ECE 6123
Advanced Signal Processing
Adaptive Filtering with LMS

Peter Willett

Fall 2017

1 The Gradient Descent Idea

1.1 Finding the Wiener Filter

Suppose we begin with the Weiner filtering cost function

$$J(\mathbf{w}) = \mathcal{E}\{|d[n] - \mathbf{w}^H \mathbf{u}_n\} \tag{1}$$
$$= \sigma_d^2 - 2\Re\{\mathbf{w}^H \mathbf{p}\} + \mathbf{w}^H \mathbf{R}\mathbf{w} \tag{2}$$

where the terms are as usual. Suppose further that we wish to achieve this iteratively, as opposed to in one step as before. One might reasonably: Why do this when you manifestly *can* get to the solution in one step? The answer is that we will attempt to see how this can work when the correlations $\mathbf{p}$ and $\mathbf{R}$ are not knows or are changing. But that will come.

The basic idea is to observed that one can reduce the error by moving in a direction of "steepest-descent" which is

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \mu \nabla J_{\mathbf{w}}(\mathbf{w}_{n-1}) \tag{3}$$

where μ is some small step size and of course $\nabla_{\mathbf{w}}$ represents the gradient with respect to $\mathbf{w}$.

1.2 Interlude about Complex Gradients

This subsection is eminently skippable. However, it is not perhaps obvious how to take a gradient of a linear of quadratic form when complex vectors are involved. The answer.turns out to be: it's exactly what you think it is. We begin with (2), and write

$$\mathbf{w} \equiv \mathbf{w}_r + j\mathbf{w}_i \tag{4}$$
$$\mathbf{p} \equiv \mathbf{p}_r + j\mathbf{p}_i \tag{5}$$
$$\mathbf{R} \equiv \mathbf{R}_r + j\mathbf{R}_i \tag{6}$$

where all RHS vectors and matrices are real and clearly the subscript refer to real and imaginary parts. We have (2) as

$$\begin{aligned} J(\mathbf{w}) &= \sigma_d^2 - 2\mathbf{w}_r^T\mathbf{p}_r - 2\mathbf{w}_i^T\mathbf{p}_i + \mathbf{w}_r^T\mathbf{R}_r\mathbf{w}_r \\ &\quad - \mathbf{w}_r^T\mathbf{R}_i\mathbf{w}_i + \mathbf{w}_i^T\mathbf{R}_i\mathbf{w}_r + \mathbf{w}_i^T\mathbf{R}_r\mathbf{w}_i \qquad (7) \\ &= \sigma_d^2 - 2\mathbf{w}_r^T\mathbf{p}_r - 2\mathbf{w}_i^T\mathbf{p}_i + \mathbf{w}_r^T\mathbf{R}_r\mathbf{w}_r \\ &\quad - 2\mathbf{w}_r^T\mathbf{R}_i\mathbf{w}_i + \mathbf{w}_i^T\mathbf{R}_r\mathbf{w}_i \qquad (8) \\ &= \sigma_d^2 - 2\mathbf{w}_r^T\mathbf{p}_r - 2\mathbf{w}_i^T\mathbf{p}_i + \mathbf{w}_r^T\mathbf{R}_r\mathbf{w}_r \\ &\quad + 2\mathbf{w}_i^T\mathbf{R}_i\mathbf{w}_r + \mathbf{w}_i^T\mathbf{R}_r\mathbf{w}_i \qquad (9) \end{aligned}$$

Our aim in going from (7) to (8) & (9) is to isolate the real or imaginary parts as row-vectors in the the inner products, and we have used the fact that since $\mathbf{R}^H = \mathbf{R}$ we have $\mathbf{R}_r^T = \mathbf{R}_r$ and $\mathbf{R}_i^T = -\mathbf{R}_i$. We have

$$\nabla\mathbf{w}_r J(\mathbf{w}) = -2\mathbf{p}_r + 2\mathbf{R}_r\mathbf{w}_r - 2\mathbf{R}_i\mathbf{w}_r \qquad (10)$$

