SAR Algorithms

Recap: What is SAR processing?

- **SAR processing algorithms model the scene as a set of discrete point targets that do not interact with each other (aka Born approximation)**
    - **No multibounce**
    - **The electric field at the target comes only from the incident wave and not from surrounding scatterers**
    - **The target model is linear because the scattered response from point target P1 and point target P2 is modelled as the response from point target P1 *by itself* + response from point target P2 *by itself***
    - **We can apply the principle of superposition!!!**
- SAR processing is the application of a matched filter for each pixel in the image where the matched filter coefficients are the response from a single isolated point target
    - We will assume noise is whitened (decorrelated)
- Equivalently, we can say:
    - SAR processing is a correlation filter between a single isolated point target response and the raw data
    - SAR processing is an inner product between our model of a single isolated point target and the raw data

Recap: What is SAR processing?

- SAR processing algorithms model the scene as a set of discrete point targets that do not interact with each other (aka Born approximation)
  - No multibounce
  - The target's electric field is only from the incident wave and not from surrounding scatterers
  - The target model is linear because the scattered response from point target P1 and point target P2 is modelled as the response from point target P1 by itself + response from point target P2 by itself
  - We can apply the principle of superposition!!!
- **SAR processing is the application of a matched filter for each pixel in the image where the matched filter coefficients are the single isolated point target response**
  - **We will assume noise is whitened (decorrelated)**
- Equivalently, we can say:
  - SAR processing is a correlation filter between a single isolated point target response and the raw data
  - SAR processing is an inner product between our model of a single isolated point target and the raw data

Recap: What is SAR processing?

- SAR processing algorithms model the scene as a set of discrete point targets that do not interact with each other (aka Born approximation)
  - No multibounce
  - The target's electric field is only from the incident wave and not from surrounding scatterers
  - The target model is linear because the scattered response from point target P1 and point target P2 is modelled as the response from point target P1 ***by itself*** + response from point target P2 ***by itself***
  - We can apply the principle of superposition!!!
- SAR processing is the application of a matched filter for each pixel in the image where the matched filter coefficients are the single isolated point target response
  - We will assume noise is whitened (decorrelated)
- **Equivalently, we can say:**
  - **SAR processing is a correlation filter between a single isolated point target response and the raw data**
  - **SAR processing is an inner product between our model of a single isolated point target and the raw data**

Recap: What is SAR processing?

- So… SAR processing is a **matched filter** and the **filter is linear**
- If the filter was also space invariant we could apply it in the frequency domain
- But: the filter is not space invariant. The point target's shape changes depending on the range to the radar.

Why do we care that it is not space invariant?

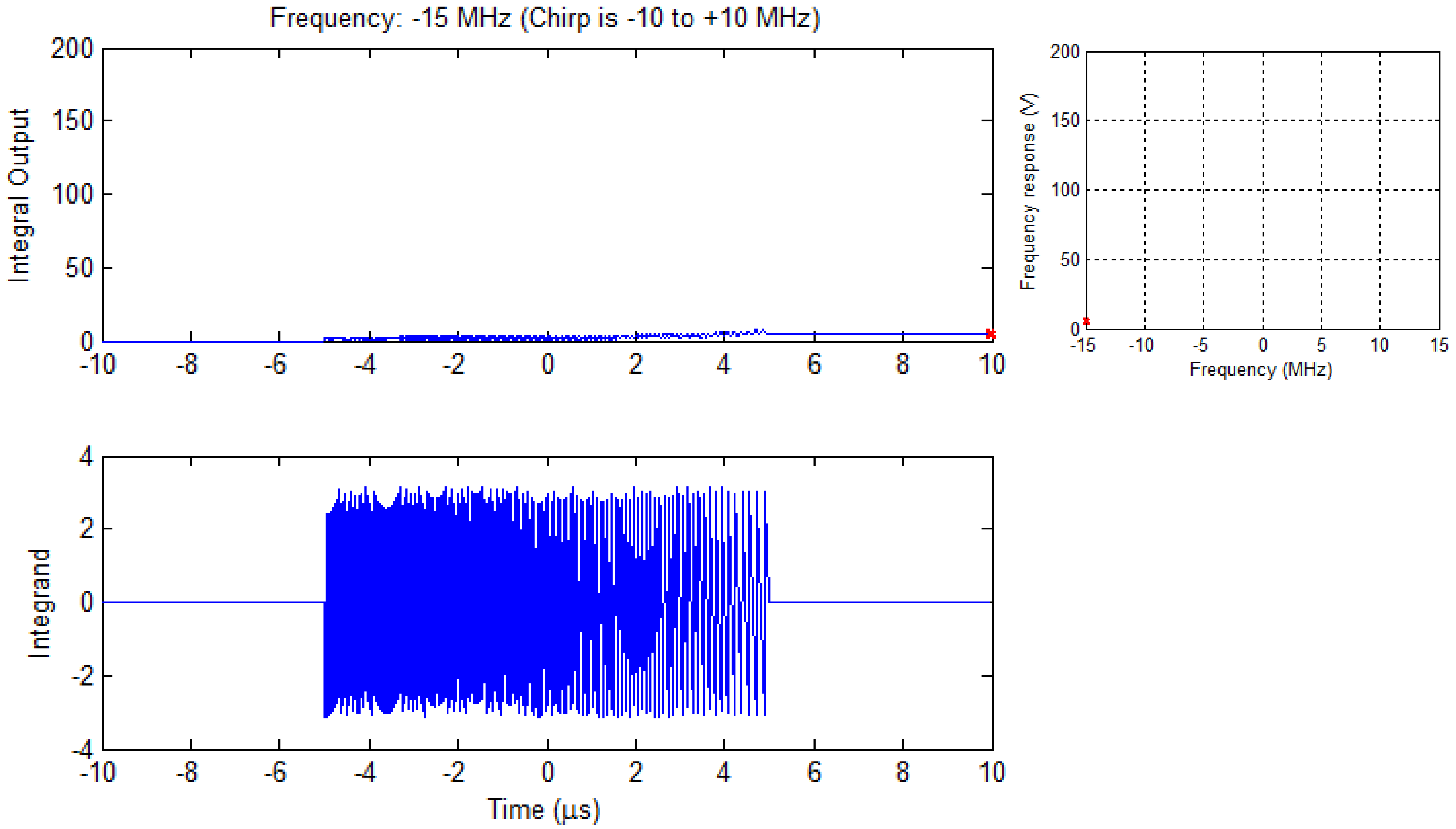
- Recall linear time invariant (LTIV) systems have complex exponentials as their Eigenfunctions. A change of basis of the input and output to complex exponentials means that a simple component-wise multiply is all that is needed to apply the filter. A change of basis to complex exponentials can be efficiently implemented using a Fast Fourier Transform (FFT) assuming data are uniformly sampled.
- Without Fourier method, $O(N^2M^2)$ operations are required instead of $O(N*\log_2(N) M*\log_2(M))$ where N and M are the dimensions of the image and are usually on the order of thousands of pixels each. The direct application of “slow” convolution could be more than 100x slower than “fast” or Fourier based convolution.
- Good news: we can exploit the structure of the signal to transform (usually through interpolation) the data into a domain where the signal is space invariant! To do this, we require properly sampled raw data and image pixels.

Principle of Stationary Phase (PSOP)

- PSOP is used to approximately solve integrals of the form

$$I = \int_A^B F(x) e^{-j\phi(t)}\, dx$$

  where the phase function, $\phi(t)$, is rapidly varying over the range of integration except for a few points where the derivative is zero (aka stationary points) AND $F(x)$ is a slowly varying function by comparison.
- With A and B equal to -∞ and ∞, the integration looks a lot like a **1-D Fourier integral**
- SAR chirp signals are similar to quadratics. Quadratic functions vary quickly everywhere and have a **single stationary point.**
- **The envelope of a SAR signal varies slowly with time.**


Frequency: -15 MHz (Chirp is -10 to +10 MHz)
Integral Output
Integrand
Time (µs)
Frequency response (V)
Frequency (MHz)

The stationary phase method is a procedure for evaluation of integrals of the form

$$I = \int_{-\infty}^{\infty} F(x) e^{-j\phi(x)} dx \tag{1}$$

where $\phi(x)$ is a rapidly-varying function of x over most of the range of integration, and $F(x)$ is slowly-varying (by comparison). Such integrals frequently arise in radiation or scattering problems. Rapid oscillations of the exponential term mean that I is approximately zero over those regions of the integrand; the only significant non-zero contributions to the integral occur in regions of the integration range where $d\phi/dx = 0$, *i.e.* at points of *stationary phase*. Points of stationary phase are labeled x_s and defined by

$$\phi'(x_s) = 0 \tag{2}$$

Note that $F(x) \approx F(x_s)$ in the vicinity of the statinary phase points, since $F(x)$ is assumed to be slowly varying, and hence this term can be pulled outside the integral. Expanding $\phi(x)$ in a Taylor series near the point x_s and keeping only the first two non-zero terms gives

$$\phi(x) \approx \phi(x_s) + \frac{1}{2}\phi''(x_s)(x - x_s)^2 \tag{3}$$

Substituting this into the integral (1) gives

Complex Gaussian has a closed form solution!

$$\begin{aligned} I &\approx F(x_s) e^{-j\phi(x_s)} \int_{-\infty}^{\infty} e^{-j\phi''(x_s)(x-x_s)^2/2} dx \\ &\approx \sqrt{\frac{2\pi}{j\phi''(x_s)}} F(x_s) e^{-j\phi(x_s)} \end{aligned} \tag{4}$$

Remember:

$$\phi(t)$$

must include your original phase function being integrated AND the Fourier term:

$$-2\pi f t$$

1. Write out envelope and phase function
2. Determine derivative of phase function.
3. Solve for the stationary point, t_s, in terms of f. This is the first messy part…

$$t_s(f) = \cdots$$

4. Determine second derivative of phase function. IGNORED IN OUR DERIVATIONS!
5. Plug t(f) into (4) wherever the stationary point occurs.
6. Simplify! This is the second messy part…

Process is the same for inverse Fourier transform except replace eqns above with:

$$\phi(f) \qquad 2\pi f t \qquad f_s(t) = \cdots$$

Good online SAR Resource

- https://saredu.dlr.de/unit

Satellite and Low Squint Airborne SAR Algorithms

- Lower squint (often <4-5 deg)
- Narrow azimuth bandwidth (usually 0.5 deg to 10 deg azimuth beamwidth)
- Range Doppler Algorithm
    - Used by the Canadian Space Agency to process RADARSAT-1 and RADARSAT-2 satellite SAR data
- Chirp Scaling Algorithm
    - Used by the European Space Agency and the German Aerospace Center (DLR) to process TerraSAR-X satellite SAR data
- These two algorithms (RDA and CSA) are very similar with the primary difference being how range cell migration correction is done.
    - RDA works with any waveform, CSA requires the use of a chirp waveform

Satellite and Low Squint Airborne SAR Algorithms

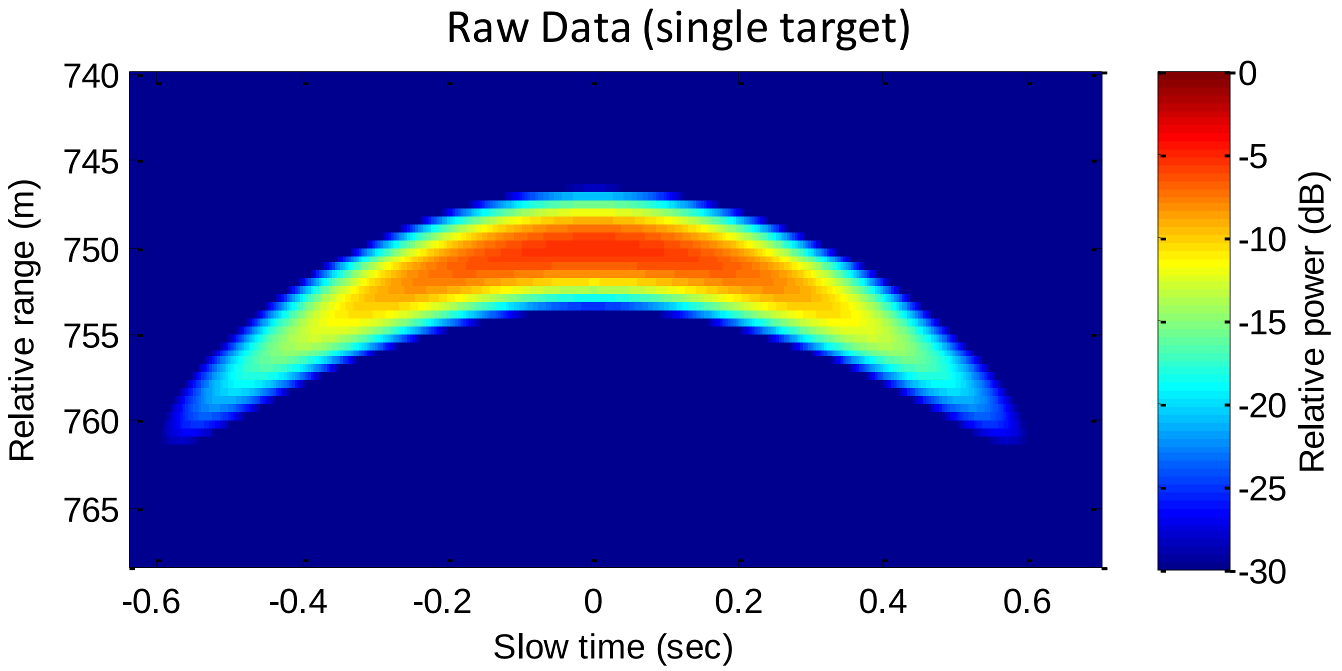
- The SAR filter is azimuth-space-invariant but it is range-variant
- The primary structure exploited by these two algorithms is that the 2-D energy from the point target lies along a 1-D contour. This energy will be interpolated or scaled/shifted to lie on a 1-D line that does not cross range bins. By converting the range varying dimension to lie on a single range bin, convolution will no longer be required in the range dimension.

