, the amplitude at the position of the scatterers increases faster than those of the ghost scatterers. With the addition of an extra receiver spaced outside of the two monostatic radars the appearance of ghost scatterers can even be further reduced. This is shown in figure 5.11.



(a) Adding bistatic contribution of monstatic radars

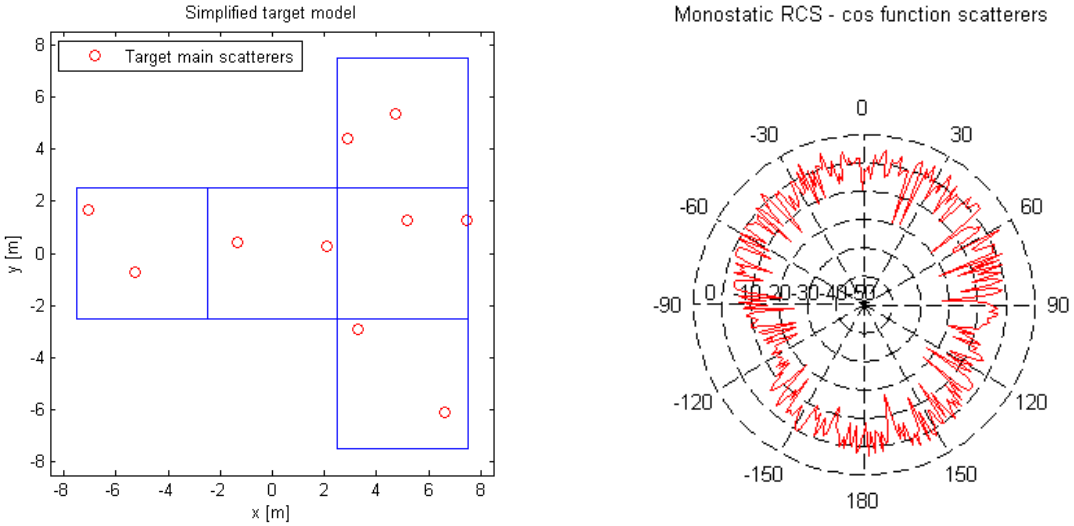
(b) Adding an extra receiver

Figure 5.11: Mitigating ghost scatterers

With the combination of the different ISAR images an estimation of the target size and shape can be made. This due to the fact that all the ISAR images have maxima at the same locations, which after summation will be the dominant spots on the ISAR map.

5.4.1 Complex RCS model

All the simulations in this thesis are based on a circular Radar Cross Section model. This means that all the scatterers have a RCS that is uniform. In reality it is impossible to see scatterers that are on the the opposite side of a target and the scatterers that are visible do not have an equal RCS. To check if the designed method still function a cosine RCS model is used to calculate the response of a target. Both the monostatic and bistatic RCS values of all scatterers are calculated. This is all the information that is needed to perform a multistatic ISAR simulation. In figure 5.12 a model of a target is shown and next to it the RCS of the target can be seen.



(a) Target scatter positions

(b) Target Radar Cross Section

Figure 5.12: Scatter positions of a target and its corresponding RCS

The situation sketch is shown in figure 5.13.

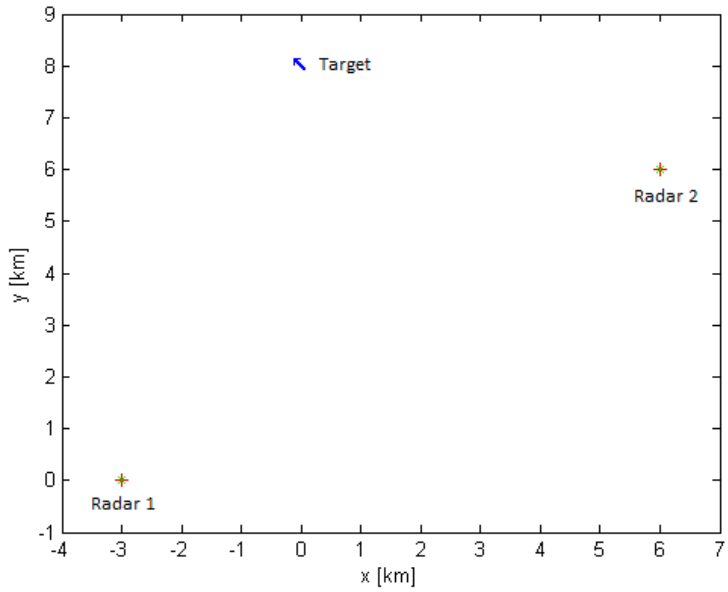
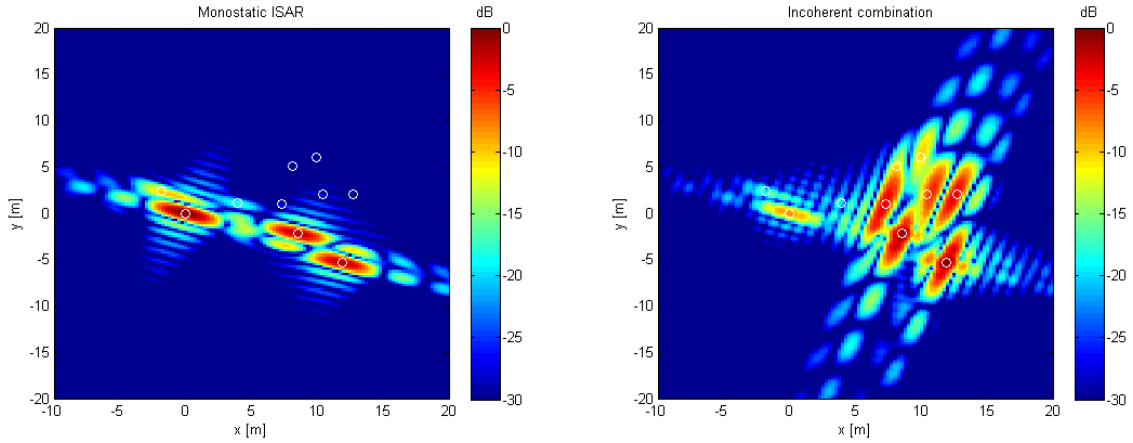


Figure 5.13: Situation Sketch

One of the radars, #2, is positioned in a rather bad place, although it is the closest radar to the target. Because the target is almost moving in a straight line away from it, the observed Doppler bandwidth is small. Combining the two radars incoherently can significantly improve the end result. In figure 5.14 the image made by the second radar and the combined image are shown.



(a) Monostatic ISAR image

(b) Incoherent multistatic ISAR image

Figure 5.14: Monostatic and multistatic ISAR response of the target from figure 5.12

It can be clearly seen that the incoherent combination does add information. The monostatic radar itself was capable of determining the position of three scatterers, while the multistatic configuration was able the distinguish seven different scatterers. With this additional information a better estimation of the target size can be made, and thus improving the upcoming classification of the target. The plot shows an incoherent combination of the two radars, because a coherent combination will show the same defects as explained in section 5.2. Grating lobes will appear in ISAR image, which deteriorates the end result. The radars are spaced too far apart from each other to observe a coherent target. Also there are gaps in the observation, which could degrade the final ISAR image if the same scatterers where observed. Even the integration method described before cannot fill up the gaps, because both radars see a completely different target in terms of scatterers. This shows that even though the geometrical restrictions can be eased, there is still a maximum gap that can be filled with the interpolation. The same scatterers must be observed by both radars. In figure 5.15 a coherent combination of the radars is shown, where the interference pattern can be seen very clearly. To illustrate the interference pattern better, the bistatic component is also added to the picture. As can be seen, due to the angle between the velocity vector and the baseline, which is almost 90°, there is a very low resolution in the bistatic component.

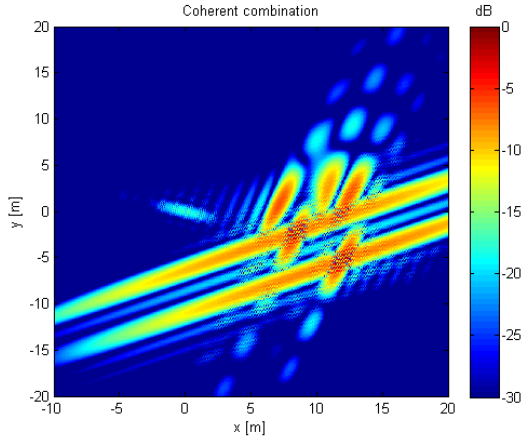


Figure 5.15: Coherent combination of the ISAR images

6. Cohererent multistatic ISAR

In chapter 4 monostatic ISAR and the extension to bistatic ISAR was discussed. In this chapter the extension to coherent multistatic configurations will be discussed and it is not so straightforward as the multistatic range resolution extension. In the coherent multistatic scenario the available sensors can also be decomposed into sets of bistatic and monostatic pairs as with the incoherent multistatic approach. If the radars are placed in a smart manner the observed rotation can be much larger than with bistatic and monostatic configurations alone. This holds only if the observed rotations are adjacent or overlap each other. Because overlapping parts of the target have to be imaged, the geometrical constraints for multistatic ISAR are very strict. In figure 6.1 two received observations are drawn in a circle. In this case the observed rotations are continuous and the observations can be merged. The resolution gain can be as large as the number of virtual receivers [6]. A virtual receiver is a phase point that represents a monostatic radar that is formed from a combination of a bistatic transmitter and receiver. For small bistatic angles the phase point will be right in between the transmitter and receiver. When the angle becomes too large, the virtual point cannot be constructed in such a way.

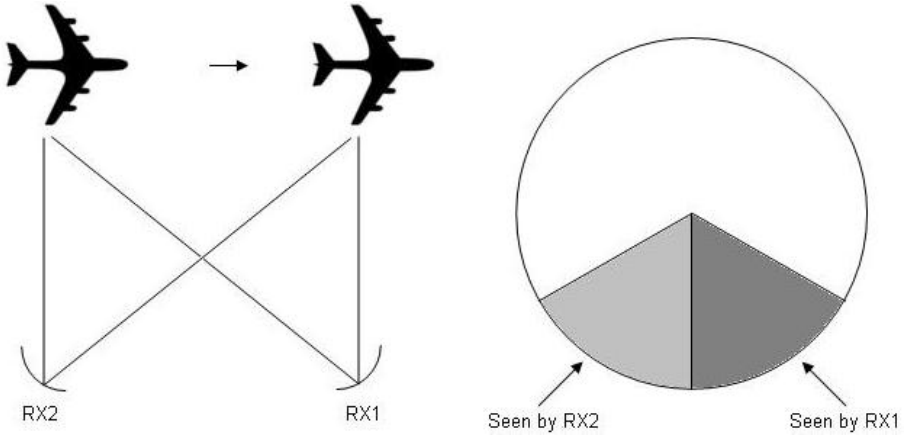


Figure 6.1: Multistatic ISAR

In figure 6.1 only the monostatic components are drawn. *RX1* and *RX2* both represent the receivers of a monostatic radar. In reality the distances between the radars could be larger, because the bistatic component could fill that gap. For two monostatic radars, the maximum allowable distance between the receivers is bounded by the speed of the target and the begin geometry. This can be deduced from figure 6.1. To make sure that the final observation angle of the first sensor coincides with the starting observation angle of the second sensor, the two sensors must be displaced by the speed of the target times the observation time. To extend this to all possible configurations, the sensor displacement is calculated with the aspect-ratios of a triangle that can be formed between the radars and the target. A right-angled triangle always has the same angles if the ratio between two of its sides stay equal. With this ratio and the displacement of the radars, the minimum observation time of a target can be calculated, before it can be coherently combined.

