Computationally Efficient Joint-Domain Clutter Cancellation for Waveform-Agile Radar

Christian Jones[1,3], Brandon Ravenscroft[1], James Vogel[2], Suzanne M. Shontz[1], Thomas Higgins[3], Kevin Wagner[3], Shannon Blunt[1]

[1]Radar Systems Lab (RSL), EECS Dept., University of Kansas, Lawrence, KS, USA

[2]Systems & Technology Research, Woburn, MA, USA

[3]Naval Research Laboratory - Radar Division, Washington, DC, USA

This work was supported by the Office of Naval Research under Contract #N00014-20-C-1006.

Motivation

- Standard MTI radar involves illumination by a **repeated waveform**, such as the commonly used linear frequency modulated (LFM) chirp
  - Range and slow-time Doppler domains are decoupled, permitting sequential receive processing

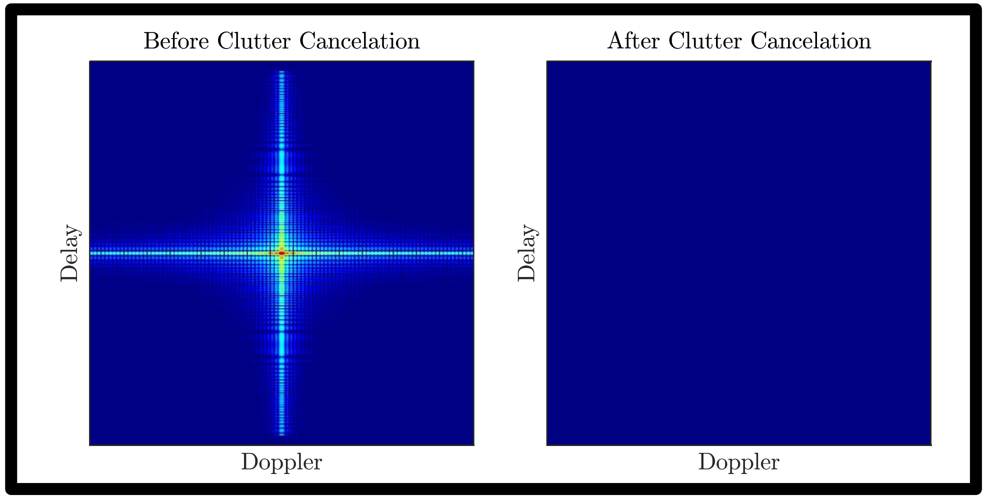
- In contrast, a CPI of **nonrepeating** waveforms provides a **multiplicative increase** in design degrees-of-freedom (DoF)
  - enables new **waveform-agile** sensing modes such as dynamic spectral notching

- The DoF increase is due to coupling of range and slow-time Doppler domains … which introduces the phenomenon of **range sidelobe modulation (RSM)**

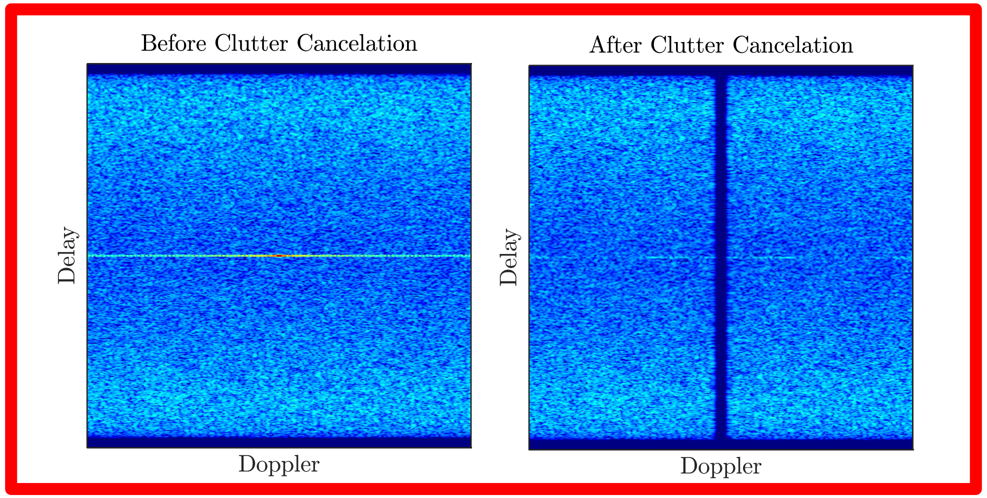
Range Sidelobe Modulation

- When performing sequential range/Doppler processing on agile waveforms, RSM yields a **smearing of clutter across the entire Doppler domain**, hindering clutter cancellation

Repeated (standard)



Waveform-Agile



- Simply put, the greater DoF for waveform-agile operation comes at the cost of requiring some manner of **compensation for clutter cancellation**

Waveform-Agile Clutter Cancellation

- Initial approaches sought to compensate for RSM by **"homogenizing" pulse compression responses** on a pulse-to-pulse basis … but the trade-off is higher sidelobes
- Non-Identical Multiple Pulse Compression (NIMPC) [1] was developed to perform **joint delay-Doppler clutter cancellation** by forming a structured (clutter + noise) covariance matrix
  - But joint domain processing can be **prohibitively expensive** => simply the curse of dimensionality
- However, it was recently shown [2] that NIMPC can be refactored to reveal a **block-Toeplitz structure**, enabling efficient solvers

Here we leverage this block-Toeplitz structure and a projection formulation (akin to [3]) to achieve **multiple orders-of-magnitude reduction in computational cost** … potentially reaching real-time

[1] T. Higgins, et al., "Aspects of non-identical multiple pulse compression," *IEEE Radar Conf*., Kansas City, MO, May 2011.

[2] C. Sahin, et al., "Reduced complexity maximum SINR receiver processing for transmit-encoded radar-embedded communications," *IEEE Radar Conf*., Oklahoma City, OK, May 2018.

[3] F. Colone, et al., "A multistage processing algorithm for disturbance removal and target detection in passive bistatic radar," *IEEE Trans. Aerospace & Electronic Systems*, vol. 45, no. 2, pp. 698-722, Apr. 2009.

… and now for the math …

Single-Pulse Snapshot Model

- Denote an $N\times 1$ collection of contiguous fast-time received samples induced by the mth pulse as

$$\mathbf{y}_m(l) = \sum_{\omega} \mathbf{S}_{m,\omega}\,\mathbf{x}_{\omega}(l) + \mathbf{n}_m(l)$$

where l indicates the range cell index and $\mathbf{n}_m(l)$ contains the N associated noise samples

- The $N\times(2N-1)$ matrix $\mathbf{S}_{m,\omega} = \mathbf{S}_m e^{j(m-1)\omega}$ accounts for the phase shift from Doppler frequency ω
- The Toeplitz structure of

$$\mathbf{S}_m = \begin{bmatrix} s_{m,N} & \cdots & \cdots & s_{m,1} & 0 & \cdots & 0 \\ 0 & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & & \ddots & 0 \\ 0 & \cdots & 0 & s_{m,N} & \cdots & \cdots & s_{m,1} \end{bmatrix}$$

models convolution of N discretized samples of waveform $s_m(t)$, denoted as $\mathbf{s}_m = [s_{m,1}\ s_{m,2}\ \cdots\ s_{m,N}]^T$, with $(2N-1)$ discretized range samples of complex scattering in $\mathbf{x}_{\omega}(l)$

Multiple-Pulse Model

- Extending to M pulses, the previous $N\times1$ receive vector becomes the $NM\times1$ vector

$$\bar{\mathbf{y}}(l)=[\mathbf{y}_1^T(l)\;\;\mathbf{y}_2^T(l)\;\;\cdots\;\;\mathbf{y}_M^T(l)]^T=\sum_{\omega}\bar{\mathbf{S}}_{\omega}\,\mathbf{x}_{\omega}(l)+\bar{\mathbf{n}}(l)$$

where $\bar{\mathbf{n}}(l)$ is the concatenation of the M noise vectors $\mathbf{n}_m(l)$ and

$$\bar{\mathbf{S}}_{\omega}=[\mathbf{S}_{1,\omega}^T\;\;\mathbf{S}_{2,\omega}^T\;\;\cdots\;\;\mathbf{S}_{M,\omega}^T]^T$$

is the $NM\times(2N-1)$ block-Toeplitz matrix of range & Doppler shifts for each scattering vector $\mathbf{x}_{\omega}(l)$