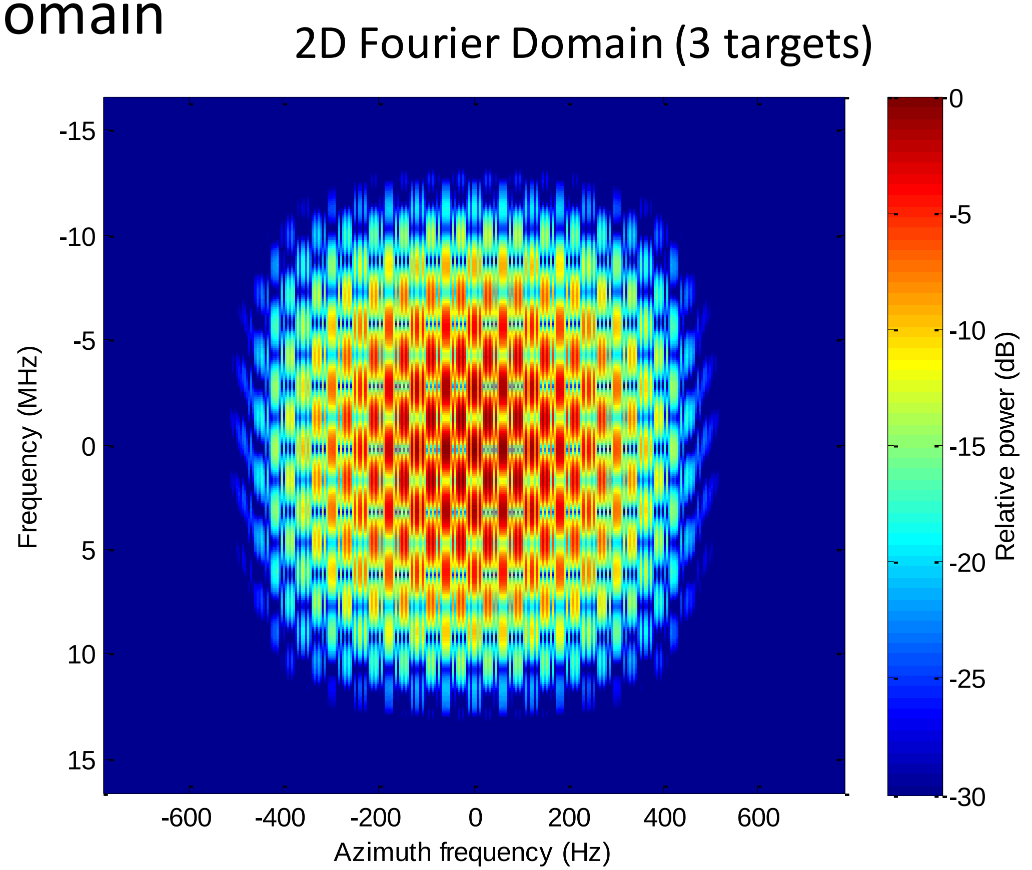
Range Doppler Algorithm (RDA) STEP 1

- Pulse compression is a LTIV filter. It is straight forward to implement in the Fourier domain.
  - Range FFT on raw data to transform to range-frequency / azimuth-space domain
  - Apply range-domain matched filter for pulse compression
- Do not take the IFFT in the range dimension when finished.

Range Doppler Algorithm (RDA) STEP 2

- Azimuth FFT
  - Transform to range-frequency / Doppler domain

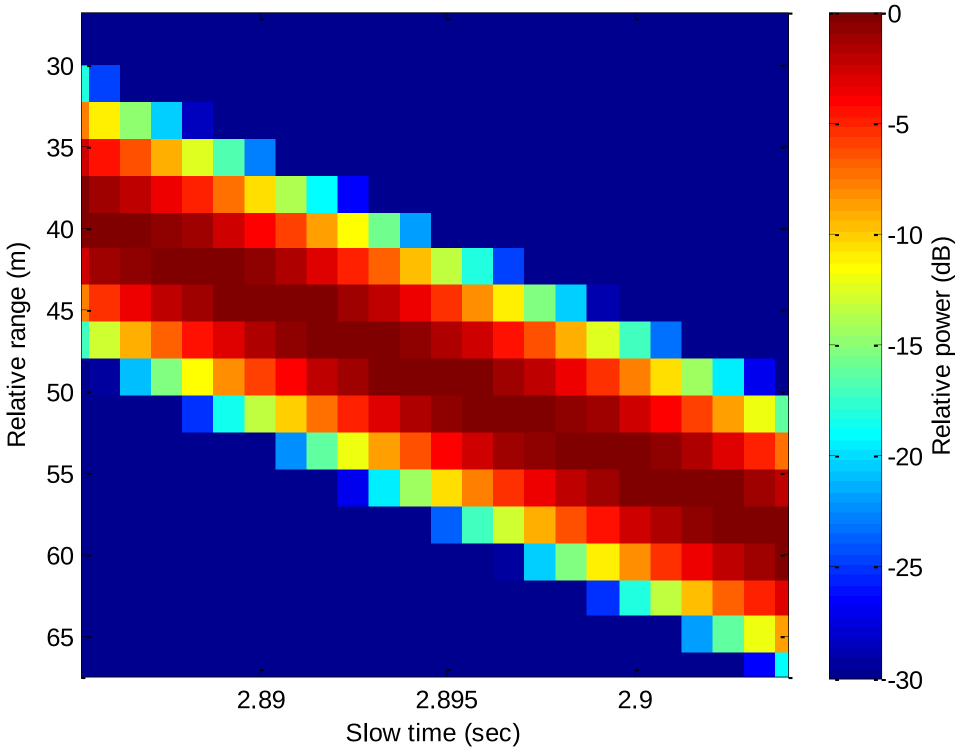




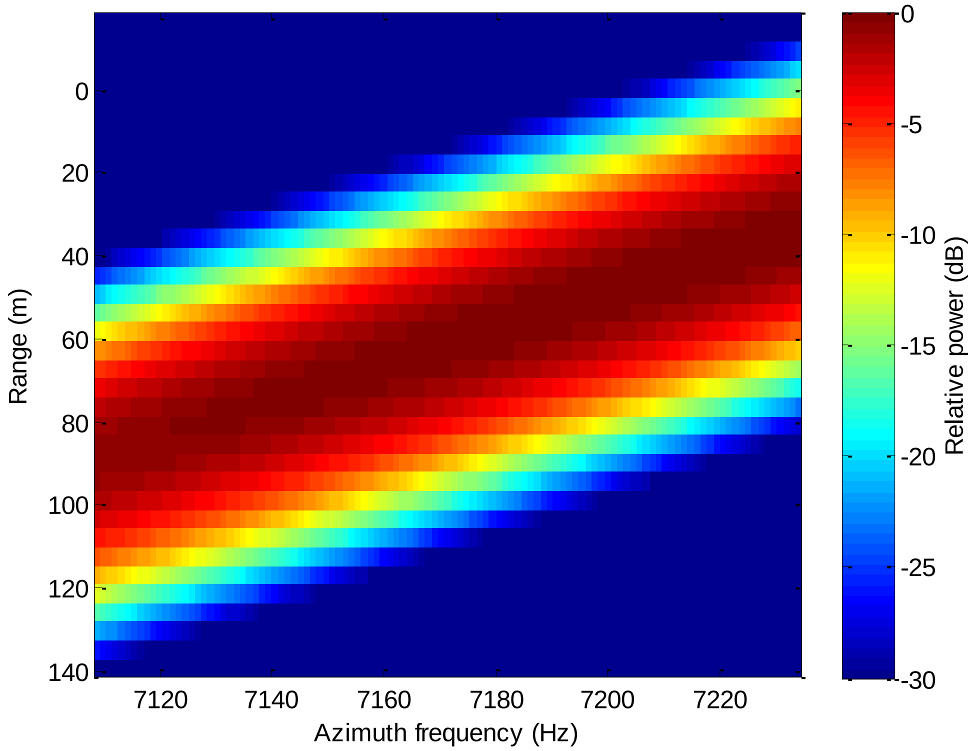
Range Doppler Algorithm (RDA): STEP 3

- Blurring occurs during the Doppler Fourier transform so that the point target “contour” is broadened. This affect is worse for large squint angles.
- This blurring can be approximated by a frequency chirp in the range domain… so to correct we need to do pulse compression again.
- This process is called Secondary Range Compression
  - For an approximate solution, this second range compression can be applied during the regular pulse compression… this is suboptimal because the Fourier transform to the Doppler domain blurs the correction so it is better to apply in the range-Doppler domain.

Range Doppler Algorithm (RDA): STEP 3



Range Space Domain (i.e. Raw Data)

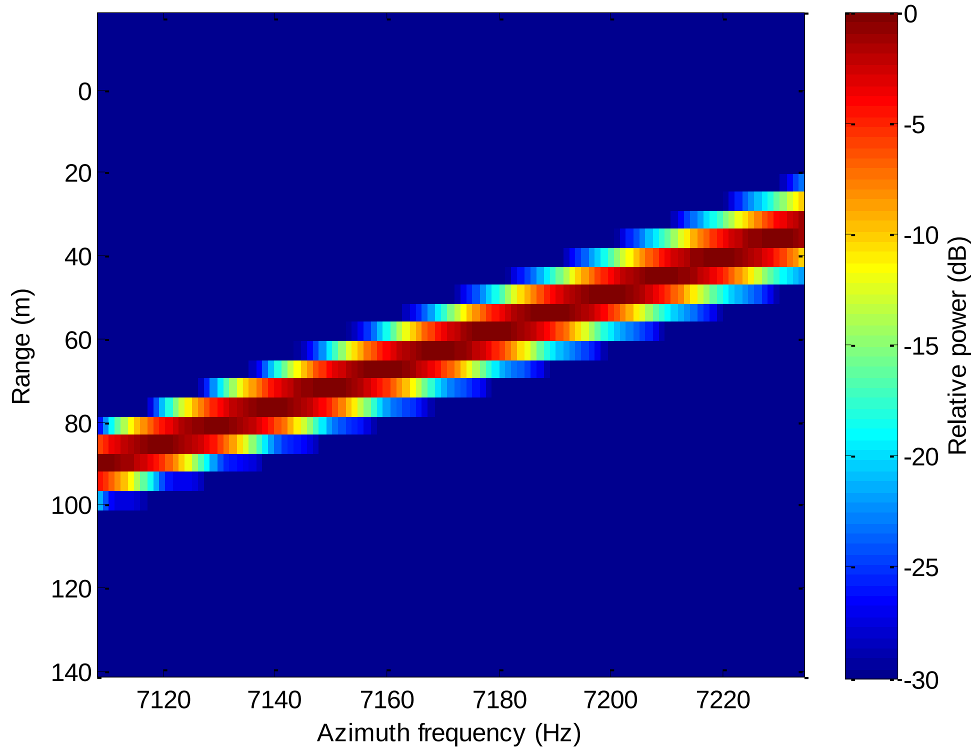


Range Doppler Domain
(note the blurring)

