Case Study: Real-Time Demonstration of a Knowledge-Aided STAP Algorithm Using PVL

High Performance Embedded Computing Workshop
Lexington, MA

September 20-22, 2005

Bob Giannaris
John Don Carlos
Jameson Bergin

8130 Boone Blvd. Suite 500
Vienna, Virginia 22182
(703)448-1116   FAX: (703)356-3103
www.islinc.com

INFORMATION SYSTEMS LABORATORIES, INC.

Outline

- **Background**
- **Knowledge-Aided Signal Processing**
- **Real-time Implementation**
- **Summary and Future Work**

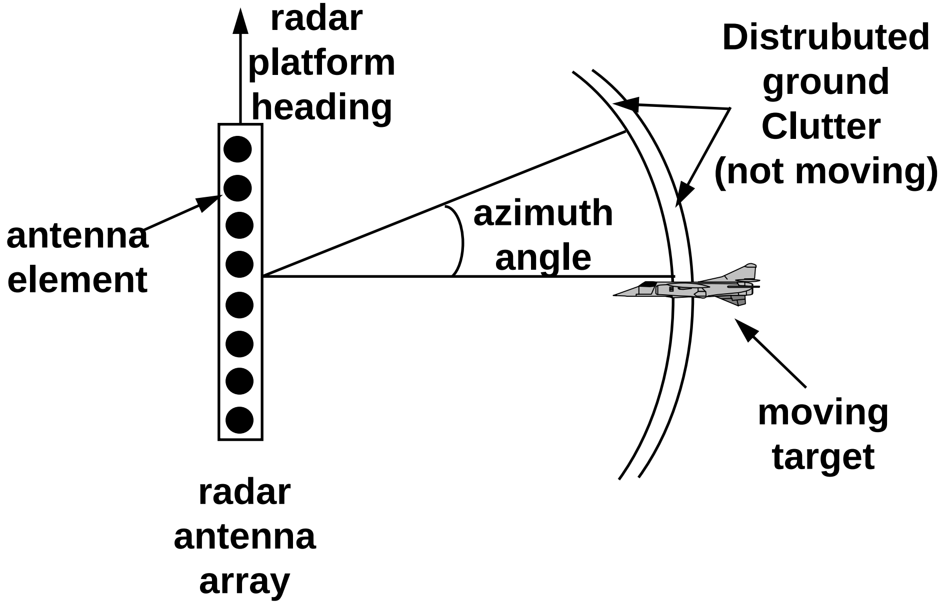
Background

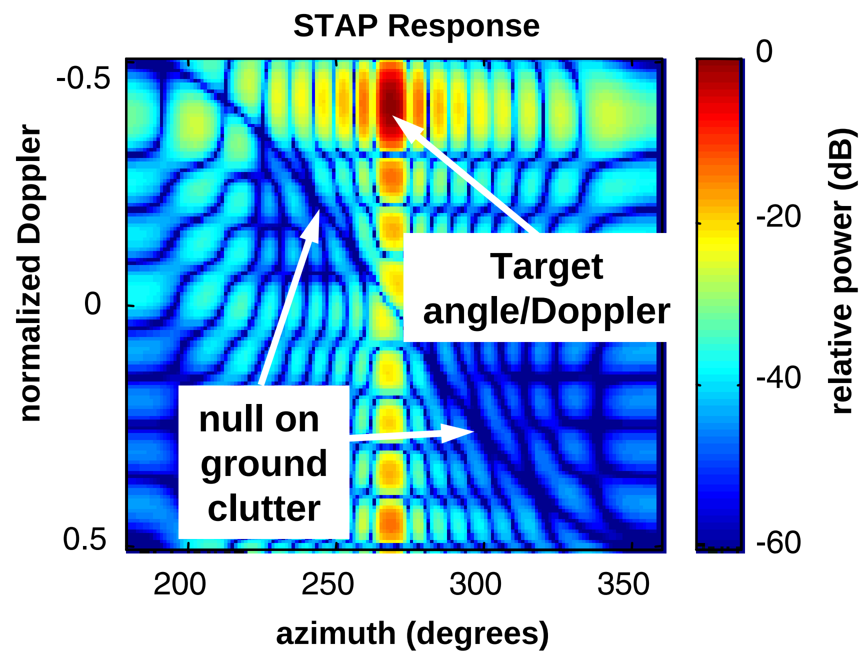
- **As part of the DARPA Knowledge-Aided Sensor Signal Processing and Expert Reasoning (KASSPER) program ISL has developed an advanced implementation of space-time adaptive processing (STAP) termed 'colored loading'**
- **Colored Loading incorporates *a priori* knowledge sources to improve clutter mitigate performance in airborne GMTI radar systems**
- **Goal: Demonstrate that the ISL-developed Colored Loading techniques can be implemented in real-time and still achieve the performance gains previously demonstrated in off line processing**
  - Set up Software Development Environment
  - Build MIT Lincoln Laboratory version of STAP
  - Analyze current STAP implementation
  - Parallelize MATLAB based linear Colored Loading Algorithms
  - Integrate converted Colored Loading Algorithm into KASSPER
  - Validate results/Evaluate Performance