- Thus the joint-domain matched filter for Doppler ω can be defined as

$$\bar{\mathbf{b}}_{\omega}=[\mathbf{s}_1^T\;\;e^{j\omega}\mathbf{s}_2^T\;\;\cdots\;\;e^{j(M-1)\omega}\mathbf{s}_M^T]^T=\begin{bmatrix}\mathbf{s}_1 & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{s}_2 & \ddots & \vdots\\ \vdots & \ddots & \ddots & \mathbf{0}\\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{s}_M\end{bmatrix}\begin{bmatrix}1\\ e^{j\omega}\\ \vdots\\ e^{j(M-1)\omega}\end{bmatrix}=\mathbf{C}\mathbf{v}_{\omega}$$

$M\times1$ Doppler steering vector for frequency ω

NIMPC Filters

- NIMPC uses this joint model to specify a bank of K clutter Doppler frequencies as

$$\bar{\mathbf{S}}_{\Omega} = [\bar{\mathbf{S}}_{\omega_1} \ \ \bar{\mathbf{S}}_{\omega_2} \ \ \cdots \ \ \bar{\mathbf{S}}_{\omega_K}]$$

from which is formed the structured (clutter + noise) correlation matrix

$$\bar{\mathbf{R}} = \bar{\mathbf{S}}_{\Omega} \bar{\mathbf{S}}_{\Omega}^{H} + \bar{\mathbf{R}}_{\text{nse}}$$

that is used to construct the NIMPC filter for frequency ω as

$$\bar{\mathbf{w}}_{\omega} = \bar{\mathbf{R}}^{-1} \bar{\mathbf{b}}_{\omega} = \bar{\mathbf{R}}^{-1} \mathbf{C} \mathbf{v}_{\omega}$$

- Therefore, the matched filter and NIMPC filter for frequency ω are respectively applied as

$$\hat{x}(l,\omega) = \frac{\bar{\mathbf{b}}_{\omega}^{H} \bar{\mathbf{y}}(l)}{\bar{\mathbf{b}}_{\omega}^{H} \bar{\mathbf{b}}_{\omega}} \qquad\qquad \hat{x}(l,\omega) = \bar{\mathbf{w}}_{\omega}^{H} \bar{\mathbf{y}}(l)$$

Reformulating NIMPC

- Solving for NIMPC filters directly involves construction and inversion of an $NM \times NM$ matrix

- Even when using efficient block-Toeplitz solvers, cost is prohibitive even for modest N and M

- Rather than solving directly, <u>the problem can be reformulated as the solution to *M* linear systems of equations</u>

$$\bar{\mathbf{w}}_{\omega} = \bar{\mathbf{R}}^{-1}\bar{\mathbf{b}}_{\omega} = \bar{\mathbf{R}}^{-1}\mathbf{C}\mathbf{v}_{\omega}$$

$$\downarrow$$

$$\bar{\mathbf{R}}\bar{\mathbf{w}}_{\omega} = \bar{\mathbf{R}}\,\bar{\mathbf{D}}\,\mathbf{v}_{\omega} = \mathbf{C}\mathbf{v}_{\omega}$$

$$\downarrow$$

$$\bar{\mathbf{R}}\,\bar{\mathbf{D}} = \bar{\mathbf{R}}[\bar{\mathbf{d}}_1 \quad \bar{\mathbf{d}}_2 \quad \cdots \quad \bar{\mathbf{d}}_M] = \mathbf{C} = [\mathbf{c}_1 \quad \mathbf{c}_2 \quad \cdots \quad \mathbf{c}_M]$$

Preconditioned Conjugate Gradient

- Separating each of the M systems of equations, the matrix inverse can be posed as an equivalent convex optimization problem
- I iterations of **linear conjugate gradient** (CG) or **preconditioned linear conjugate gradient** (PCG) can then be performed to iteratively converge to solution of the system

Equivalent Objective Function

Objective: $f(\bar{\mathbf{d}}_m) = \bar{\mathbf{d}}_m^H \bar{\mathbf{R}} \bar{\mathbf{d}}_m - \bar{\mathbf{d}}_m^H \mathbf{c}_m - \mathbf{c}_m^H \bar{\mathbf{d}}_m$

Gradient: $\nabla_{\bar{\mathbf{d}}_m^*} f(\bar{\mathbf{d}}_m) = \bar{\mathbf{R}} \bar{\mathbf{d}}_m - \mathbf{c}_m$

Hessian: $\nabla^2 f(\bar{\mathbf{d}}_m) = \bar{\mathbf{R}}$

Initialization

Preconditioner: $\bar{\mathbf{M}}$

$\bar{\mathbf{d}}_{m,0} = \mathbf{0}$ or $\bar{\mathbf{d}}_{m,0} = \bar{\mathbf{M}}^{-1} \mathbf{c}_m$

$$\mathbf{r}_0 = \mathbf{c}_m - \bar{\mathbf{R}} \bar{\mathbf{d}}_{m,0}$$
$$\mathbf{u}_0 = \bar{\mathbf{M}}^{-1} \mathbf{r}_0$$
$$\mathbf{p}_0 = \mathbf{u}_0$$
$$i = 0$$

Descent Routine

while $i < I$ and $\|\mathbf{r}_i\| > \phi$

$$\alpha_i = \frac{\mathbf{r}_i^H \mathbf{u}_i}{\mathbf{p}_i^H \bar{\mathbf{R}} \mathbf{p}_i}$$
$$\bar{\mathbf{d}}_{m,i+1} = \bar{\mathbf{d}}_{m,i} + \alpha_i \mathbf{p}_i$$
$$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \bar{\mathbf{R}} \mathbf{p}_i$$
$$\mathbf{u}_{i+1} = \bar{\mathbf{M}}^{-1} \mathbf{r}_{i+1}$$
$$\beta_i = \frac{\mathbf{r}_{i+1}^H \mathbf{u}_{i+1}}{\mathbf{r}_i^H \mathbf{u}_i}$$
$$\mathbf{p}_{i+1} = \mathbf{u}_{i+1} + \beta_i \mathbf{p}_i$$
$$i = i + 1$$

end

Efficient Matrix Operations

- Using the block-Toeplitz structure of $\bar{\mathbf{S}}_\Omega$, storage of $\bar{\mathbf{R}}$ can be avoided by performing Toeplitz matrix multiplies (**<u>equivalent to convolution</u>**) as frequency-domain multiplication

$$\begin{aligned}\bar{\mathbf{R}}\,\mathbf{f} &= (\bar{\mathbf{S}}_\Omega \bar{\mathbf{S}}_\Omega^H + \bar{\mathbf{R}}_{\text{nse}})\mathbf{f} \\ &= \bar{\mathbf{S}}_\Omega \bar{\mathbf{S}}_\Omega^H \mathbf{f} + \bar{\mathbf{R}}_{\text{nse}}\,\mathbf{f} \\ &= \bar{\mathbf{S}}_\Omega \mathbf{g} + \bar{\mathbf{R}}_{\text{nse}}\,\mathbf{f} \\ &= \mathbf{h} + \sigma_{\text{nse}}^2\,\mathbf{f}\,,\end{aligned}$$

$$\mathbf{g}_k = \left[\sum_{m=1}^{M} \mathbf{A}^H\left(e^{-j(m-1)\omega_k}(\mathbf{q})^* \odot (\mathbf{A}\tilde{\mathbf{s}}_m)^* \odot (\mathbf{A}\tilde{\mathbf{f}}_m)\right)\right]$$

$$\mathbf{g} = \begin{bmatrix}\mathbf{g}_1^T & \mathbf{g}_2^T & \cdots & \mathbf{g}_K^T\end{bmatrix}^T = \bar{\mathbf{S}}_\Omega^H \mathbf{f}$$

$$\mathbf{h}_m = \tilde{\mathbf{I}}_{N\times(2N-1)}\left[\sum_{k=1}^{K} \mathbf{A}^H\left(e^{j(m-1)\omega_k}(\mathbf{q}) \odot (\mathbf{A}\tilde{\mathbf{s}}_m) \odot (\mathbf{A}\mathbf{g}_k)\right)\right]$$

$$\mathbf{h} = \begin{bmatrix}\mathbf{h}_1^T & \mathbf{h}_2^T & \cdots & \mathbf{h}_M^T\end{bmatrix}^T = \bar{\mathbf{S}}_\Omega \mathbf{g}\,,$$