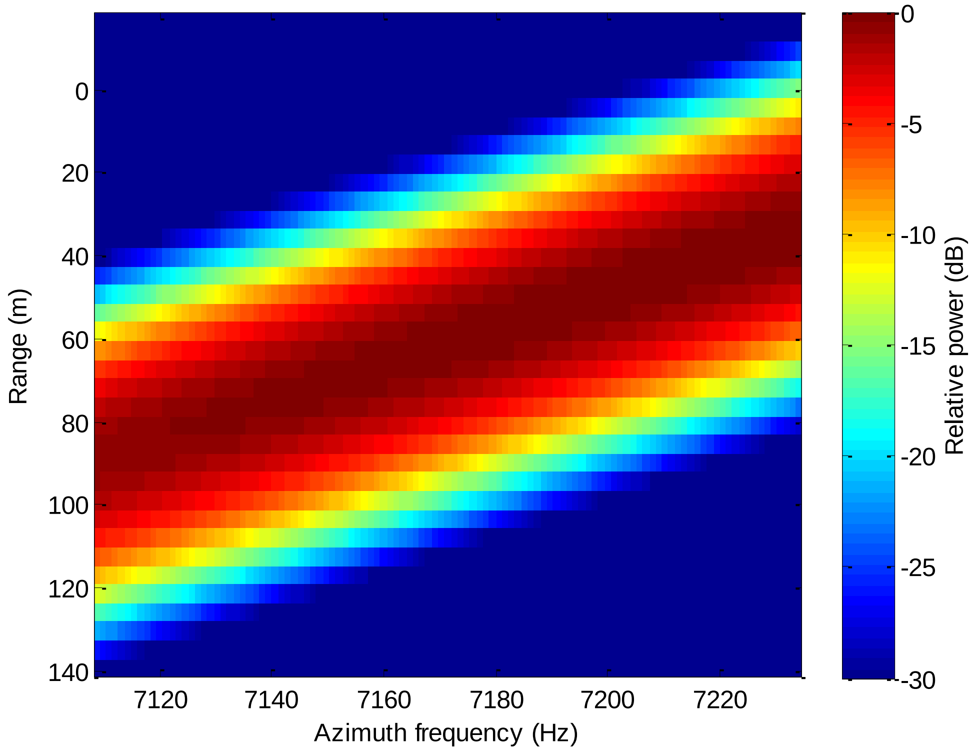
Range Doppler Algorithm (RDA): STEP 3

- The SRC correction is derived from our range Doppler representation of the signal:
  - $K_{src}(R_0, f_\eta) = \frac{2V_r^2 f_0^3 D^3(f_\eta, V_r)}{c R_0 f_\eta^2}$
  - $H_{src}(f_\eta) = exp\left(-j\pi \frac{f_\tau^2}{K_{src}(R_0, f_\eta)}\right)$
  - R_0: Range of closest approach
    - Note that this should be R_{ref} (midpoint of scene) if applied in the range-frequency domain as described here. Improved performance can be seen by applying the SRC chirp compression with the RCMC interpolating kernel since both are range varying filters at that point. If this is done, then R_0 can be used since RCMC interpolation is done in the range-Doppler domain.
  - f_η: Doppler frequency
  - V_r: Effective velocity (rectilinear coordinate system)
  - f_τ: Baseband range frequency
  - f_0: Center frequency
  - D: Cosine of the squint angle, $D = \cos(\theta_s) = (1 - \sin^2(\theta_s))^{0.5} = \left(1 - \frac{c^2 f_\eta^2}{4 f_c^2 V^2}\right)^{0.5}$

Range Doppler Algorithm (RDA): STEP 3



Range Doppler Domain
(After Secondary Range Compression)



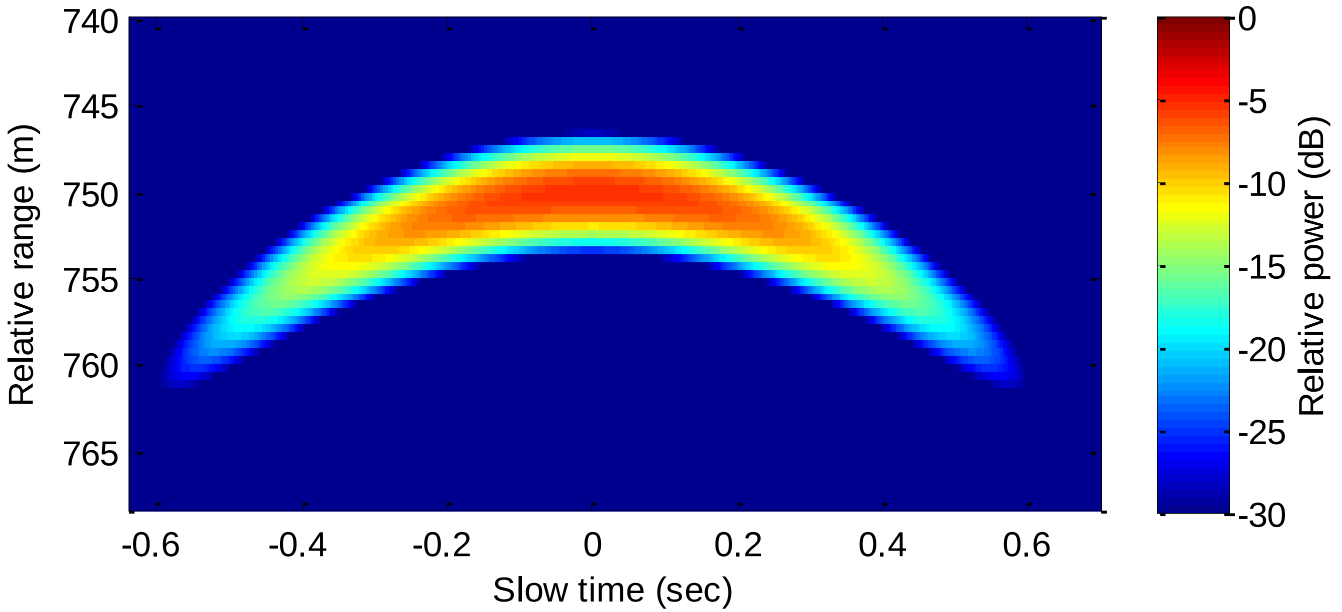
Range Doppler Domain
(note the blurring)

Range Doppler Algorithm (RDA): STEP 4

- Range IFFT
  - Transform to range / Doppler domain

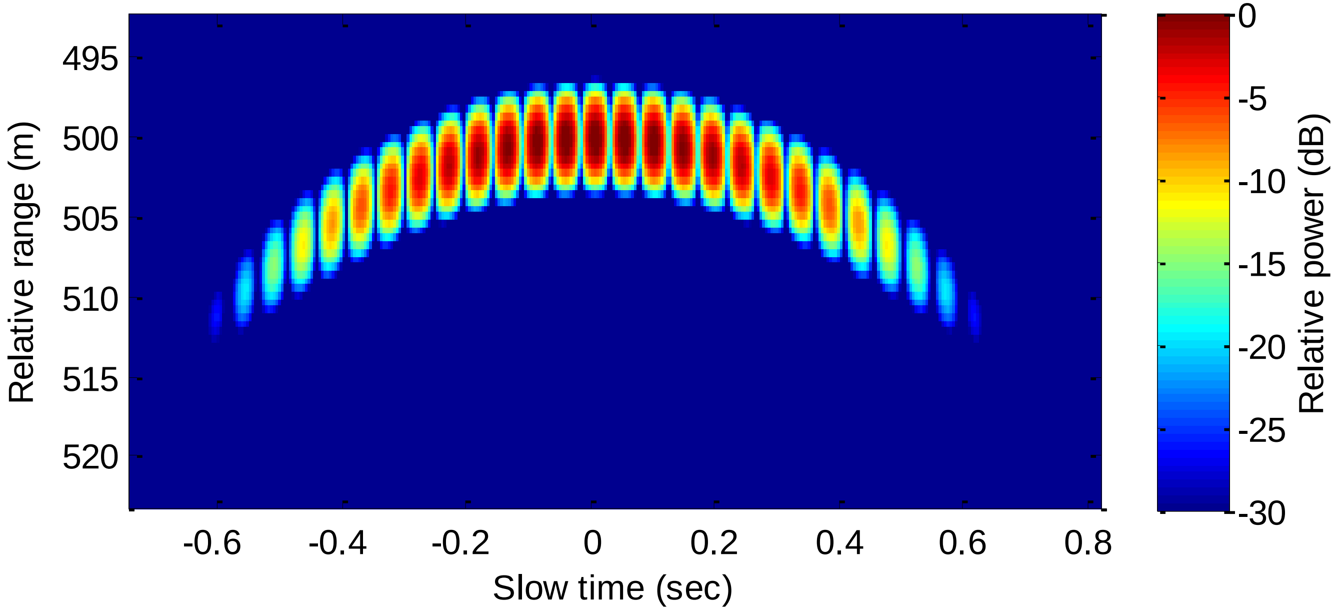
Range Doppler Algorithm (RDA): STEP 5

- Range Cell Migration Correction (RCMC) in Doppler domain
  - SAR processing is a 2-D filter, but the energy is focused along a single hyperbolic contour.
    - Contour is range dependent
  - The idea is to flatten the contour using a process called RCMC
- Example point target response:
- RCMC easy to apply for a single point target.



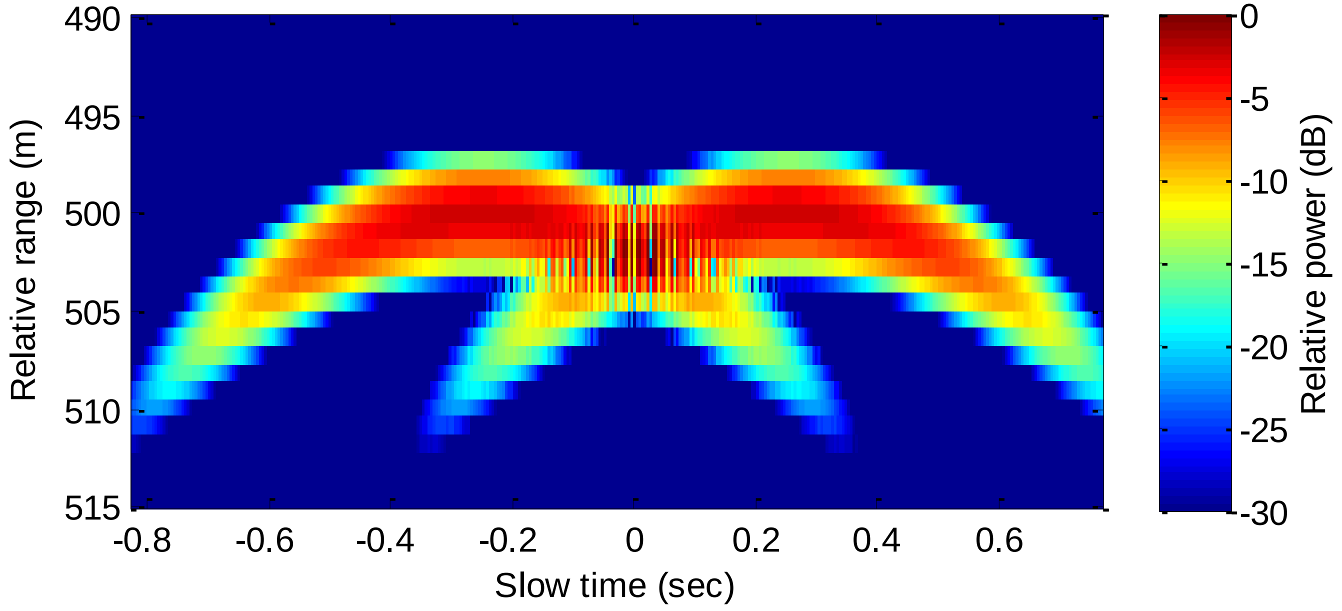
Range Doppler Algorithm (RDA): STEP 5

- Example of two point targets at the same range and next to each other. Envelope is about the same for both but the phases are offset (think of two tones and what you see is the beat frequency… double side band suppressed carrier).
- Could apply RCMC for this case as well.



