Computation of Signal Gain and Noise Gain of a SAR Processor

Ulrich Balss, Helko Breit, Andreas Niedermeier

German Aerospace Center (DLR)
Remote Sensing Technology Institute (IMF)
Outline

- Introduction
- Energy Normalization vs. Power Normalization
- Range (Pre-)Compression and Chirp Energy
- Azimuth Compression and Aperture Length
- Signal Gain vs. Noise Gain
- White and Colored Noise
- Tracking Signal and Noise Power through the Focusing Chain
- Conclusions
Signal Gain of a SAR Processor
(Simplified Model)

\[ g_{\text{SAR}}(i,j) = \frac{1}{N' M'} \sum_{i,j} \text{S}_{\text{img}}^2(i,j) \]

\[ g_{\text{SRAW}}(i,j) = \frac{1}{N M} \sum_{i,j} \text{S}_{\text{raw}}^2(i,j) \]
Signal and Noise Gain
(More Detailed Sight)

\[ S_{\text{raw}}(i,j) + n_{\text{rec}}(i,j) \rightarrow \cdots \rightarrow S_{\text{img}}(i,j) + n_{\text{img}}(i,j) \]

Input consists of SAR raw data and receiver noise

Several different processing steps

output consists of SAR data and image noise
Extended Chirp Scaling:
Processing Chain for Sliding Spotlight Data

Read Raw Data
Range Precompression
Azimuth Beam Gain Correction
Az. Forward FFT + Zero Padding
Range Forward FFT
Range Compression
Range Bandpass
Range Backward FFT
Azimuth Compression
Azimuth Scaling
Azimuth Backward FFT
Deramping for Pattern Correction
Azimuth Forward FFT
Antenna Pattern Correction
Sidelobe Suppression
Azimuth Bandpass
Forward FFT
Deramping
Recombination

Legend:
- per subaperture
- once for the entire datatake
- impact on processor gain
Energy versus Power Normalization

Changing the sampling rate of a signal:

\[ s_1(n) \]  
\[ S_1(k) \]
\[ S_2(k') \]
\[ s_2(n') \]

Transform to spectral domain

Spectral zero padding *(energy normalized)*

Transform back to time domain
Energy versus Power Normalization

Changing the sampling rate of a signal:

\[ s_1(n) \rightarrow \rightarrow S_1(k) \rightarrow \rightarrow S_2(k') \rightarrow \rightarrow s_2(n') \]

Transform to spectral domain

Spectral zero padding (power normalized)

Transform back to time domain
Range Precompression

\[ s_{rc}(t, \tau) = s_{raw}(t, \tau) \ast r(-\tau) \]

Convolution with time reversed chirp replica in time domain
Range Precompression

\[ s_{rc}(t, \tau) = s_{raw}(t, \tau) \ast r(-\tau) \]

Convolution with time reversed chirp replica in time domain

\[ S_{rc}(t, \nu) = S_{raw}(t, \nu) \cdot R^*(\nu) \]

Multiplication by conjugate complete spectrum of chirp replica
Range Precompression

\[ s_{rc}(t, \tau) = s_{raw}(t, \tau) \ast r(-\tau) \]

Convolution with time reversed chirp replica in time domain

\[ S_{rc}(t, \nu) = S_{raw}(t, \nu) \cdot R^*(\nu) \]

Multiplication by conjugate complete spectrum of chirp replica

\[ S_{rc}(t, \nu) = \frac{S_{raw}(t, \nu) \cdot R^*(\nu)}{\|R(\nu)\|^2} \]

Ditto. with normalization:
Chirp energy occurs twice (in replica \textit{and in the signal})
Range Precompression

\[ s_{rc}(t, \tau) = s_{raw}(t, \tau) \ast r(-\tau) \]

Convolution with time reversed chirp replica in time domain

\[ S_{rc}(t, \nu) = S_{raw}(t, \nu) \cdot R^*(\nu) \]

Multiplication by conjugate complete spectrum of chirp replica

\[ S_{rc}(t, \nu) = \frac{S_{raw}(t, \nu) \cdot R^*(\nu)}{\|R(\nu)\|^2} \]

Ditto. with normalization:
Chirp energy occurs twice
(in replica and in the signal)

\[ S_{rc}(t, \nu) = \frac{S_{raw}(t, \nu)}{R(\nu)} \]

This is eqv. to a complex division
(in time domain: deconvolution)
Range Precompression

\[ s_{rc}(t, \tau) = s_{raw}(t, \tau) \ast r(-\tau) \]

Convolution with time reversed chirp replica in time domain

\[ S_{rc}(t, \nu) = S_{raw}(t, \nu) \cdot R^*(\nu) \]

Multiplication by conjugate complete spectrum of chirp replica

\[ S_{rc}(t, \nu) = \frac{S_{raw}(t, \nu) \cdot R^*(\nu)}{\|R(\nu)\|^2} \]