In figure 6.2 a scenario is sketched in which the angle towards the target increases.

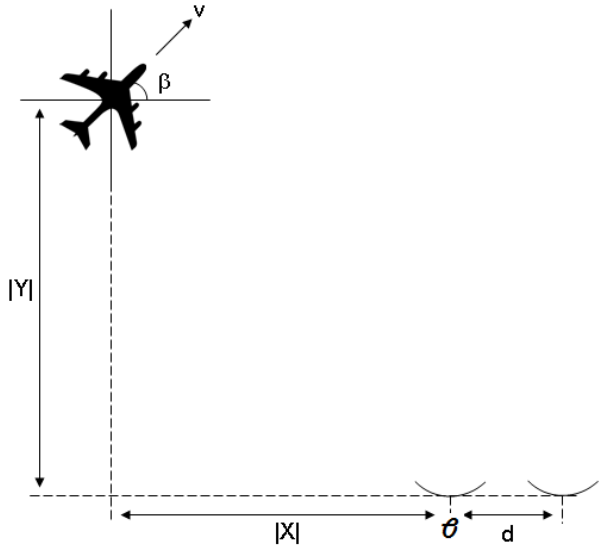


Figure 6.2: ISAR constraints

$$\frac{|Y| + \sin(\beta)v}{|X| - \cos(\beta)v + d} = \frac{|Y|}{|X|} \qquad \text{for } -0.25\pi < \beta < 0.75\pi \tag{6.1}$$

or

$$\frac{|Y| + \sin(\beta)v}{|X| - \cos(\beta)v} = \frac{|Y|}{|X| + d} \qquad \text{for } 0.75\pi < \beta < 1.75\pi \tag{6.2}$$

The equations 6.1 & 6.2 are only valid in the quadrant shown in figure 6.2. Thus for negative x-axis and positive y-axis assuming the origin at the left transmitter. For the other quadrants the only thing that has to be determined is if the target angle towards the closest sensor is decreasing (6.1) or increasing (6.2).
Multistatic ISAR can be a large improvement over the monostatic and bistatic case. Not only it is possible to increase the cross-range resolution for the same integration time as a monostatic radar, as well as it is possible to decrease the observation time and keeping the cross-range resolution constant equal to the monostatic case. Decreasing the observation time is very useful when using phased array radars, because imaging techniques like ISAR have a great impact on the time/energy budget that is available in such radar systems. Reducing the integration time however has a large impact on the geometry of the transmitters, receivers and target. This will be discussed in the following section.

6.1 Bandwidth interpolation

The geometrical constraints with multistatic ISAR are very stringent. The constraints are so stringent due to the fact that adjacent or overlapping areas of the target must be imaged. This means that by combining different sensors a continuous Doppler spectrum is observed. When a discontinuous Doppler spectrum is observed, the sidelobe levels of the response will greatly increase and thus deteriorate the end result. Figure 6.3 gives an exaggerated overview of the situation. In the upcoming simulations the target has a speed of $5m/s$, the two monostatic radars are spaced 10 meters apart from each other and the distance towards the target at time instance zero is 2000 meter from the second receiver. The spacing between the radars already results in a gap of 50% of the total observed angle. The observation time is one second.

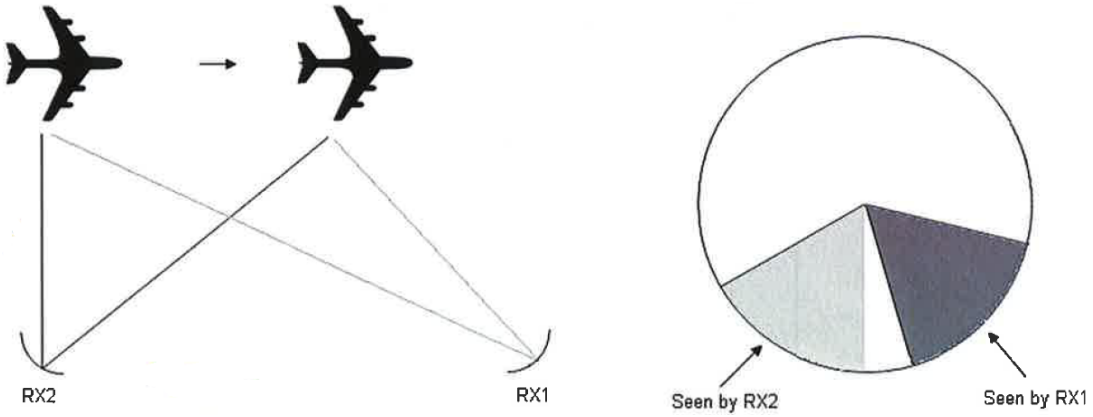


Figure 6.3: Multistatic ISAR with a gap in the observation

In figure 6.4 two simulations are shown with a discontinuous Doppler spectrum. The target rotated only 2.5 millirad. This rotation is seen by both radars, so in this case the total observed rotation is 5 millirad. Between the two observed rotations a gap is added, a gap of 1 millirad and one of 2.5 millirad. The normal sidelobe levels are -13 dB, but in these two simulations the sidelobes are -7 dB and -3.5 dB respectively. The gaps seem very small, but in the scenario described here, the target only rotates 5 millirad, so in this case a relatively large gap is present in the data.

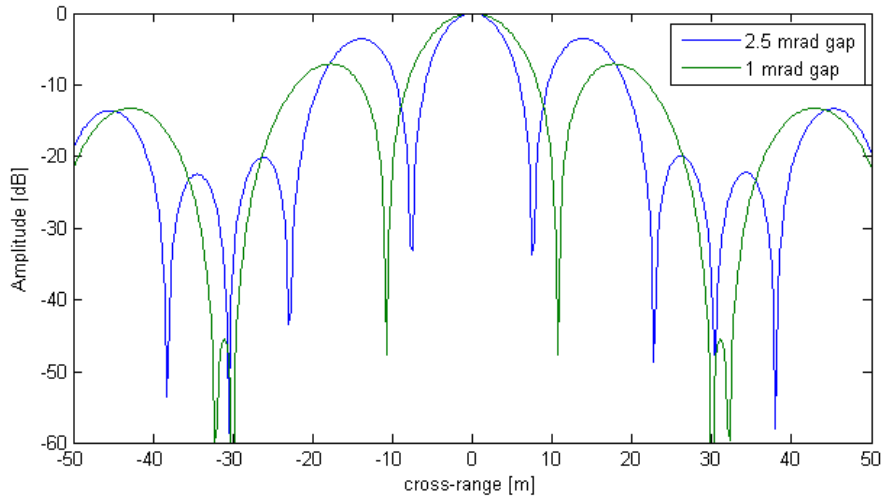


Figure 6.4: Combination of non-overlapping ISAR images

The apparent increase in resolution comes from the fact that the data is windowed with some form of block like pulse. Transformed to the frequency domain this windows also looks like a sinc function. Combining two sinc functions results in an increased resolution, but with higher sidelobes. A possible solution to the deterioration caused by the higher sidelobe levels is trying to interpolate the data between the different observations, assuming the target has a finite complexity. By interpolating the data the theoretical resolution may be achieved again in the simulations, while maintaining the -13 dB sidelobe levels. This approach reduces the geometrical demands on the multistatic ISAR configuration. The interpolation is done with an autoregressive (AR) model, where the number of poles is dependent on the available data. If there is more data available, more poles can be calculated and the more precise the interpolation will be. Also the number of scatterers that can be interpolated is dependent on the available amount of data. When the gap between the observations is closed by interpolation, the sidelobes will return to their original level of -13 dB. If overlapping areas are imaged or the interpolation is done in such a way, that overlapping portion of the data is available twice, the resolution will deteriorate, but the sidelobe levels will decrease even further, which is especially useful when there is a lot of noise present in the measurements.

Figure 6.5: Left: Two overlapping measurements.
Right: Corresponding window.

Imaging overlapping areas can be compared to the usage of a window over the available data. The data midway the observations have a higher amplitude than the data at both ends. It can be seen as a discrete window that has an amplitude of $\frac{1}{2}$ at both ends of the data and an amplitude of 1 where the samples that overlap. Windowing suppresses sidelobes, but also decreases the resolution of a measurement.

Derivation of an autoregressive model

In this subsection a quick derivation of an autoregressive (AR) model will be given. An AR model is also known as an all pole filter and is used to model time series. Here the model will be used to model two ISAR images and interpolate the missing data in between. An AR model is defined with the following equation

$$y_n = -\sum_{m=1}^{m=p} a_m \; y_{n-m} + e_n \tag{6.3}$$

where p is the model order, y_n are the samples and a_m are the coefficients. e_n is the residual and the goal of the Burg method is to minimize the total energy in the residual:

$$E = \sum_{n=1}^{n=N} |e_n|^2 \tag{6.4}$$

To find the filter coefficients usually the Yule-Walker equations have to be solved. But when there are periodic signals present in the data, the Yule-Walker equations can lead to incorrect

coefficients estimation and it is better to use the Burg method. The Burg method is also much more stable than the Yule-Walker method [8]. The Yule-Walker equations directly estimate the AR parameters, where the Burg method estimates the reflection coefficients. The reflection coefficients represent an unbiased estimate of the partial correlation coefficients and can be computed sequentially by minimizing the sum of forward and backward prediction errors. To visualize what the reflection coefficients do, a lattice filter is shown in figure 6.6. It can be seen that the higher order errors are a combination of the previous errors.

