Matlab + CUDA Bistatic SAR Simulation: IPR, PSF, and K-Space Transfer Function Across Agile and Non-Agile Waveforms

John Summerfield

October 9, 2025

Outline

Motivation

- Unified Matlab+CUDA simulation for non-traditional SAR geometries (forward-looking, highly squinted, bistatic) and waveform strategies (constant vs. frequency-agile).
- Predict/interpret impulse response (IPR), spatially variant point spread function (PSF), and the Fourier-domain transfer function (TF/passband).
- Provide geometry–waveform design knobs to shape sidelobe axes and mitigate skew in passband coverage.

Key Publications

- J. Summerfield, D. Kasilingam, and A. Gatesman, “Bistatic SAR Point Spread Function Analysis for Close Proximity Geometries,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, Art. no. 5236715.
- J. Summerfield, L. Harcke, B. Conder, and E. Steinbach, “Bistatic SAR Flight Demonstration Using Agile Frequency Waveforms to Achieve Orthogonal Sidelobe Axes,” *IEEE RadarConf24*, Denver, CO, 2024, pp. 1–6.
- J. Summerfield, L. Harcke, B. Conder, and D. Kasilingam, “Maximizing the Radar Generalized Image Quality Equation for Bistatic SAR Using Waveform Frequency Agility,” *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–18, 2024, Art. no. 5218918.

Platforms and Trajectories

Two platforms: transmitter (TX) and receiver (RX). Trajectories include:

- Straight-and-level, constant-velocity lines at altitude.
- Circles about scene center (at altitude).
- Conical log-spiral paths (constant squint, constant elevation).
- Zero-squint, constant-elevation direct passes.

Kinematics and LOS Quantities

$$\overrightarrow{Pos}_{\text{TX}}(t),\ \overrightarrow{Vel}_{\text{TX}}(t),\ \overrightarrow{LOS}_{\text{TX}}(t),\ \dot{\overrightarrow{LOS}}_{\text{TX}}(t)$$
$$\overrightarrow{Pos}_{\text{RX}}(t),\ \overrightarrow{Vel}_{\text{RX}}(t),\ \overrightarrow{LOS}_{\text{RX}}(t),\ \dot{\overrightarrow{LOS}}_{\text{RX}}(t)$$

Bistatic range gradient and rate gradient:

$$\overrightarrow{\nabla R}(t) = \overrightarrow{LOS}_{\text{TX}}(t) + \overrightarrow{LOS}_{\text{RX}}(t),$$
$$\overrightarrow{\nabla \dot{R}}(t) = \dot{\overrightarrow{LOS}}_{\text{TX}}(t) + \dot{\overrightarrow{LOS}}_{\text{RX}}(t).$$

Move–Stop–Move and Stepped-Frequency Modeling

- **Move–Stop–Move**: At each slow-time sample, platforms treated as stationary for RF propagation; phase delay from bistatic range, no Doppler time-scaling term.
- **Wideband via Stepped-Frequency**: Wideband waveforms approximated by stepped narrowband tones across $f \in [f_{\text{min}}(t), f_{\text{max}}(t)]$.
- Valid for slow-moving platforms and moderate bandwidth–time products; to be extended for high-speed and modern wideband designs.

Fourier-Domain Surfaces and Modulation Vectors

Define surfaces (passband parameterizations):

$$\overrightarrow{F}(f,t) = \frac{2\pi f}{c}\,\overrightarrow{\nabla R}(t), \quad \text{(constant waveform parameters)}$$
$$\overrightarrow{G}(f,t) = \text{agile variant with time-varying } f\_\text{min}(t), f\_\text{max}(t).$$

Instantaneous modulation vectors:

$$\overrightarrow{F}\_f(f,t) = \frac{\partial \overrightarrow{F}}{\partial f}, \qquad \overrightarrow{F}\_t(f,t) = \frac{\partial \overrightarrow{F}}{\partial t},$$
$$\overrightarrow{G}\_f(f,t) = \frac{\partial \overrightarrow{G}}{\partial f}, \qquad \overrightarrow{G}\_t(f,t) = \frac{\partial \overrightarrow{G}}{\partial t}.$$

