Short Course on
Space-Time Adaptive Processing

Raviraj S. Adve
Department of Electrical and Computer Engineering
University of Toronto
10 King's College Road
Toronto, ON M5S 3G4, Canada

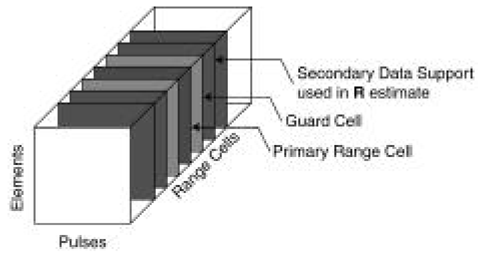
Tel: (416) 946 7350
E-mail: *rsadve@comm.utoronto.ca*

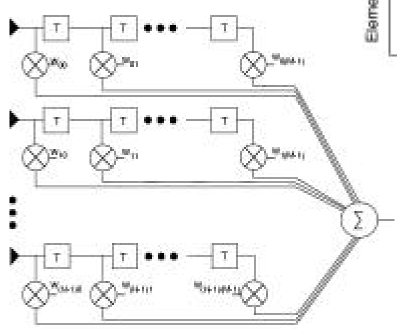
Overview

- STAP: Detection of weak signals in stressful environments

- The data cube
- Steering Vectors
- Beam Patterns
- Components of the data
  - Clutter ridge, noise and discrete jammers, noise
- Non-adaptive techniques
  - MTI, DPCA, Matched Filter
- Optimal techniques
  - Wiener Filter, Minimum Output Energy
- Fully Adaptive STAP
  - SMI, MSMI, Kelly's Test
- Reduced dimension techniques
  - Principal Components, CSM
  - Factored approaches
- Original JDL, Sigma-Delta Σ-Δ
- JDL for practical arrays
  - Transformation matrix
  - Implementation
- Non-homogeneity detection
  - GIP
  - STAP based NHD
  - Implementation
- D^3 Processing
  - Sarkar's original method
  - Two dimensional extension
  - Implementation
- Hybrid Algorithm
  - Implementation
- KB-STAP
  - Implementation

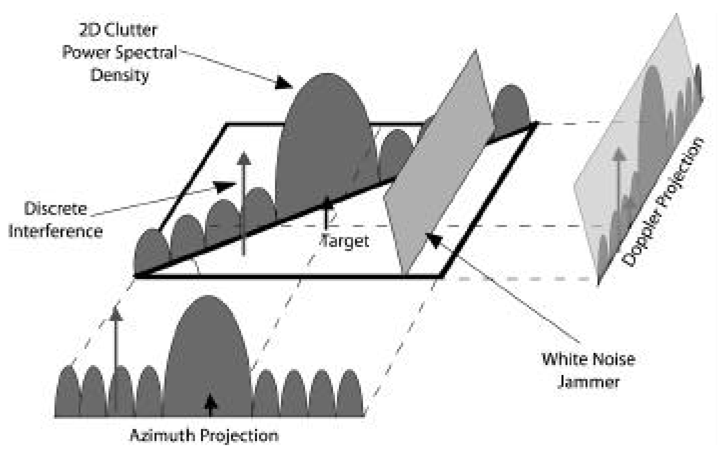
Space-Time Adaptive Processing





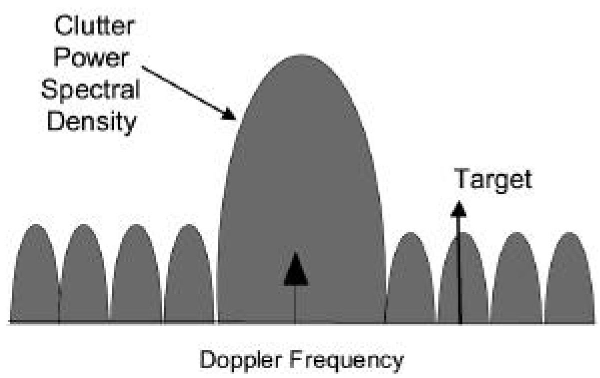
Goal: Find **w**

Airborne Interference Scenario

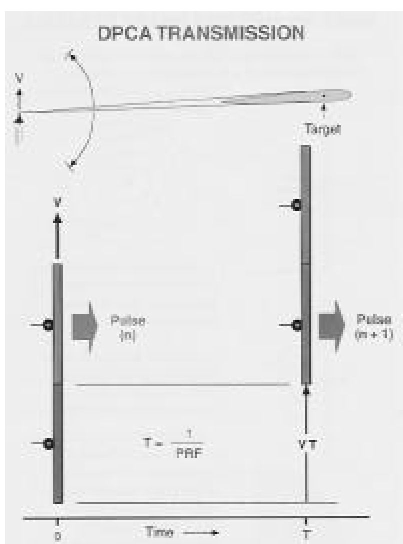


Airborne Moving Target Indication

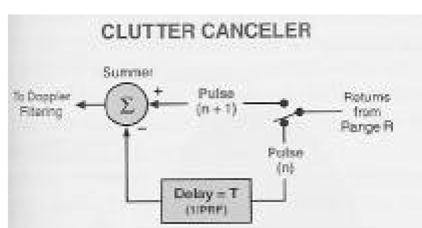
AMTI: Discriminate between clutter and target using Doppler



Displaced Phased Center Antenna



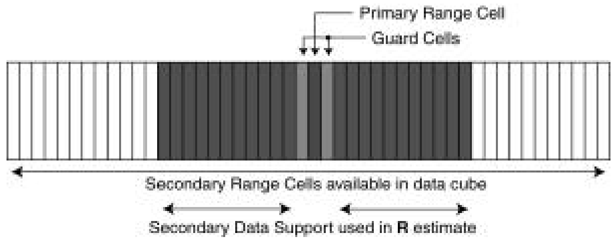
- DPCA:Maintain constant phase center over successive pulses and subtract sections appropriately
- Follow with Doppler filtering
- Velocity and PRF must match



Pictures from "Introduction to Airborne Radar", G.W. Stimson, SciTech Publishing Inc.