Outline

- Background
- **Knowledge-Aided Signal Processing**
- Real-time Implementation
- Summary and Future Work

Space-Time Adaptive Processing

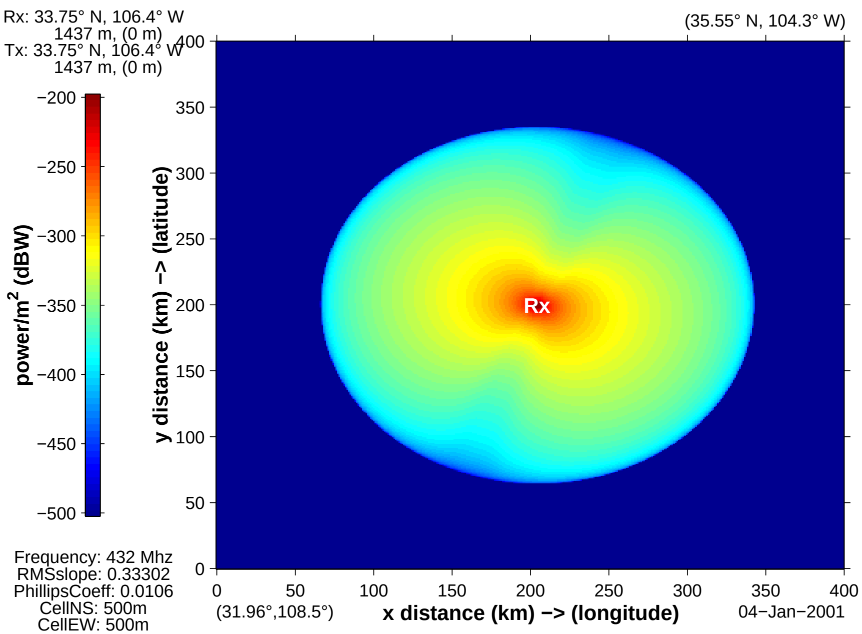




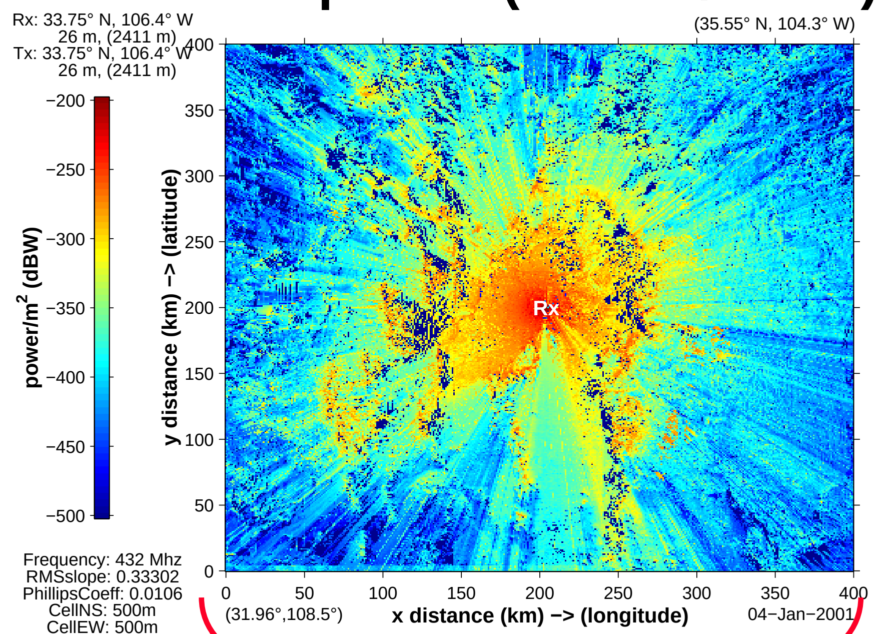
- Ground clutter response for airborne radars is spread in Doppler due to ownship motion
- Requires a 2D null in angle-Doppler space
- STAP uses an estimate of the observed clutter angle-Doppler response to form precise nulls for clutter cancellation
- STAP offers signal-to-clutter gains on the order of 40-50 dB
- Enabling technology for airborne radars → difficult to implement in the real world → KASSPER is addressing this problem