Differential motion in $\overrightarrow{K}$-space:

$$d\overrightarrow{K} = \overrightarrow{F}\_f\, df + \overrightarrow{F}\_t\, dt \quad \text{(constant)},$$
$$d\overrightarrow{K} = \overrightarrow{G}\_f\, df + \overrightarrow{G}\_t\, dt \quad \text{(agile)}.$$

Skew and Orthogonality

- With fixed per-pulse frequency parameters, highly squinted monostatic and most bistatic geometries yield **skewed** passbands.
- Frequency agility can be designed such that $\left\langle \vec{F}\_f, \vec{F}\_t \right\rangle = 0$ (or $\left\langle \vec{G}\_f, \vec{G}\_t \right\rangle = 0$) *for all* (f, t), producing non-skewed coverage and orthogonal sidelobe axes.

Spatially Variant PSF

SAR image formation is expressed as a non-stationary spatial integral:

$$\tilde{\rho}(\vec{r}) = \int_{\vec{r}'} \rho(\vec{r}')\, psf(\vec{r}', \vec{r})\, |d\vec{r}'|.$$

Point Spread Function (PSF): Neglecting amplitude terms,

$$psf(\vec{r}', \vec{r}) = \frac{\displaystyle\int_{t_{\min}}^{t_{\max}} \int_{f_{\min}(t)}^{f_{\max}(t)} \exp\left[j\, \frac{2\pi f}{c} \left(R(t, \vec{r}) - R(t, \vec{r}')\right)\right] df\, dt}{\displaystyle\int_{t_{\min}}^{t_{\max}} BW(t)\, dt}$$

Interpretation:

- *Waveform frequency diversity* is captured by integrating over frequency $f \in [f_{\min}(t), f_{\max}(t)]$, where $BW(t) = f_{\max}(t) - f_{\min}(t)$ may vary with slow-time.
- *Geometric diversity* arises from the time-varying difference in bistatic range signatures $R(t, \vec{r}) - R(t, \vec{r}')$ over the aperture interval $t \in [t_{\min}, t_{\max}]$.
- Together, these diversity sources shape the spatial extent, sidelobe structure, and anisotropy of the PSF.

Impulse Response Approximation

Localized approximation of the PSF around the origin:

$$psf\left(\overrightarrow{O} - \frac{\overrightarrow{r}}{2},\ \overrightarrow{O} + \frac{\overrightarrow{r}}{2}\right) \approx ipr(\overrightarrow{r})$$

$$ipr(\overrightarrow{r}) = \frac{\displaystyle\int_{t_{\min}}^{t_{\max}} \int_{f_{\min}(t)}^{f_{\max}(t)} \exp\left[j \left\langle \overrightarrow{r}, \overrightarrow{F}(f,t) \right\rangle\right] df\, dt}{\displaystyle\int_{t_{\min}}^{t_{\max}} BW(t)\, dt} \qquad \overrightarrow{F}(f,t) = \frac{2\pi f}{c} \overrightarrow{\nabla R}(t)$$

Interpretation:

- *Waveform frequency diversity* is described by integrating over $f \in [f_{\min}(t), f_{\max}(t)]$, where $BW(t)$ may vary with slow-time t.
- *Geometric diversity* is captured by slow-time variation in the length and direction of the bistatic range gradient $\overrightarrow{\nabla R}(t)$, which modulates $\overrightarrow{F}(f,t)$ and shapes the impulse response.
- Together, frequency and geometric diversity determine the IPR mainlobe width, sidelobe structure, and anisotropy.

Transfer Function / Passband

The transfer function (TF) is the spatial Fourier transform of the impulse response:

$$TF(\overrightarrow{K}) = \mathcal{F}_{\overrightarrow{r}}\left[\, ipr(\overrightarrow{r})\,\right] = \int_{t_{\min}}^{t_{\max}} \int_{f_{\min}(t)}^{f_{\max}(t)} \delta\Big(\overrightarrow{K} - \overrightarrow{F}(f,t)\Big)\, df\, dt$$

Interpretation:

- The set of instantaneous spatial frequency positions $\{\overrightarrow{F}(f,t)\}$ (or $\{\overrightarrow{G}(f,t)\}$) traces a surface in $\overrightarrow{K}$-space.
- The transfer function is the energy distribution along this surface — the SAR system's spatial frequency response.
- Density depends on *geometric diversity* through the variation of $\overrightarrow{\nabla R}(t)$ and on *frequency diversity* through $f \in [f_{\min}(t), f_{\max}(t)]$.