$\mathbf{A}$: $(2N-1)\times(2N-1)$ DFT matrix

$\tilde{\mathbf{k}}$: append $(N\text{-}1)$ zeros to the end of arbitrary vector $\mathbf{k}$

$\tilde{\mathbf{I}}_{N\times(2N-1)} = [\mathbf{I}_{N\times N}\ \ \mathbf{0}_{N\times(N-1)}]$: Remove zero pad

- The complexity of the matrix inverse goes from $O(N^3M^3)$ to $O(2\,IMK\,(2N-1)\log_2(2N-1))$ for I iterations and no preconditioner
- While CG and PCG converge superlinearly, iterating for fixed I does not guarantee the exact solution … <u>but the iterative nature provides a trade-space between accuracy and computation time</u>

A Note on Preconditioning

- Optimal preconditioning can greatly **improve rate of convergence**, but likewise increases per-iteration computational cost

- Rather than an optimal preconditioner, here it is more practical to select a preconditioner that **improves rate of convergence without significantly altering complexity** ($\sim M(2N-1)\log_2(2N-1)$)

- A block-circulant preconditioner that approximates $\bar{\mathbf{R}}$ has been shown to improve the rate of convergence, but **increases per-iteration cost** by $2M^2N\log_2(N)$ => factor of M greater than CG

- Alternatively, if clutter is localized in Doppler, a **block-circulant preconditioner that approximates the diagonal blocks** of $\bar{\mathbf{R}}$ improves convergence (to a lesser degree) while only increasing per-iteration cost by $2MN\log_2(N)$ => strictly less than the complexity of CG

An "All-of-Range" Collective Model

- Where the previous model operated on $N\times 1$ range-indexed snapshots, the entire $L\times 1$ receive interval (in range) can be employed to form a projection onto the nullspace of the clutter
- The $L\times 1$ measurement vector induced by the mth pulse is denoted in a similar form as

$$\dot{\mathbf{y}}_m = \sum_{\omega} \dot{\mathbf{S}}_{m,\omega}\, \dot{\mathbf{x}}_{\omega} + \dot{\mathbf{n}}_m \qquad \dot{\mathbf{S}}_{m,\omega} = e^{j\omega(m-1)} \begin{bmatrix} s_{m,1} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & & & & \vdots \\ s_{m,N} & & s_{m,1} & & & \vdots \\ 0 & \ddots & \vdots & \ddots & & \vdots \\ \vdots & & s_{m,N} & & \ddots & 0 \\ \vdots & & & \ddots & & s_{m,1} \\ \vdots & & & & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & s_{m,N} \end{bmatrix}$$

$L\times(L-N+1)$ banded Toeplitz matrix

which is likewise extended to M pulses via the $ML\times 1$ aggregate receive vector

$$\ddot{\mathbf{y}} = \sum_{\omega} \ddot{\mathbf{S}}_{\omega}\, \dot{\mathbf{x}}_{\omega} + \ddot{\mathbf{n}} \qquad \ddot{\mathbf{S}}_{\omega} = [\dot{\mathbf{S}}_{1,\omega}^T \;\; \dot{\mathbf{S}}_{2,\omega}^T \;\; \cdots \;\; \dot{\mathbf{S}}_{M,\omega}^T]^T$$

$ML\times(L-N+1)$ block-Toeplitz matrix

- This representation increases the dimensionality of each Toeplitz matrix, <u>thereby improving FFT efficiency and accuracy of the circulant approximation</u>

Projection NIMPC (Proj-NIMPC)

- This collective model is then used to define the bank of K Doppler frequencies and subsequent projection

$$\ddot{\mathbf{S}}_{\Omega} = [\ddot{\mathbf{S}}_{\omega_1} \;\; \ddot{\mathbf{S}}_{\omega_2} \;\; \cdots \;\; \ddot{\mathbf{S}}_{\omega_K}] \qquad\qquad \ddot{\mathbf{P}} = \ddot{\mathbf{I}} - \ddot{\mathbf{S}}_{\Omega}(\ddot{\mathbf{S}}_{\Omega}^{H}\ddot{\mathbf{S}}_{\Omega})^{-1}\ddot{\mathbf{S}}_{\Omega}^{H}$$

which projects onto the null space of the clutter in a manner akin to the extensive cancellation algorithm (ECA) [3], with matched filtering applied thereafter

- As before, each operation in the projection can be <u>implemented efficiently via frequency domain multiplication and PCG:</u>

$$\begin{aligned}\ddot{\mathbf{z}} = \ddot{\mathbf{P}}\,\ddot{\mathbf{y}} &= (\ddot{\mathbf{I}} - \ddot{\mathbf{S}}_{\Omega}(\ddot{\mathbf{S}}_{\Omega}^{H}\ddot{\mathbf{S}}_{\Omega})^{-1}\ddot{\mathbf{S}}_{\Omega}^{H})\,\ddot{\mathbf{y}} \\ &= \ddot{\mathbf{y}} - \ddot{\mathbf{S}}_{\Omega}(\ddot{\mathbf{S}}_{\Omega}^{H}\ddot{\mathbf{S}}_{\Omega})^{-1}\ddot{\mathbf{S}}_{\Omega}^{H}\,\ddot{\mathbf{y}} \\ &= \ddot{\mathbf{y}} - \ddot{\mathbf{S}}_{\Omega}(\ddot{\mathbf{S}}_{\Omega}^{H}\ddot{\mathbf{S}}_{\Omega})^{-1}\ddot{\mathbf{g}} \\ &= \ddot{\mathbf{y}} - \ddot{\mathbf{S}}_{\Omega}\ddot{\mathbf{a}} \\ &= \ddot{\mathbf{y}} - \ddot{\mathbf{h}}\end{aligned}$$

Freq. Multiplication

$$\ddot{\mathbf{g}} = \ddot{\mathbf{S}}_{\Omega}^{H}\,\ddot{\mathbf{y}} \qquad O(2\,KML\,\log_2(L))$$

$$\ddot{\mathbf{h}} = \ddot{\mathbf{S}}_{\Omega}\ddot{\mathbf{a}} \qquad O(2\,KML\,\log_2(L))$$

PCG for I iterations

$$\ddot{\mathbf{a}} = (\ddot{\mathbf{S}}_{\Omega}^{H}\ddot{\mathbf{S}}_{\Omega})^{-1}\ddot{\mathbf{g}}$$

$$(\ddot{\mathbf{S}}_{\Omega}^{H}\ddot{\mathbf{S}}_{\Omega})\ddot{\mathbf{a}} = \ddot{\mathbf{g}}$$

$$O(4\,(2\,I+1)\,KML\,\log_2(L))$$

Computational Complexity

- Computational cost for direct NIMPC and the block-diagonal PCG versions were determined to get a sense of overall complexity

- In addition to substantial efficiency gains as N and M grow, the PCG implementations avoid the need to store large matrices

- Proj-NIMPC scales linearly with M while all the NIMPC approaches scale by at least M^2

- The larger Toeplitz matrices in Proj-NIMPC yield a more accurate circulant approximation

Order of Complex Operations	
NIMPC (direct)	$K(2N-1)\,(MN)^2 + (MN)^3 + D\,(MN)^2$
NIMPC (direct w. efficient solver)	$K(2N-1)\,(MN)^2 + M(MN)^2 + (MN)^2$
NIMPC PCG	$2IM^2K(2N-1)\log_2(2N-1) + 2IM^2N\log_2(N)$
Proj-NIMPC (direct w. efficient solver)	$K^3(L-N+1)^2 + 2K^2M(L-N+1)\,L\log_2(L)$
Proj-NIMPC PCG	$4(2I+1)\,KML\log_2(L)$

Variables Definitions	
N	Samples per Waveform
M	Pulses per CPI
K	# Doppler Frequencies to Suppress
D	# Doppler Frequencies to Estimate (direct NIMPC only)
L	Samples per PRI
I	Conjugate Gradient Iterations