Ditto. with normalization:
Chirp energy occurs twice
(in replica \textit{and in the signal})

\[ S_{rc}(t, \nu) = \frac{S_{raw}(t, \nu)}{R(\nu)} \]

This is eqv. to a complex division
(in time domain: deconvolution)

\textbf{A threshold avoids division by low values (or even zero)!}
Azimuth Compression

short aperture (e.g. ScanSAR):

long aperture (e.g. Staring Spotlight):

short aperture (e.g. near range beam):

long aperture (e.g. far range beam):

low PRF:

high PRF:

Received signal energy depends on:
- backscatter coefficient \((\text{wanted})\)
- number of echo pulses in aperture \((\text{unwanted})\)
Azimuth Compression

short aperture (e.g. ScanSAR):

long aperture (e.g. Staring Spotlight):

short aperture (e.g. near range beam):

long aperture (e.g. far range beam):

low PRF:

high PRF:

Received signal energy depends on:

- backscatter coefficient \( \text{(wanted)} \)
- number of echo pulses in aperture \( \text{(unwanted)} \)
Signal Gain versus Noise Gain

Input:
narrow-band signal + wideband noise

Output:
narrow band signal + narrow-band noise
Signal Gain versus Noise Gain

Input:
narrow-band signal + wideband noise

Output:
narrow band signal + narrow-band noise

If bandwidth of signal and noise differs:
Different portions of signal and of noise energy are removed.
Consequently, signal and noise gain of the filter are different.
White and Colored Noise

signal

Antenna pattern removal

noise

Hamming windowing

?
White and Colored Noise

\[ \text{noise gain}_{\text{APR}} = \frac{\int (N(f) \cdot A_{\text{APR}}(f))^2 df}{\int N(f)^2 df} \]
White and Colored Noise

\[
\text{noise gain}_{HW} = \frac{\int (N(f) \cdot A_{APR}(f) \cdot A_{HW}(f))^2 df}{\int (N(f) \cdot A_{APR}(f))^2 df}
\]
Tracking the Signal through the SAR Processor (Range Compressed Data)
Tracking the Signal through the SAR Processor (Azimuth Compression)
Tracking the Signal through the SAR Processor (Azimuth Scaling)
Tracking the Signal through the SAR Processor (Deramping for Antenna Pattern Removal)
Tracking the Signal through the SAR Processor (Antenna Pattern Removal)
Tracking the Signal through the SAR Processor (Multiply by Hamming Window)
Tracking the Signal through the SAR Processor (Residual Deramping)
Tracking the Signal through the SAR Processor (Recombination)

\[ f = t_{foc} \]

(...)

(...)

foc
Tracking the Noise through the SAR Processor (Range Compressed Data)
Tracking the Noise through the SAR Processor (Azimuth Compression)
Tracking the Noise through the SAR Processor (Azimuth Scaling)
Tracking the Noise through the SAR Processor (Deramping for Antenna Pattern Removal)
Tracking the Noise through the SAR Processor (Antenna Pattern Removal)
Tracking the Noise through the SAR Processor (Multiply by Hamming Window)
Tracking the Noise through the SAR Processor (Residual Deramping)
Tracking the Noise through the SAR Processor (Image in SPECAN Domain)
Conclusions

- There is no fast-track to processor calibration
Conclusions

- **There is no fast-track to processor calibration**
  - Each processing step has to be analyzed separately w.r.t. to its impact on signal gain
Conclusions

- There is no fast-track to processor calibration
  - Each processing step has to be analyzed separately w.r.t. to its impact on signal gain
  - Signal and noise gain might differ according to their different spectral properties
Conclusions

- There is **no fast-track to processor calibration**
  - Each processing step has to be analyzed separately w.r.t. to its impact on signal gain
  - Signal and noise gain might differ according to their different spectral properties
  - Keep in mind that in the processing chain, the input signal and noise is already changed by the predecessor steps
Conclusions

- There is **no fast-track** to processor calibration
  - Each processing step has to be analyzed separately w.r.t. to its impact on signal gain
  - Signal and noise gain might differ according to their different spectral properties
  - Keep in mind that in the processing chain, the input signal and noise is already changed by the predecessor steps
    - Don’t trust that white noise remains white after a filter step!
Conclusions

- There is no fast-track to processor calibration
  - Each processing step has to be analyzed separately w.r.t. to its impact on signal gain
  - Signal and noise gain might differ according to their different spectral properties
  - Keep in mind that in the processing chain, the input signal and noise is already changed by the predecessor steps
    - Don’t trust that white noise remains white after a filter step!
- A correct processor normalization is indispensable:
  - It is required from the first days of the mission!
Thank you for your attention!

Photo: © Robert Metzig, DLR-DFD 2014