Frequency Limits

$$\textbf{Constant:} \quad f\_\mathrm{min}(t) = f\_\mathrm{min},\ f\_\mathrm{max}(t) = f\_\mathrm{max}.$$

$$\textbf{Agile:} \quad f\_\mathrm{min}(t) = \frac{C\_\mathrm{min}}{\|\overrightarrow{\nabla R}(t) \times \overrightarrow{\hat{z}}\|},\ f\_\mathrm{max}(t) = \frac{C\_\mathrm{max}}{\|\overrightarrow{\nabla R}(t) \times \overrightarrow{\hat{z}}\|}.$$

Design goal: make $\overrightarrow{F}\_f$ and $\overrightarrow{F}\_t$ (or $\overrightarrow{G}\_f$ and $\overrightarrow{G}\_t$) orthogonal over (f, t) to reduce passband skew and align sidelobe axes.

Batch Wrapper: `AmbiguityGenBatch.m`

- Starting point: lets the user define **geometry**, **frequency**, **scene**, and **desired resolution** parameters.
- Supports running **sets of parameters**; a `for`-loop iterates each combination in the set.
- For each configuration, packages inputs and calls `Sim_Main.m`.

Core Orchestrator: `Sim_Main.m`

- Generates TX/RX trajectories and slow-time grid; constructs **K-space sample points**.
- Manages device memory for per-kernel inputs and dispatch order.
- Calls CUDA sequence: PSF $\rightarrow$ IPR $\rightarrow$ Phase History $\rightarrow$ Matched-Filter image.

CUDA PSF: `PSF_Kernel.cu`

Purpose: Simulate spatially variant PSF by integrating phase over (t, f) for each PSF pixel.

- Inputs to GPU: platform position timelines (TX/RX), frequency signatures, PSF pixel coordinates.
- Kernel: accumulates complex phase terms and writes PSF tiles / full grid.

CUDA IPR: `IPR_Kernel.cu`

Purpose: Simulate localized impulse response (IPR).

- Inputs to GPU: bistatic range gradient samples $\overrightarrow{\nabla R}(t)$, frequency signatures, IPR pixel coordinates.
- Kernel: accumulates complex exponentials across (t, f) for each IPR pixel.

CUDA Phase History: `PhaseHist_Kernel.cu`

Purpose: Generate raw phase history from the scene and platform timelines.

- Inputs to GPU: target positions/reflectivities, platform positions, frequency signatures, slow-time sampling.
- Kernel: computes bistatic ranges to each target and accumulates complex samples per (t, f).

CUDA Matched Filter Image: `MF_Kernel.cu`

Purpose: Form the SAR image using the simulated phase history.

- Inputs to GPU: phase history (already on device), platform positions, image pixel grid, frequency signatures.
- Kernel: matched filter / back-projection accumulation per pixel over (t, f); outputs complex image.

What is Simulated

1. Sparse scene: a few point targets near scene center.
2. TX/RX flight paths and velocity vectors.
3. LOS vectors and time-derivatives for both platforms.
4. $\overrightarrow{\nabla R}(t)$ and $\overrightarrow{\nabla \dot{R}}(t)$.
5. Two stepped-frequency waveform sets: constant vs agile.
6. $\overrightarrow{K}$-space surfaces $\overrightarrow{F}(f, t)$ and $\overrightarrow{G}(f, t)$.
7. Modulation vectors $\overrightarrow{F}\_f$, $\overrightarrow{F}\_t$ and $\overrightarrow{G}\_f$, $\overrightarrow{G}\_t$.
8. Spatially variant PSF and localized IPR.
9. Phase history from point targets.
10. Matched-filter (back-projection) image formation.