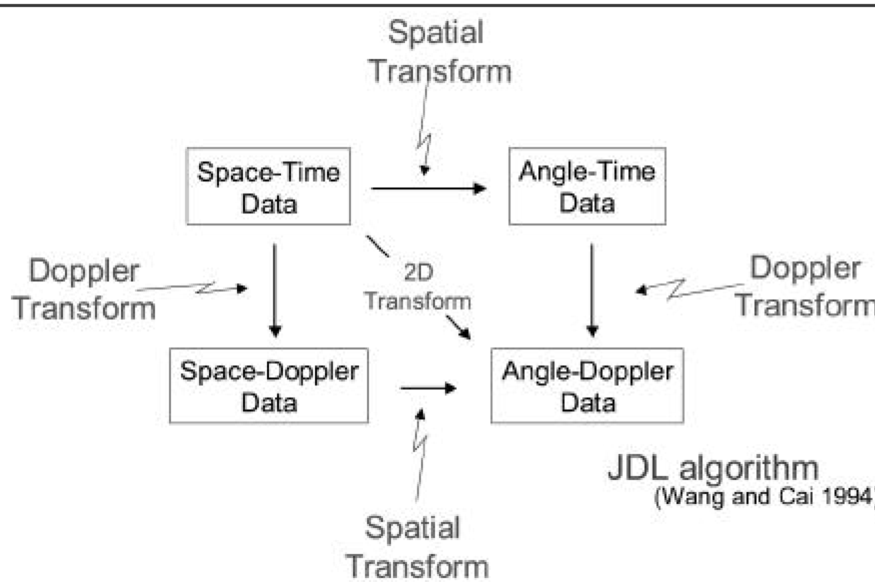
Fully Adaptive STAP

Estimating the covariance matrix

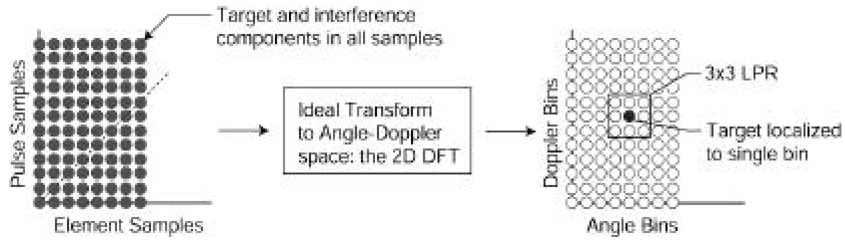


- Issues of
  - Availability of secondary data
  - Computation load
- Fully adaptive STAP is impossible to implement in practice

STAP Processing Domains



Joint Domain Localized Processing



Transformation shown for primary range only. All ranges are transformed.

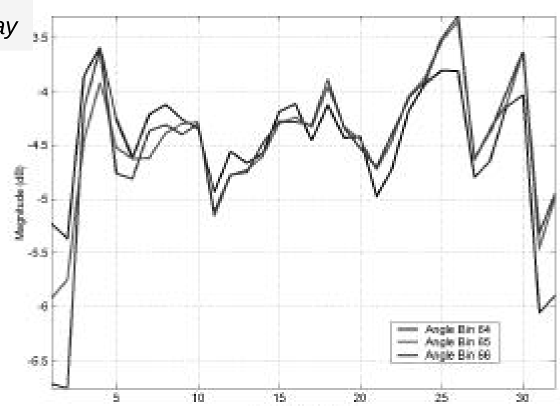
Measured Steering Vectors

Ideal Array

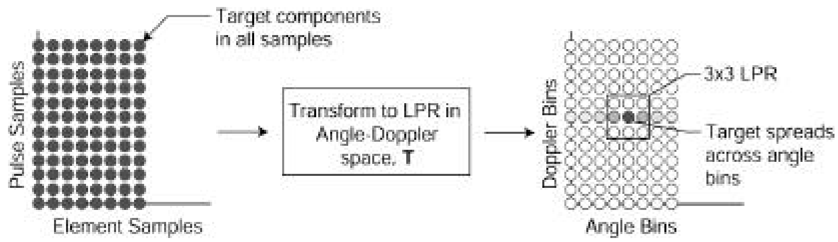
$\mathbf{s} = [\ 1\ e^{jkd\,sin(\theta)}\ e^{j2kd\,sin(\theta)} \ldots\ e^{j(N-1)kd\,sin(\theta)}\ ]^T$

– *Constant Magnitude*

MCARM Array



JDL For Real Arrays



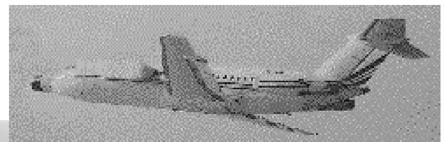
Transformation shown for primary range only. All ranges are transformed.

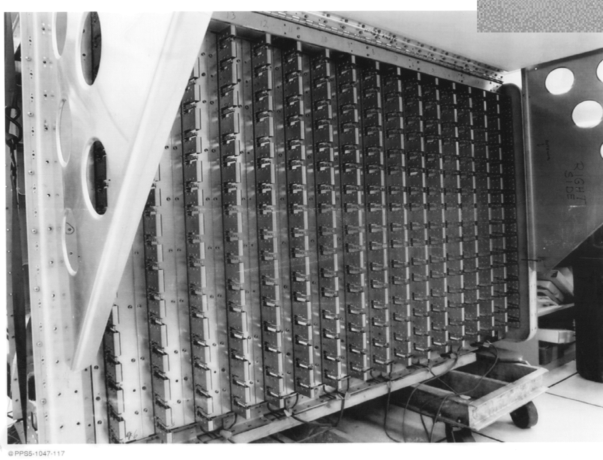
Implementation of JDL

1. Choose the size of the LPR, i.e. η_a and η_d. These numbers are usually odd. Common choices are 3, 5 or 7.
2. Choose the number of secondary data vectors that will be used to estimate the covariance matrix (usually of the order of $2\eta_a\eta_d$ - $4\eta_a\eta_d$.) and the number of guard cells (usually 2-4).
3. If using tapers in space and time, choose a length *N* taper in the space domain $\mathbf{t}_a$ and a length *M* taper $\mathbf{t}_d$ in the temporal domain.
4. Set the angle bin to be the direction in which the radar signal is transmitted. Choose a set of η_a angles centered around (and including) the look angle.
5. For each Doppler bin of interest, repeat the following steps:
6. Choose a set of η_d Doppler bins centered around (and including) the look Doppler. Use the set of angles and Doppler bins to form the transformation matrix **T**.
7. Transform *the entire datacube* to the angle-Doppler space.
8. Form the JDL steering vector. This vector is fixed for all range bins.
9. For each range bin, repeat the following steps:
10. Estimate an angle-Doppler covariance matrix using *transformed secondary data*.
11. Obtain the angle-Doppler adaptive weights.
12. Obtain a decision statistic. The data vector is the *transformed primary data*.
13. Compare the decision statistic to the threshold. If statistic is greater than the threshold, declare a target to be present at the look angle-Doppler, if not a target is said to be absent.