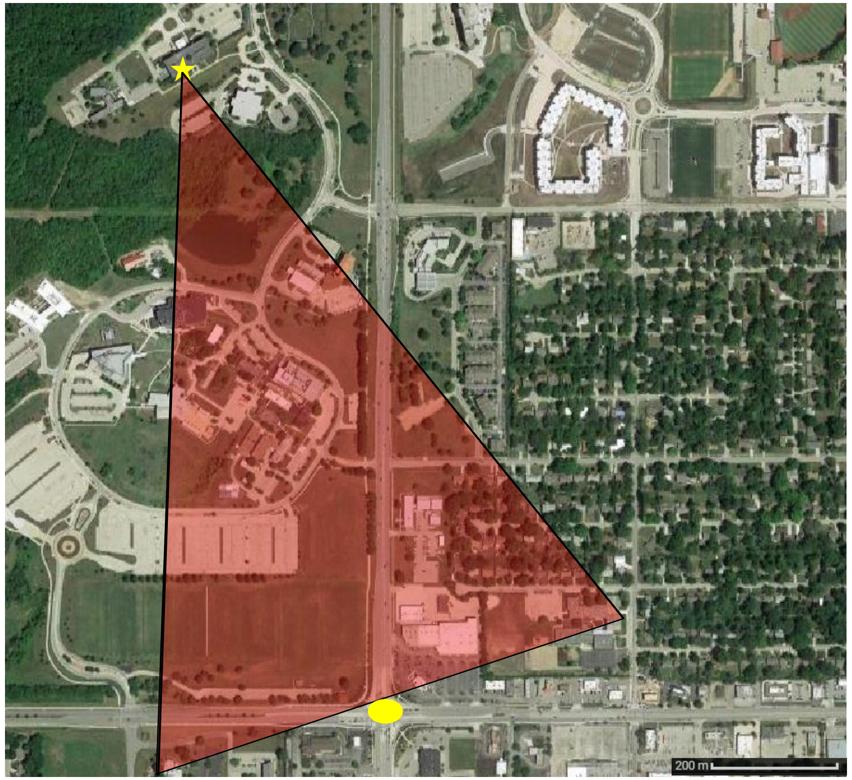
Now consider an application example with open-air measurements that assigns values to these parameters for a more meaningful comparison of efficiency improvement

Experimental Setup

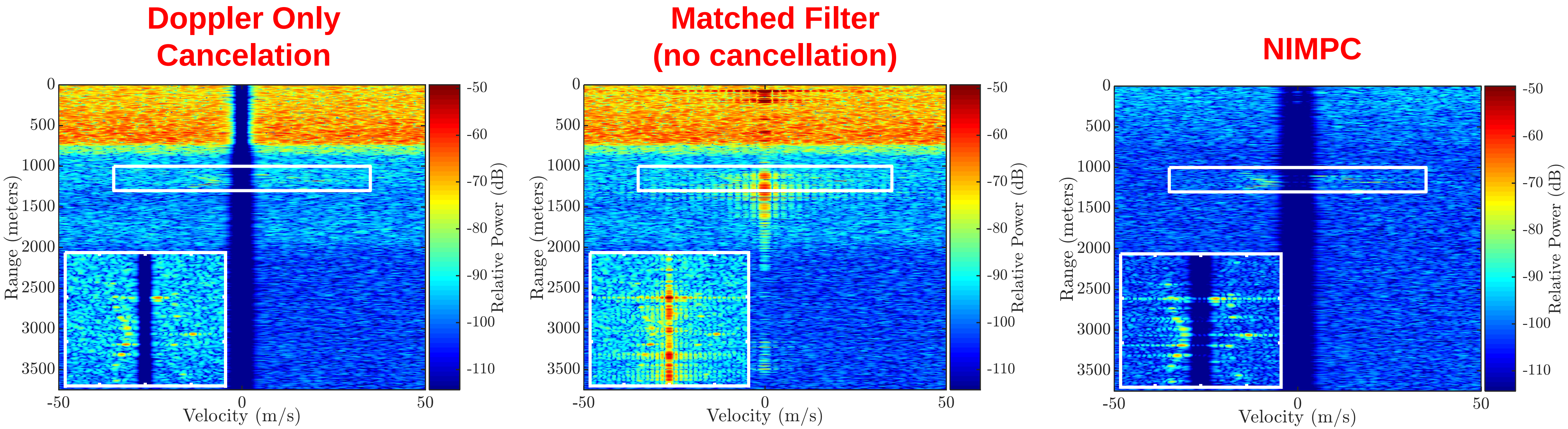
- Open-air measurements were collected with an S-band radar testbed to assess the different NIMPC joint-domain implementations on real MTI data
  - 3.55 GHz center frequency, 33.3 MHz 3-dB bandwidth, 4.5 μs pulses, 5 kHz PRF
  - Separate (but collocated) Tx and Rx antennas produce significant direct-path leakage

- M =100 pulsed, random FM waveforms (i.e. no repeat during the CPI) illuminated a traffic intersection in Lawrence, KS (from roof of Nichols Hall on KU campus)
  - Receive sample rate yielded $N = 900$ and $L = 5.9\times10^3$ range cells
  - NIMPC clutter cancellation uses K=10 bins equally spaced over ±150 Hz in Doppler
  - NIMPC filters diagonal loaded with $\sigma^2_{\text{nse}} = 10^{-4}$
  - I = 10 iterations were performed for both NIMPC PCG and Proj-NIMPC PCG

Annotated field of view for measured results

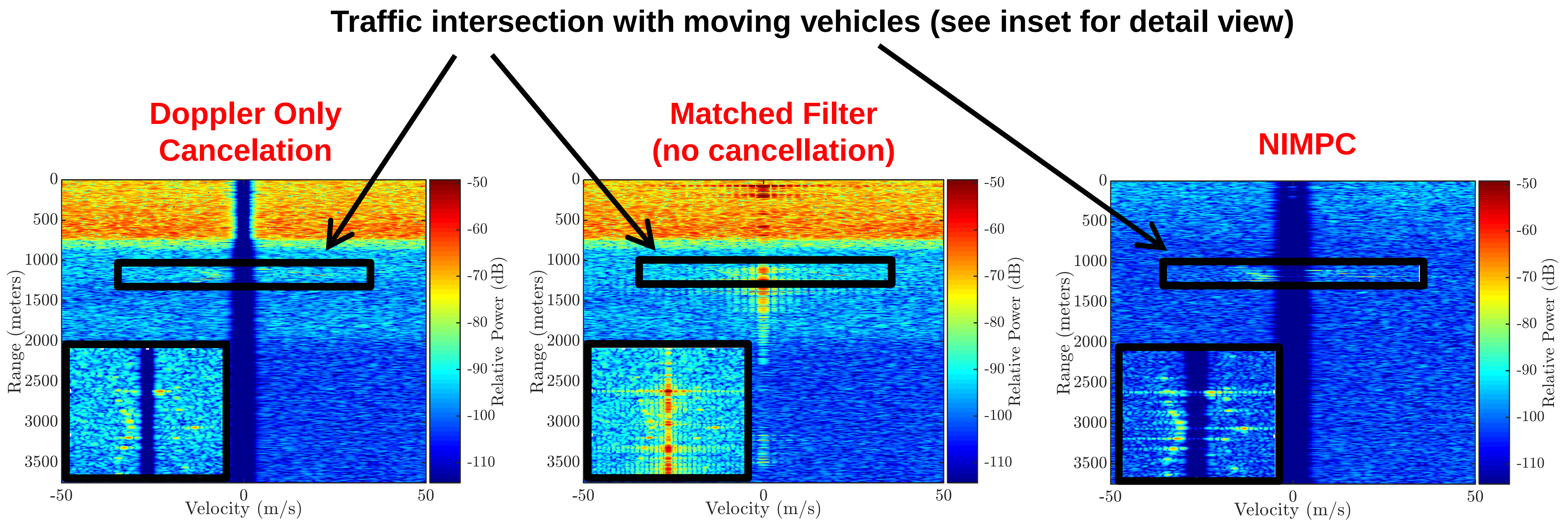


- First compare original NIMPC with standard processing (matched filter & Doppler processing), with and without simple projection-based clutter cancellation