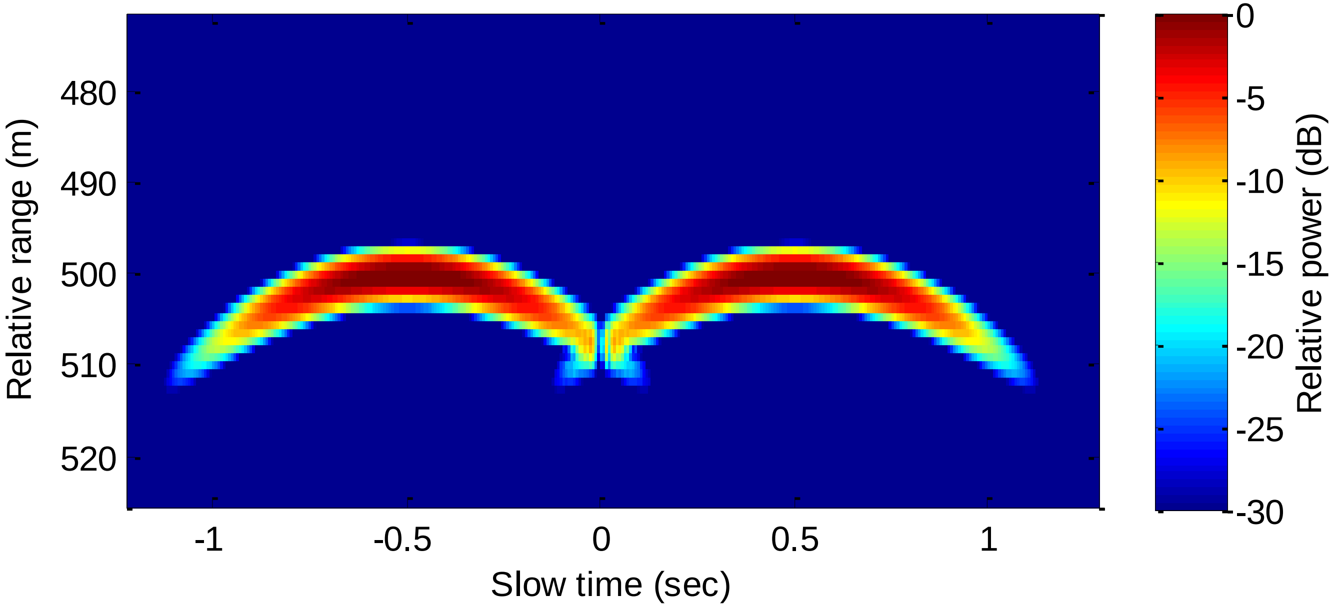
Range Doppler Algorithm (RDA): STEP 5

- Example of two point targets far apart from each other… RCMC not possible because each target needs a different correction.



Range Doppler Algorithm (RDA): STEP 5

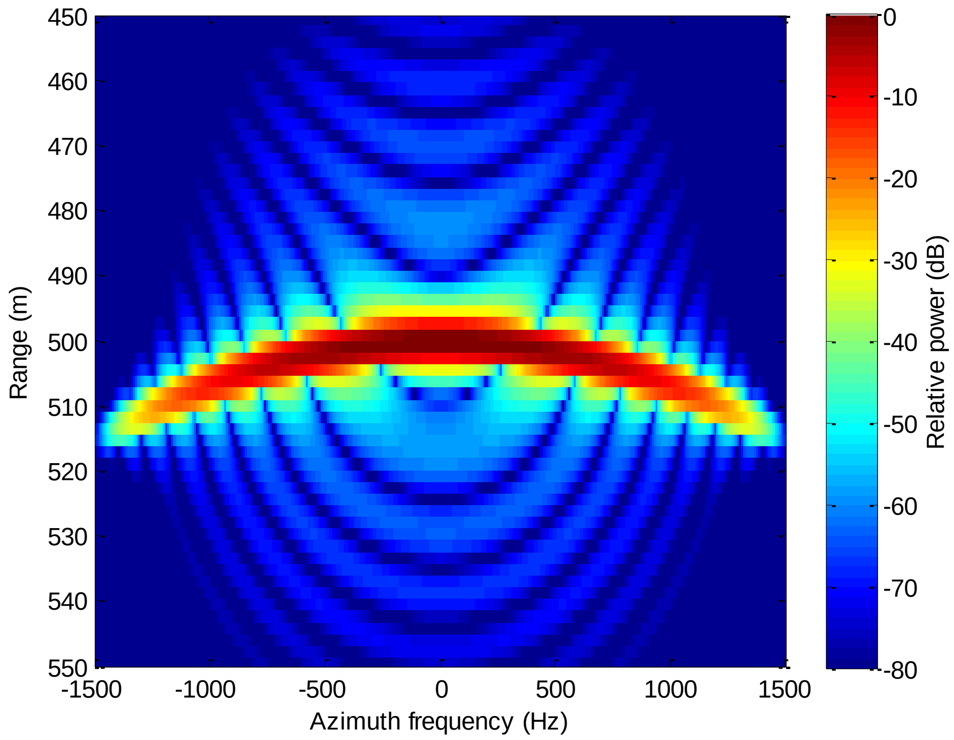
- Example of two point targets far apart from each other:



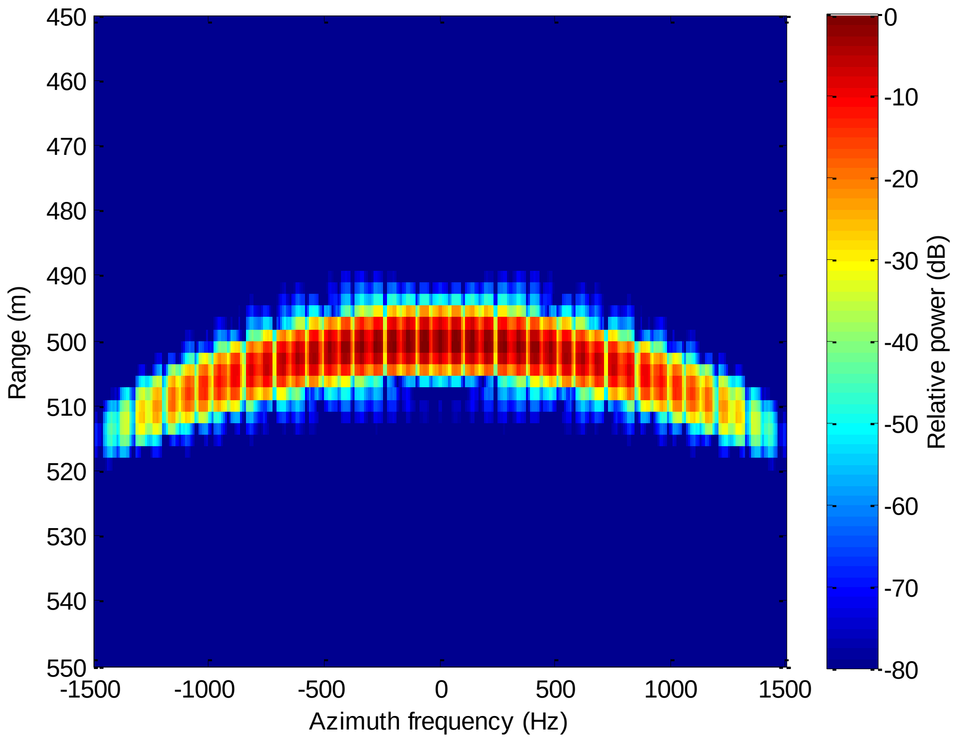
Range Doppler Algorithm (RDA): STEP 5

- RCMC cannot be applied in the range-space domain because RCMC is dependent on the relative along-track position rather than the absolute along-track position.
- Hmmm… we know that the range cell migration is a function of incidence angle (i.e. Doppler).
- RCMC can be applied in the range-Doppler domain because RCMC depends on the absolute Doppler.
- **Every target at the same range has the same envelope in the range-Doppler domain!!!**

Range Doppler Algorithm (RDA): STEP 5



Single Target



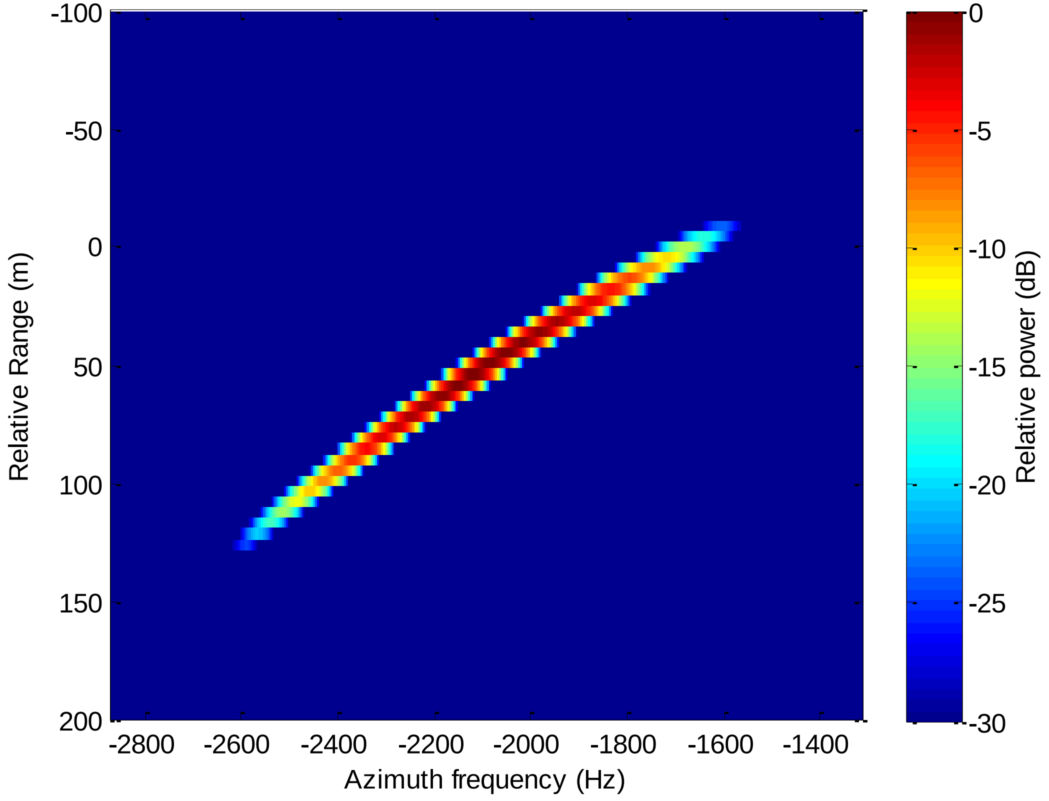
Both Targets… envelope has not changed, but interference pattern has.