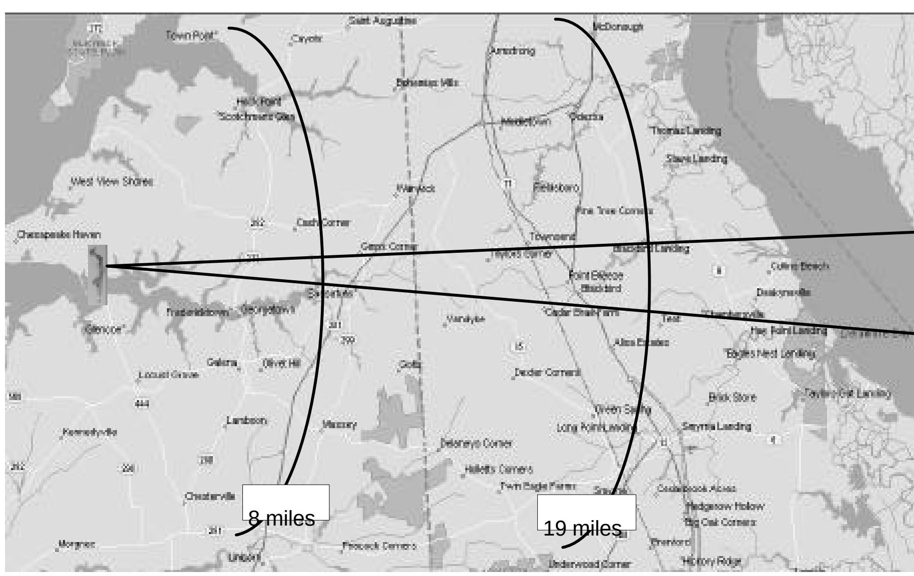
Steps 5 and 9 form nested loops

The MCARM Array





Data Source



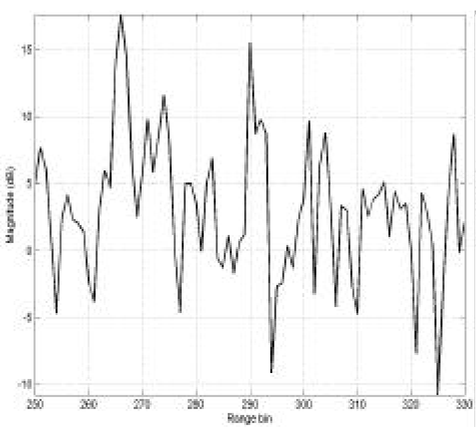
Example: Injected Target

- 22 elements, 128 pulses
- Amplitude = 0.0003
- Angle bin = 0 (broadside)
- Doppler bin = -9
- Range bin = 290
- 3x3 JDL Processing
- No window in space.
- $\mathbf{t}_d$ = *kaiser(M,log10(M));*

