High Performance Parallel Implementation of Adaptive Beamforming Using Sinusoidal Dithers

High Performance Embedded Computing Workshop

Peter Vouras and Trac D. Tran

Johns Hopkins University

September 19, 2007

Overview

- **Gradient Estimation Using Sinusoidal Dithers**
- **Application to Adaptive Beamforming**
- **Parallel Implementation**
- **Results**
- **Conclusion**

Gradient Estimation Using Dithers

- **Consider any objective function *f*(*x*) where *x* is an *N*-by-1 vector. For each component of *x*, superimpose a sinusoidal dither of different frequency, as in**

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\theta} = \boldsymbol{x} + \alpha\left[\cos\left(\omega_1 t\right), \quad \cos\left(\omega_2 t\right), \ldots, \cos\left(\omega_N t\right)\right]^T$$

where *α* is a small scalar. The Taylor series expansion of *f*(*x*) yields,

$$f\left(\boldsymbol{x}'\right) = f\left(\boldsymbol{x} + \boldsymbol{\theta}\right) = f\left(\boldsymbol{x}\right) + \nabla f\left(\boldsymbol{x}\right)^T \boldsymbol{\theta} + \tfrac{1}{2}\boldsymbol{\theta}^T \nabla^2 f\left(\boldsymbol{x}\right)\boldsymbol{\theta} + \ldots$$

$$= f\left(\boldsymbol{x}\right) + \alpha \sum_{i=1}^{N} \left.\frac{\partial f}{\partial x_i}\right|_{\boldsymbol{x}} \cos\left(\omega_i t\right) + \ldots$$

Gradient Estimation Using Dithers - 2

- **The components of the gradient vector can be determined exactly after we multiply *f*(*x'*) by cos($\omega_j t$), for *j* = 1, 2, …, *N*. The result after using trigonometric identities is,**

$$f(\boldsymbol{x}')\cos(\omega_j t) = f(\boldsymbol{x})\cos(\omega_j t) + \frac{\alpha}{2}\left.\frac{\partial f}{\partial x_j}\right|_{\boldsymbol{x}}$$

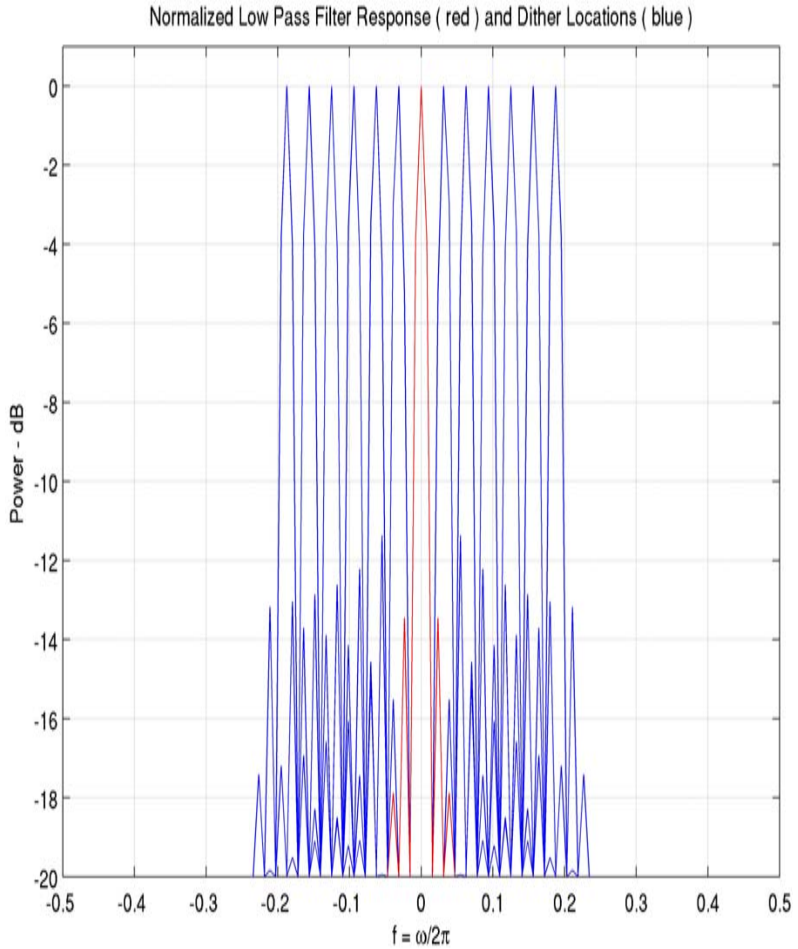
$$+\frac{\alpha}{2}\left.\frac{\partial f}{\partial x_j}\right|_{\boldsymbol{x}}\cos(2\omega_j t) + \frac{\alpha}{2}\sum_{i\neq j}\left.\frac{\partial f}{\partial x_i}\right|_{\boldsymbol{x}}\cos([\omega_i - \omega_j]t)$$