$$\nabla\mathbf{w}_i J(\mathbf{w}) = -2\mathbf{p}_i + 2\mathbf{R}_i\mathbf{w}_r + 2\mathbf{R}_r\mathbf{w}_i \qquad (11)$$

via (8) and (9) respectively. We therefore write

$$\begin{aligned} \nabla\mathbf{w} J(\mathbf{w}) &= \nabla\mathbf{w}_r J(\mathbf{w}) + j\nabla\mathbf{w}_i J(\mathbf{w}) \qquad (12) \\ &= -2\mathbf{p}_r + 2\mathbf{R}_r\mathbf{w}_r - 2\mathbf{R}_i\mathbf{w}_r \\ &\quad j(-2\mathbf{p}_i + 2\mathbf{R}_i\mathbf{w}_r + 2\mathbf{R}_r\mathbf{w}_i) \qquad (13) \\ &= -2(\mathbf{p}_r + j\mathbf{p}_i) + 2\mathbf{R}_r(\mathbf{w}_r + j\mathbf{w}_i) \\ &\quad -2\mathbf{R}_i(j\mathbf{w}_r - \mathbf{w}_i) \qquad (14) \\ &= -2\mathbf{p} + 2\mathbf{R}\mathbf{w} \qquad (15) \end{aligned}$$

This is the answer you might expect and might even know.

2 The LMS Algorithm

Equation (15) might seem to give us the way to update (3). One approach might be to estimate – say, by a block average – the required correlations $\mathbf{p}$ and $\mathbf{R}$ and perform exactly that[1]. However, This does, however, require a certain amount of computation overhead in terms of the solution to a set of linear equations; and the block-average idea is not especially reactive to changes. A better idea in terms of the latter would be a "forgetting factor" sort of average. On the surface one is left with the $\mathcal{O}(M^3)$ computational

[1] Of course this would beg the question as to why not simply go directly to the Wiener solution directly via the linear equations.

load, but in fact the exponential average is exactly what we shall see when we discuss the RLS algorithm, and its update will be shown cleverly to be $\mathcal{O}(M^2)$. The LMS update is $\mathcal{O}(M)$, meaning that quite lengthy filters are easily in reach.

We need estimators for $\mathbf{p}$ and $\mathbf{R}$, and LMS espouses the very simplest:

$$\hat{\mathbf{p}} = \mathbf{u}_n d[n]^* \tag{16}$$

$$\hat{\mathbf{R}} = \mathbf{u}_n \mathbf{u}_n^H \tag{17}$$

Note that both are (by definition) unbiased estimators. We thus have the LMS update

$$\begin{aligned} \mathbf{w}_n &= \mathbf{w}_{n-1} - \mu(-\mathbf{u}_n d[n]^* + \mathbf{u}_n \mathbf{u}_n^H \mathbf{w}_{n-1}) && (18) \\ &= \mathbf{w}_{n-1} + \mu \mathbf{u}_n (d[n]^* - \mathbf{u}_n^H \mathbf{w}_{n-1}) && (19) \\ &= \mathbf{w}_{n-1} + \mu \mathbf{u}_n e[n]^* && (20) \end{aligned}$$

where we have absorbed the constant 2 into the unspecified μ, and of course we have

$$\hat{d}[n] = \mathbf{w}_{n-1}^H \mathbf{u}_n \tag{21}$$

as the filter output at time n.

3 LMS Analysis

3.1 Discussion

There are many ways to analyze the LMS algorithm. The way shown in the text is excellent but complicated. I used to teach it, but in later years I've come to the conclusion that it provides inexact but good answers for very restrictive assumptions. In short, it is a lot of effort that provides a very sharp answer that is not especially intuitive, isn't exact nor applies when things are not Gaussian[2]. What we seek is guidance about μ: how large should it be? Clearly a large μ is a concern in that it may "go unstable" and throw $\mathbf{w}_n$ wildly in various directions. A small μ avoids this, and has the additional benefit that the added "gradient noise" in steady state (due to continual vacillations in $\mathbf{w}_n$) can be reduced. But a filter with a small μ may take a long time to converge.