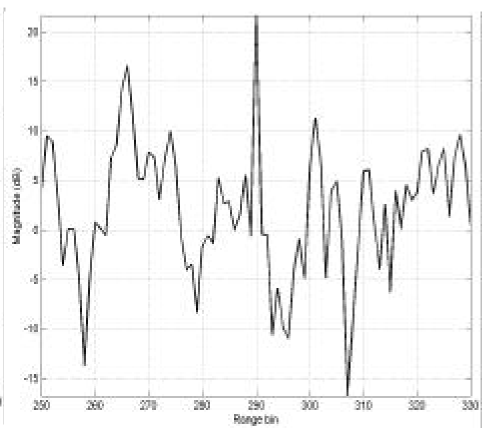
BRSC November 12^{th} 2001

JDL For Real Arrays

Assuming ideal array



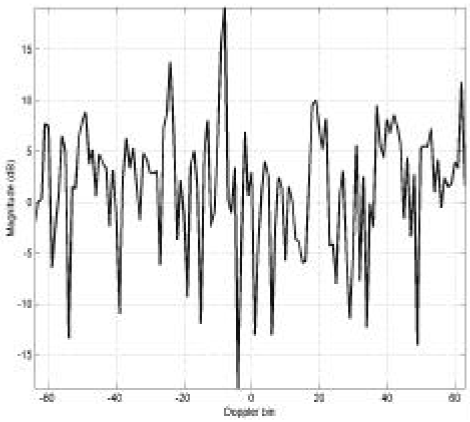
Accounting for array effects



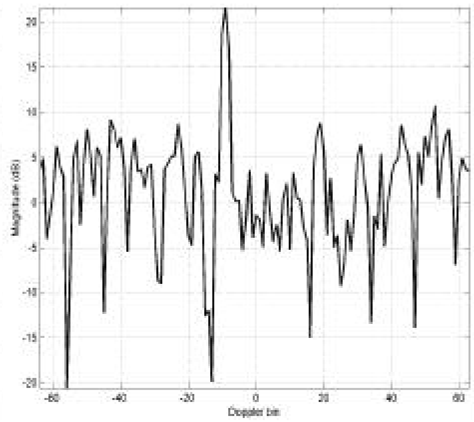
BRSC November 12^{th} 2001

JDL For Real Arrays

Assuming ideal array



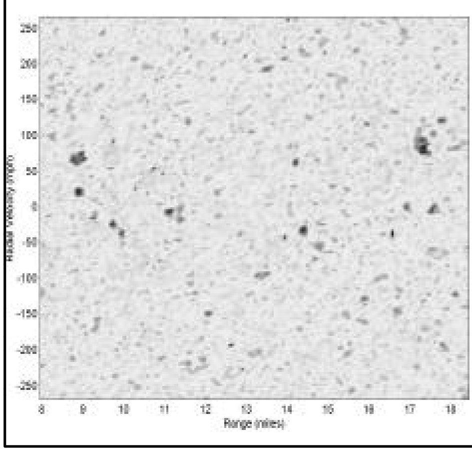
Accounting for array effects



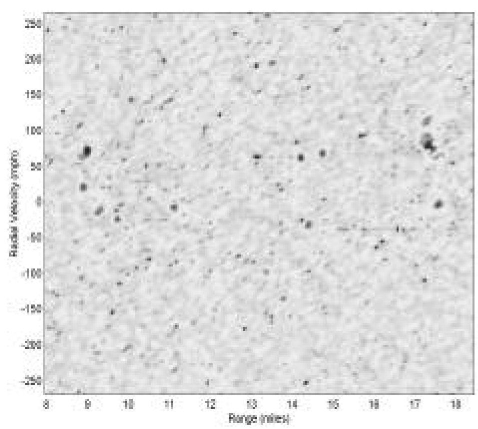
JDL For Real Arrays

Accounting for array effects significantly improves STAP performance

Assuming ideal array



Accounting for array effects

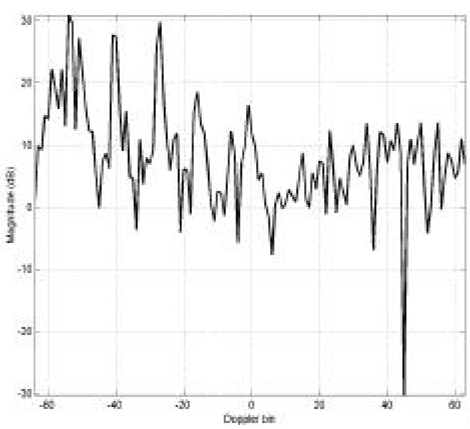


MTS Tones

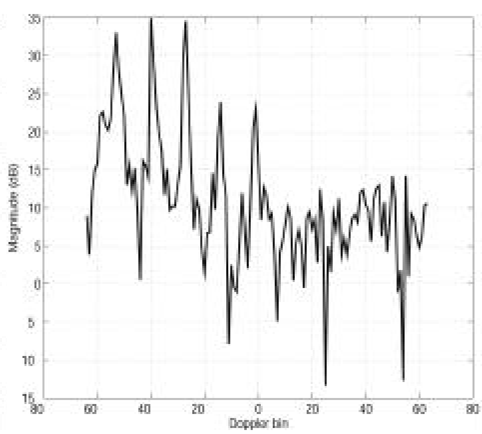
- Moving Target Simulator
- 22 elements, 128 pulses
- 5 simulated targets
  - Located at 0, -200, -400, -600, & -800 Hz
- 200 Hz ≡ 13 Doppler bins
- 3x3 JDL Processing
- No window in space.
- $\mathbf{t}_d$ = *kaiser(M,log10(M));*

JDL For Real Arrays

Assuming ideal array

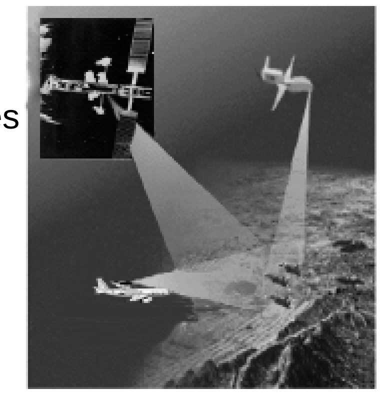


Accounting for array effects



Sources of Non-Homogeneities

- Statistical outliers
- Real world radar returns contain
  - Changes in terrain over short distances
    - Urban areas
    - Land-Sea interfaces
  - Manmade non-homogeneities
    - Vehicular traffic
    - Corner reflectors
    - Blinking jammers
  - Dense target environments
    - Large sidelobe targets



Non-Homogeneities

- Incorrect estimate of covariance matrix
  - Poor interference suppression (more false alarms)
- Target-like signals
  - In secondary data, cause target nulling
  - In sidelobes, are false alarms and improper ID
- Non-Homogeneity Detection (NHD)
  - Identifies all cells that deviate from the “norm”
  - Separates data cube into homogeneous and non-homogeneous cells
    - Generalized Inner Product (GIP)
    - STAP as NHD

Generalized Inner Product

- Estimates **R** using all the data (entire data cube)

$$\eta_{GIP} = \mathbf{x}^H \mathbf{R} \mathbf{x}$$

- Find *K* range cells closest to mean of GIP statistic
  - Estimate **R** using these *K* range cells
- Can be applied in original or transform domains

- Computationally efficient
- Mathematically sound (T^2 test, Chen)

- Loose local info.
  - Handles several clutter types?
- Wasting a lot of data

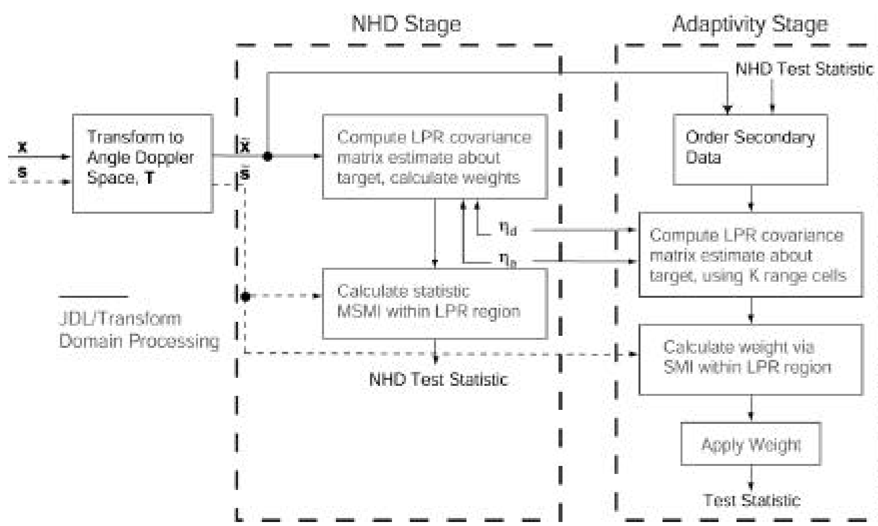
GIP Variants

- Cell assumed homogeneous if within tolerance of mean
  - Use all homogeneous cells to estimate **R**
    - good estimate of **R** (many, many cells used)
    - no local information
  - Use *K* homogeneous cells close to the primary range cell to estimate **R**
    - regain local information
    - is this local information valid
    - computationally more complex

