Constraint Adaptive Natural Gradient Algorithms for Adaptive Array Processing

Ijteba-ul-Hasnain Shah & TS Durrani
Constraint optimization is not trivial and generally involves manifolds with non linear surfaces. Working with a search space that carries the nonlinear manifolds introduces certain challenges in implementation of the algorithm. Any optimisation scheme that exploits the given structure of the underlying space is deemed successful with better convergence and accuracy. Since the procedure of finding optimisers is exactly a search based on the geometric information of both the constraints and the cost function, it is very important to develop search techniques using the intrinsic geometry properties of both functions. Novel Constraint Adaptive Natural Gradient Algorithms (CANA) have been developed exploiting the Natural Gradient technique, and have been applied to problems in Array Processing. The Algorithms are capable of rapidly adjusting the response of an array of sensors to a signal coming from Direction of Interest (DoI) and suppressing signals incident on the array from other directions. The algorithms provide more uniform performance than those based upon conventional tangential gradients, yet are simple to implement. Along with the ability of natural gradient algorithm to follow the exact surface of the optimization space, self correcting feature of constraint optimization makes the algorithms rapidly converge to steady feasible points. Numerical simulations confirm its suitability for Beam Forming, Null Steering and mitigating against hostile interferences.
To “receive” specified signals and “reject” rest of it in the presence of “noise”
Constraint Adaptive Array

• Consider a array processor with the output $y(n)$,
  \[ y(n) = w^T x \]
  Where $w$ is the weight vector and $x$ is the incoming data vector.

• Minimize $E[y^2(n)]$ with some look direction constraint.
  \[ c^T w = f \]
Constraint Adaptive Natural Gradient Algorithm

\[ w_{n+1} = P[w_n + \mu G^{-1} \left( \frac{\Delta J(w)}{\Delta w_n} \right)] + K \]

\[ P = \left[ I - G^{-1} c c^T \right] / c^T G^{-1} c \]

\[ K = G^{-1} c / c^T G^{-1} c \]

G is the Riemannian metric Constraint Adaptive Natural Gradient Algorithm

If \( G = I \) then the above weight equation become CLMS or Frost’s solution.

What is G?
Riemannian metric

• The cost function optimisation involves finding “minimum” or “feasible” point of the surface.
• The optimisation surfaces are often non linear manifolds.
• Conventional gradients are unable to approximate the surfaces.
• Hence Natural gradient- based upon Riemannian metric is required to approximate the surface.
Parallel transpositions along a) plane and b) curved manifolds
\[
G^{-1} = [I - w_n w_n^H]
\]
\[
G^{-1} = [\epsilon I - \nu w_n w_n^H]
\]
\[
\nu = [\epsilon / (1 + \epsilon w_n^H w_n)]
\]
\[
\nabla J_w = G^{-1} \nabla J_w
\]

Natural Gradient
Convergence Characteristics of CANA

Convergence characteristics of CANA, CLMS and Newton algorithms
Broadside Beam Forming & Cancellation of Interference

Broadside Beam Forming with 8 element Array-Equal power Interference sources
Broadside Beam Forming with 8 element Array-Unequal power Interference sources
<table>
<thead>
<tr>
<th>S/No.</th>
<th>Input I/N ratio dB</th>
<th>Null power at the position of Interferer dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>-57.87</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>-60.0</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>-82.92</td>
</tr>
<tr>
<td>4</td>
<td>23.0103</td>
<td>-89.85</td>
</tr>
<tr>
<td>5</td>
<td>24.7712</td>
<td>-93.9</td>
</tr>
<tr>
<td>6</td>
<td>26.0206</td>
<td>-96.77</td>
</tr>
<tr>
<td>7</td>
<td>26.98</td>
<td>-99.00</td>
</tr>
<tr>
<td>8</td>
<td>27.7815</td>
<td>-100.8</td>
</tr>
<tr>
<td>9</td>
<td>28.4510</td>
<td>-102.4</td>
</tr>
<tr>
<td>10</td>
<td>29.0309</td>
<td>-103.7</td>
</tr>
<tr>
<td>11</td>
<td>29.5424</td>
<td>-104.9</td>
</tr>
<tr>
<td>12</td>
<td>30.00</td>
<td>-105.9</td>
</tr>
<tr>
<td>13</td>
<td>40.00</td>
<td>-129</td>
</tr>
<tr>
<td>14</td>
<td>43.0103</td>
<td>-135</td>
</tr>
<tr>
<td>15</td>
<td>44.7712</td>
<td>-139.9</td>
</tr>
<tr>
<td>16</td>
<td>50.00</td>
<td>-152.0</td>
</tr>
</tbody>
</table>