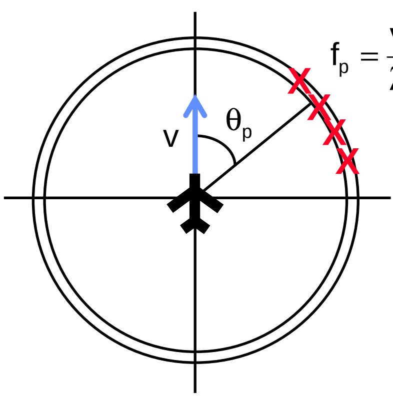
Site-specific Clutter Modeling (prior knowledge)

bald earth



site-specific (SCATS/DTED)





$$f_p = \frac{v}{\lambda}\cos(\theta_p)$$

ICM
calibration errors
channel mismatch

$$\mathbf{R}_c = \sum_{p=1}^{P_c} \left|\alpha_p\right|^2 \mathbf{v}(\theta_p, f_p)\mathbf{v}^H(\theta_p, f_p) \circ \mathbf{T}_p$$

sum over all scattering patches in a given range bin

Colored Loading

- **The prior knowledge covariance model is incorporated using a generalization of traditional diagonal loading**

$$\mathbf{w}_m = \alpha(\mathbf{R}_m + \beta_{d,m}\mathbf{R}_{c,m} + \beta_{L,m}\mathbf{I})^{-1}\mathbf{v}_m = \alpha(\mathbf{R}_m + \mathbf{Q}_m)^{-1}\mathbf{v}_m$$

$\mathbf{R}_m$: covariance estimated from data

$\mathbf{R}_{c,m}$: prior clutter model

$\beta_{L,m}\mathbf{I}$: diagonal loading

$\mathbf{Q}_m$: **"colored loading"**

- **The loading factors are chosen to a given desired level of performance**
- **The subscript m indicates that the loading matrix is different for each Doppler bin**
- **Diagonal loading will generally remain constant over time → colored loading will change as the platform moves**
- **The formulation shown here is in the covariance domain**

Data Domain Implementation

- **STAP is typically implemented in the 'data domain'**
  - Weights are computed directly from the data without computing an estimated covariance matrix
  - More numerically stable than covariance domain implementations
- **Diagonal loading is accomplished by augmenting the data matrix**

$$\mathbf{X}_a = [\mathbf{X} \quad \sqrt{\beta}\mathbf{I}] \Rightarrow \mathbf{R}_{s,L} = \mathbf{X}_a \mathbf{X}_a^H = \mathbf{R}_s + \beta\mathbf{I}$$

- **Typically, the *colored* loading matrix Q_m is Hermitian and positive-definite so that its Cholesky decomposition exists**

$$\mathbf{Q}_m = \beta_\mathrm{d}\mathbf{R}_{c,m} + \beta_\mathrm{L}\mathbf{I} = \mathbf{Q}_m^{1/2}\mathbf{Q}_m^{1/2} = \mathbf{C}_m^H \mathbf{C}_m$$

$\mathbf{C}_m$ is upper triangular

- **Efficient implementation achieved in the data domain by augmenting data matrix and using QR decomposition**

$$\mathbf{X}_a = [\mathbf{X}_m \quad \sqrt{K}\mathbf{C}^H] \rightarrow (1/K)\mathbf{X}_a \mathbf{X}_a^H = \mathbf{R}_s + \mathbf{Q}_m$$

- **Once the matrix C is computed, the STAP implementation is identical to existing STAP w/ diagonal loading**