Matlab + CUDA Implementation (High Level)

Matlab

- Geometry generation, trajectories, LOS/$\dot{\text{LOS}}$.
- Waveform scheduling and stepped-frequency grids.
- Phase-history synthesis from point targets.
- IPR/PSF/TF integrals (CPU vectorized where feasible).
- Orchestration of GPU kernels and data movement.

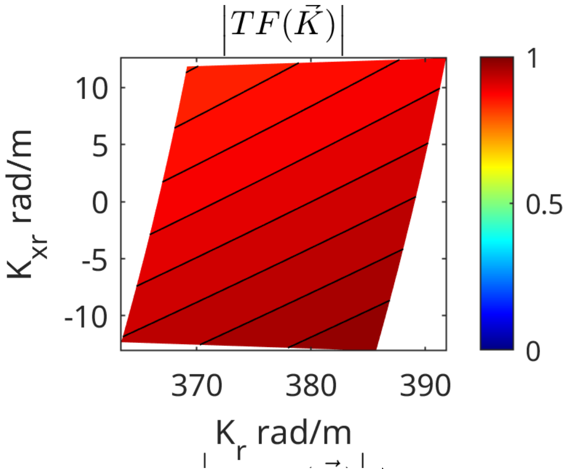
CUDA Kernels

- Range-phase accumulation per scatterer per (f, t) tile.
- Back-projection / matched filtering accumulator.
- Optional K-space density estimation for passband visualization.

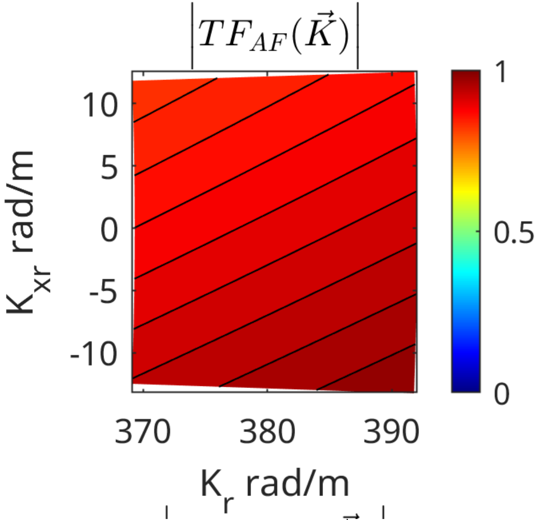
CUDA Back-Projection Sketch

```
// Kernel signature (illustrative)
__global__ void backproject(const float* __restrict__ ph_real,
                            const float* __restrict__ ph_imag,
                            const float* __restrict__ posTX,
                            const float* __restrict__ posRX,
                            const float* __restrict__ gridX,
                            const float* __restrict__ gridY,
                            cuFloatComplex* __restrict__ img,
                            int Npix, int Nt, int Nf) {
  int p = blockIdx.x * blockDim.x + threadIdx.x;
  if (p >= Npix) return;
  cuFloatComplex acc = make_cuFloatComplex(0.f, 0.f);
  // loop over (t,f) tiles
  for (int it=0; it<Nt; ++it) {
    for (int k=0; k<Nf; ++k) {
      // compute bistatic range for pixel p, time it
      // fetch phase history sample and accumulate
    }
  }
  img[p] = acc;
}
```

Passband Surfaces (Placeholders)