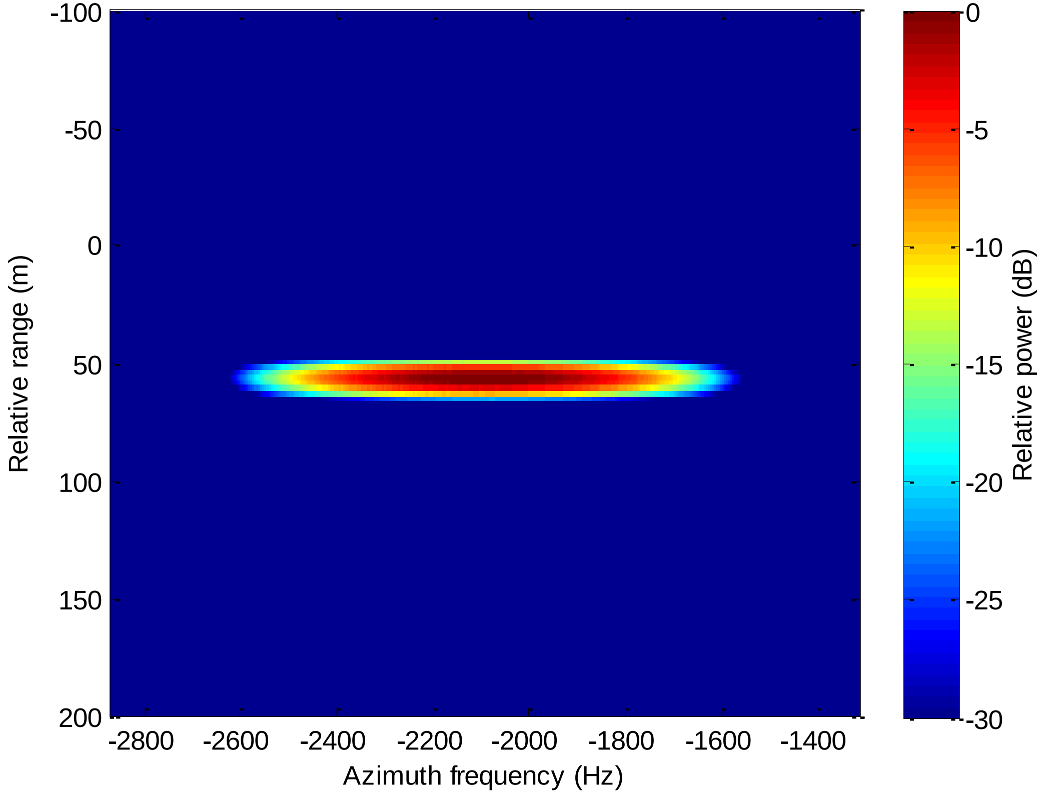
Range Doppler Algorithm (RDA): STEP 5

- We need to remove this much delay (this turns out to be simple geometry):
  - $R_0\left(\frac{1-D\left(f\,\eta,V_r\right)}{D\left(f\,\eta,V_r\right)}\right)$
  - R_0: Range of closest approach
  - f_η: Doppler frequency
  - V_r: Effective velocity (rectilinear coordinate system)
  - D: Cosine of the squint angle

Range Doppler Algorithm (RDA): STEP 5

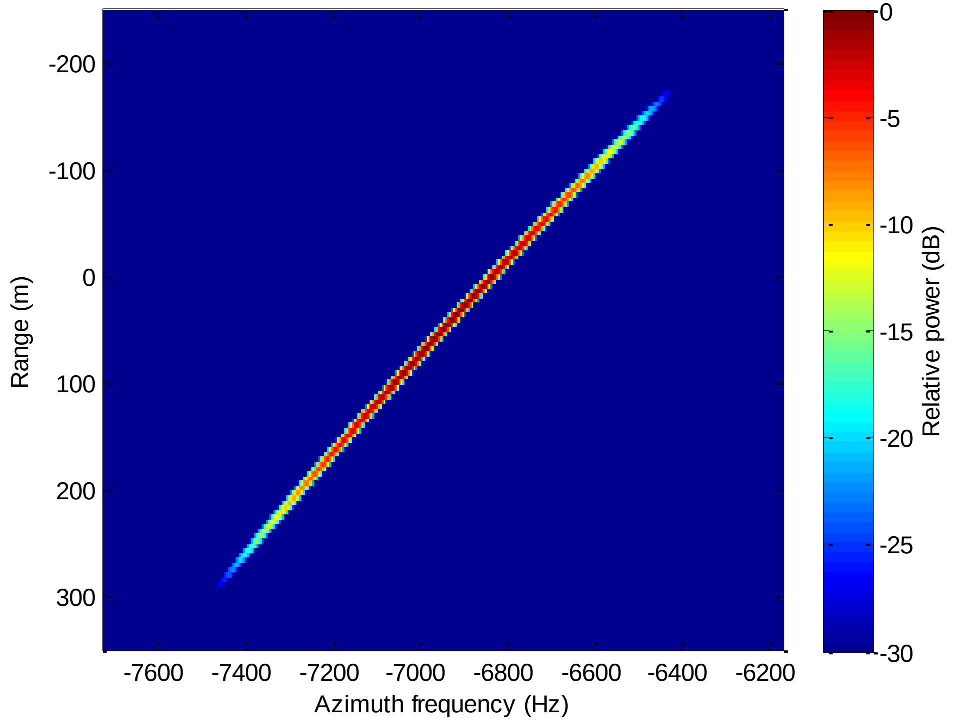
- Use the truncated and windowed sinc interpolation method to do the time shift. Example of 3 deg squint:

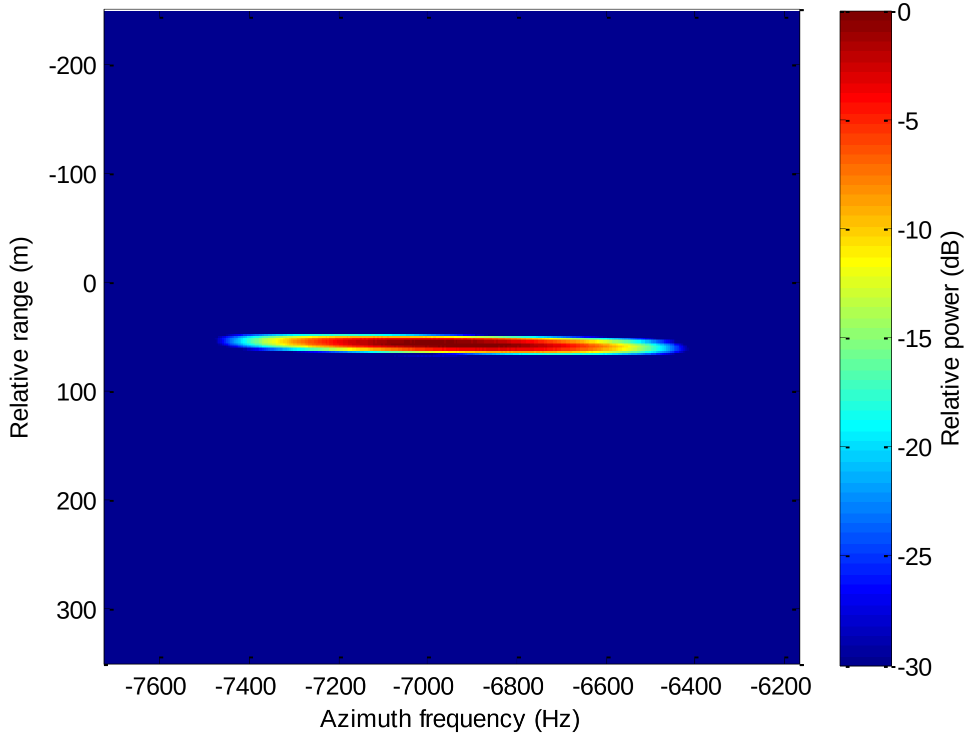




Range Doppler Algorithm (RDA): STEP 5

- Use the truncated and windowed sinc interpolation method to do the time shift. Example of 10 deg squint:





Range Doppler Algorithm (RDA): STEP 6

- All targets have been interpolated so that they occupy a single range bin in the range-Doppler domain.
- Originally the problem was that the range cell migration changed as a function of range → This prevented a simple application of Fourier methods since the response was space-variant.
- Now it is no longer a 2-D filter so the space variance does not matter and we only need to apply a 1-D azimuth filter.

Range Doppler Algorithm (RDA): STEP 6

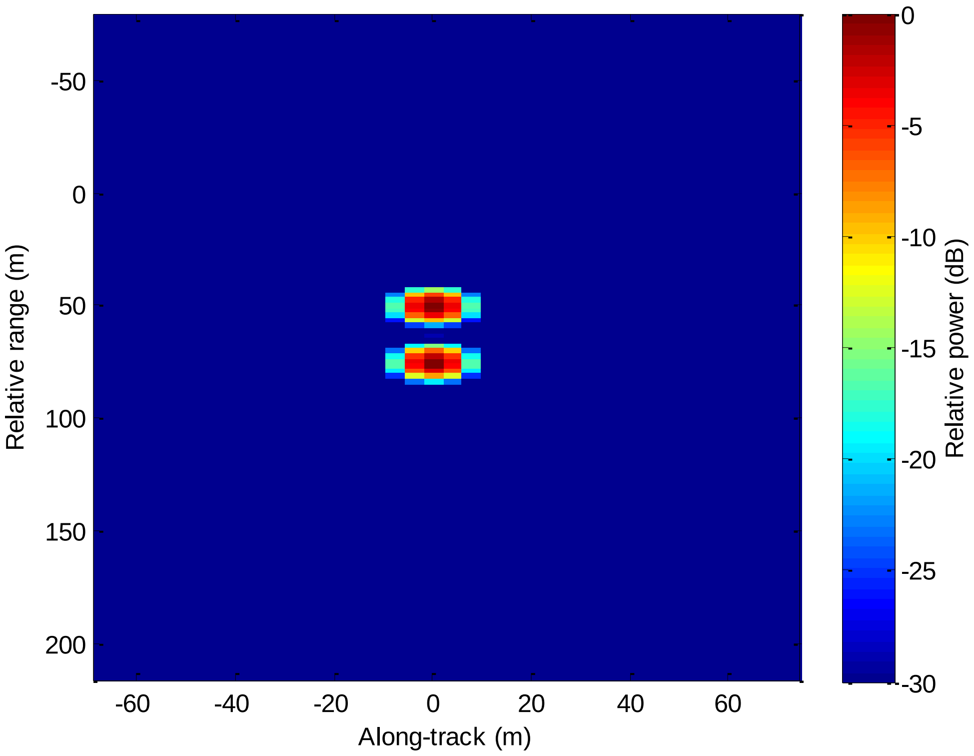
- Using the range-Doppler representation of the signal after RCMC, the azimuth compression filter is:
  - $exp\left\{j\,\frac{4\pi R_0 D\left(f\,\eta, V_r\right) f_c}{c}\right\}$
  - R_0: Range of closest approach
  - $f\eta$: Doppler frequency
  - V_r: Effective velocity (rectilinear coordinate system)
  - D: Cosine of the squint angle
  - f_c: Center frequency
  - c: Speed of light

Range Doppler Algorithm (RDA): STEP 7

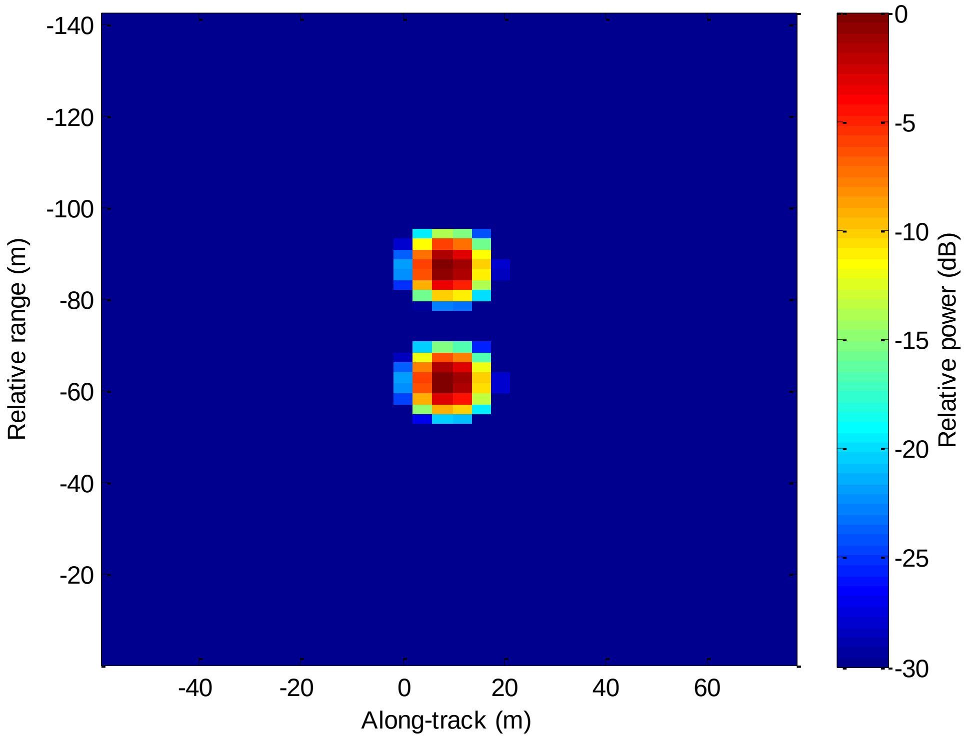
- Azimuth IFFT
  - Transform into range / azimuth-space domain

Range Doppler Algorithm (RDA): STEP 7

- Example (side note: range dependent Doppler centroid correction and relative range cell migration correction when there is squint).