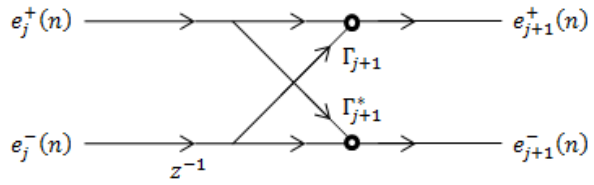


Figure 6.6: Example of a Lattice filter

With the Levinson-Durbin recursion and the corresponding reflection coefficients the AR parameters can be determined. The derivation of the Burg method [9] will be described in this subsection starting with the forward and backward prediction errors that can be written as:

$$\xi_j^B = \xi_j^+ + \xi_j^- = \sum_{n=j}^{N} |e_j^+(n)|^2 + \sum_{n=j}^{N} |e_j^-(n)|^2 \tag{6.5}$$

This equation blends the forward and backward errors with equal emphasis and minimizing this error may be justified on the statistical equivalence of $y(n)$ and $y^*(N-n)$. To find the reflection coefficient that minimizes ξ_j^B the derivative of ξ_j^B towards the reflection coefficients Γ_j^B is set equal to zero:

$$\begin{aligned} \frac{\delta}{\delta\left(\Gamma_j^B\right)^*}\xi_j^B &= \frac{\delta}{\delta\left(\Gamma_j^B\right)^*}\sum_{n=j}^{N}\left\{|e_j^+(n)|^2 + |e_j^-(n)|^2\right\} \\ &= \sum_{n=j}^{N}\left\{e_j^+(n)\left[e_{j-1}^-(n-1)\right]^* + \left[e_j^-\right]^* e_{j-1}^+(n)\right\} \end{aligned} \tag{6.6}$$

The update equations for the prediction errors are linked to the reflection coefficients. The forward and backward update equations for the prediction errors are as follows:

$$e_j^+ = e_{j-1}^+ + \Gamma_j^B e_{j-1}^-(n-1) \qquad e_j^- = e_{j-1}^-(n-1) + \Gamma_j^B e_{j-1}^+ \tag{6.7}$$

Combining equations 6.6 and 6.7 and solving for the reflection coefficients Γ_j^B, the value of Γ_j^B that minimizes the error is found:

$$\Gamma_j^B = -\frac{2\sum_{n=j}^{N} e_{j-1}^+(n)\left[e_{j-1}^-(n-1)\right]^*}{\sum_{n=j}^{N}\left\{|e_{j-1}^+(n)|^2 + |e_{j-1}^-(n-1)|^2\right\}} \tag{6.8}$$

When the reflection coefficients are available the autoregressive model can be very efficiently programmed in the form of a Lattice structure or with the Levinson-Durbin recursion the poles

of the filter can be found. When all the model parameters are available it is possible to determine the value of samples outside the sampled domain and thus interpolate the available data. The implementation of the Burg algorithm used here is dependent on the amount of available data. If more data is present, more reflection coefficients can be calculated and the more precise the all pole filter will be. The model uses a number of poles that is one third of the available number of samples. Because the number of poles is dependent on the number of samples, the amount of samples should be made as large as possible. This can be done by increasing the observation time, or decreasing the step size (grid resolution) in the domain where the samples are taken. The second method will not alway increase the available amount of information. For example in case that the theoretical resolution of the picture is already smaller than the resolution on the grid. In section 6.2 the size and position of the response in the frequency domain will be discussed, and hereby the boundaries on the number of poles.

The interpolation method is derived from a bandwidth interpolation model to increase the down-range resolution made by [7]. The method described in that paper uses the information from both transmissions, in this case the ISAR images, at the same time to calculate the reflection coefficients. This results in a modification of equation 6.5 and 6.8. The new error function becomes:

$$\xi_j^B = \sum_{m=j}^{M}\sum_{n=j}^{N} |e_j^+(m,n)|^2 + \sum_{m=j}^{M}\sum_{n=j}^{N} |e_j^-(m,n)|^2 \tag{6.9}$$

With this new error function the reflection coefficients can again be formulated by taking the partial derivative to Γ_j^B. The new equation for the two-way reflection coefficients is:

$$\Gamma_j^B = -\frac{2\sum_{m=1}^{M}\sum_{n=j}^{N} e_{j-1}^+(m,n)\left[e_{j-1}^-(m,n-1)\right]^*}{\sum_{m=1}^{M}\sum_{n=j}^{N}\left\{|e_{j-1}^+(m,n)|^2 + |e_{j-1}^-(m,n-1)|^2\right\}} \tag{6.10}$$

With the two-way reflection coefficients the interpolation of the data can be performed. The reflection coefficients can be used to calculate the poles of the filter. The order of the model, i.e. the number of poles, determines how much data is used to calculate an interpolation sample between the two sampled domains. The higher the order of the model, the more samples are used to calculate the new samples. Each pole is a weight that is multiplied with a certain sample. Adding multiple of these samples multiplied with their corresponding weight will result in a new interpolation sample. The simplest form of a model is seen in equation 6.11, with $x(n-i)$ are the samples and a_i the value of the respective pole.

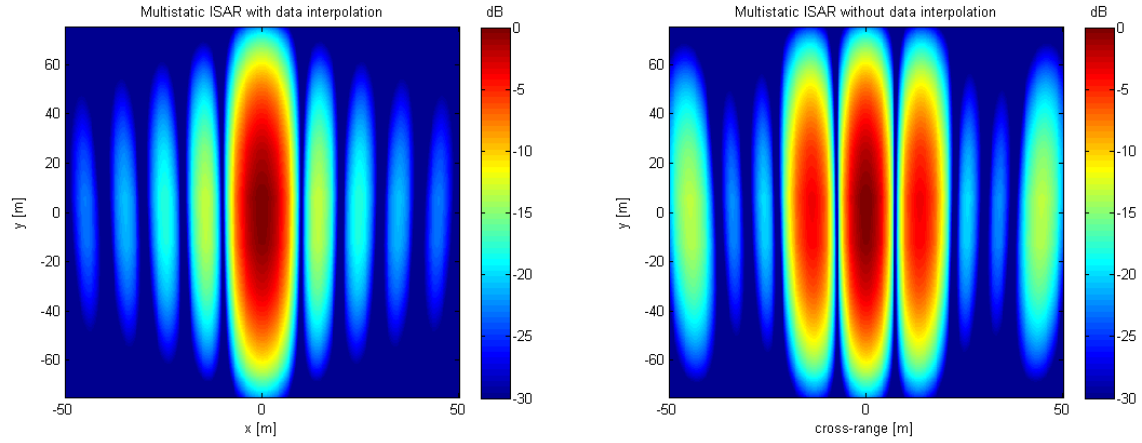
$$x(n) = a_1 \cdot x(n-1) + a_2 \cdot x(n-2) + \ldots + a_i x(n-i) \tag{6.11}$$

Bandwidth interpolation

All the interpolations done in the upcoming sections are based on the reflection coefficients calculated with equation 6.10. The interpolation increases the observed rotation, i.e. Doppler bandwidth or integration time, instead of the transmission bandwidth, for which this method was originally designed. Interpolation of the data in this section is done on the raw data that

is neither range nor azimuth compressed. The interpolation requires that new range and phase grids have to be calculated with an assumption for target speed and direction, because the time domain processing technique requires that additional information. The increase in observation time will be added to one of the radars that is performing ISAR. This means that one radar has the real observation time and the other radar receives all the extra information that is required to observe adjacent rotations. Close attention to the part of the target that is being imaged is important in placing the received signals in the correct order before interpolating them, the data may be swapped.

In figure 6.7 two multistatic ISAR simulations are made, where only the monostatic components are used. In 6.7a the gap between the two radars is interpolated and in 6.7b the two individual parts are summed without any additional actions. A cross-section of both figures is shown in figure 6.8. It can be clearly seen that the interpolated data has a better performance in terms of sidelobe levels, but the resolution is lower than when the two real data parts are summed.



(a) With interpolation

(b) Without interpolation

Figure 6.7: Range compressed signal and the interpolation of the signal on a grid

In the simulations in figures 6.7 and 6.8 a gap size of 50% was introduced. The response returns to the theoretical expected values when interpolation is being used, so it seems possible the loosen the stringent geometrical demands in order to perform multistatic ISAR.

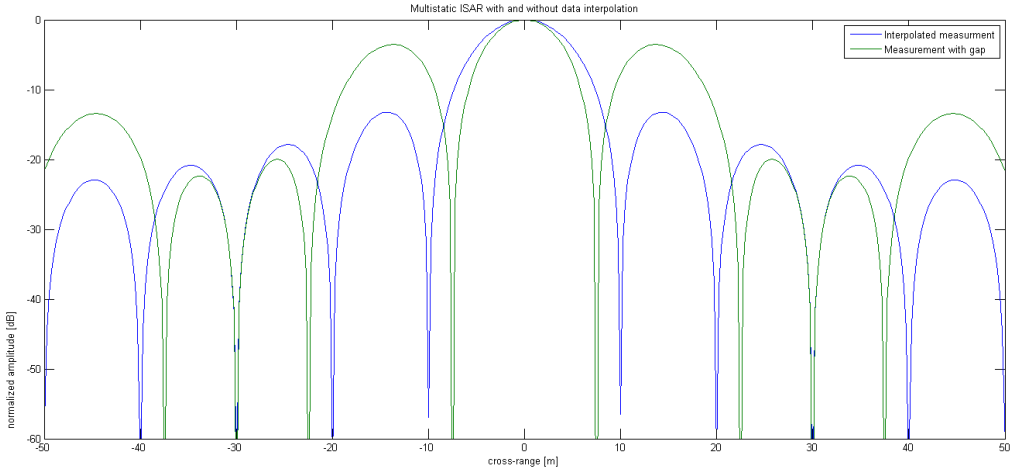


Figure 6.8: Cross section of the multistatic ISAR simulation of figure 6.7

6.2 Image interpolation

In section 6.1 an interpolation method based on the raw data is proposed. As explained when doing such an interpolation, the sequence of the received signal is important. Another weakness of that method is that it requires relatively more processing then needed, because of the fact that none of the information is compressed and additional range and phase grids have to be calculated. In this subsection the interpolation will take place on the final ISAR images of both radars. This method also has its limits, although it is now bound to the size of the grid on which the target is imaged and the transmission bandwidth. Furthermore with this approach the interpolation can be much better visualized and the interpolation takes place much faster. Also the interpolation parameters can be found more easily with this approach of the interpolation. A disadvantage is that due to the compression the number of poles that can be determined is not very large. This means that a smaller number of scatters can be interpolated than with the method described in section 6.1. Since a typical target only has a few dominant scatters this method may be used without a noteworthy degradation on the final image. All things considered, the interpolation on the final ISAR images is a more preferable method than the method described in the previous section. The interpolation will now take place in the frequency domain, after the ISAR images are completely constructed. To make the frequency domain images more clear, a situation sketch is shown in figure 6.9. The target consist of four scatterers that are positioned on a square with sides that are 20 meters long.