$$+\frac{\alpha}{2}\sum_{i\neq j}\left.\frac{\partial f}{\partial x_i}\right|_{\boldsymbol{x}}\cos([\omega_i + \omega_j]t) + H.O.T.$$

- **Note that the only constant term on the right hand side is the *j*th component of the gradient vector scaled by the factor $\alpha/2$, and can be recovered exactly by low-pass filtering *f*(*x'*)cos($\omega_j t$).**

Dither Application

- Dithers can be applied sequentially or in parallel
- If applied in parallel, care must be taken to avoid crosstalk between adjacent channels which may corrupt the gradient estimate
  - Requires high sampling rate, and narrow low pass filter with steep transition regions



Advantages of Using Dithers in Adaptive Beamforming

PROs

- **Estimates gradient exactly and results in higher SINR than other recursive algorithms**
- **Very scalable algorithm**
  - **Each dither computation is independent of others and can therefore be distributed across parallel processors**

CONs

- **Increased computations and FLOP count**

Brief Overview of Adaptive Beamforming

- **Linearly Constrained Minimum Variance (LCMV) Beamformer**
- **Minimize output power of an array, $y(n) = \boldsymbol{w}^H \boldsymbol{u}(n)$, subject to a set of linear constraints**

$$\underset{\boldsymbol{w}}{\textbf{minimize}} \quad J(n) = E\left[\left|y(n)\right|^2\right]$$

such that $\boldsymbol{C}^H \boldsymbol{w} = \boldsymbol{f}$.

Decompose $\boldsymbol{w}$ into,

$$\boldsymbol{w} = \boldsymbol{w}_q - \boldsymbol{B}\boldsymbol{w}_a$$

$$y(n) = \boldsymbol{w}_q^H \boldsymbol{u}(\boldsymbol{n}) - \boldsymbol{w}_a^H \boldsymbol{B}^H \boldsymbol{u}(\boldsymbol{n}).$$