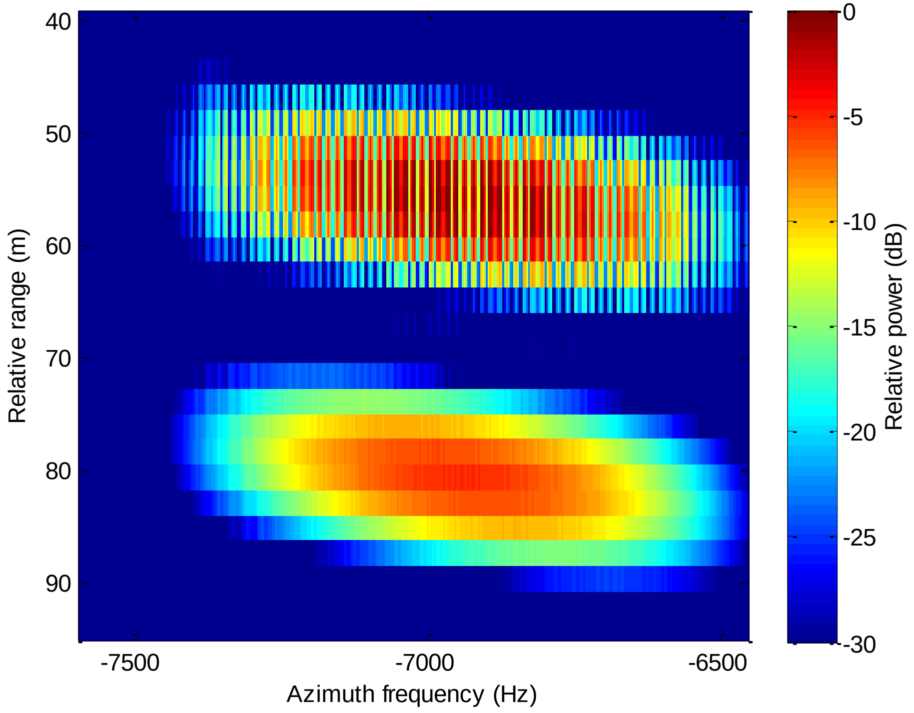
No squint: Position is perfect



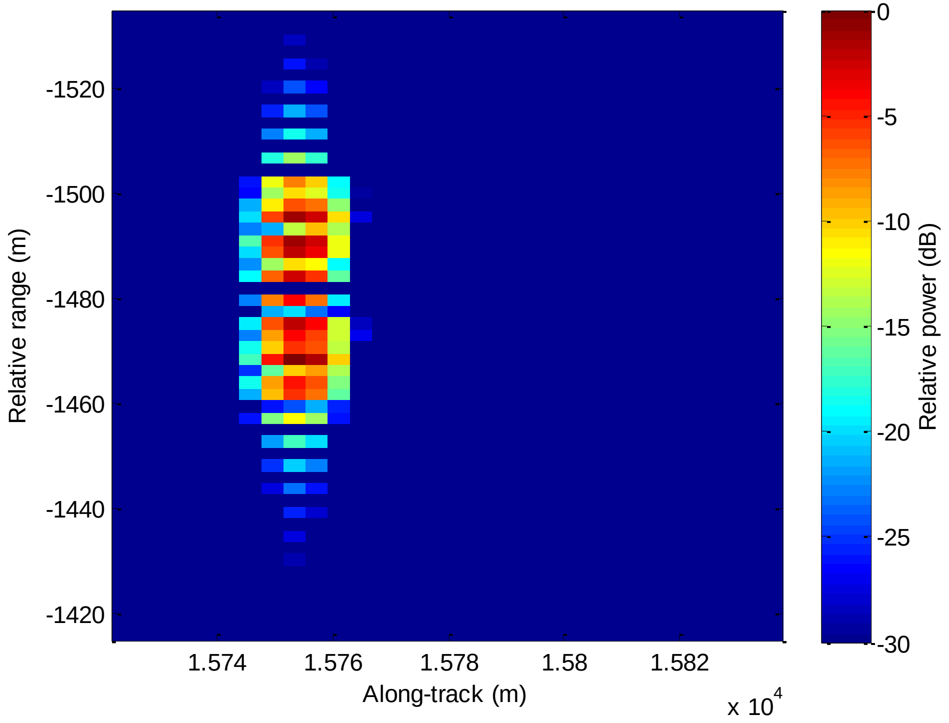
3 deg squint: range is correct, but azimuth is off by one pixel

Range Doppler Algorithm (RDA): STEP 7

10 deg squint (RCMC not perfect)



Azimuth correction ends with smeared range bins

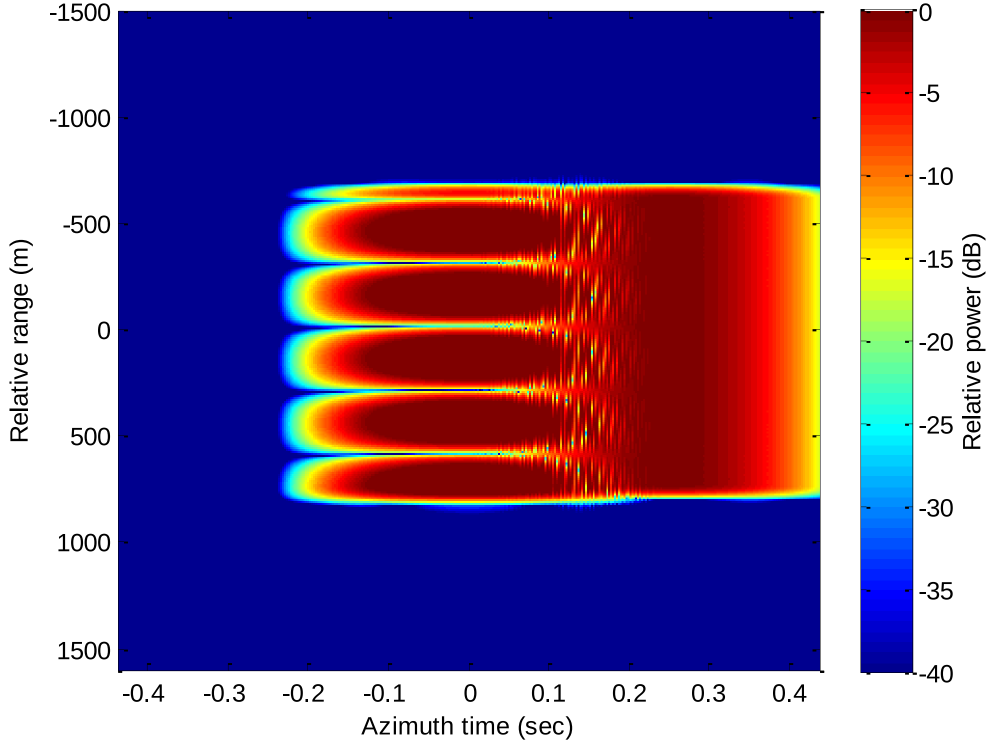


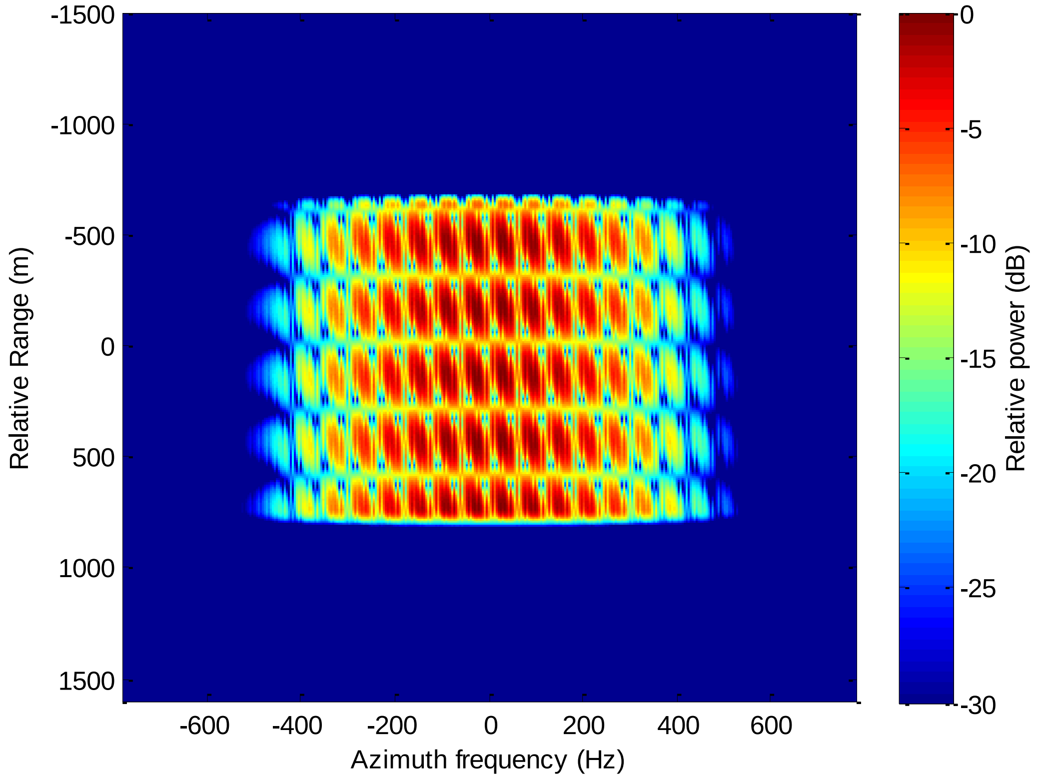
Chirp Scaling Algorithm (CSA)

- The problem with RDA is that the RCMC interpolation is slow and requires SRC.
- Chirp scaling does the same thing as RDA, but does the RCMC with chirp scaling which also makes the blurring from the Doppler Fourier transform smaller.
  - Greater efficiency + range/azimuth decoupling built into range compression (analogous to range Doppler algorithms secondary range compression)

Chirp Scaling Algorithm (CSA): Step 1

- Azimuth FFT
  - Transform to range / Doppler domain



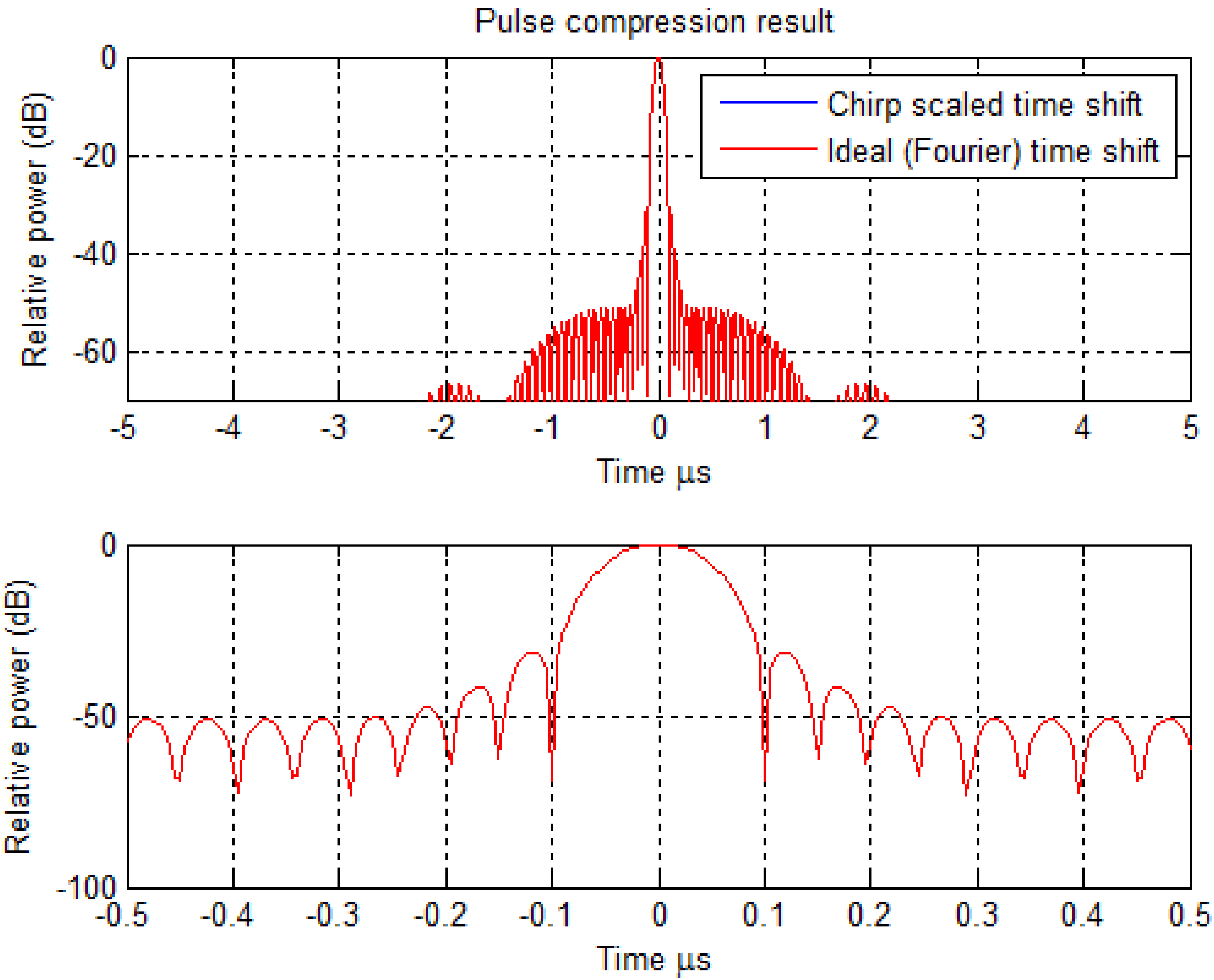


Chirp Scaling Algorithm (CSA): Step 2

- Apply chirp scaling… multiply by:
  - $exp\left\{j\pi K_m\left[\frac{D\left(f_{\eta_{ref}},V_r\right)}{D\left(f_\eta,V_r\right)}-1\right]\left(\tau-\frac{2R_{ref}}{cD\left(f_\eta,V_r\right)}\right)^2\right\}$
  - R_{ref}: Range of closest approach for reference range for bulk RCM (usually the midpoint in the range)
  - f_η: Doppler frequency
  - $f_{\eta_{ref}}$: Doppler frequency at reference (usually Doppler centroid)
  - V_r: Effective velocity (rectilinear coordinate system)
  - D: Cosine of the squint angle
  - τ: Time
  - c: Speed of light