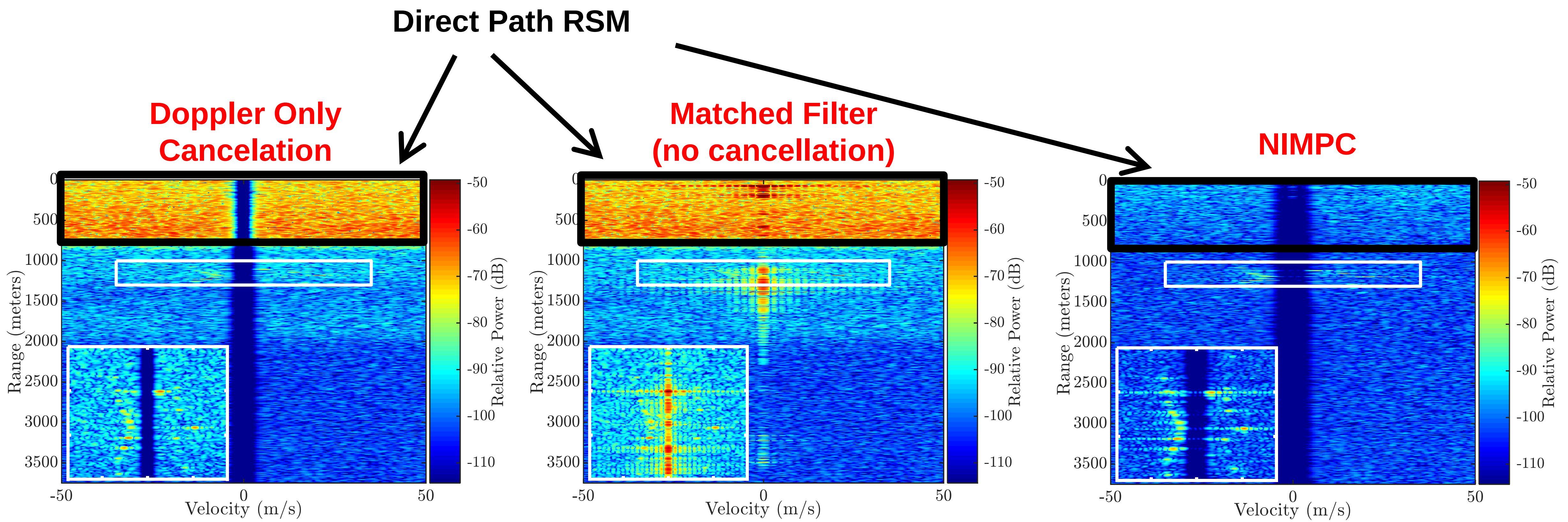
Experimental Validation: NIMPC Baseline

- The RSM speckle in the intersection is pushed below the noise floor by NIMPC



Experimental Validation: NIMPC Baseline

- NIMPC provides significant suppression of clutter RSM (here caused by direct-path leakage)

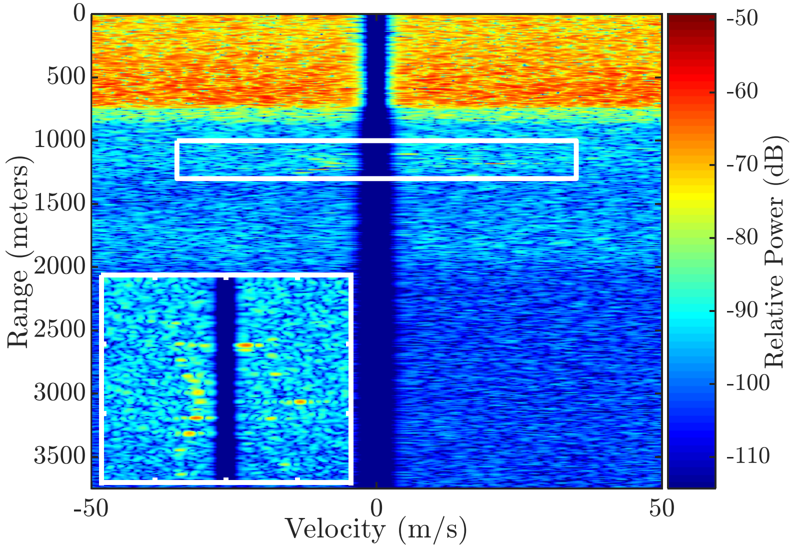


Unlike NIMPC, Doppler-only cancellation does not account for RSM

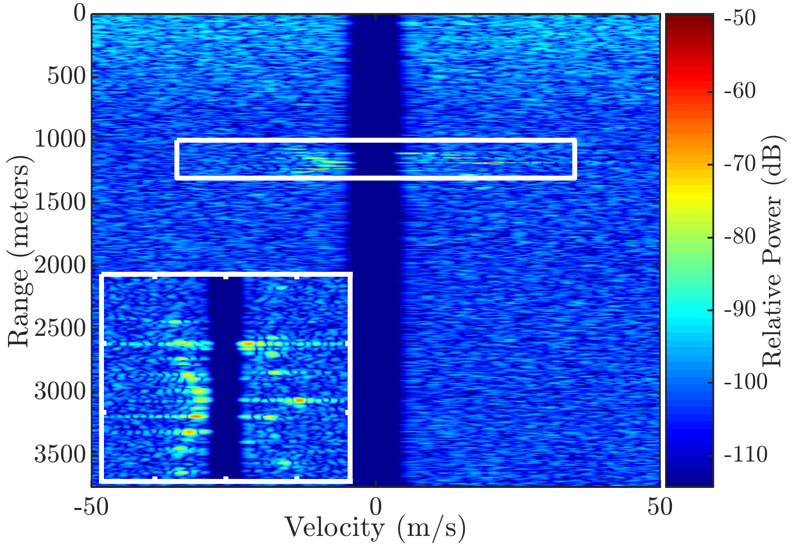
Experimental Validation: Proj-NIMPC (Direct)

- Now compare the direct form of Proj-NIMPC with standard processing (with clutter cancellation) and original NIMPC

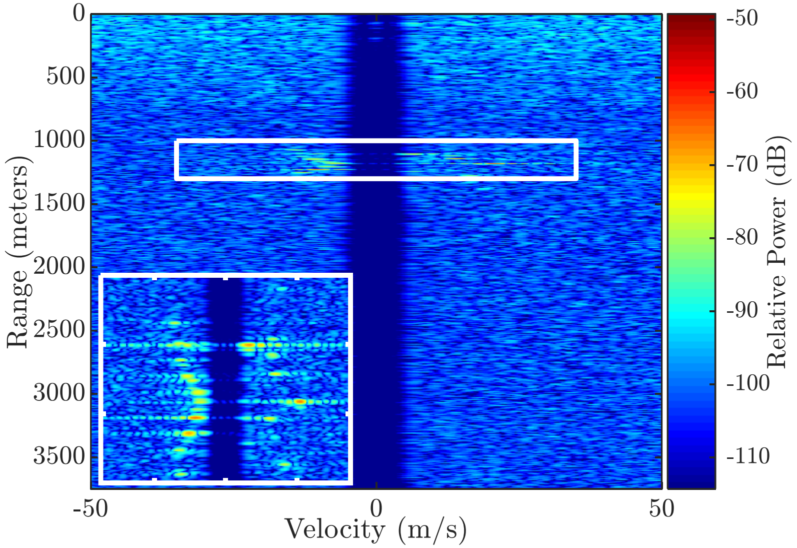
Doppler Only Cancelation



Direct Proj-NIMPC



NIMPC



Proj-NIMPC yields RSM suppression on par with NIMPC for a <u>markedly lower cost</u>

Evaluation of Efficient Direct Solvers

- For the direct NIMPC approaches, an efficient block-Toeplitz solver reduces computational cost by **an order of magnitude**

- Using the block-Toeplitz structure for the Proj-NIMPC implementation reduces complexity by **almost two additional orders of magnitude**