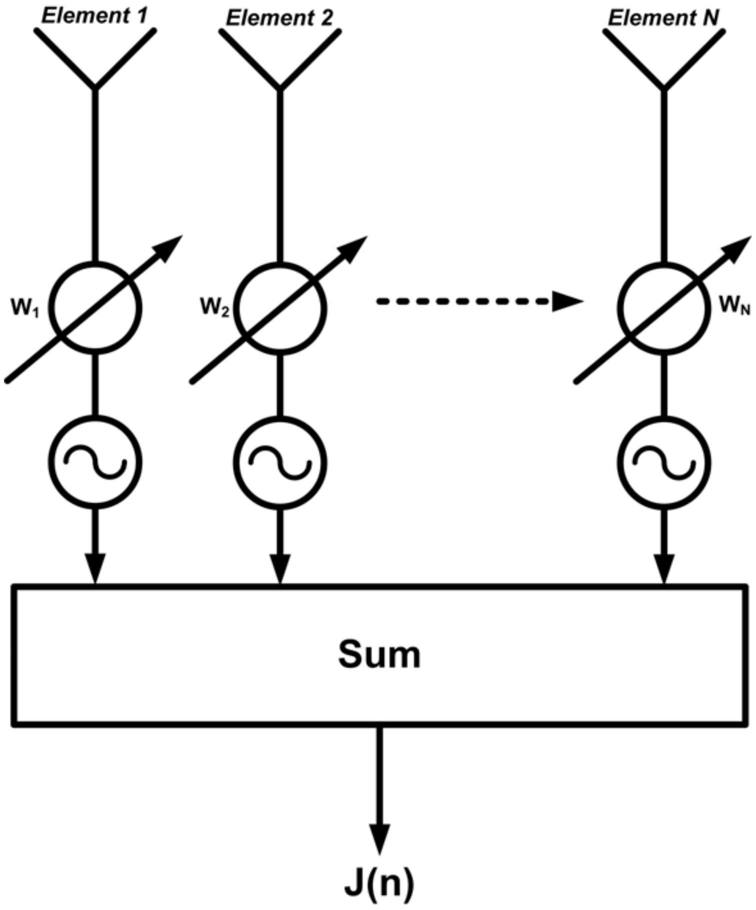
Rewrite as unconstrained program,

$$\underset{\boldsymbol{w}_a}{\textbf{minimize}} \quad J(n)$$

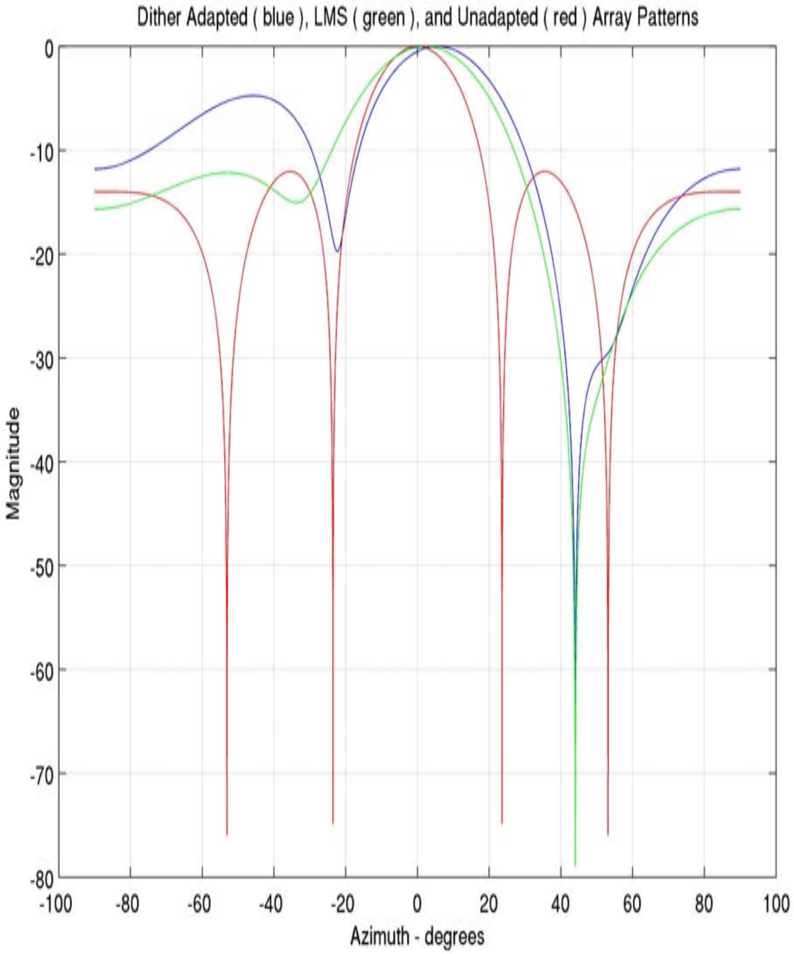
$$\boldsymbol{w}_a(n+1) = \boldsymbol{w}_a(n) - \mu \nabla_{\boldsymbol{w}_a} J(n).$$