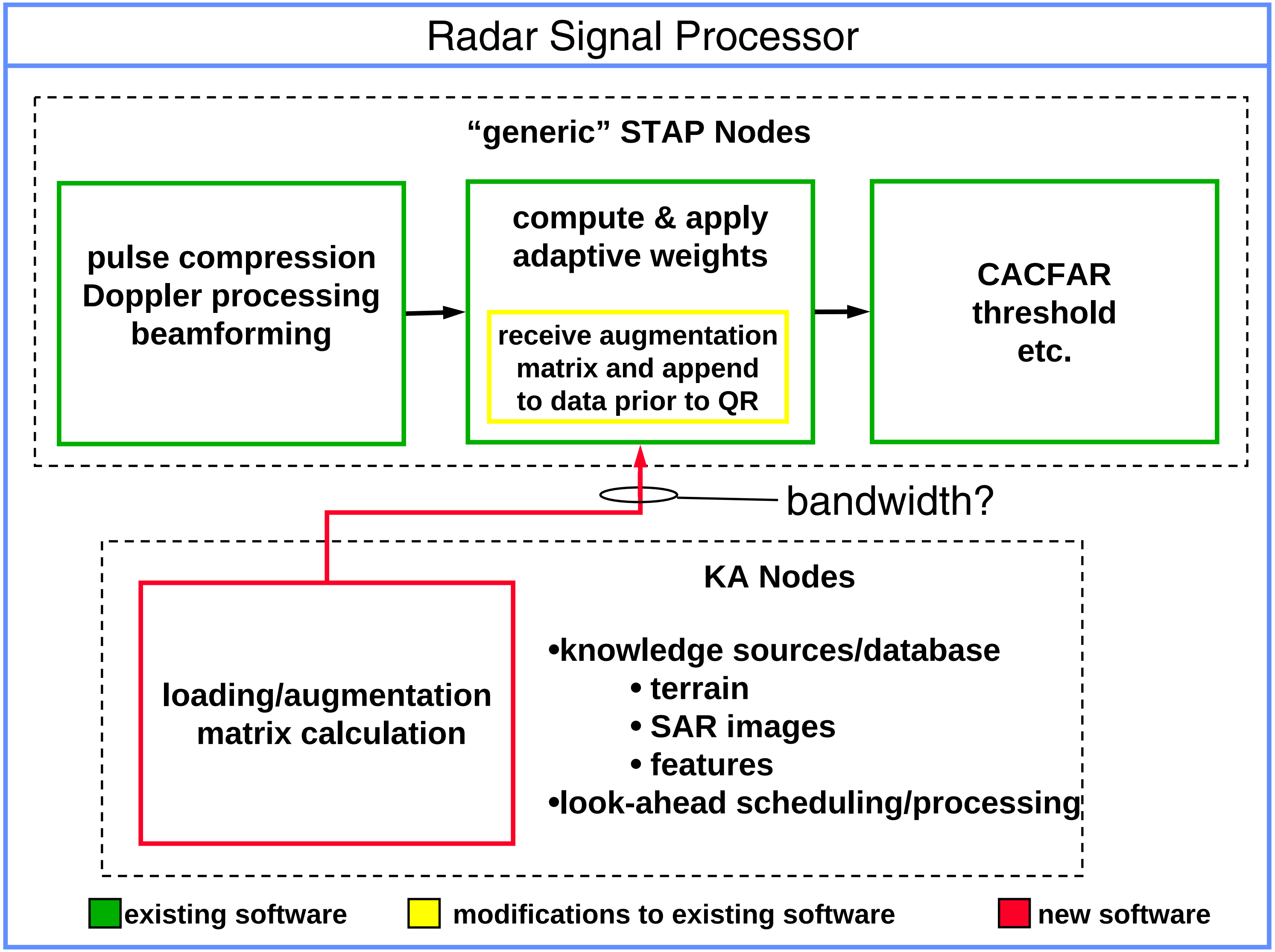
Implementation Issues

- **The information used in the colored loading matrix will often come from databases such as DTED, DFAD, SAR maps, etc.**
  – Large volumes of data → frequent updates as platform moves
  – Traditionally this information has not be available in the radar beamformer
- **A different loading matrix is required for each Doppler bin**
  – High fidelity forms of the clutter covariance model can be computationally complex
  – Bandwidth requirements

Outline

- **Background**
- **Knowledge-Aided Signal Processing**
- **Real-time Implementation**
- **Summary and Future Work**