STAP as NHD

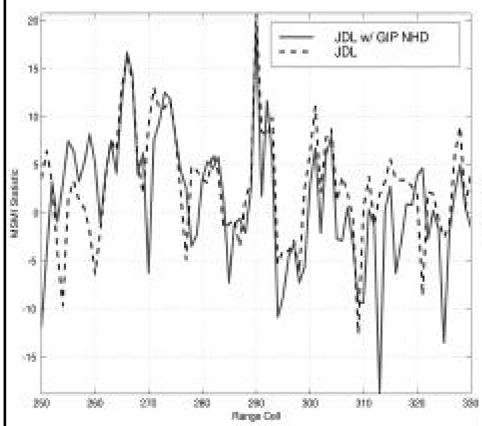
- Apply STAP assuming homogeneous data
  - Choosing threshold for statistic sets tolerance to non-homogeneities
    - trade-off between sensitivity and numbers
  - Retains local information
  - Only *cells that impact* on STAP processing identified as non-homogeneous
  - Use *K* homogeneous cells close to the primary range cell to estimate **R**
  - Computationally intensive

JDL-NHD

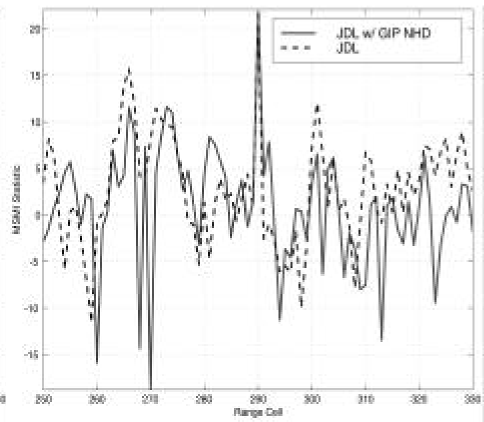


LPR GIP Performance

Assuming Ideal Array



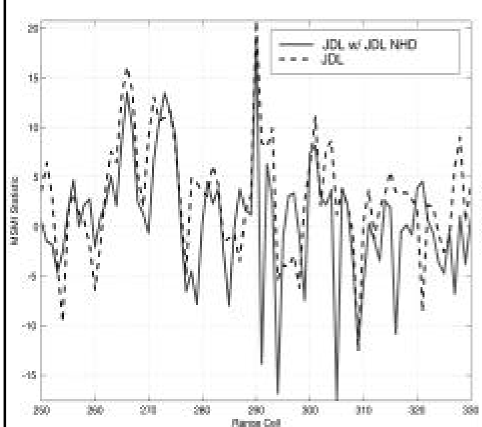
Accounting for Array Effects



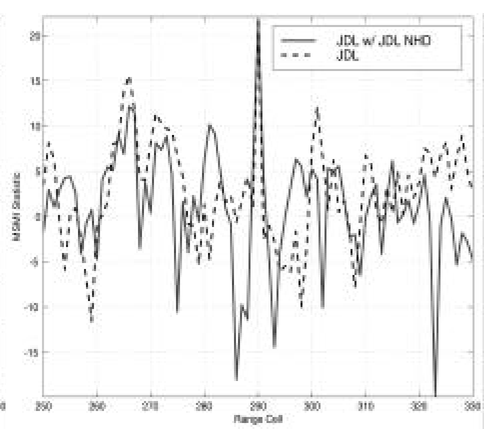
BRSC November 12^{th} 2001

JDL-NHD Performance

Assuming Ideal Array



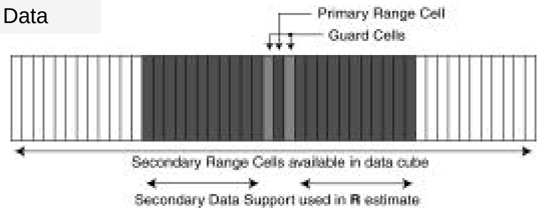
Accounting for Array Effects



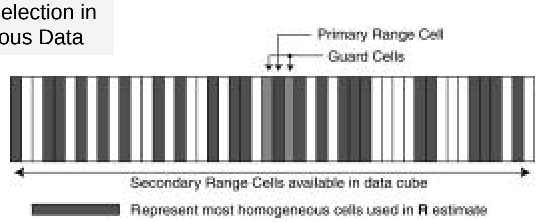
BRSC November 12^{th} 2001

STAP in Homogeneous Cells

Sample Support Selection in Homogeneous Data

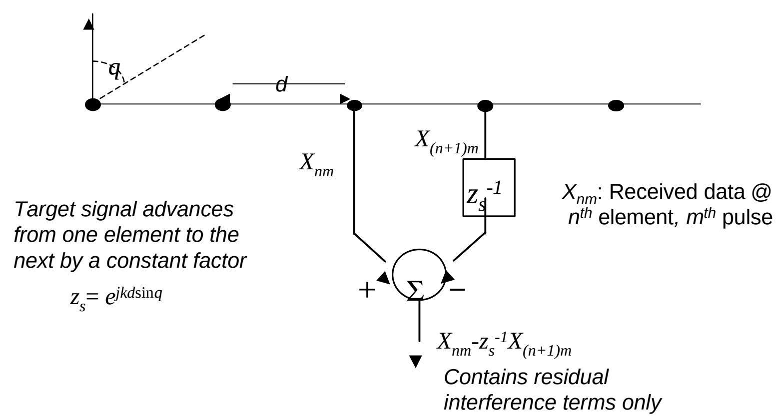


Sample Support Selection in Non-Homogeneous Data



What do you do in non-homogeneous cells?