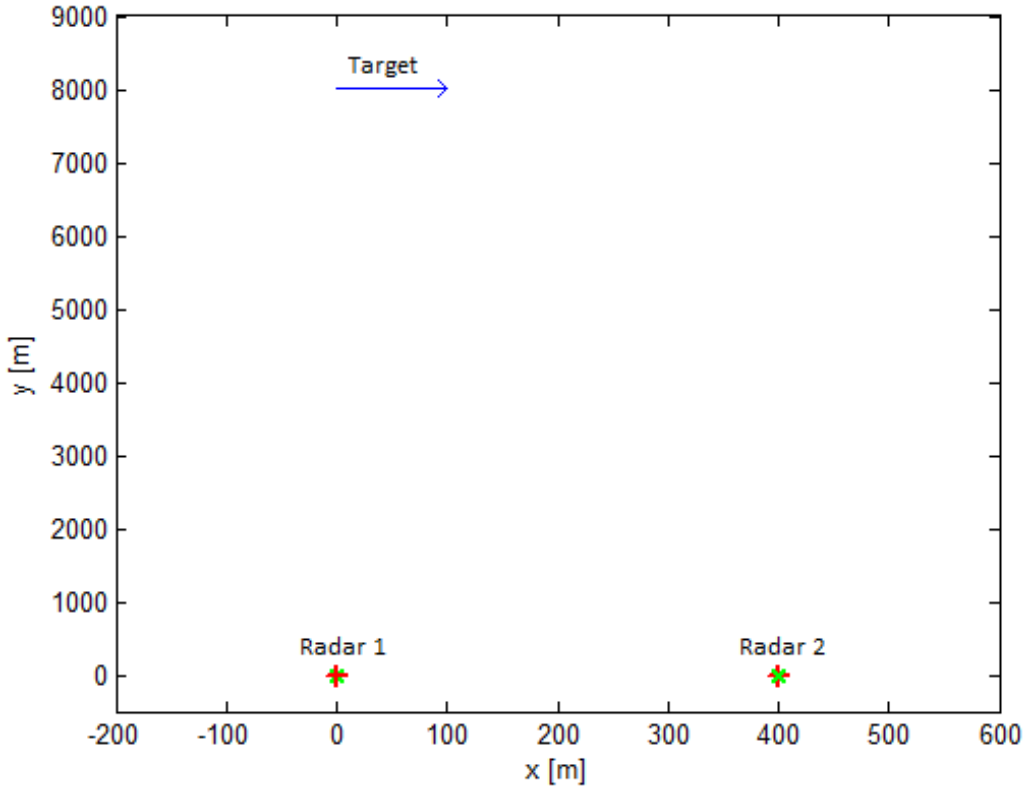


Figure 6.9: Situation sketch used in section 6.2

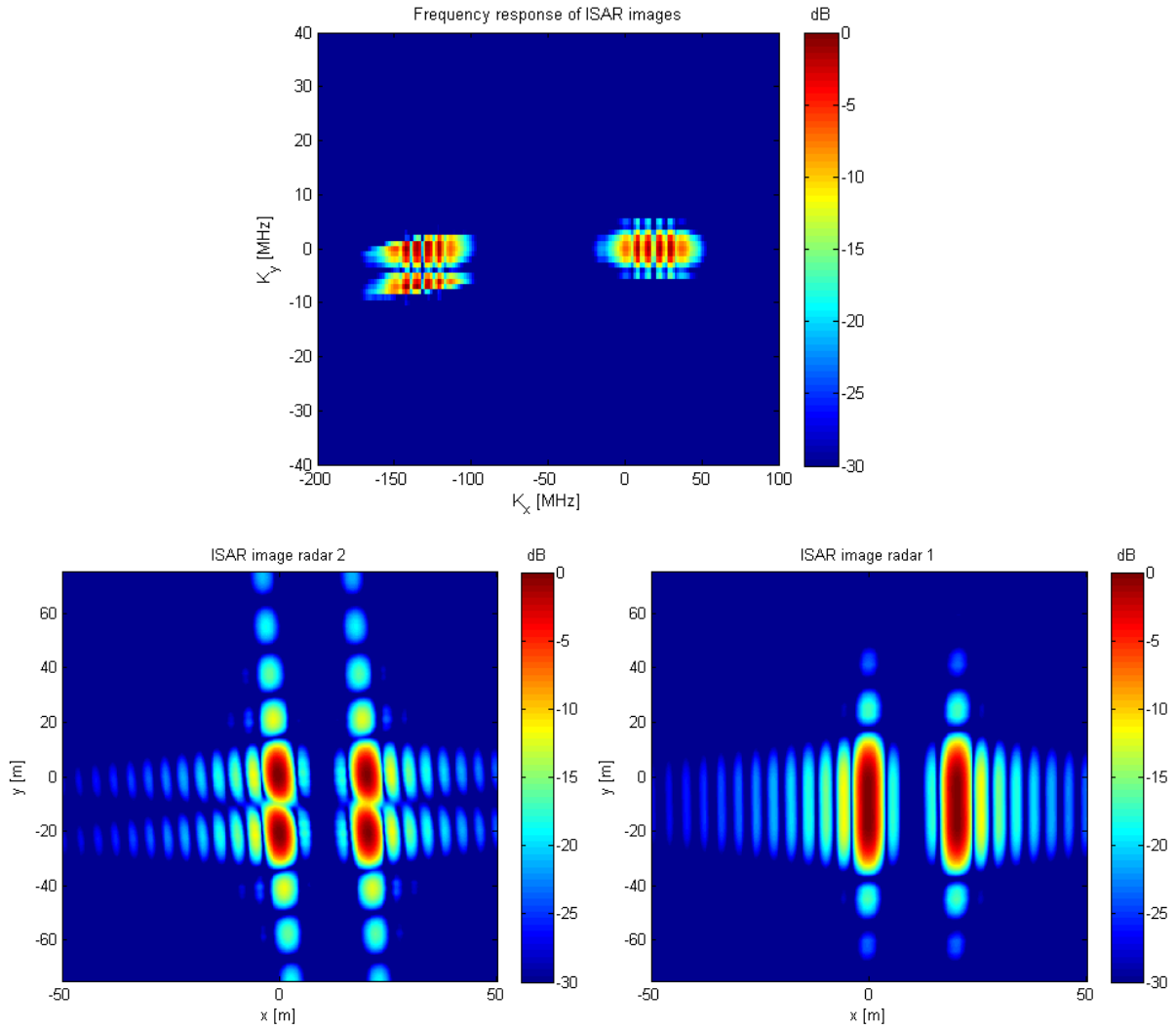
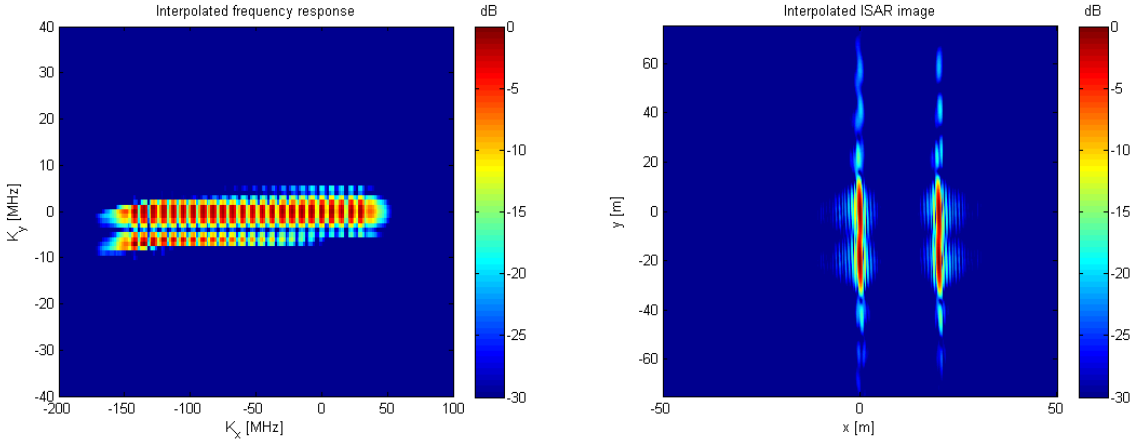


Figure 6.10: Above: Frequency response of both ISAR images
Below: Time domain ISAR images

In figure 6.10 the two separate ISAR images can be seen and the combined frequency response of the two ISAR images. The frequency response is made by taking the 2D inverse Fourier transform. After the 2D IFFT the axis on the frequency grid become k_x and k_y. The transformation to the frequency domain clearly shows which part of the data is missing and what should be interpolated to get the theoretical results. The method used here calculates the poles of the AR model and interpolates the data by using the Multi-band AR filter estimation that has been discussed in the previous section. Because time domain processing is a linear method, this averaged interpolated part can be added to the summation. One way to see it is that in the end three images are summed; the original two images and the image that is made by interpolation. The end result will of course be made by taking the Fourier transform of the interpolated frequency domain image, so no additional summation takes really place.



(a) Frequency domain response

(b) Interpolated ISAR image

Figure 6.11: Interpolated frequency response and ISAR image of the data seen in figure 6.10

In figure 6.11 the end result of the interpolation can be seen. On the left the interpolated frequency response is seen and on the right the final ISAR image. When a cut is made at the response of the scatterers the cross-range can be determined. According to the simulation the cross-range resolution is about $0.75[m]$. When calculating the cross-range resolution with equation 4.6 the cross-range should be $0.7[m]$, which is well within the resolution of the ISAR image. The resolution of the grid is $0.25[m]$. Also the sidelobes appear at the -13 dB levels as expected.

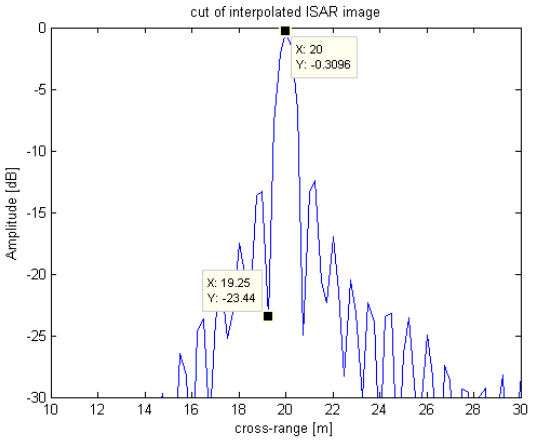


Figure 6.12: Cut of interpolated ISAR image

To aid the interpolation, the bistatic component can also be used as a fulcrum for the interpolation. When using the bistatic component a smaller gap has to be interpolated, which has a positive effect on the interpolation errors that may occur. In figure 6.13 the bistatic component of the two monostatic radars seen in figure 6.10 is added to the frequency response. As expected the frequency response is in between the two monostatic responses, because the virtual monostatic phase point of the bistatic component is in between the two radars.