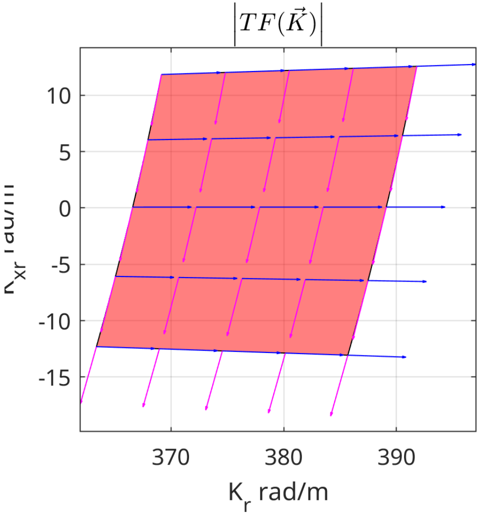


Constant parameters $\vec{F}(f,t)$

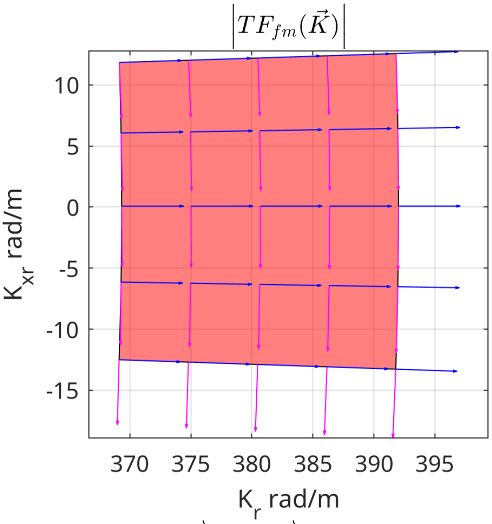


Agile parameters $\vec{G}(f,t)$

Modulation Vectors and Orthogonality (Placeholders)

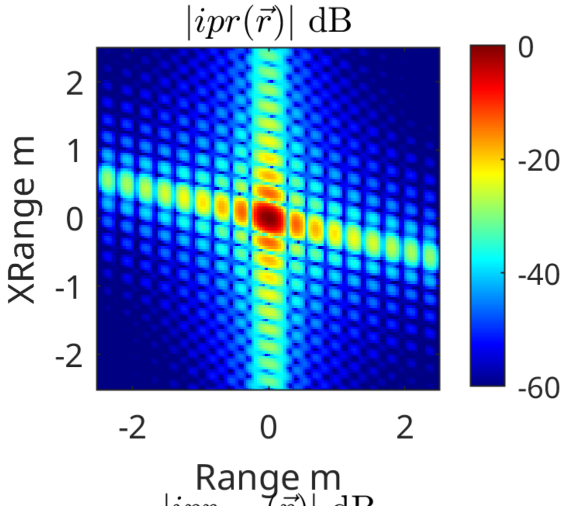


$\overrightarrow{F}\_f$ vs $\overrightarrow{F}\_t$

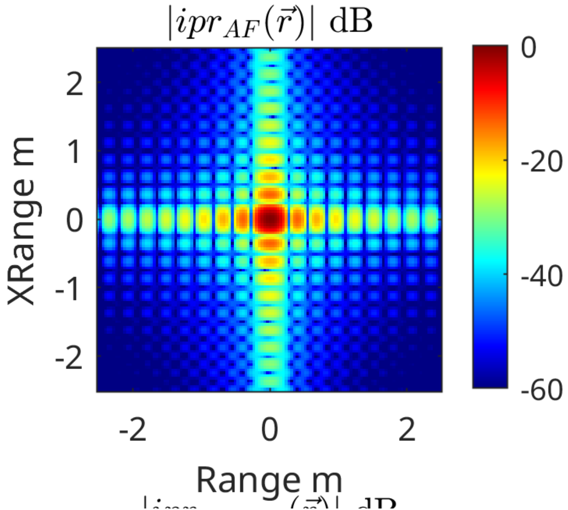


$\overrightarrow{G}\_f$ vs $\overrightarrow{G}\_t$

IPR/PSF Comparisons (Placeholders)

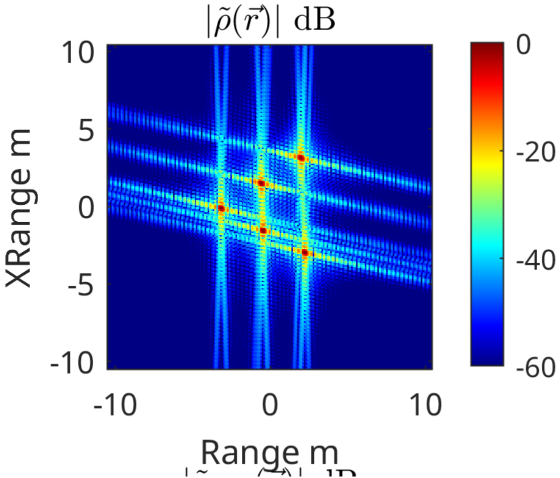


IPR (constant)