Non-statistical (D^3) Processing



D^3 Algorithm: Maximize gain of array while minimizing residual interference terms

Implementation of D^3 Algorithm

1. If using measured data, "rotate" the data using the measured steering vector
2. Arrange the data from the primary range cell in the form of a $N \times M$ matrix
3. Choose the emphasis parameter k
4. Form matrix **A** and matrix $[\mathbf{a}_{(0:N-2)}^H \mathbf{a}_{(0:N-2)} - \mathbf{A}^T \mathbf{A}^*]$. If using rotated $\mathbf{a}_{(0:N-2)} = [1\ 1\ 1\ \ldots\ 1]^T$
5. Find the eigenvector corresponding to the largest eigenvalue of this matrix. This is $\mathbf{w}_s$.
6. For each Doppler bin, repeat the following steps:
7. Form matrix **B** and matrix $[\mathbf{b}_{(0:M-2)}^H \mathbf{b}_{(0:M-2)} - \mathbf{B}^T \mathbf{B}^*]$
8. Find the eigenvector corresponding to the largest eigenvalue of this matrix. This is $\mathbf{w}_t$.
9. The D^3 adaptive weights are given by

$$\mathbf{w}(?, f_d) = \begin{bmatrix} \mathbf{w}_t \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{w}_s \\ 0 \end{bmatrix}$$

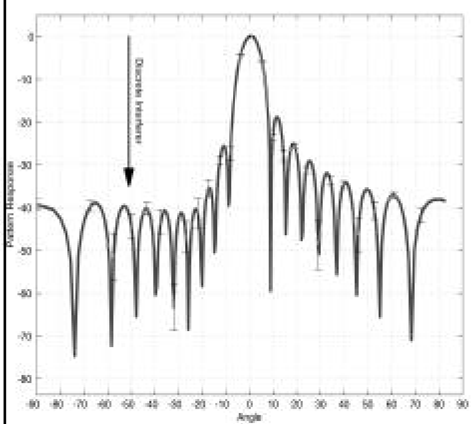
Simulation Example

- 18 elements, 18 pulses
- Mainbeam azimuth – broadside
- Jammer azimuth angles – 45^o, -20^o
- Normalized Doppler – 1/3
- Discrete Interferer
  - Azimuth angle – -51^o
  - Normalized Doppler – 1/3