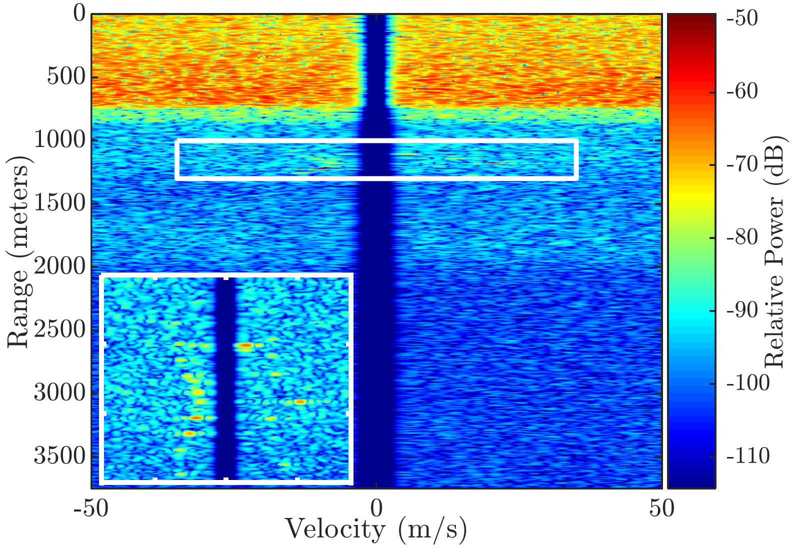
- However, these direct solutions require storage of extremely large matrices … and further computational reduction is possible

Number of Complex Operations For Experiment	
NIMPC (direct)	9.5×10^{14}
NIMPC (direct w. efficient solver)	8.7×10^{13}
NIMPC PCG	
Proj-NIMPC (direct w. efficient solver)	1.9×10^{12}
Proj-NIMPC PCG	