Chirp Scaling Algorithm (CSA): Step 2

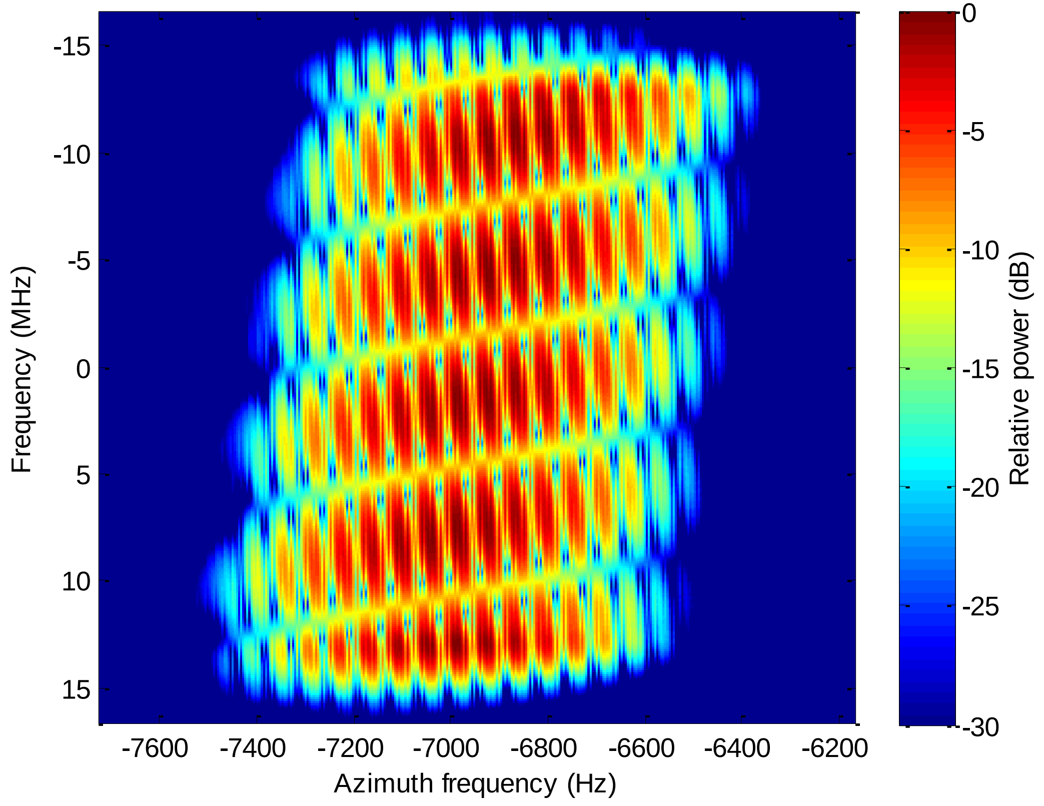
- Continued…
  - $K_m = \frac{K_r}{1 - K_r \frac{c R_0 f_\eta^2}{2 V_r^2 f_c^3 D^3(f_\eta, V_r)}}$
  - K_r: Range chirp rate
  - c: Speed of light
  - R_0: Range of closest approach
  - f_η: Doppler frequency
  - V_r: Effective velocity (rectilinear coordinate system)
  - f_c: Center frequency
  - D: Cosine of the squint angle



Pulse compression result
Chirp scaled time shift
Ideal (Fourier) time shift
Relative power (dB)
0
-20
-40
-60
-5
-4
-3
-2
-1
0
1
2
3
4
5
Time µs
Relative power (dB)
0
-50
-100
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
Time µs

Chirp Scaling Algorithm (CSA): Step 3

- Range FFT
  - Transform to range-frequency / Doppler domain

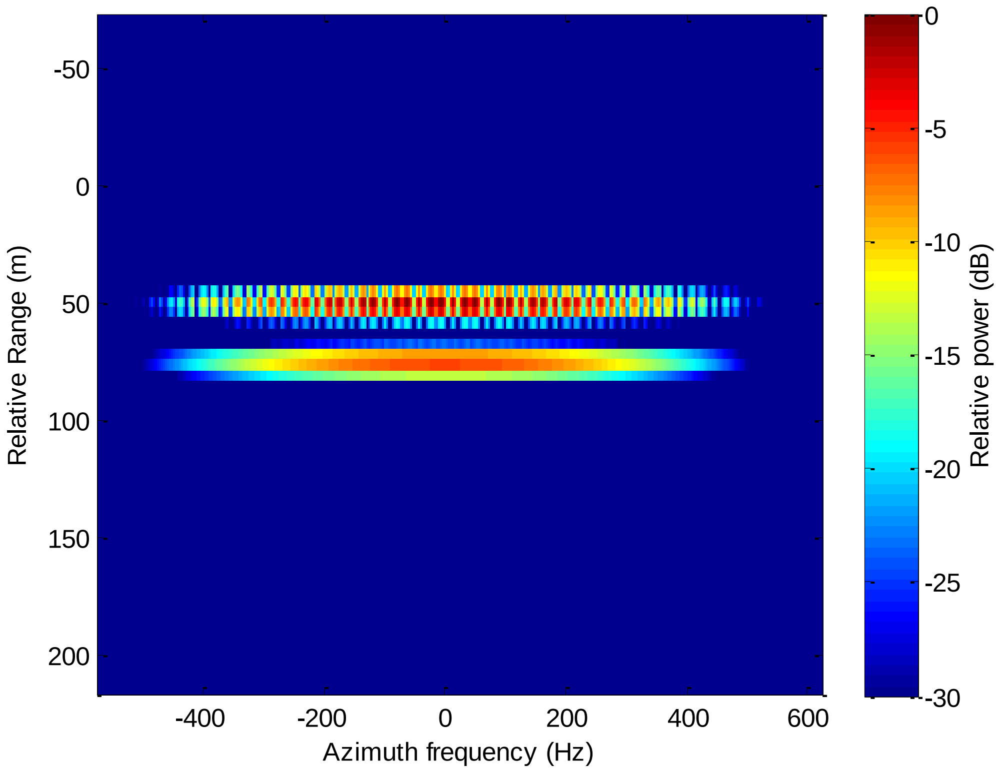


Chirp Scaling Algorithm (CSA): Step 4

- Range Compression (including range/azimuth decoupling) + bulk range cell migration correction
  - $exp\left\{j\frac{\pi D(f\eta,V_r)}{K_m D(f\eta_{ref},V_r)}f_\tau^2\right\}exp\left\{j\frac{4\pi}{c}\left[\frac{1}{D(f\eta,V_r)}-\frac{1}{D(\eta_{ref},V_r)}\right]R_{ref}f_\tau\right\}$
  - R_{ref}: Range of closest approach for reference range for bulk RCM (usually the midpoint in the range)
  - f_η: Doppler frequency
  - $f_{\eta_{ref}}$: Doppler frequency at reference (usually Doppler centroid)
  - V_r: Effective velocity (rectilinear coordinate system)
  - D: Cosine of the squint angle
  - f_τ: Baseband range frequency
  - c: Speed of light
  - K_m: From before but evaluated at R_{ref}