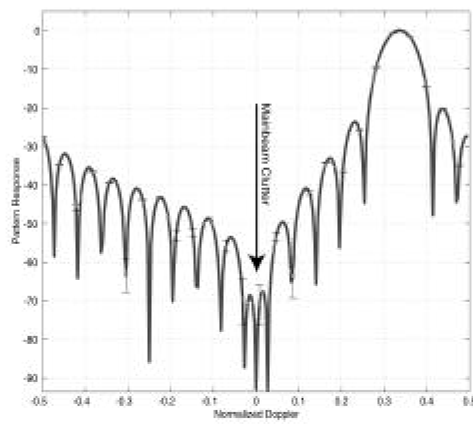
Performance Simulations

JDL Alone

Angle Response



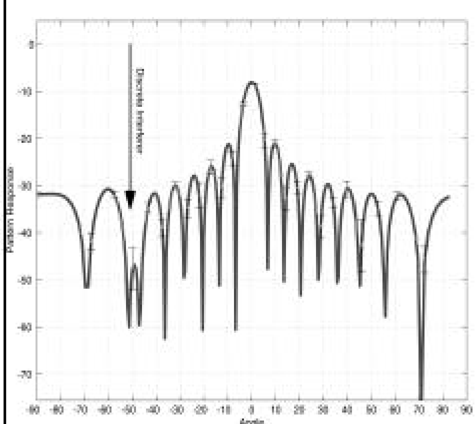
Doppler Response



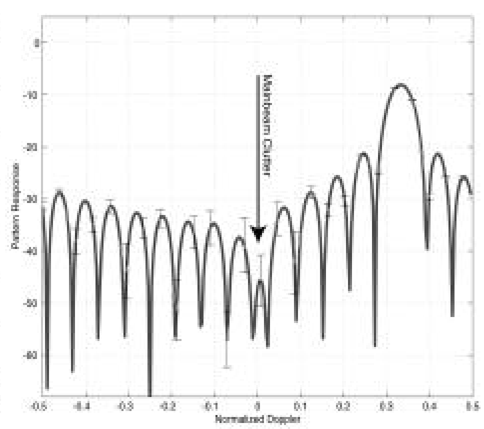
Performance Simulations

Direct Data Domain Method Alone

Angle Response

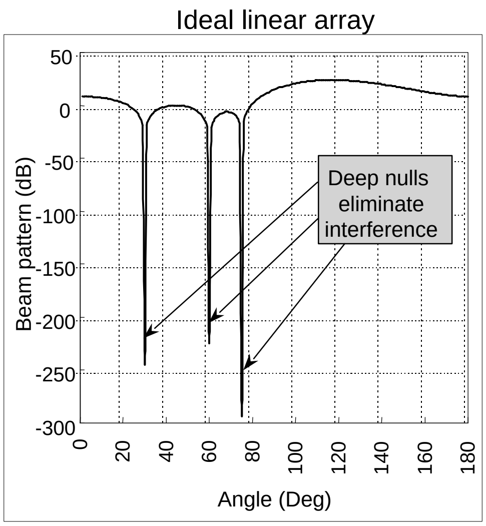


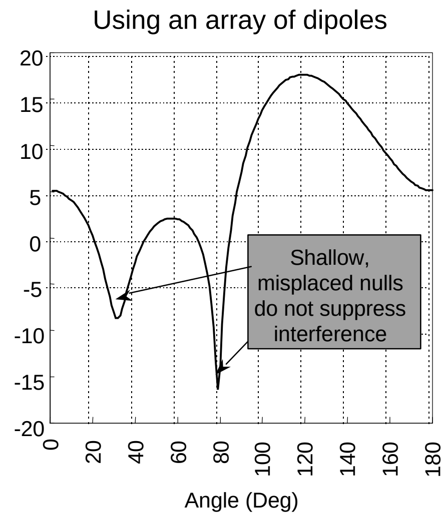
Doppler Response



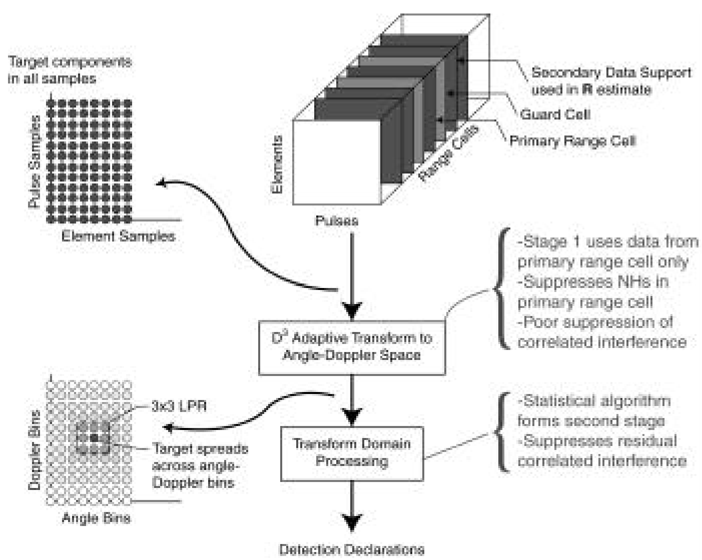
Poor suppression of correlated interference

Array Effects (Mutual Coupling)



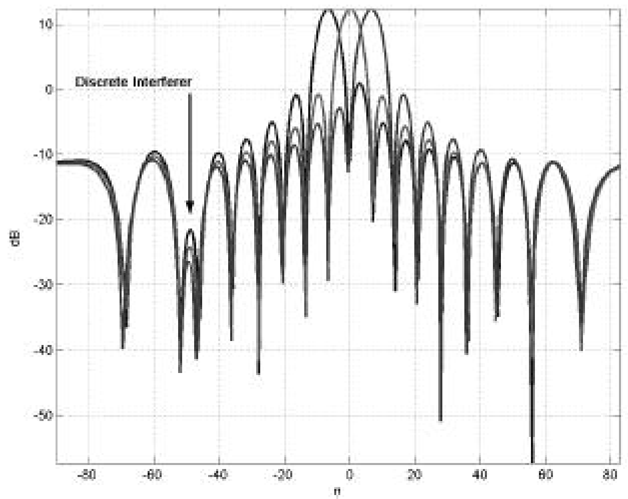


Hybrid Algorithm



Stage 1: Repeated D^3

$$\mathbf{T} = \left[\mathbf{w}(\theta_{-1}, f_{-1})\ \mathbf{w}(\theta_{-1}, f_0)\ \mathbf{w}(\theta_{-1}, f_1)\ \mathbf{w}(\theta_0, f_{-1})\ \mathbf{w}(\theta_0, f_0)\ \mathbf{w}(\theta_0, f_1)\ \ \mathbf{w}(\theta_1, f_{-1})\ \ \mathbf{w}(\theta_1, f_0)\ \ \mathbf{w}(\theta_1, f_1)\right]$$

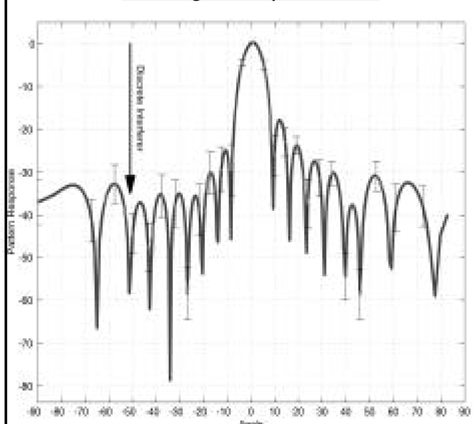


- D^3 applied $\eta_a\eta_d$ times using the *same* primary data
- (θ_0, f_0) the actual look angle and Doppler

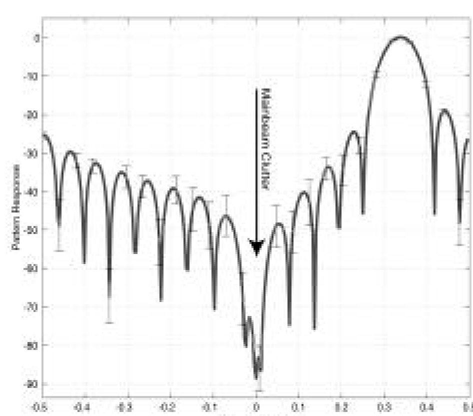
Performance Simulations

Two-Stage Hybrid Method

Angle Response



Doppler Response

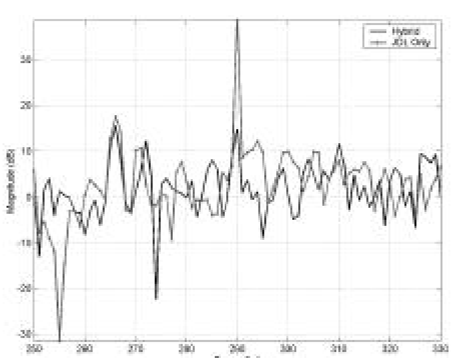


Implementation of Hybrid Algorithm

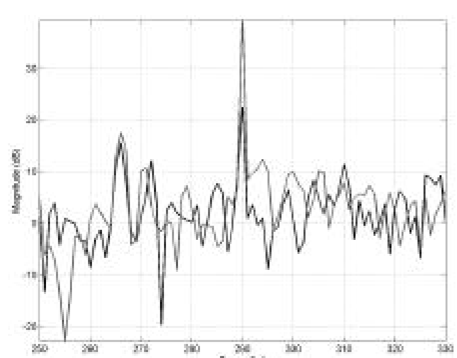
1. Choose the size of the LPR, i.e. η_a and η_d. Choose the set of η_a auxiliary look angles centered about look angle. Usually 3, 5, or 7
2. Choose *K*, the number of secondary data vectors to be used in estimating the angle-Doppler covariance matrix (**R**)
3. For each Doppler bin in each range bin, repeat the following steps:
4. Choose η_d auxiliary Dopplers centered around the look Doppler
5. For each auxiliary Doppler and angle bin, execute the D^3 algorithm
   - The $\eta_a\eta_d$ weight vectors form the transformation matrix **T**
6. Choose *K* homogeneous range cells closest to the primary range cell as secondary data. Transform the primary data, the secondary data and the steering vector using this transformation matrix
7. Estimate the angle-Doppler covariance matrix **R**
8. Obtain the adaptive weights using the transformed steering vector
9. Obtain a decision statistic