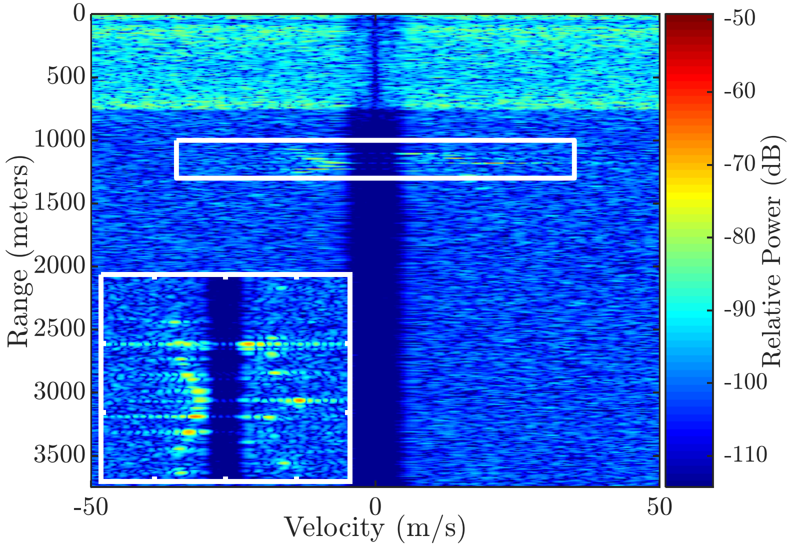
Experimental Validation: NIMPC PCG

- Compare to the NIMPC PCG implementation after 10 iterations

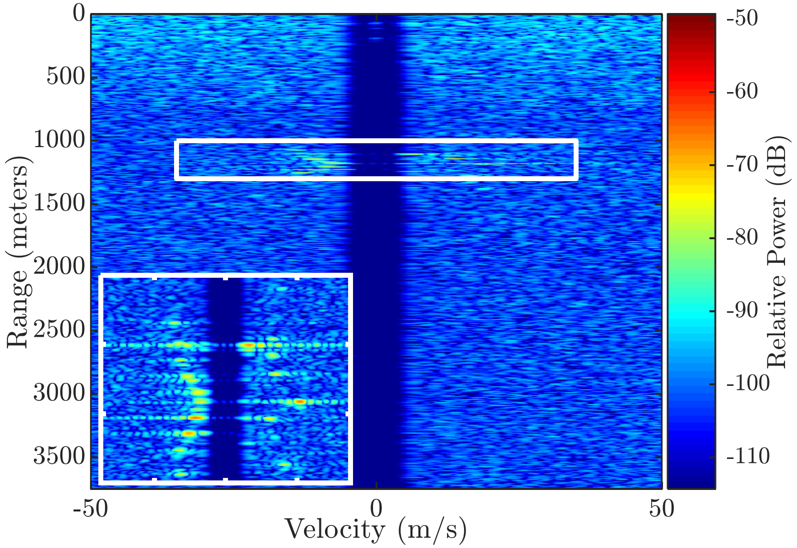
Doppler Only Cancelation



NIMPC PCG *I* = 10

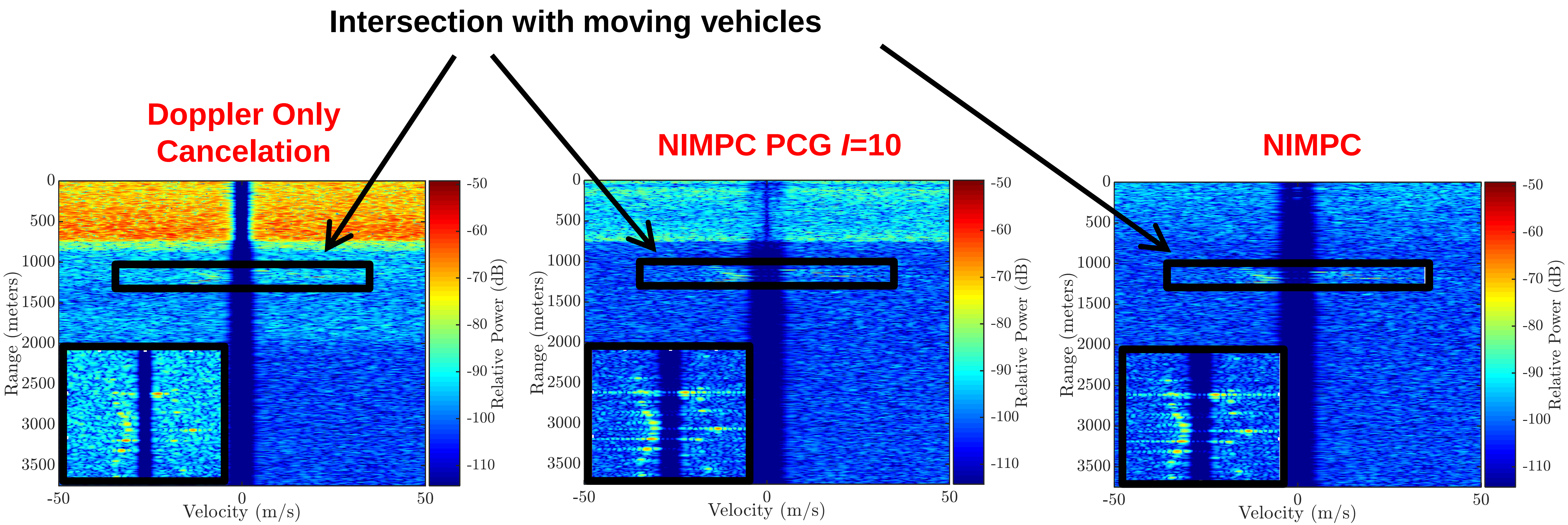


NIMPC



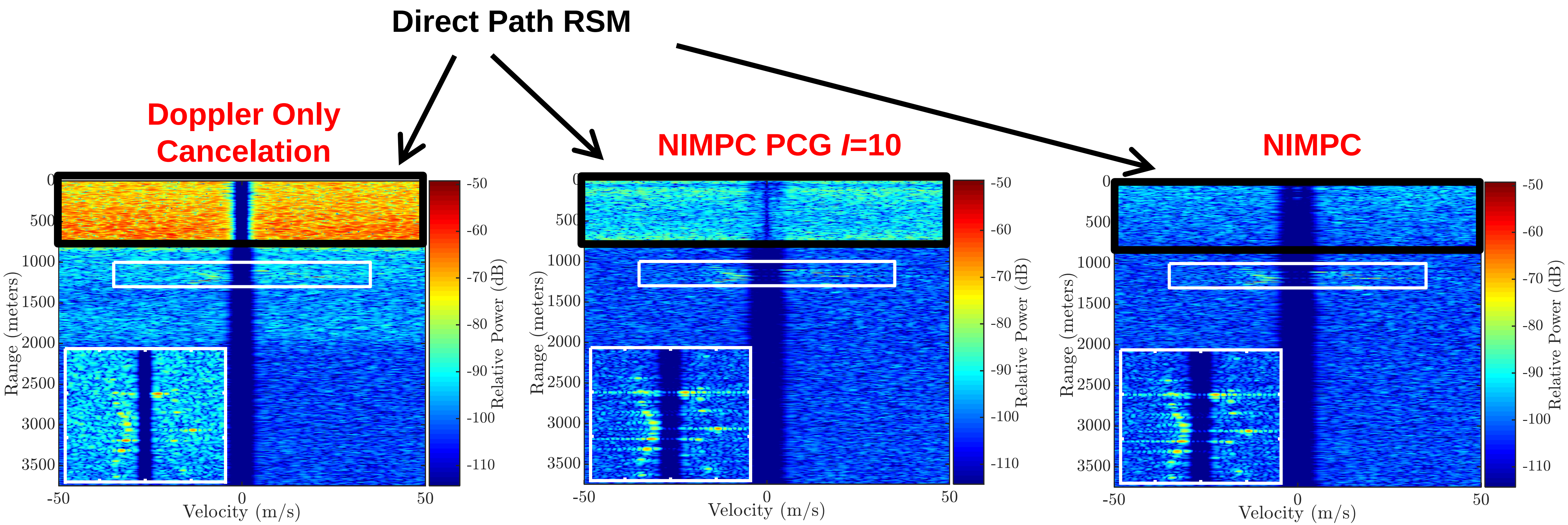
Experimental Validation: NIMPC PCG

- 10 iterations is enough to push the intersection RSM below the noise floor



Experimental Validation: NIMPC PCG

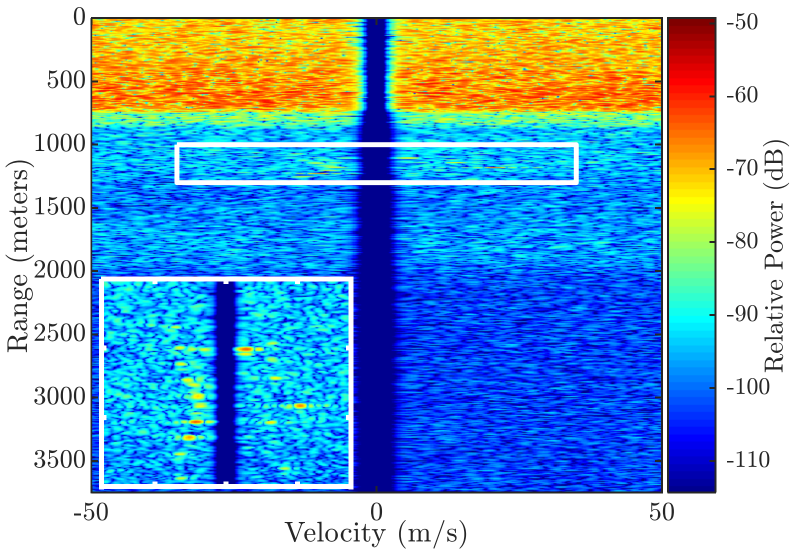
- However, 10 iterations is not enough to fully suppress the RSM from direct path leakage



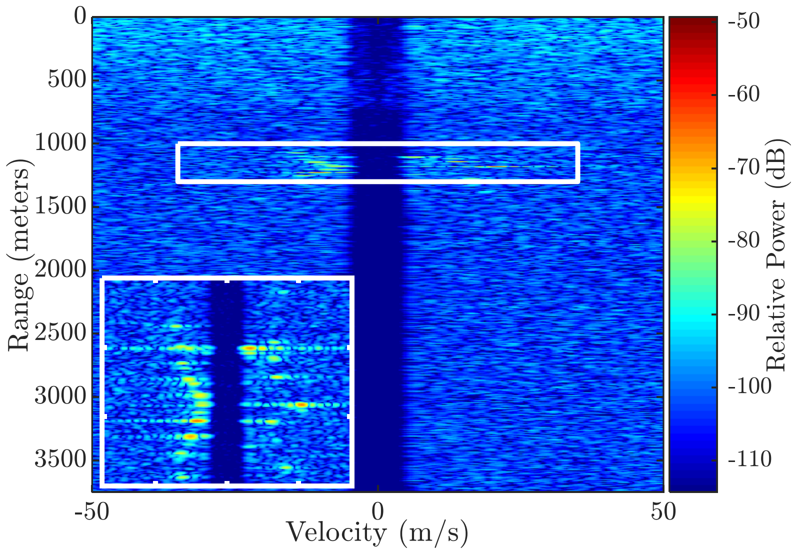
10 iterations of PCG has mitigated RSM at the intersection, but some direct path RSM remains