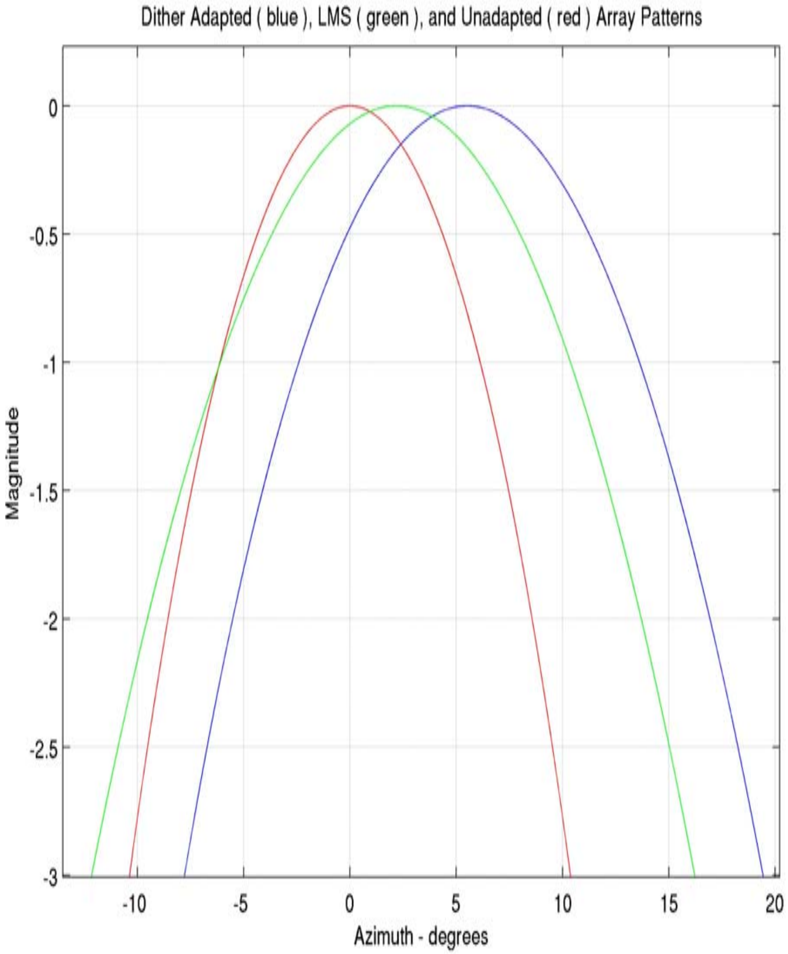
Adaptive Array Configuration With Dithers

- **Separate dither must be applied to real and imaginary part of each array element weight**