MCARM Example

No target, non-homogeneity



Target and non-homogeneity



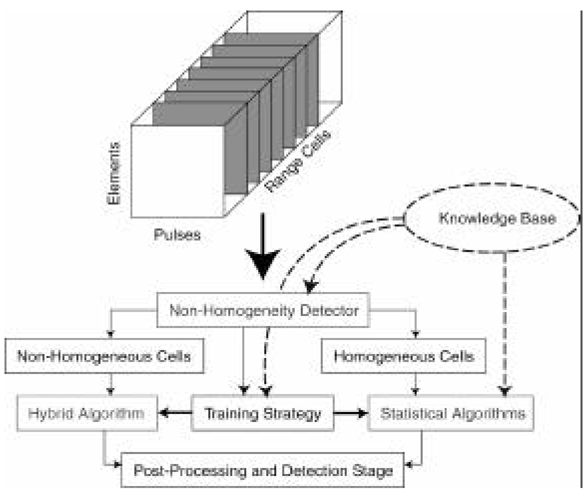
Non-homogeneity at angle bin –35^o, look angle: broadside

Hybrid Algorithm

- Very effective against discrete *and* correlated interference (clutter)
  - Non-statistical D^3 processing followed by statistical JDL processing
- Computationally intensive
  - D^3 algorithm must be executed $\eta_a \eta_d$ times
  - Use hybrid algorithm sparingly

Knowledge Based STAP

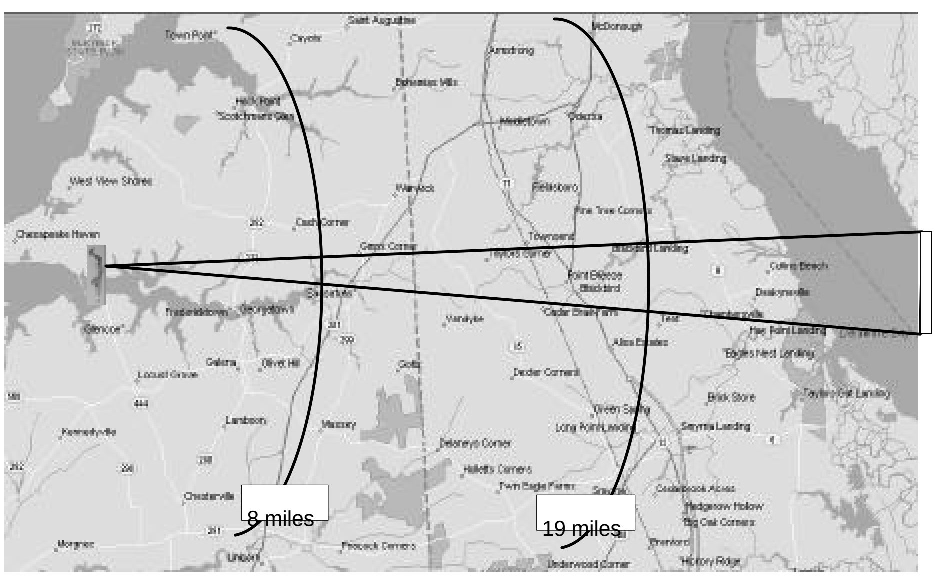
Put it all together, what do you get?
- A comprehensive, practical, approach to STAP



Implementation of KB-STAP

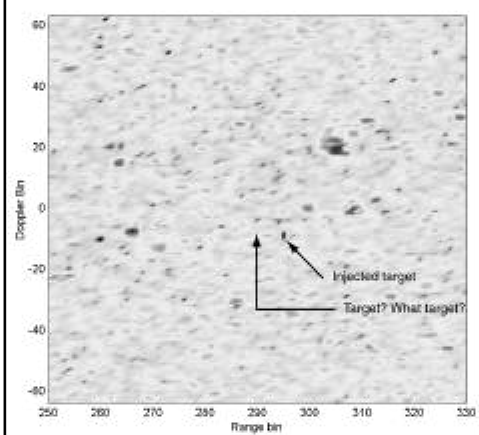
1. For each Doppler bin, repeat the following steps:
2. For all range bins, identify the homogeneous and non-homogeneous range bins using the JDL-NHD
3. For each range cell, repeat the following steps:
4. If the range cell is homogeneous, use JDL with other homogeneous cells for sample support
5. If the range cell is non-homogeneous, use the hybrid algorithm with other homogeneous range cells as sample support
6. In either case, compare with a threshold to determine if target is present or absent.

Data Source

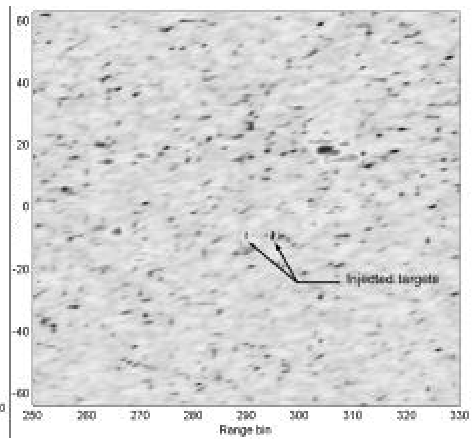


Ground and Air Moving Target Indication

Classical STAP



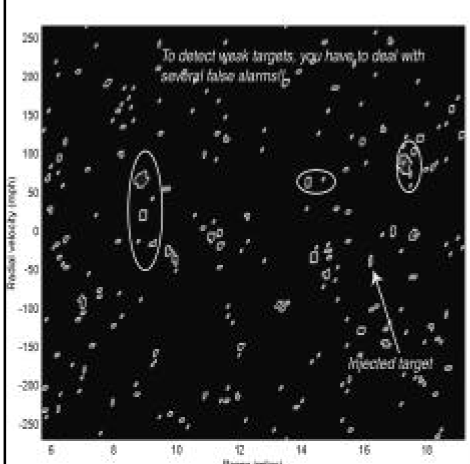
KB-STAP



Note: the NHD identifies the first target range cell as homogeneous

GMTI/AMTI After Thresholding

Classical STAP



KB-STAP