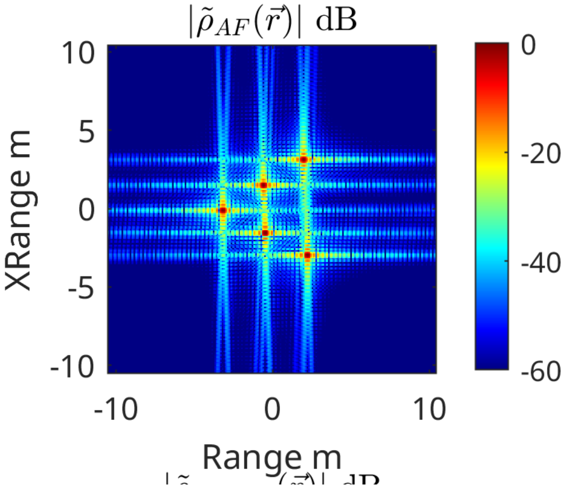


IPR (agile)

Matched-Filter Images (Placeholders)



MF image (constant)



MF image (agile)

Representative Geometries

- Forward-looking monostatic (log-spiral, constant squint/elevation).
- Highly squinted monostatic (straight pass, large yaw).
- Bistatic close-proximity (TX/RX near scene, non-colocated).

Each case run with constant and agile waveform sets for contrast.

Example Parameters (Edit as Needed)

Center frequency	1.0000×10^{10} Hz
Bandwidth (const)	5.9567×10^{8} Hz
Pulses per aperture	256
TX range	20 km
RX range	8 km
Platform speed (TX/RX)	100 m/s
Scene size	$20\,\mathrm{m} \times 20\,\mathrm{m}$
Agile constants	$C_{\mathrm{min}},\ C_{\mathrm{max}}$ (user-defined)

Current Approximations

- Move–stop–move (no Doppler time-scaling during each tone).
- Wideband modeled as stepped narrowband tones.

Planned updates:

- High-speed platform corrections (time-scaling, higher-order kinematics).
- Continuous wideband models and modern waveform families.
- Rigorous PSF stationarity analysis and compensation.

Summary

- Unified pipeline links geometry, waveform agility, and $\overrightarrow{K}$-space passband structure.
- Orthogonal modulation-vector design reduces passband skew and aligns sidelobe axes.
- CUDA-accelerated back-projection enables tractable exploration of non-traditional geometries.

Notation

$\overrightarrow{Pos}$_TX/RX(t)	Platform position vectors
$\overrightarrow{Vel}$_TX/RX(t)	Platform velocity vectors
$\overrightarrow{LOS}$_TX/RX(t)	Unit LOS vectors to scene center
$\overrightarrow{\dot{LOS}}$_TX/RX($t$)	Slow-time derivatives of LOS
$\overrightarrow{\nabla R}(t)$	Bistatic range gradient $=$ LOS _TX $+$ LOS _RX
$\overrightarrow{\nabla \dot{R}}(t)$	Bistatic range-rate gradient $=$ $\dot{\text{L}}$OS_TX $+$ $\dot{\text{L}}$OS_RX
$\overrightarrow{F}(f,t)$	$\frac{2\pi f}{c}\,\overrightarrow{\nabla R}(t)$ (constant waveform case)
$\overrightarrow{G}(f,t)$	Agile surface (time-varying freq. limits)
$\overrightarrow{F}\_f$, $\overrightarrow{F}\_t$	$\partial\overrightarrow{F}/\partial f$, $\partial\overrightarrow{F}/\partial t$
$\overrightarrow{G}\_f$, $\overrightarrow{G}\_t$	$\partial\overrightarrow{G}/\partial f$, $\partial\overrightarrow{G}/\partial t$

References

See *Key Publications* slide for full citations. Add additional related works as needed.