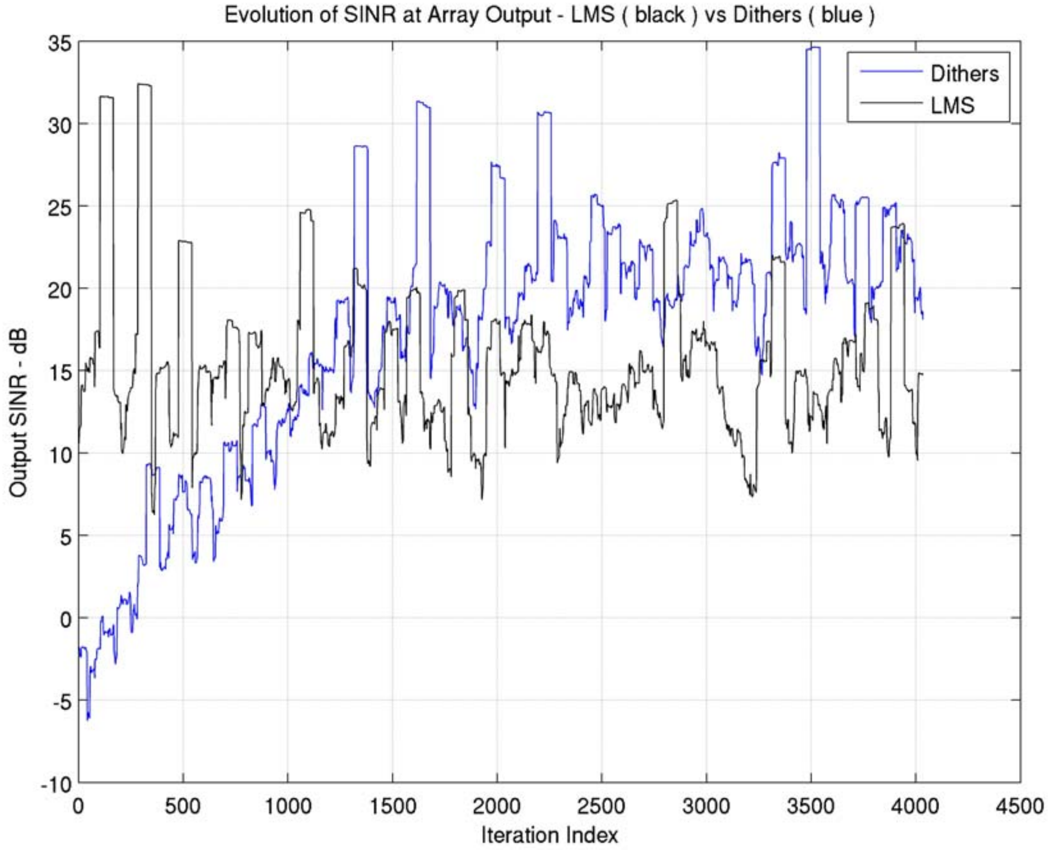
Adaptive Array Performance



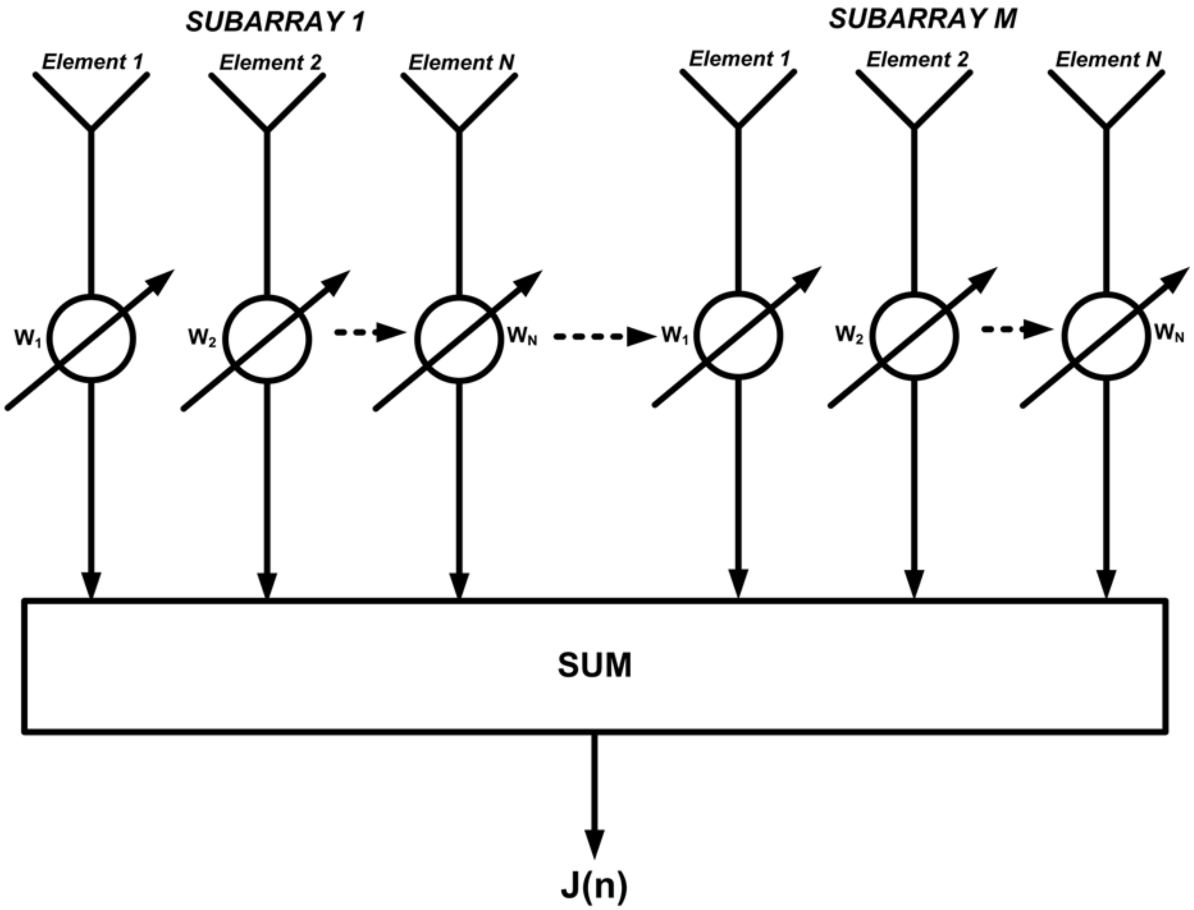


Jammer at 44º azimuth

Adaptive Array Performance - 2

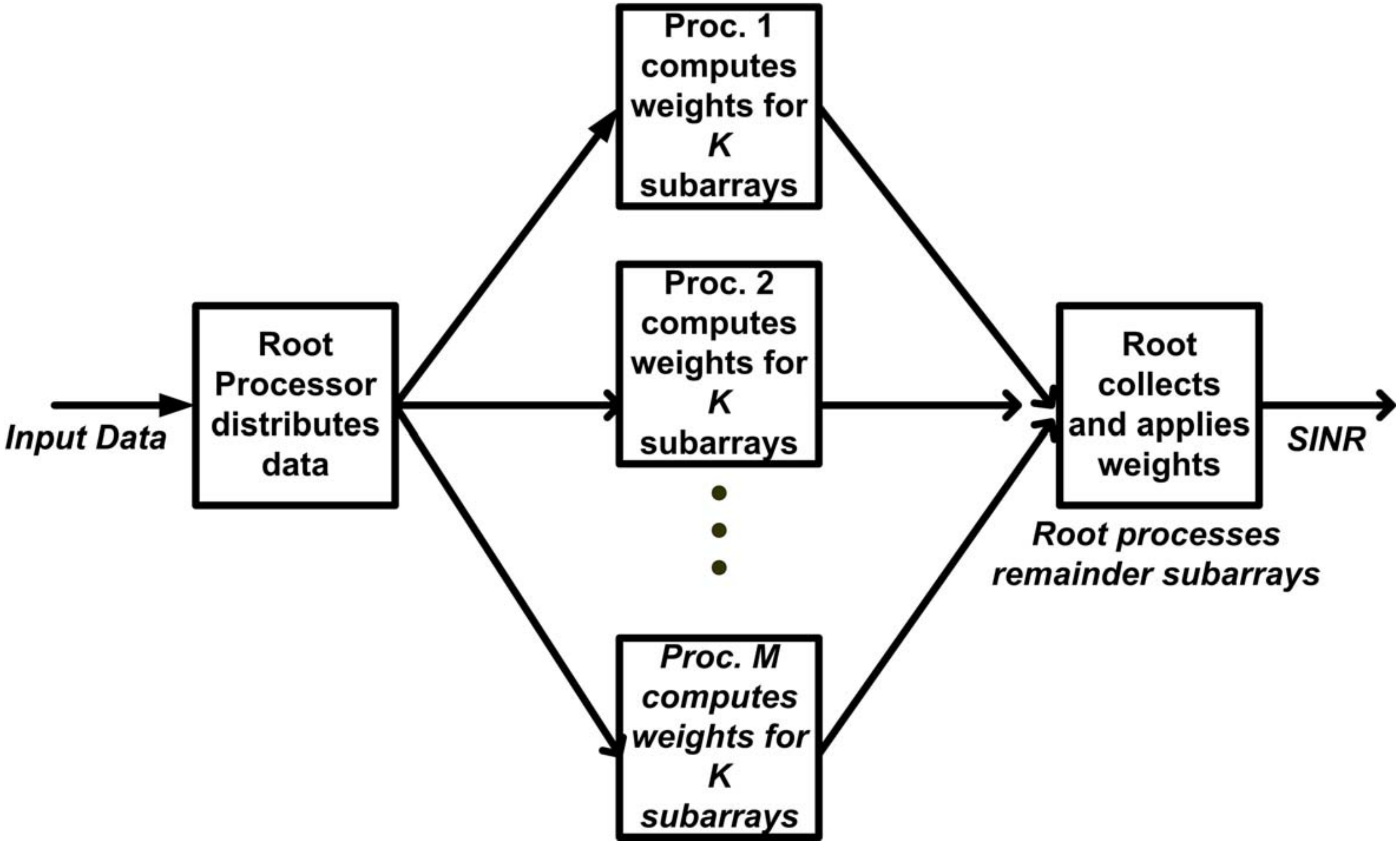


Subarray Architecture



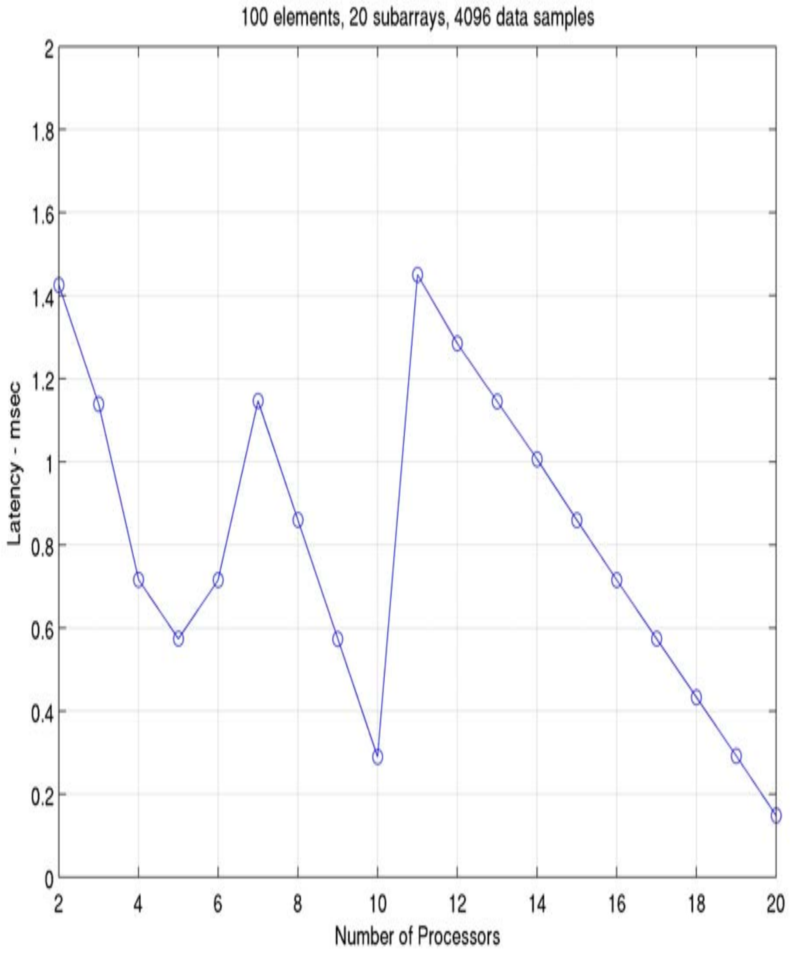