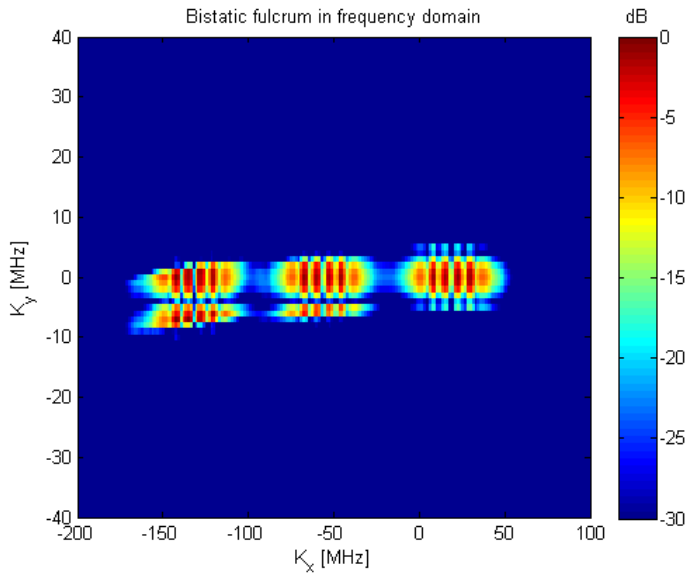
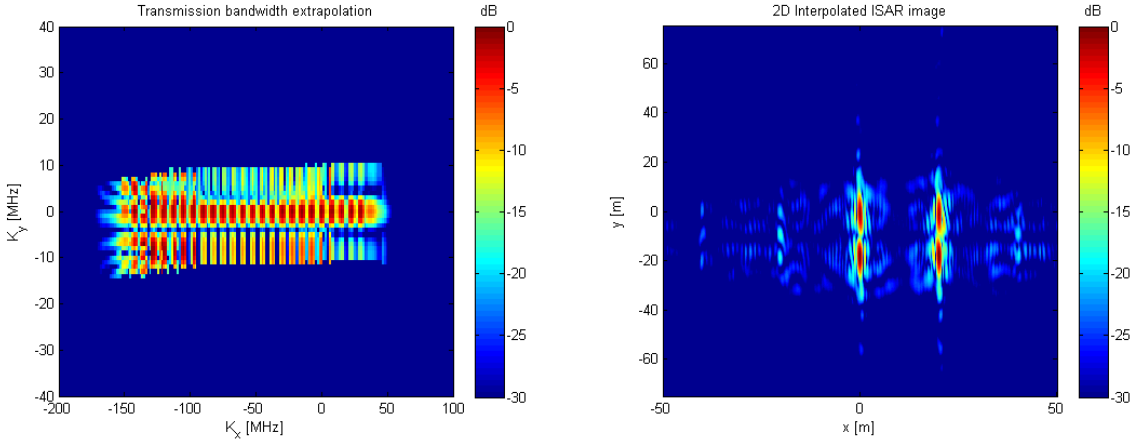


Figure 6.13: Bistatic fulcrum

The bandwidth interpolation used in this chapter was originally designed to combine different transmission frequencies to increase the down-range resolution. A possible improvement in the current simulation could be the combination of different transmission frequencies of the different radar sets and combining the transmission bands to increase the down-range resolution. To give an example of the increase in down-range resolution an extrapolation on the transmission bandwidth is made with the AR model described in this chapter. In figure 6.11b it it still difficult to separate the four scatterers that are present due to the fact that the down-range resolution is not sufficient. In other words the transmission bandwidth is too small. In figure 6.14b the same four scatterers are present in the picture, but with the transmission bandwidth extrapolation the scatterers can now be seen separately. More research into this possible improvement has to be done, and is outside the boundaries of this thesis.



(a) Frequency domain response

(b) Inter- and extrapolated ISAR image

Figure 6.14: Inter- and extrapolated frequency response of the data seen in figure 6.10

Parameters image interpolation

To determine where the section of data is located that has to be interpolated, the position of the ISAR responses in the frequency domain have to be determined. To get from the ISAR image to the frequency domain an IFFT and the Matlab routine *fftshift* are used. The shift is used to position the response around the center of the map. When an (I)FFT is performed on a range map, the domain to which the transform is made is the k_x and k_y domain.

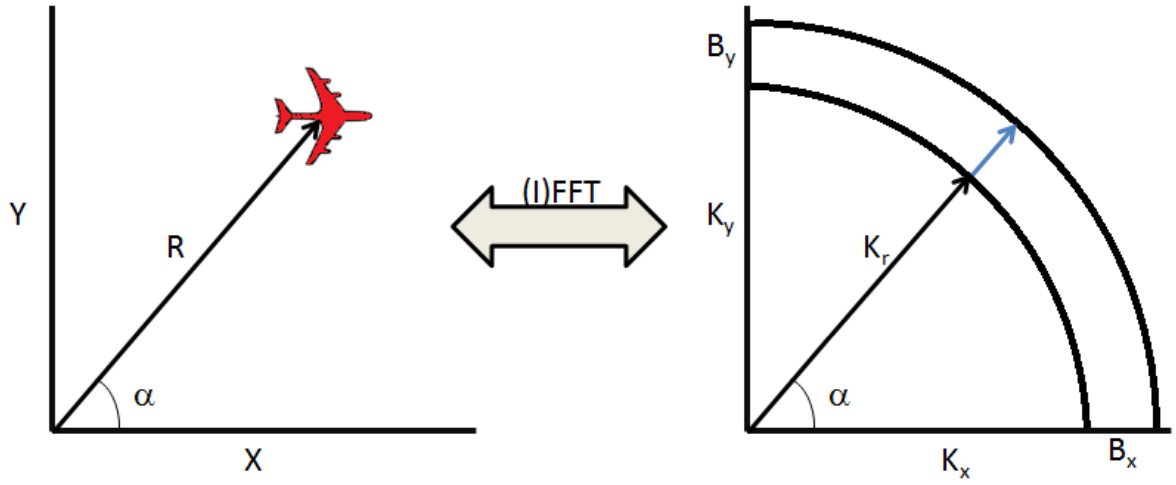


Figure 6.15: Domain transformation

In figure 6.15, k_x and k_y are the frequency components related to the transmit frequency and B_x and B_y the frequency components related to the transmission bandwidth. When an ISAR image is transformed, which is a range map, the axis in the IFFT domain are also k_x and k_y, so both frequency axis related to the transmit frequency and bandwidth. It is now possible to determine the limits and the resolution of both axis. The resolution in the frequency domain is dependent on the limits in the range domain and vice versa. The step size on the frequency axis are described in the following way:

$$\Delta f_{x/y} = \frac{c}{2 \cdot grid_{x/y}} \tag{6.12}$$

where $grid_{x/y}$ is the upper limit minus the lower limit for the x-axis and y-axis respectively. For most of the simulations made in this chapter $grid_x = 100[m]$ and $grid_y = 150[m]$. The limits in the frequency domain are calculated with the next equation:

$$limit_{upper} = \frac{c}{4 \cdot \Delta R_{d/c}} \qquad \& \qquad limit_{lower} = \frac{-c}{4 \cdot \Delta R_{d/c}} \tag{6.13}$$

where $\Delta R_{d/c}$ is the step size in the down-range and cross-range respectively. To clarify what is seen in the frequency domain images a situation sketch of a radar performing ISAR is shown in figure 6.16. The imaging is already translated to the frequency domain.

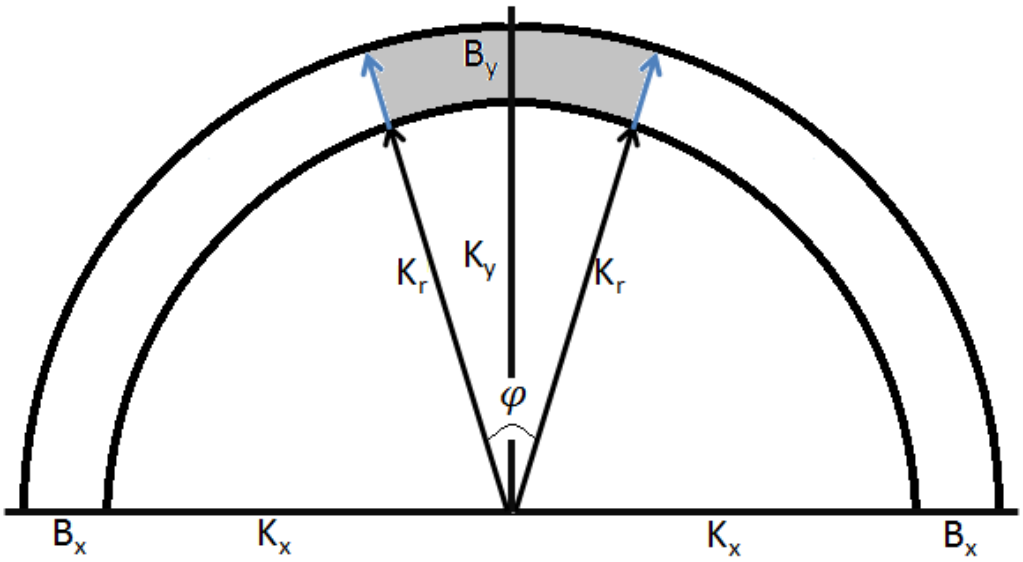


Figure 6.16: ISAR imaging in frequency domain

The light gray area that is being imaged by the radar can be compared to the response that is seen in figure 6.11a. Now that it is known what the axis represent, the position of the ISAR images in the frequency domain can be determined and the interpolation can be done. When an (I)FFT is used, the information that is outside the transformation domain, will be folded back into that domain. This is called aliasing. The position of the ISAR image after the IFFT is thus a modulo position of the target. In figure 6.17 an illustration of the effect of the folding that happens in the frequency domain is shown. The lines drawn there represent the upper part of a circle, which is folded back at the edge of the image.

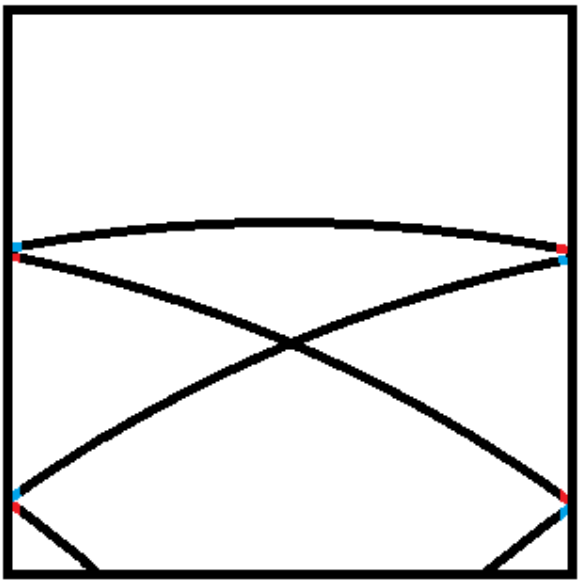


Figure 6.17: Folding in the frequency domain

To find the ISAR image in the frequency domain, the angle between the radar and the target is needed, as well as the transmit- frequency and bandwidth. The four equations for the corner points of the response are

$$\begin{aligned} k_{x1} &= \cos(\phi_1) \cdot (f_c + f_B) \quad k_{y1} = \sin(\phi_1) \cdot (f_c + f_B) \\ k_{x2} &= \cos(\phi_2) \cdot (f_c + f_B) \quad k_{y2} = \sin(\phi_2) \cdot (f_c + f_b) \\ \\ k_{x3} &= \cos(\phi_1) \cdot (f_c) \qquad k_{y3} = \sin(\phi_1) \cdot (f_c) \\ k_{x4} &= \cos(\phi_2) \cdot (f_c) \qquad k_{y3} = \sin(\phi_2) \cdot (f_c) \end{aligned} \tag{6.14}$$

Where the points are clockwise the corners of a rectangle starting with 1 (one) in the upper left corner. These four points can be fed into the interpolation routine and the interpolation can be automated. With these corner points it also became possible to calculate the maximum gap between the two images. The boundary condition on the interpolation is that at least 50 % overlaps in the horizontal plane, so a k_y corner point of one image is in between two k_y corner points of the other image.

$$\begin{aligned} \sin(\alpha_1) \cdot f_c = \sin(\alpha_2) \cdot (f_c + \frac{1}{2} f_B) \quad &\rightarrow \frac{\sin(\alpha_1)}{\sin(\alpha_2)} = \frac{f_c + \frac{1}{2} f_B}{f_c} \quad if \sin(\alpha_1) > \sin(\alpha_2) \\ \sin(\alpha_1) \cdot (f_c + \frac{1}{2} f_B) = \sin(\alpha_2) \cdot f_c \quad &\rightarrow \frac{\sin(\alpha_1)}{\sin(\alpha_2)} = \frac{f_c}{f_c + \frac{1}{2} f_B} \quad if \sin(\alpha_1) < \sin(\alpha_2) \end{aligned} \tag{6.15}$$

Equation 6.15 shows that the gap between the observation is bounded by the transmit bandwidth and the starting angle of the image. The first constraint, transmission bandwidth, due to the fact that a large angle is required, before the response has shifted enough in the k_y domain before it is outside the response of the other image. And the starting angle constraint because of the fact that the slope of the responses increases as the angle increases. This is clearly visible in figure 6.17.