- Finally, consider 10 iterations of Proj-NIMPC PCG

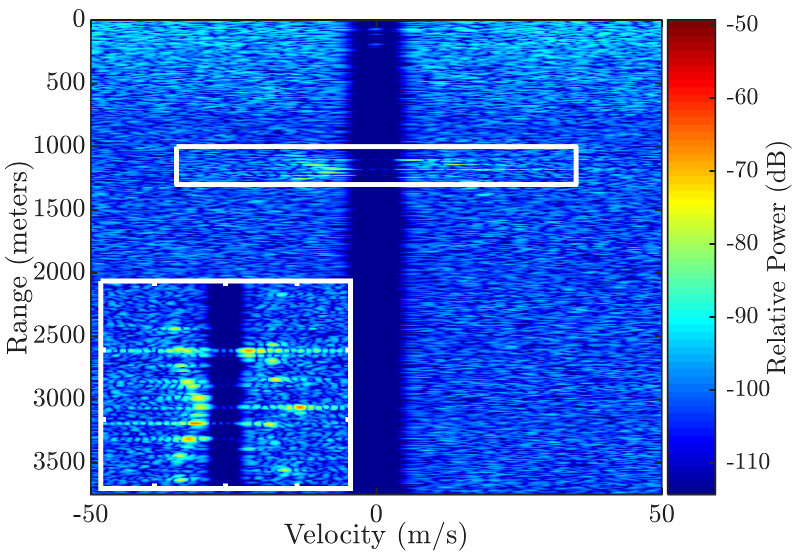
Doppler Only Cancelation



Proj-NIMPC PCG *l* = 10

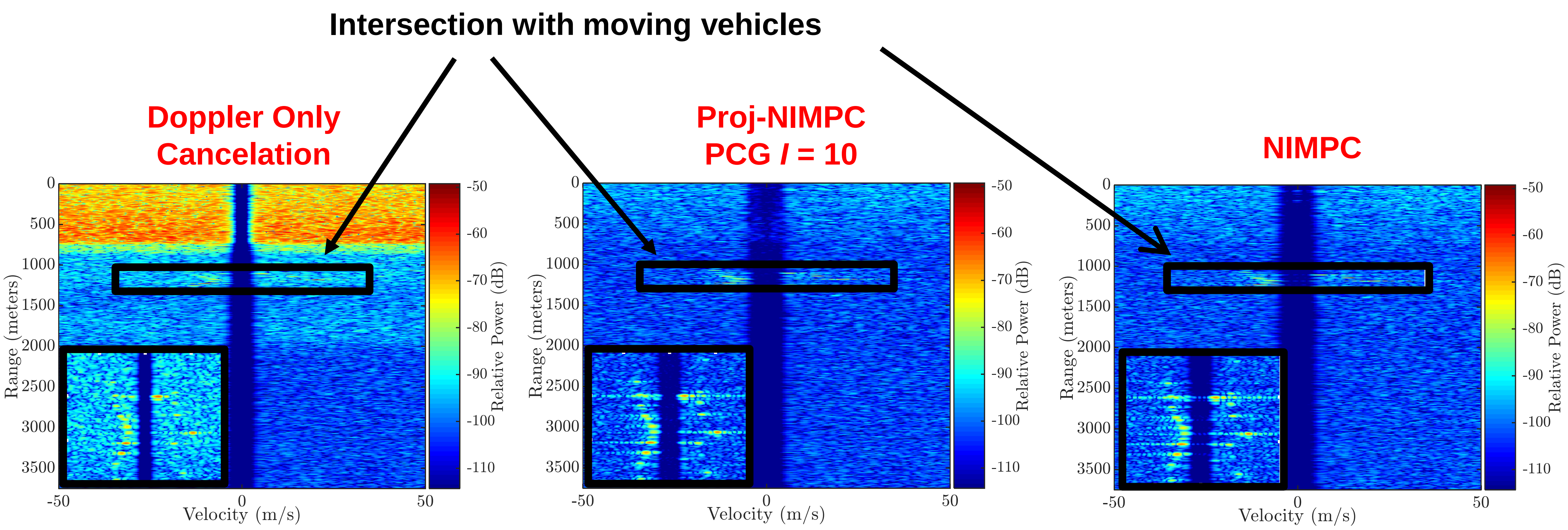


NIMPC



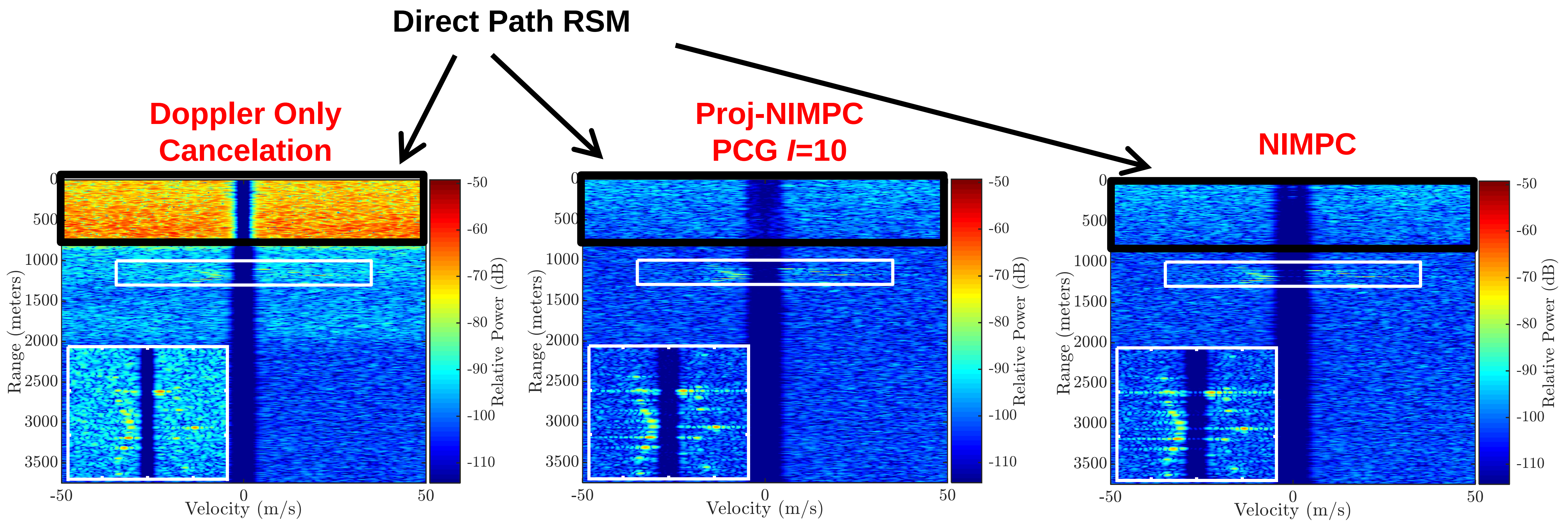
Experimental Validation: Proj-NIMPC PCG

- 10 iterations is again enough to push the intersection RSM below the noise floor



Experimental Validation: Proj-NIMPC PCG

- Moreover, now the direct path RSM is suppressed to the same degree as NIMPC



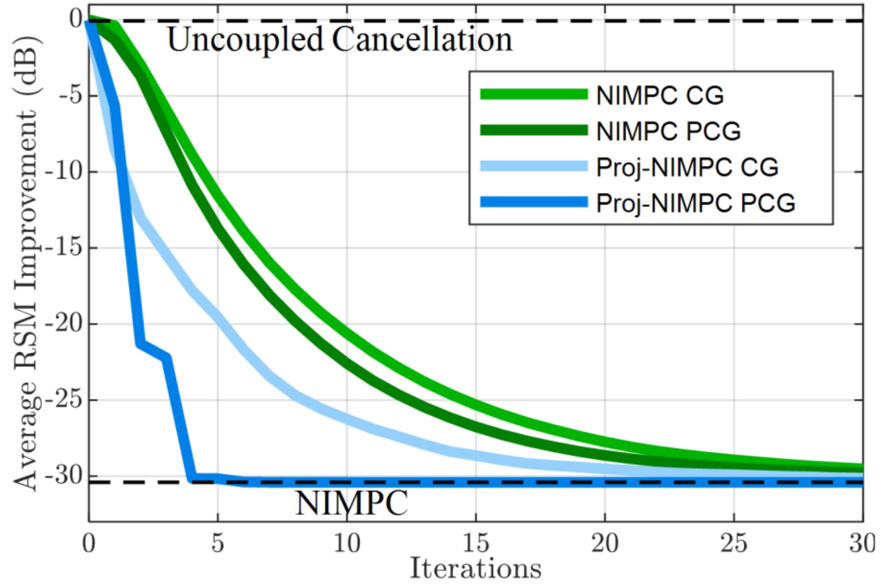
Proj-NIMPC PCG suppresses RSM on par with NIMPC for a significantly lower computational cost

Evaluation of Iterative Solvers

- 10 iterations of NIMPC PCG reduces computational cost by 4 orders of magnitude relative to original NIMPC
- However, 10 iterations is not enough for NIMPC PCG to sufficiently converge so some residual RSM remains
- Alternatively, with reduction in computational cost more than 5 orders of magnitude, 10 iterations of Proj-NIMPC PCG yields RSM suppression as well as NIMPC

Number of Complex Operations For Experiment	
NIMPC (direct)	9.5×10^{14}
NIMPC (direct w. efficient solver)	8.7×10^{13}
NIMPC PCG	**2.3×10^{10}**
Proj-NIMPC (direct w. efficient solver)	1.9×10^{12}
Proj-NIMPC PCG	**3.1×10^{9}**

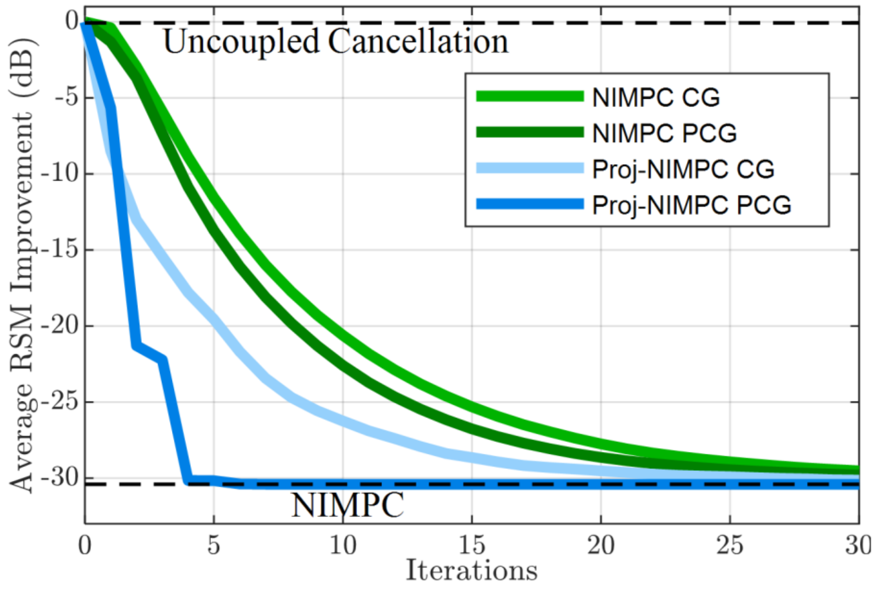
Direct Path RSM Improvement vs Iteration



Conclusions

- Traditional clutter cancellation does not account for range-Doppler coupling that occurs with waveform agility
- NIMPC performs joint range-Doppler clutter cancellation … but incurs a high computational cost
- Using efficient solvers and reformulating NIMPC as a projection can each reduce computational overhead
- Further, both NIMPC and Proj-NIMPC can be iteratively implemented with PCG to significantly improve efficiency … potentially to the point of achieving real-time processing

Direct Path RSM Improvement



Number of Complex Operations For Experiment	
NIMPC (direct no structure)	9.5×10^{14}
NIMPC (direct with structure)	8.7×10^{13}
NIMPC PCG	**2.3×10^{10}**
Proj-NIMPC (direct with structure)	1.9×10^{12}
Proj-NIMPC PCG	**3.1×10^{9}**