Architecture



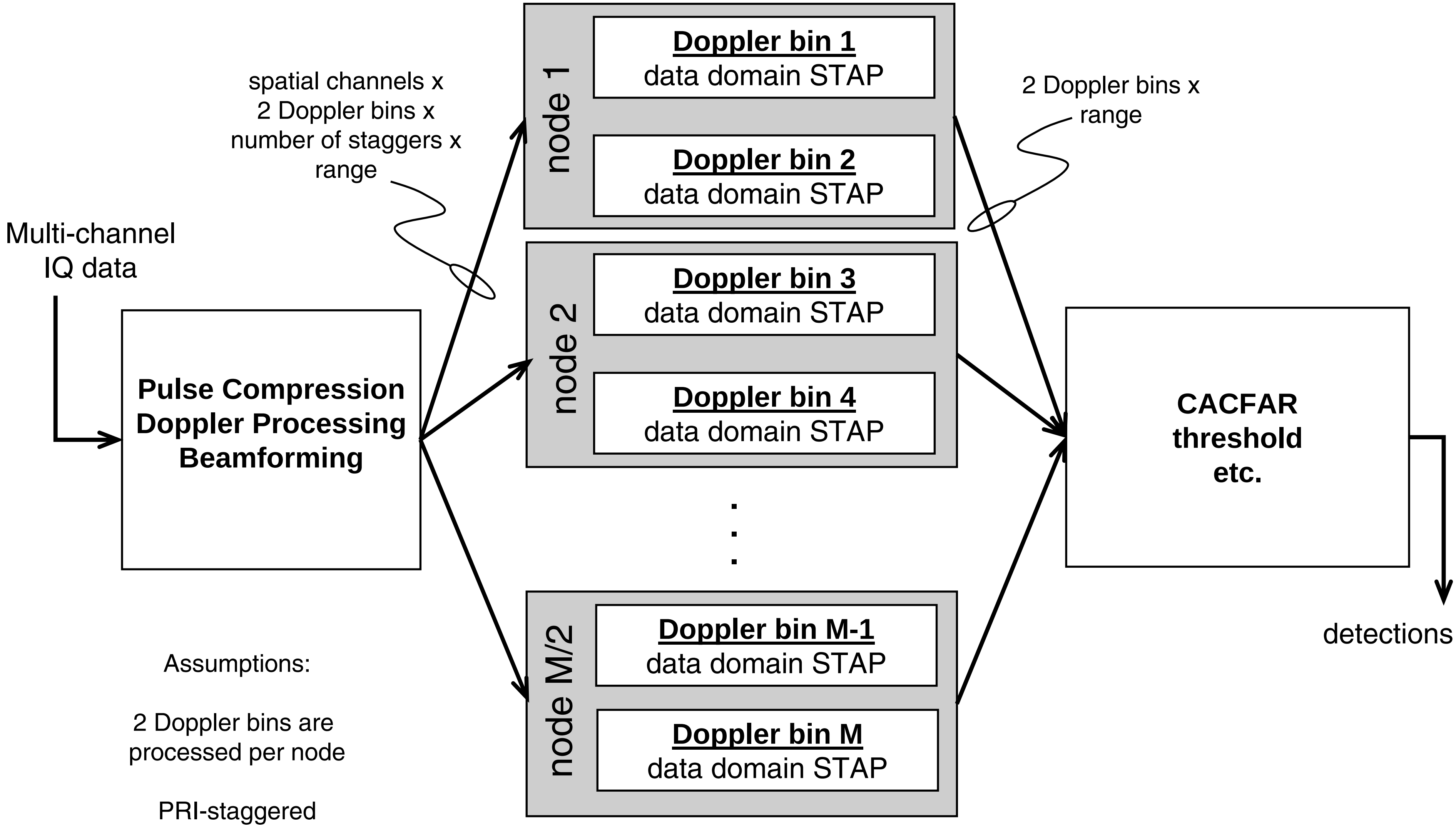
Bandwidth Requirements

- **Assumptions:**
  - One complex *D x D* loading/augmentation matrix is required per Doppler bin per CPI
    - » In covariance domain matrix is Hermitian symmetric
    - » In data domain matrix is all zeros below main diagonal
    - » Only need to send upper (or lower) triangular part of the matrix
  - 128 Doppler bins
  - 100 ms CPI length
  - 2 staggers, 3 beams (D = 6)
  - Double precision real and imaginary data
- **Bandwidth required to communicate augmentation matrices**