To summarize what is done in this latest section a block diagram of the necessary steps made is shown in figure 6.18.

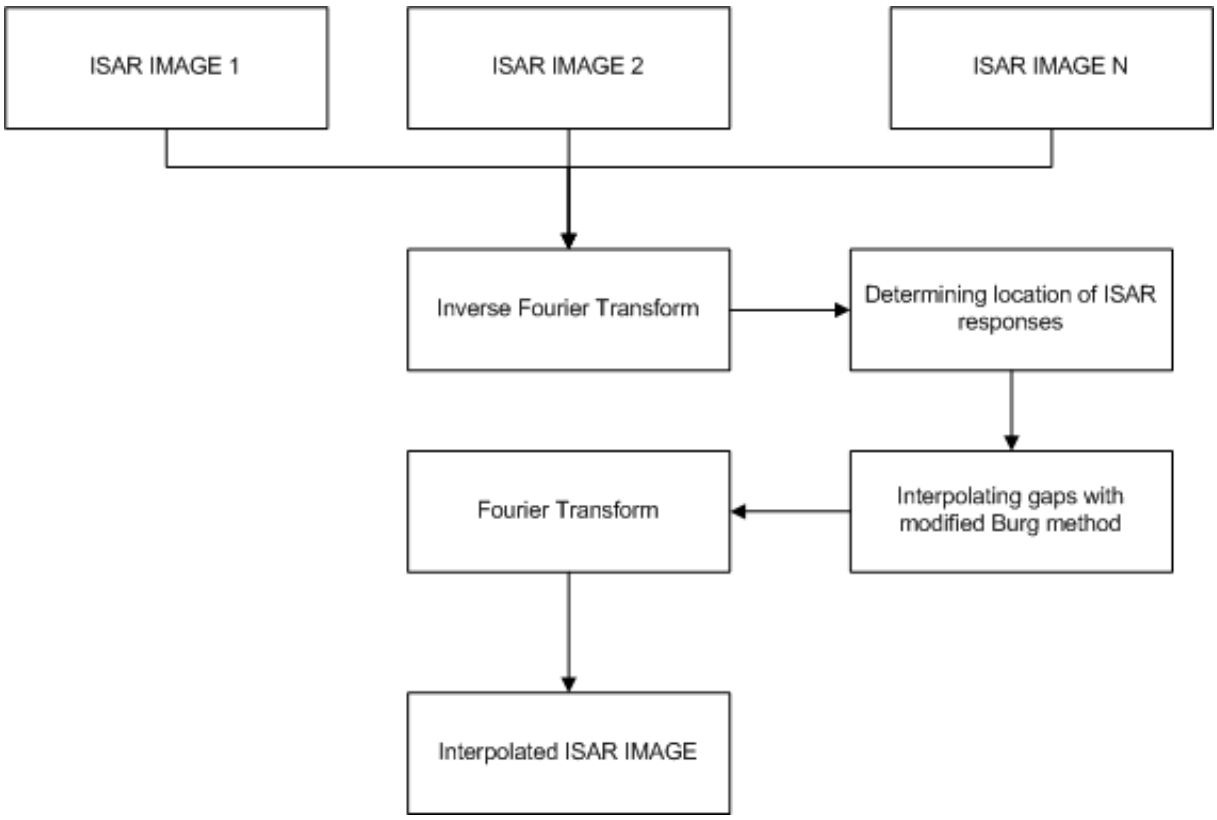


Figure 6.18: Block diagram for interpolating ISAR images

The shape of the ISAR response in the k_x and k_y domain is now known, and it looks like the interpolation can even be further improved by first performing a Stolt interpolation over the transformed data. This interpolation compensates for the range curvature. After the Stolt interpolation the data is lying on straight lines, instead of on the curvatures we now see. This additional improvement is not further investigated.

7. Conclusions

The most important reason to make use of a multistatic radar system is the possibility to overcome some weaknesses of monostatic and bistatic radar as well as complying to the stringent demands set by the radar end users. In chapter 2 & 3 the advantages and disadvantages on the range and Doppler resolution and target localization are shown. The most important aspects are:

+ cross-range decoupling
+ 3D target velocity vector decomposition
- geometry restrictions
- radar management

With multistatic radar the cross-range resolution can in some cases be decoupled from the range towards the target, as the down-range of one radar pair can be the cross-range of another radar pair. With multistatic radar a complete 3D decomposition of the target speed is possible, allowing a better way of tracking and a faster initialization of the track. Not only tracking can be improved, but also imaging techniques like ISAR require the target velocity vector to calculate the grid on which the target is being imaged. There are also some disadvantages with bi- and multistatic range and Doppler measurements. The geometry between the radars and the target is very important in whether a certain radar pair increases the resolution or if it deteriorates the measurement. The key is the bistatic angle φ_{bi} which determines the resolution of the bistatic responses. Keeping this is mind it becomes clear that radar management plays a vital role in multistatic radar configurations, because the usage of the radar can be optimized and radar set that deteriorates the end result can be excluded from the final image. In this thesis the focus was on the possible gains of multistatic radar, hereby doing no research into radar management.

The target can also be imaged to increase the resolution even further so that it becomes possible to identify the different scatters on a target. With Inverse Synthetic Aperture Radar (ISAR) the cross-range resolution of the radar can be greatly improved by integration of target responses over a certain amount of time. Two different approaches on ISAR are done in this thesis, a coherent and an incoherent approach. In chapter 5 an incoherent approach of multistatic ISAR in the form of 2D ISAR is developed and in chapter 4 the coherent ISAR approach is discussed. The advantages and disadvantages are:

+ High resolution imaging with lower transmission bandwidth
+ decreased integration with same resolution compared to monostatic ISAR
+ time domain ISAR processing allows imaging in arbitrary grid
- geometry restrictions

The main idea behind 2D ISAR is imaging targets with systems that do not make use of a large transmission bandwidth. The loss of resolution in down-range can be partially compensated by a second radar that sees the target under a different angle. Combining the two images incoherently, the amplitude at position of the scatters will increase more than the other parts of the image. This way a better estimation on target size and shape can be made. When the angle between two observation becomes too large, the target itself is no longer coherent. This can be due to the fact that different scatterers are imaged by different radars. The incoherent approach is likely the only approach that gives correct results in this case, possibly with an increased knowledge of the target, because the different radars see other scatterers.

Before it was possible to combine several different ISAR images, another method of calculating these images was necessary. To combine the images, all the images should use the same reference. Time domain techniques are used to calculate the ISAR images, because of a property that allowed that all the images could be made on a grid of your own choice. With time domain processing the received signals are interpolated over a grid of your own choice, making the summation of the different response very easy.

The strength of multistatic ISAR lays in the fact that it can achieve much higher resolutions with the same integration time than monostatic ISAR or achieves the same resolution as with monostatic radar, but with a much shorter integration time. The observed rotation is in the multistatic case larger than in the monostatic case. But there is a very stringent requirement on observed rotation; the rotations must overlap or form a continuous rotation. When there are gaps in the observation, the ISAR image can severely deteriorate. The demand that adjacent rotations should be observed also imposes a heavy requirement on the radar geometry. To loosen this requirement an interpolation on the available data is proposed. If there are gaps present in the observed rotation, an interpolation fills the gaps and the ISAR image will be the same as without a gap. The interpolated gap cannot be too large, as the same scatterers must be present in both observations. Also if a large gap has to be interpolated, the errors in the interpolation become larger. The gaps arise from a limited observation time in combination with a certain radar geometry. So not only the geometry of the radars can be less strict, also the integration time could be shortened. This is a large advantage for phased array radars, because the available time and energy is limited. When the integration time can be reduced, the radar can more often perform other tasks, like target localization and velocity vector estimation.

Recommendations

One of the main improvements that can be done in a possible extension to this thesis, is the simulation with real RCS models. A small research into 2D ISAR simulations with real RCS models is currently done at TNO The Hague, with some very promising results. The fact that it encompasses an incoherent summation of the data, the RCS fluctuations do not impact the end result a lot. As already said in chapter 6, the interpolation of the data can only take place if the same scatterers are present in both ISAR images. This means that the bistatic angle between the different transmitters and receivers cannot be too large. The fact that the bistatic angle should be small, the chance that the two measurements are not correlated anymore is small and even with RCS fluctuations the interpolation can be done.

A second recommendation is a research into radar management, which plays a vital role in optimizing the usage of multistatic radar network. Not all transmit and receive pairs can contribute to an improvement of the resolution or target localization, they can even make the measurement even worse. Important points with radar management are the bistatic angle, the range and the returned power from the target.

Furthermore the down-range interpolation technique that is used for ISAR cross-range interpolation, may also be used in its original form for multistatic configurations as well. One of the possible ways to create orthogonal transmission sets is a frequency division multiplex technique, where the different transmitters use a different transmit frequency. Combining the different transmission frequencies and interpolate the possible gaps between them may be used to increase the down-range resolution, for which the interpolation technique was originally developed.