For this study, techniques to minimize grating lobes were not considered

Software Design

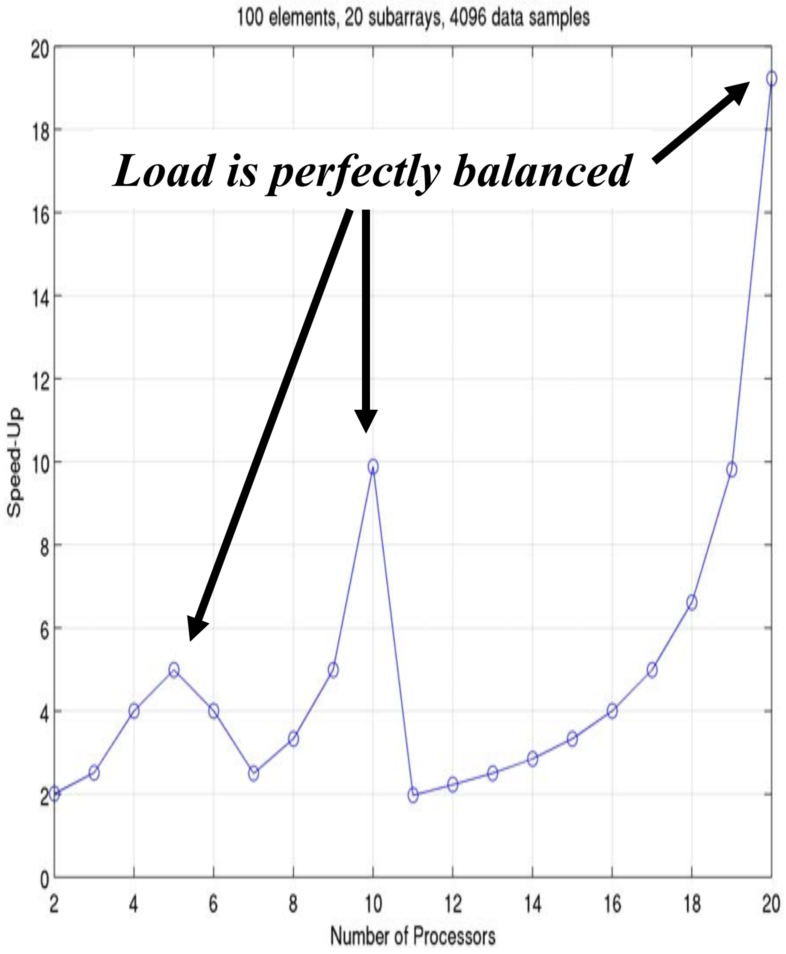


MPI Implementation on Cray XD1

Measured Results



Scalability



Speed-Up

Conclusions

- **Directly estimating the components of the gradient vector using sinusoidal dithers may improve the performance of steepest descent algorithms**
  - **Adaptive beamforming**
- **Subarray implementation of adaptive beamformer evaluated on Cray XD1 using parallel processors**
  - **Convolution (dot product) and mixing operations may be implemented on FPGAs for further performance improvements**