$$BW = \frac{(128\text{bits})(128)(D(D+1)/2)}{100\text{ms}} = 3.4\text{Mbps}$$

- **May want to only apply this technique to a small number of Doppler bins**

STAP Algorithm Partitioning Example

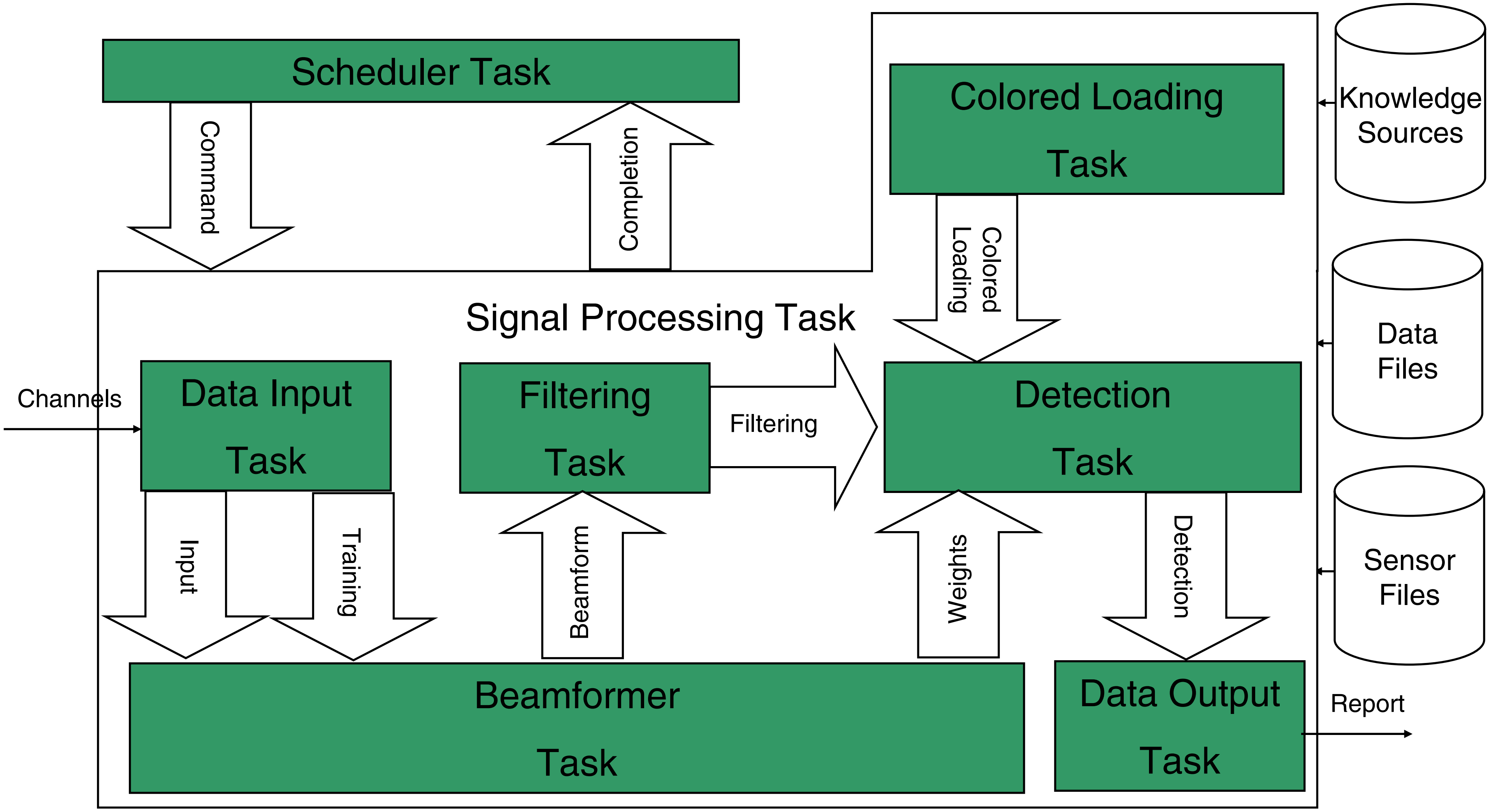


Assumptions:

2 Doppler bins are processed per node

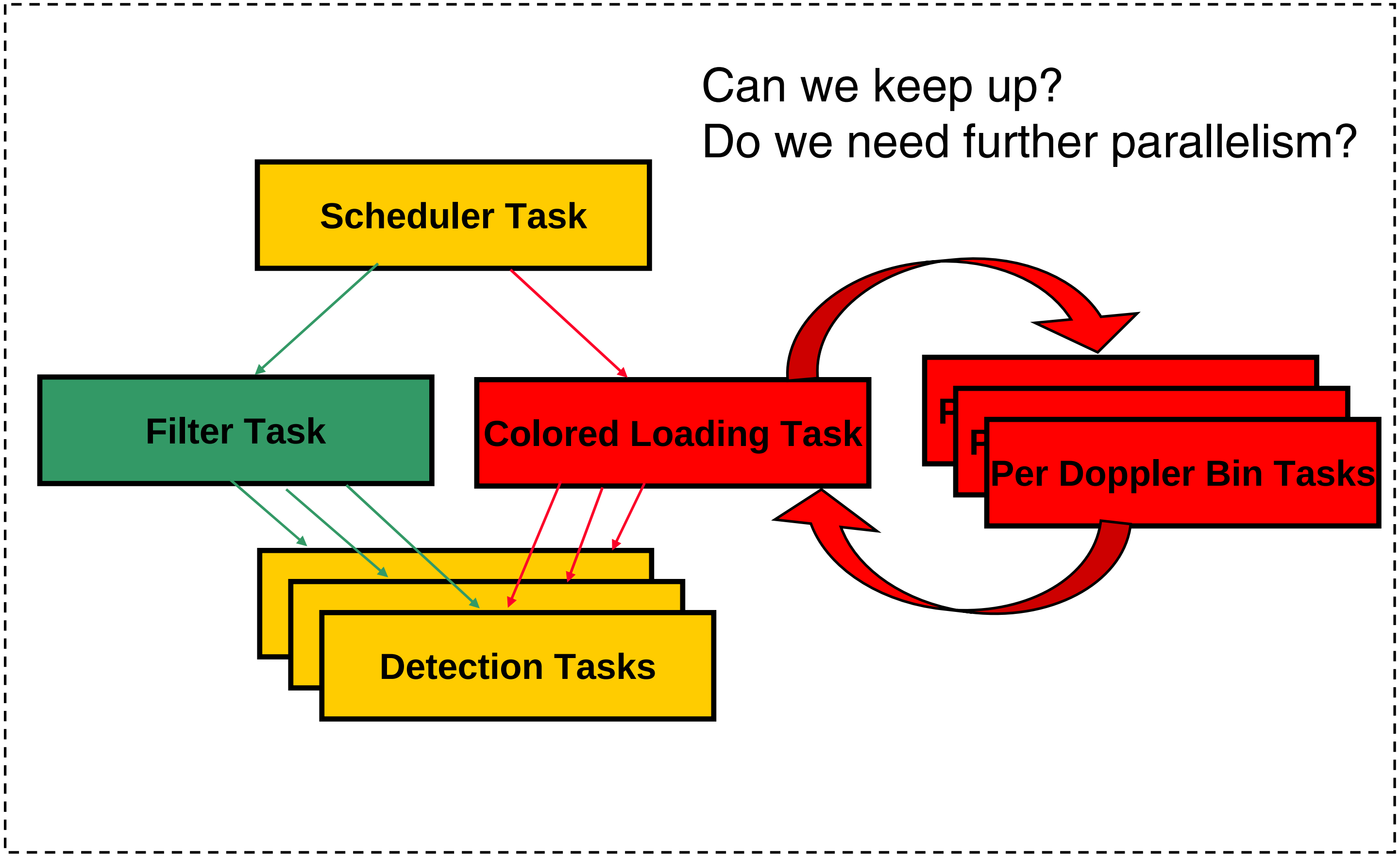
PRI-staggered post-Doppler STAP

STAP Application Top Level Diagram + Colored Loading



Architecture Changes

Colored Loading Integration

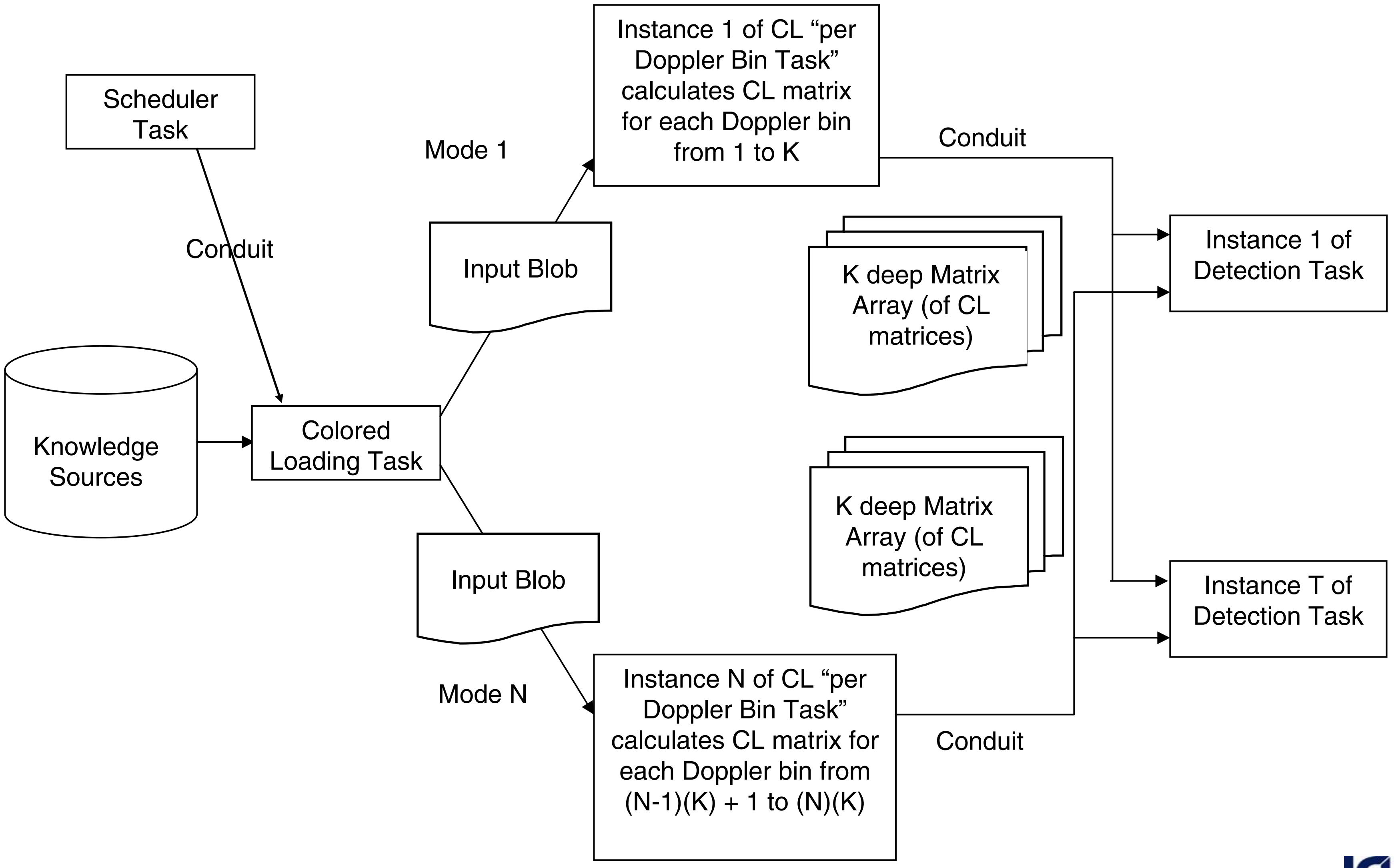


Unchanged

Modified

New

Colored Loading Flow Diagram

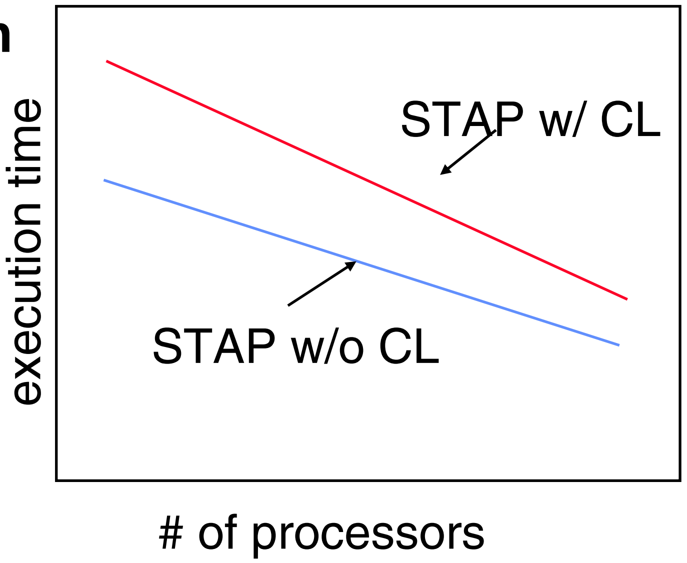


Outline

- Background
- Knowledge-Aided Signal Processing
- Real-time Implementation
- **Summary and Future Work**

Summary and Future Work

- **Timing Tests/Bandwidth Validation**
- **MIT Cluster Implementation**
- **VSIPL++ Implementation**
- **Lessons Learned**



Questions?

- **Bob Giannaris** bgiannaris@islinc.com
- **Jamie Bergin** jsb@islinc.com
- **John Don Carlos** jdc@islinc.com

Data Passing via Conduits