8. Matlab code

```
%-------------------------------------------------------------------------%
%                 Range and Doppler resolution functions                  %
%-------------------------------------------------------------------------%
%
% Author:         JMM Verzeilberg
% Date:           16-03-2011
% Matlab version: 2010B
%
% Implementation of a mono-, bi- and multistatic range and Doppler
% ambiguity functions. At most one transmitter and three receivers can be
% used. If identical receiver positions are used, the program wil set the
% received signal to zero for one of the receivers.

clear all; close all; clc

c       = 3e8;                  % Speed of light [m/s]
TX      = [0 0];                % TX position [m]
RX1     = [0 0];                % First RX Position
RX2     = [-500 500];           % Second RX position [m]
RX3     = [500 500];            % Third RX position [m]

PRF     = 2500;                 % Pulse repetition frequency [Hz]
Int     = 2;                    % Integration time [s]
NP      = 30;                   % Number of pulses used in doppler processing
FC      = 2e9;                  % Center frequency [Hz]
B       = 50e6;                 % Bandwidth [Hz]
Tstep   = 1/(2*B);              % Sampling time [s]
t       = -2.5e-6:Tstep:2.5e-6;% Pulse length [s]
alpha   = (pi*B)/(2*t(end));    % parameter for chirp [-]

V       = [20 50];              % Target speed [m/s]
Target  = [0 500];              % Target position [m] @t=0
scaling= 1;                     % Step size range ambiguity
i_max   = 600;                  % Parameter for area size
j_max   = 600;                  % Parameter for area size

%-------------------------------------------------------------------------%
%                          Range ambiguity functions                      %
%-------------------------------------------------------------------------%

Rt=norm(Target-TX);                             % Distance TX - target
Rr1=norm(Target-RX1); t01=(Rt+Rr1)/c;           % Distance target - RX1
Rr2=norm(Target-RX2); t02=(Rt+Rr2)/c;           % Distance target - RX2
Rr3=norm(Target-RX3); t03=(Rt+Rr3)/c;           % Distance target - RX3

sRX1=exp(2*1i*pi*FC*t01+1i*alpha.*(t-t01).^2);        % Received signal RX1
if norm(RX1-RX2)==0
    sRX2=zeros(1,length(sRX1));
    sRX3=sRX2;
else
    sRX2=exp(2*1i*pi*FC*t02+1i*alpha.*(t-t02).^2);    % Received signal RX2
```

```
    if norm(RX2-RX3)==0
        sRX3=zeros(1,length(sRX1));
    else
        sRX3=exp(2*1i*pi*FC*t03+1i*alpha.*(t-t03).^2);% Received signal RX3
    end
end

xstart      = -(i_max)/(2/scaling);
ystart      = -(j_max)/(2/scaling);
correlation = zeros(i_max+1,j_max+1);

    for i=0:i_max
        for j=0:j_max
            Targeth=[xstart+i*scaling ystart+j*scaling];
            Rth=norm(Targeth-TX);
            Rrh1=norm(Targeth-RX1);
            Rrh2=norm(Targeth-RX2);
            Rrh3=norm(Targeth-RX3);
            t01h=(Rth+Rrh1)/c;
            t02h=(Rth+Rrh2)/c;
            t03h=(Rth+Rrh3)/c;
            sRh1=exp(-1i*alpha.*(t'-t01h).^2);
            sRh2=exp(-1i*alpha.*(t'-t02h).^2);
            sRh3=exp(-1i*alpha.*(t'-t03h).^2);
            correlation(j+1,i+1)=sRX1*(sRh1)+sRX2*(sRh2)+sRX3*(sRh3);
        end
    end

xas=-i_max/2:1/scaling:i_max/2;
yas=-j_max/2:1:j_max/2;
figure(1);
imagesc(xas,yas,20*log10(abs(correlation))...
    -max(max(20*log10(abs(correlation)))))
set(gca,'YDir','normal')
xlabel('x [m]')
ylabel('y [m]')
h=colorbar;
title(h,'dB')

%------------------------------------------------------------------------%
%                         Doppler ambiguity functions                    %
%------------------------------------------------------------------------%
%
% The radial speed of the target towards the first receiver is calculated
% using the FFT across NP pulses. Maximum unambigous speed is
%   PRF/4*(c/FC) =  93.75 [m/s]

sRX1=zeros(NP,length(t));
for i=0:NP-1
    Targeth=[Target(1)+(V(1)/PRF)*i Target(2)+(V(2)/PRF)*i];
    Rt=norm(Targeth-TX);
    Rr=norm(Targeth-RX1);
    t0=(Rt+Rr)/c;
    sRX1(i+1,:)=exp(2*1i*pi*FC*t0+1i*alpha.*(t-t0).^2);
end

Speed=fftshift(fft(sRX1,256));
as=linspace(-PRF/4*(c/FC),PRF/4*(c/FC),256);
figure(2); plot(as,mean(abs(Speed),2))

% The speed of the target is calculated by cross referencing it with
% possible speeds, i.e. an hypothesis. This way a Vx Vy image can be made
% and added up incoherently to determine the speed of the target
```

```
k_max          = 200;                  % Number of Doppler velocities in V_x
l_max          = 200;                  % Number of Doppler velocities in V_y
scalingD       = 2;                    % Step size for Doppler velocities
sRXh1          = zeros(NP,length(t)); % Vector for Hypothesis speed
sRXh2          = zeros(NP,length(t)); % Vector for Hypothesis speed
sRXh3          = zeros(NP,length(t)); % Vector for Hypothesis speed
sRX2           = zeros(NP,length(t));
sRX3           = zeros(NP,length(t));
doppler_ambi1  = zeros(l_max+1,k_max+1);
doppler_ambi2  = zeros(l_max+1,k_max+1);
doppler_ambi3  = zeros(l_max+1,k_max+1);


if norm(RX1-RX2)==0

else

        for i=0:NP-1
            Targeth=[Target(1)+(V(1)/PRF)*i Target(2)+(V(2)/PRF)*i];
            Rt=norm(Targeth-TX);
            Rr=norm(Targeth-RX2);
            t0=(Rt+Rr)/c;
            sRX2(i+1,:)=exp(2*1i*pi*FC*t0+1i*alpha.*(t-t0).^2);
        end

    if norm(RX2-RX3)==0
        sRX3=zeros(1,length(sRX1));
    else

        for i=0:NP-1
            Targeth=[Target(1)+(V(1)/PRF)*i Target(2)+(V(2)/PRF)*i];
            Rt=norm(Targeth-TX);
            Rr=norm(Targeth-RX3);
            t0=(Rt+Rr)/c;
            sRX3(i+1,:)=exp(2*1i*pi*FC*t0+1i*alpha.*(t-t0).^2);
        end

    end

end



for k=0:k_max
    for l=0:l_max

        for m=0:NP-1
            Targeth=[Target(1)+(k/scalingD/PRF)*m ...
                Target(2)+(l/scalingD/PRF)*m];
            Rt=norm(Targeth-TX);
            Rr=norm(Targeth-RX1);
            t0=(Rt+Rr)/c;
            sRXh1(m+1,:)=exp(2*1i*pi*FC*t0+1i*alpha.*(t-t0).^2);
            if m==(NP-1)
                doppler_ambi1(l+1,k+1)=sum(sum(sRX1.*conj(sRXh1)));
            end
        end

        if norm(RX1-RX2)~=0
            for m=0:NP-1
                Targeth=[Target(1)+(k/scalingD/PRF)*m ...
                Target(2)+(l/scalingD/PRF)*m];
```

```
                Rt=norm(Targeth-TX);
                Rr=norm(Targeth-RX2);
                t0=(Rt+Rr)/c;
                sRXh2(m+1,:)=exp(2*1i*pi*FC*t0+1i*alpha.*(t-t0).^2);
                if m==(NP-1)
                    doppler_ambi2(l+1,k+1)=sum(sum(sRX2.*conj(sRXh2)));
                end
            end
        end
        if norm(RX2-RX3)~=0
            for m=0:NP-1
                Targeth=[Target(1)+(k/scalingD/PRF)*m ...
                Target(2)+(l/scalingD/PRF)*m];
                Rt=norm(Targeth-TX);
                Rr=norm(Targeth-RX3);
                t0=(Rt+Rr)/c;
                sRXh3(m+1,:)=exp(2*1i*pi*FC*t0+1i*alpha.*(t-t0).^2);
                if m==(NP-1)
                    doppler_ambi3(l+1,k+1)=sum(sum(sRX3.*conj(sRXh3)));
                end
            end
        end
    end
end
coherent=doppler_ambi1+doppler_ambi2+doppler_ambi3;
incoherent=abs(doppler_ambi1)+abs(doppler_ambi2)+abs(doppler_ambi3);
xas=0:1/scalingD:k_max/scalingD;
yas=0:1/scalingD:l_max/scalingD;
figure, imagesc(xas,yas,20*log10(abs(coherent))...
    -max(max(20*log10(abs(coherent)))))
ylabel('V_y [m/s]')
xlabel('V_x [m/s]')
figure, imagesc(xas,yas,20*log10(abs(incoherent))...
    -max(max(20*log10(abs(incoherent)))))
ylabel('V_y [m/s]')
xlabel('V_x [m/s]')
```

```
%-------------------------------------------------------------------------%
%                  Frequency domain ISAR Processing                       %
%-------------------------------------------------------------------------%
%
% Author:          JMM Verzeilberg
% Date:            16-03-2011
% Matlab version: 2010B
%
% Implementation to calculate the ISAR response of a single scatterer via
% the frequency domain approach. The maximum distance of the target is
% related to the length of the transmitted signal, defined in de second
% part of the program.

clear all; clc

c       = 3e8;                  % Speed of light [m/s]
TX      = [0 0];                % TX position [m]
RX      = [0 0];                % RX position [m]

PRF     = 1250;                 % Pulse repetition frequency [Hz]
Int     = 1;                    % Integration time [s]
FC      = 2e9;                  % Center frequency [Hz]
B       = 4e6;                  % Bandwidth [Hz]
Tstep   = 1/(12*B);             % Sampling time [s]
t       = -2.5e-6:Tstep:2.5e-6; % Pulse length [s]
alpha   = (pi*B)/(2*t(end));    % parameter for chirp [-]

V       = [60 0];               % Target speed [m/s]
Target  = [0 1990];             % Target position [m] @t=0
Rdelta  = Tstep*(c/2);          % Size of a cell in the range response

%-------------------------------------------------------------------------%
%              Allocating memory for matrices and vectors                 %
%                     and defining some parameters                        %
%-------------------------------------------------------------------------%

s0=[exp(1i*(alpha*t.^2-pi))...
    zeros(1,5*length(t))];                  % Transmitted signal
Range=zeros(1,PRF*Int);                     % Vector for Range info
dRc=zeros(PRF*Int,length(s0));              % Matrix for received signals

%-------------------------------------------------------------------------%
%                       Frequency Domain Calculations                     %
%-------------------------------------------------------------------------%

for p=1:(PRF*Int)

Targetp     = [Target(1)+V(1)/PRF*(p-1)...
               Target(2)+V(2)/PRF*(p-1)];       % Target position @ pulse p
Range(p)    = norm(Targetp-TX)+norm(Targetp-RX);% Range
dt          = Range(p)/c;                       % Fly around time
sR1         = [exp(-2*1i*pi*FC*dt+1i*(alpha*(t).^2-pi))...
               zeros(1,5*length(t))];           % Received signal
sR1         = circshift(sR1,[0 ...
               floor(Range(p)/(1500/length(t)))]);% Shifted received signal
dRc(p,:)    = ifft(fft(sR1).*conj(fft(s0)));

end

cDop= exp(-1i*2*pi*Range/(c/FC)).'*ones(1,length(s0));% Azimuth filter
Xf  = (ifft(fft(dRc).*conj(fft(cDop))));            % Azimuth compression

%-------------------------------------------------------------------------%
```

```
%                        Plotting the end result                          %
%-------------------------------------------------------------------------%

yas    = linspace(-V(1)/(2*Int),V(1)/(2*Int),PRF*Int);
xas    = Rdelta*(0:(length(s0)-1));
figure
imagesc(xas,yas,fftshift(20*log10(abs(Xf)),1)-max(max(20*log10(abs(Xf)))));
caxis([-30 0])
ylabel('cross-range [m]')
xlabel('down-range [m]')
```

```
%------------------------------------------------------------------------%
%                     Time Domain ISAR Processing                        %
%------------------------------------------------------------------------%
%
% Author:          JMM Verzeilberg
% Date:            27-04-2011
% Matlab version: 2010B SP1
%
% Generating responses of the ISAR image on a predefined grid centered
% around the first scatterer in the input file. The position of the
% transmitters and receivers is read from a database used in a TNO
% project.


clear all; clc
load SPREWS_BASES.mat
a           = 0;
c           = 3e8;                  % Speed of light [m/s]
TX          = Base_TX(1:1,:);       % TX position [m]
RX          = Base_RX(1:1,:);       % RX position [m]

PRF         = 2000;                 % Pulse repetition frequency [Hz]
Int         = 1;                    % Integration time [s]

FC          = 3e9;                  % Center frequency [Hz]
B           = 10e6;                 % Bandwidth [Hz]
Upsamp      = 2;                    % Oversampling factor
Tstep       = 1/(2*B*Upsamp);       % Sampling time [s]
t           = -2.5e-6:Tstep:2.5e-6; % Pulse length [s]
alpha       = B/(2*t(end));         % parameter for chirp [-]

V           = [100 0];              % Target speed [m/s]
scatterer.x = [-10 10 -10 10];      % Target x-position [m] @t=0
scatterer.y = [10 10 -10 -10]+8000; % Target y-position [m] @t=0
Rdelta      = Tstep*(c);            % Cell size in the range response
RF          = 11;                   % Range factor: determines max. range
GridRes     = 0.25;                 % Grid resolution [m]
Gridsx      = 100;                  % Grid size in x-direction [m]
Gridsy      = 150;                  % Grid size in y-direction [m]

%------------------------------------------------------------------------%
%              Allocating memory for matrices and vectors                %
%                     and defining some parameters                       %
%------------------------------------------------------------------------%

s0      = conj(fft([exp(-1i*pi*(alpha*t.^2)) zeros(1,RF*length(t))]));
Dbins   = PRF*Int;
Rbins   = length(s0);
RC      = zeros(1,Rbins);
Time    = ((1:(RF+1)*length(t))-1) * Tstep - 2.5e-06;
TimeSub = [(Time(1)-Tstep) Time];
AmpPulse= [0 ones(1,length(t)) zeros(1,length(t)*RF)];
RCGrid  = zeros(Gridsy*GridRes+1,Gridsy*GridRes+1,(size(TX,1)*size(RX,1)));
xas     = Rdelta*(0:(length(s0)-1));
display(['Maximum (two-way) target range = ' num2str((Rs+1)*1.5) ' km'])



%------------------------------------------------------------------------%
%                         Time Domain Calculations                       %
%------------------------------------------------------------------------%

for r=1:size(TX,1)
```

```
        TXc=TX(r,:);
for s=1:size(RX,1)
    RXc=RX(s,:);
    for p=1:Dbins
        for q=1:length(scatterer.x);

        scattererp  = [scatterer.x(q)+V(1)/PRF*(p-1)...
                   scatterer.y(q)+V(2)/PRF*(p-1)];    % scatterer position

               if q==1
  [gridX gridY] = meshgrid((scattererp(1)-50):0.25:(scattererp(1)+50),...
                  (scattererp(2)-75):0.25:(scattererp(2)+75));
               end
        Range         = norm(scattererp-TXc)+norm(scattererp-RXc); % Range
        tao           = Range/c;                              % Fly around Time
        sR1           = exp(-2*1i*pi*(FC*(-tao)+0.5*alpha*(Time-tao).^2));
        AmpR          = interp1(TimeSub,AmpPulse,Time-tao,'linear',0);
        RC            = RC+ifft(fft((sR1 .* AmpR)).*s0);
        end

        RangeOnGrid   = sqrt((TXc(1)-gridX).^2+(TXc(2)-gridY).^2)+...
                        sqrt((RXc(1)-gridX).^2+(RXc(2)-gridY).^2);

        phase         = exp(-2*pi*1i*RangeOnGrid/(c/FC));
        RCGrid(:,:,a)= RCGrid(:,:,a)+(interp1(xas,RC,RangeOnGrid).*phase);
        RC            = zeros(1,Rbins);
    end
end
end

%-------------------------------------------------------------------------%
%                        Plotting the end result                          %
%-------------------------------------------------------------------------%

as1=-(Gridsx/2):0.25:(Gridsx/2);                    % Defining the x-axis
as2=-(Gridsy/2):0.25:(Gridsy/2);                    % Defining the y-axis
figure, imagesc(as1,as2,20*log10(abs(RCGrid(:,:,1)))...
                -max(max(20*log10(abs(RCGrid(:,:,1))))))
set(gca,'YDir','normal')
```

```
%-------------------------------------------------------------------------%
%                    ISAR interpolation routine                           %
%-------------------------------------------------------------------------%
%
% Author:          Ph. van Dorp, JMM Verzeilberg
% Date:            20-07-2011
% Matlab version: 2010B
%
% Routine to interpolate the data between two separate ISAR images. The
% interpolation takes place in the frequency domain, where the operation
% takes place faster than in the time domain. The starting points of the
% interpolation are found by the input of the transmit frequency,
% transmission bandwidth and the observation angle.


load('c:\users\philip\matlab\multi_static_isar\presentation.mat');

%-------------------------------------------------------------------------%
%                Plotting information to get a clearer view               %
%-------------------------------------------------------------------------%

ZLevel = [70 100];                  % Color axis scaling parameters

figure;                             % Plotting the first ISAR image
subplot(1,2,1)
imagesc(20*log10(abs(RC_Radar1)));
colorbar; caxis(ZLevel);
subplot(1,2,2)
imagesc(angle(RC_Radar1));
colorbar; caxis(ZLevel);

figure;                             % Plotting the second ISAR image
subplot(1,2,1)
imagesc(20*log10(abs(RC_Radar2)));
colorbar; caxis(ZLevel);
subplot(1,2,2)
imagesc(angle(RC_Radar2));
colorbar; caxis(ZLevel);

figure;                             % Plotting the combined ISAR image
imagesc(20*log10(abs(RC_Radar1 + RC_Radar2)));
colorbar; caxis(ZLevel);

RCGrid1IFFT = fftshift(ifft2(RC_Radar1));   % Transfomring ISAR images to
RCGrid2IFFT = fftshift(ifft2(RC_Radar2));   % frequency domain

figure;                             % Plotting  freq. domain ISAR images
subplot(1,2,1)
imagesc(abs(RCGrid1IFFT));
colorbar;
subplot(1,2,2)
imagesc(abs(RCGrid2IFFT));
colorbar;
pause
%-------------------------------------------------------------------------%
%                           ISAR interpolation                            %
%-------------------------------------------------------------------------%

[N,M] = size(RC_Radar1);
RCGrid3IFFT = zeros(size(RCGrid1IFFT));

% Positions of the ISAR images for the interpolation
RC1_x1= (mod(cos(phi1)*(FC+B/2),6e8))/1.5e6+1;
```

```
RC1_x2= (mod(cos(phi2)*(FC+B/2),6e8))/1.5e6-1;
RC2_x1= (mod(cos(phi3)*(FC+B/2),6e8))/1.5e6+1;
RC2_x2= (mod(cos(phi4)*(FC+B/2),6e8))/1.5e6-1;


for k = 1:N

   fprintf('%5d of %5d\n',k,N);

   Temp1IFFT = RCGrid1IFFT(k,:);
   Temp2IFFT = RCGrid2IFFT(k,:);

   Index1 = RC1_x1:RC1_x2;
   Index2 = RC2_x1:RC2_x2;
   Index3 = (Index1(end)+1):(Index2(1)-1);

   IndexLow = Index1 - Index1(1) + 1;
   IndexHigh = Index2 - Index1(1) + 1;
   IndexInterp = (IndexLow(end) + 1):(IndexHigh(1) - 1);

   [z1,z2] = bandwidth_interpolation_armasa_interp(IndexLow,...
             Temp2IFFT(Index1),IndexHigh,Temp1IFFT(Index2),5);

   Temp3IFFT = Temp1IFFT*1 + Temp2IFFT*1;
   Temp3IFFT(Index3) = z2(IndexInterp);
   RCGrid3IFFT(k,:) = Temp3IFFT;
end

NewRCGrid = fft2(fftshift(RCGrid3IFFT));

figure;
imagesc(abs(RCGrid3IFFT));
colorbar

figure;
imagesc(abs(20*log10(abs(NewRCGrid))));
caxis([100 130]);
colorbar

%-------------------------------------------------------------------------%
%                  Test: Transmission bandwidth extrapolation             %
%-------------------------------------------------------------------------%

RCGrid4IFFT = RCGrid3IFFT;
RCGridDetection = abs(RCGrid4IFFT) > 100;

for k = 1:M
   Temp = RCGrid4IFFT(:,k);
   Index = find(abs(Temp) >= 100);

   if ~isempty(Index)
      IndexStart = min(Index);
      IndexStop = max(Index);

      IndexMiddle = round((IndexStart + IndexStop)/2);

      Index = IndexMiddle + (-3:3);
      z = bandwidth_extrapolation(Temp(Index),3);

      length(z)
      Index = IndexMiddle + (-(7*1+3):(3+7*1));
      RCGrid4IFFT(Index,k) = z;
   end
```

```
end

NewRCGrid4 = fft2(fftshift(RCGrid4IFFT));

figure;
subplot(1,2,1)
imagesc(abs(RCGrid4IFFT));
colorbar
subplot(1,2,2)
imagesc(20*log10(abs(NewRCGrid4))-max(max(20*log10(abs(NewRCGrid4)))));
colorbar; caxis([-30 0]);


save('c:\users\philip\matlab\multi_static_isar\presentation_interp.mat',...
    'RCGrid4IFFT','NewRCGrid4','RCGrid3IFFT','NewRCGrid');
```