Relationship between I/N ratio and Null power/depth
Relationship between I/N ratio and Null power/depth
Multi Beam Antenna

- By changing the constraint vector $c$ and constraint set $f$
Accuracy of CANA

<table>
<thead>
<tr>
<th>Element Number</th>
<th>Optimum Weight</th>
<th>CLMS Weight</th>
<th>CANA Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.10718</td>
<td>0.13652</td>
<td>0.10718</td>
</tr>
<tr>
<td>2</td>
<td>0.12485</td>
<td>0.09797</td>
<td>0.12485</td>
</tr>
<tr>
<td>3</td>
<td>0.12755</td>
<td>0.13293</td>
<td>0.12755</td>
</tr>
<tr>
<td>4</td>
<td>0.12801</td>
<td>0.12331</td>
<td>0.12801</td>
</tr>
<tr>
<td>5</td>
<td>0.12809</td>
<td>0.15755</td>
<td>0.12809</td>
</tr>
<tr>
<td>6</td>
<td>0.12811</td>
<td>0.14669</td>
<td>0.12811</td>
</tr>
<tr>
<td>7</td>
<td>0.12811</td>
<td>0.12095</td>
<td>0.12811</td>
</tr>
<tr>
<td>8</td>
<td>0.12811</td>
<td>0.08408</td>
<td>0.12811</td>
</tr>
</tbody>
</table>

Comparison of weights calculated by CANA (NG) and CLMS (CG) algorithms
Comparison of weights calculated by Natural (NG) and Conventional Gradient based (CLMS) algorithms
Lobe structure comparison between CANA and CLMS
## Computational Complexity of CANA

<table>
<thead>
<tr>
<th>Constraint Adaptive Algorithm</th>
<th>Computational Complexity [multiplications]</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLMS</td>
<td>$3p^2 + p$</td>
</tr>
<tr>
<td>CANA</td>
<td>$5p^2 + p$</td>
</tr>
</tbody>
</table>

Complexity Analysis of Constraint Adaptive Algorithms
Per iteration complexity comparison of CANA and CLMS
Overall Computational Complexity
Constraint Adaptive Natural Gradient Algorithm with Non Adaptive Metric (CANA-NAM)

\[ w_{n+1} = P[w_n + \mu G^{-1}(\frac{\Delta J(w)}{\Delta w_n})] + K \]

G is a fixed matrix
Convergence comparison of CANA-NAM with CLMS
Effect of regularization parameter \( \zeta \) on convergence of CANA-NAM
Adaptive Step-size Algorithm

\[ \alpha_n = - \frac{d_n R w'_{n+1}}{d_n^T d_n} \]

\[ w'_{n+1} = P w_n + k \]

\[ d_n = P G^{-1} R w_n \]
Optimum value for step size $\alpha$ for Adaptive Step size Algorithm
Convergence comparison of Adaptive Stepsize Algorithm with fixed stepsize
CANA in polar space

\[ J(r, \theta) = [r \cos \theta \hspace{1em} r \sin \theta] \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} [r \cos \theta \hspace{1em} r \sin \theta]^H + \lambda [c_1 \hspace{1em} c_2]^T [r \cos \theta \hspace{1em} r \sin \theta] - 1 \]

\[ w_{n+1} = P[w_n - \mu H_1] + b \]

\[ P = [I - Bc^T] \]

\[ b = Qe \left[ e^T Qe \right]^{-1} \]

\[ G = \begin{bmatrix} 1 & 0 \\ 0 & r^2 \end{bmatrix} \]

\[ Q_1 = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta / r & \cos \theta / r \end{bmatrix} \]
Constrained Broadband Array Processing in Wave-Number Domain

\[ x_m(t) = \sum_{p=1}^{N} \int e^{jk(ct-md\sin(\theta_p))} \, d\varphi_p(k) + v_m(t) \]
$h_{opt}(k) = \frac{1}{1 + a_{opt}^r (e^{jk})}$
Optimum Array Response-Normalised
dB Pattern
Multiple Broadband Interference Cancellation
Constant Main-lobe Response
dB pattern
Wave-number Filter steering
Conclusion

• Better convergence
• Lesser overall computational complexity
• Simplicity
• Ideal for practical implementation
• Wave-number techniques for broadband