Chirp Scaling Algorithm (CSA): Step 5

- Range IFFT
  - Transform to range / Doppler domain

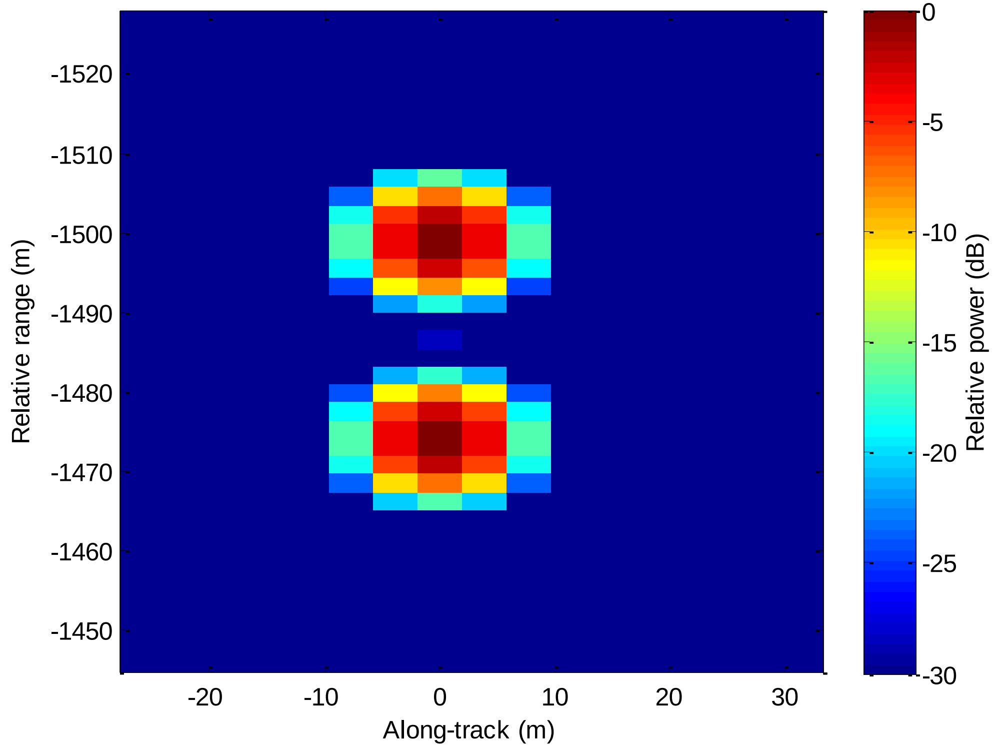


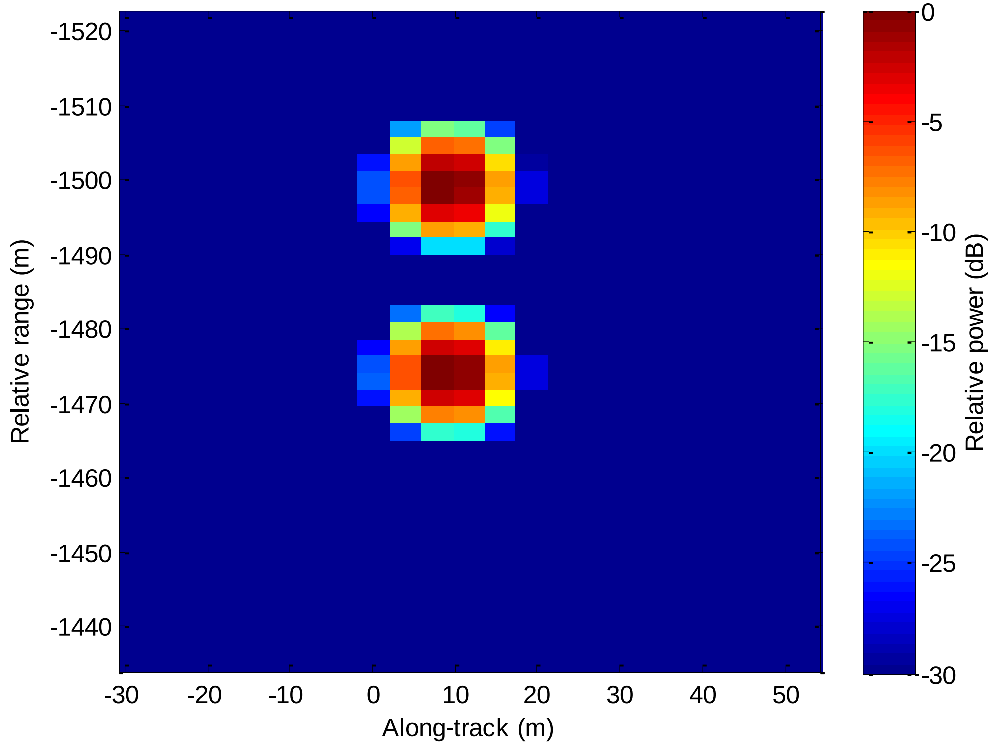
Chirp Scaling Algorithm (CSA): Step 6

- Azimuth compression and phase correction. Multiply by…
  - $exp\left\{j\frac{4\pi R_0 D\left(f_\eta, V_r\right) f_c}{c}\right\} exp\left\{-j\frac{4\pi K_m}{c^2}\left[1-\frac{D\left(f_\eta, V_r\right)}{D\left(\eta_{ref}, V_r\right)}\right]\left[\frac{R_0}{D\left(f_\eta, V_r\right)}-\frac{R_{ref}}{D\left(f_\eta, V_r\right)}\right]^2\right\}$
  - R_0: Range of closest approach
  - f_η: Doppler frequency
  - V_r: Effective velocity (rectilinear coordinate system)
  - D: Cosine of the squint angle
  - f_c: Center frequency
  - c: Speed of light
  - R_{ref}: Range of closest approach for reference range for bulk RCM (usually the midpoint in the range)
  - K_m: From before

Chirp Scaling Algorithm (CSA): Step 7

- Azimuth IFFT
  - Transform to range / azimuth-space domain





Wide Aperture (Airborne and Ground based) Algorithms

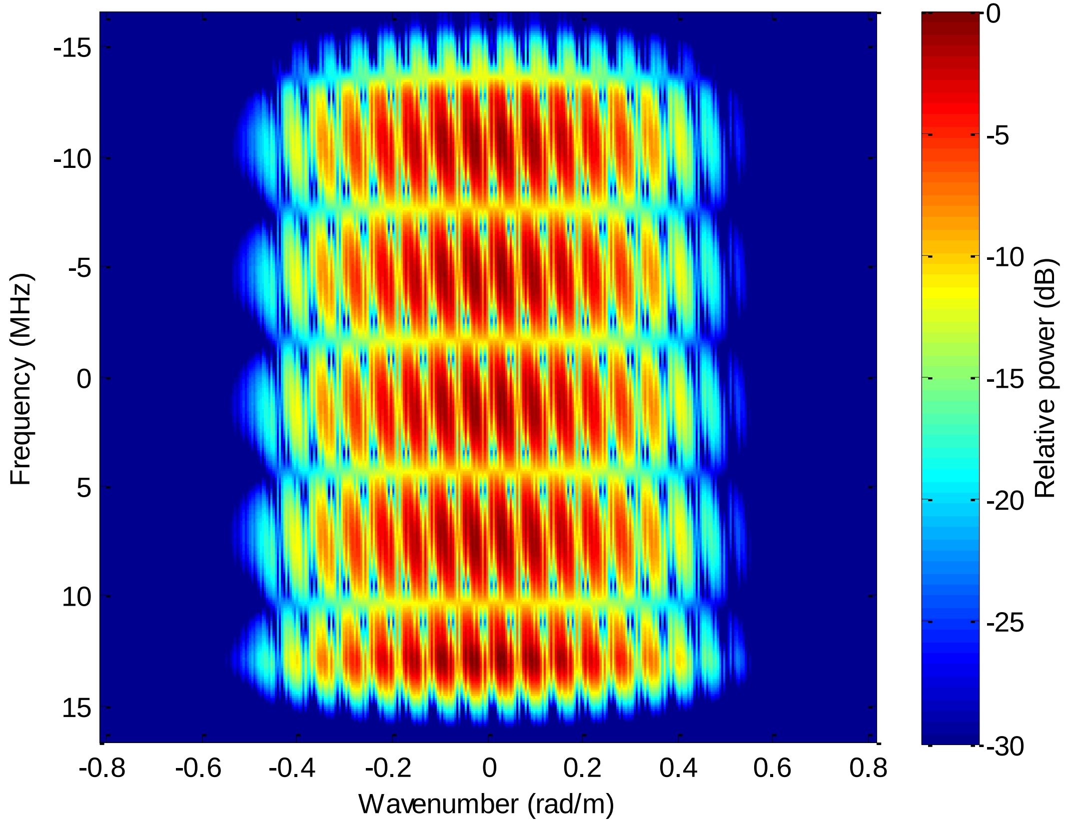
- f-k migration (AKA ω-k migration as in omega-wavenumber migration)
  - Handles strip map mode data collection with very wide apertures
  - Disadvantage is that time and space variant modifications are not handled well because processing is done in the f-k domain.
- Time domain correlation (TDC): not covered
  - Fast factorized TDC is a good and fast implementation of TDC which keeps most of the desirable properties of TDC
  - Lars M.H. Ulander et al., Synthetic-Aperture Radar Processing Using Fast Factorized Back-Projection, Transactions on Aerospace and Electronic Systems, vol. 39, no. 3, July 2003.
- Polar Format Algorithm (PFA) : not covered
  - Armin W. Doerry, Synthetic Aperture Radar Processing with Tiered Subapertures, Sandia Report SAND94-1390, 1994.
    - Very complete description of PFA
  - Jack L. Walker, Range-Doppler Imaging of Rotating Objects, IEEE Transactions on Aerospace and Electronic Systems, vol. 16, no. 1, Jan 1980.
    - Original reference.

F-k migration

- Exploding reflector model
  - The linear target model is equivalent to the exploding reflector model
  - Rather than the radar transmitting a pulse at time zero, each target is replaced by an isotropic source that radiates a pulse starting at time zero and the velocity of propagation is halved.

F-k migration: Step 1

- Two-dimensional FFT
  - Transform to range-frequency / wavenumber domain
    - (Wavenumber has a one to one mapping with Doppler domain)



F-k migration: Step 2

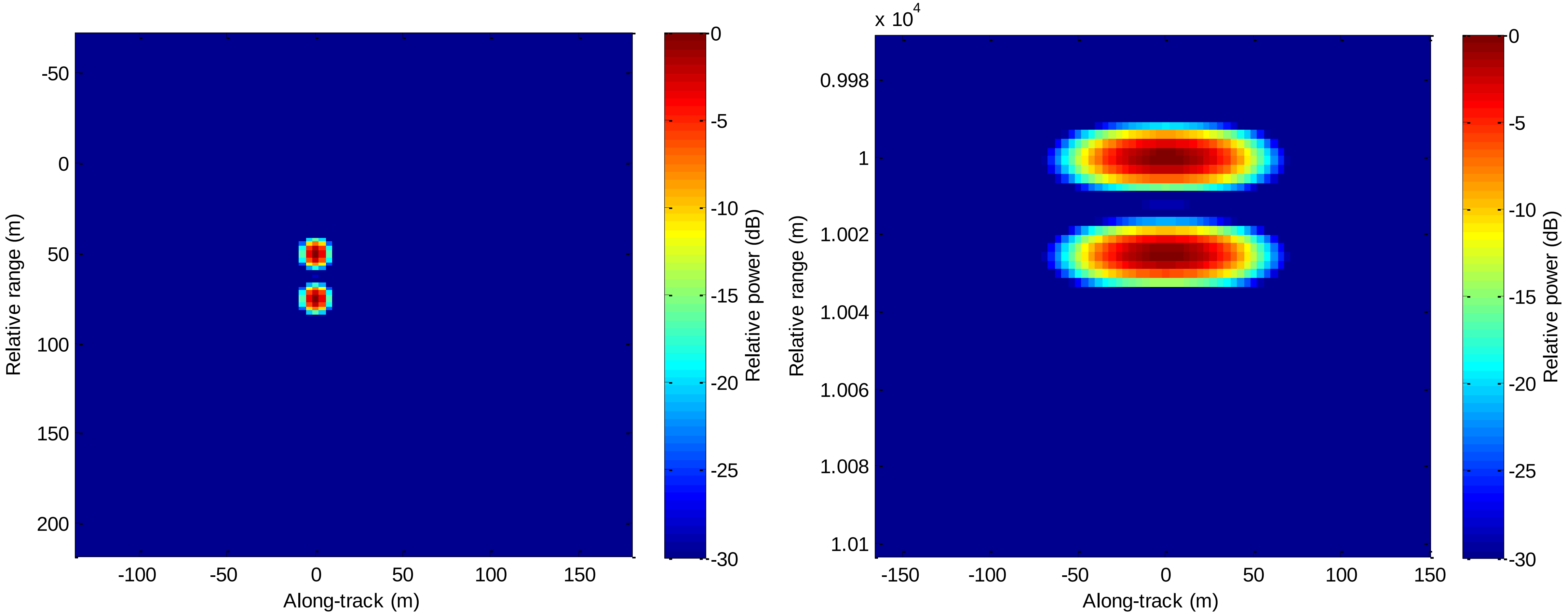
- Reference frequency multiply (RFM)
  - Applies the 2-D filter for the reference range (i.e. determine the response from a point target at the reference range and then use that as a correlation/matched filter)
    - This will apply both range and azimuth compression
  - We know that this will perfectly focus the reference range, but slowly degrade away from that range because the filter needs to be space variant to perfectly focus the targets

F-k migration: Step 2

- The RFM is a complex exponential with this phase:
  - $\frac{4\pi R_{ref}}{c}\sqrt{(f_0 + f_\tau)^2 - \frac{c^2 f_\eta^2}{4V_r^2}} + \frac{\pi f_\tau^2}{K}$
  - R_{ref}: Reference range for bulk RFM filter (usually midpoint of range)
  - c: Speed of light
  - f_O: Center frequency
  - f_τ: Baseband range frequency
  - f_η: Doppler frequency
  - V_r: Effective velocity (rectilinear coordinate system)
  - K: Chirp rate

F-k migration: Step 2

- Examples of reference range and away from reference range



F-k migration: Step 3

- Stolt Interpolation
  - First we note the residual phase after reference frequency multiply (RFM) filter is:
    - $\frac{4\pi\left(R_0-R_{ref}\right)}{c}\sqrt{(f_0+f_\tau)^2-\frac{c^2 f_\eta^2}{4V_r^2}}$
  - R_0: Range of closest approach
  - R_{ref}: Reference range for bulk RFM filter
  - c: Speed of light
  - f_O: Center frequency
  - f_τ: Baseband range frequency
  - f_η: Doppler frequency
  - V_r: Effective velocity (rectilinear coordinate system)

F-k migration: Step 3

- Stolt Interpolation
  - Data start uniformly sampled in $f = f_0 + f_\tau$
  - Define a new variable f':
    - $f' = \sqrt{(f_0 + f_\tau)^2 - \frac{c^2 f_\eta^2}{4V_r^2}}$
  - We note that there is a one to one mapping between f to f' and we can solve for f in terms of f':
    - $f = \sqrt{f'^2 + \frac{c^2 f_\eta^2}{4V_r^2}}$
  - If we do a change of variable to f' and resample the range frequency axis so that f' is uniformly sampled (instead of f), then we end up with:
    - $\frac{4\pi\left(R_0 - R_{ref}\right)}{c} f'$
  - Now the IFFT of this signal will produce a focused point at R_0 which is just what we want!
  - Resampling usually uses sinc interpolation for best results, but sometimes other interpolators are used such as linear interpolation with oversampling

F-k migration: Step 4

- Two-dimensional IFFT
  - Transform to range-space domain
  - Before and after Stolt interpolation for target a long way from the reference range.