Bibliography

[1] W. van Rossum, *MIMO Imaging*. TNO, The Hague, 2010.

[2] P. Tait, *Introduction to Radar Target Recognition*. IEE Radar series 18, 2005.

[3] S. Kingsley & S. Quegan, *Understanding Radar Systems*. SciTech Publishing, Rayleigh (NC), 1999

[4] V.S. Chernyak, *Fundamentals of Multisite Radar Systems*. Gordon and Breach Science Publishers, Amsterdam, 1998

[5] W.J. Vlothuizen & M. Ditzel, *Real-time Brute Force SAR Processing*. TNO, The Hague, 2009

[6] D.P. Pastina & M. Bucciarelli & P. Lambardo *Multistatic and MIMO Distributed ISAR for Enhanced Cross-Range Resolution of Rotation Targets*. IEEE transactions on Geoscience and remote sensing Vol. 48 No. 8, Rome August 2010

[7] Ph. van Dorp & R. Ebeling & A.G. Huizing *High Resolution Radar Imaging using Coherent MultiBand Processing Techniques*. TNO, The Hague, 2010

[8] M.J.L. de Hoon & T.H.J.J. van der Hagen & H. Schoonewelle $ H. van Dam *Why Yule-Walker should not be used for Autoregressive modelling*. Interfaculty Reactor Institute, TU Delft

[9] M.H. Hayes, *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, Hoboken (NJ), 1996

[10] W.G. Carrara & R.S. Goodman & R.M. Majewski *Spotlight Synthetic Aperture Radar*. Artech House, London, 1995