[2] I have seen conference presentations and journal papers that purport to give exact answers but are quite indigestible, both in development and solution.

3.2 Convergence

We begin with

$$\mathbf{w}_n \;=\; \mathbf{w}_{n-1} \;-\; \mu(-\mathbf{u}_n d[n]^* + \mathbf{u}_n \mathbf{u}_n^H \mathbf{w}_{n-1}) \tag{22}$$

and define $\varepsilon_n \equiv \mathbf{w}_n - \mathbf{R}^{-1}\mathbf{p}$ to be the filter error. We have

$$\begin{aligned}
\varepsilon_n &= \varepsilon_{n-1} \;-\; \mu(-\mathbf{u}_n d[n]^* + \mathbf{u}_n \mathbf{u}_n^H \mathbf{w}_{n-1}) & (23)\\
\mathcal{E}\{\varepsilon_n\} &= \mathcal{E}\{\varepsilon_{n-1}\} \;-\; \mu(-\mathcal{E}\{\mathbf{u}_n d[n]^*\} + \mathcal{E}\{\mathbf{u}_n \mathbf{u}_n^H \mathbf{w}_{n-1}\}) & (24)\\
\mathcal{E}\{\varepsilon_n\} &= \mathcal{E}\{\varepsilon_{n-1}\} \;-\; \mu(-\mathbf{p} + \mathcal{E}\{\mathbf{u}_n \mathbf{u}_n^H \mathbf{w}_{n-1}\}) & (25)
\end{aligned}$$

We claim that we can write

$$\mathcal{E}\{\mathbf{u}_n \mathbf{u}_n^H \mathbf{w}_{n-1}\} \;=\; \mathcal{E}\{\mathbf{u}_n \mathbf{u}_n^H\}\mathcal{E}\{\mathbf{w}_{n-1}\} \tag{26}$$

and offer the justification that whatever dependence there may be between $\mathbf{u}_n$ and $\mathbf{w}_{n-1}$ – and in the case that $\{\mathbf{u}_n\}$ forms an independent sequence there would be none – it is second-order. That is, the changes in $\mathbf{w}_{n-1}$ arising from recent $\mathbf{u}_n$'s are small perturbations around the expected value. As such we claim

$$\begin{aligned}
\mathcal{E}\{\varepsilon_n\} &\approx \mathcal{E}\{\varepsilon_{n-1}\} \;-\; \mu(-\mathbf{p} + \mathbf{R}\mathcal{E}\{\mathbf{w}_{n-1}\}) & (27)\\
&= \mathcal{E}\{\varepsilon_{n-1}\} \;-\; \mu(-\mathbf{p} + \mathbf{R}[\mathcal{E}\{\varepsilon_{n-1}\} - \mathbf{R}^{-1}\mathbf{p}]) & (28)\\
&= \mathcal{E}\{\varepsilon_{n-1}\} \;-\; \mu\mathbf{R}\mathcal{E}\{\varepsilon_{n-1}\} & (29)\\
&= [\mathbf{I} \;-\; \mu\mathbf{R}]\;\mathcal{E}\{\varepsilon_{n-1}\} & (30)
\end{aligned}$$

Hence a necessary condition for convergence (in the mean, and subject to our approximation) is that all eigenvalues of $\mathbf{I} \;-\; \mu\mathbf{R}$ be less than unity in magnitude; and since all eigenvalues of $\mathbf{R}$ are non-negative that means

$$\mu \;<\; \frac{2}{\lambda_{max}} \tag{31}$$

Now, the whole point of LMS is to avoid explicit knowledge of $\mathbf{R}$, much less of its eigenstructure. So a nice way to assure convergence is to note

$$Tr(\mathbf{R}) \;=\; \sum_{i=1}^{M} \lambda_i \tag{32}$$

which means

$$Tr(\mathbf{R}) \;\geq\; \lambda_{max} \tag{33}$$

As such, a reasonable way to assure convergence is to select

$$\mu < \frac{2}{Tr(\mathbf{R})} \tag{34}$$

We can use

$$\mu < \frac{2}{Mr[0]} \tag{35}$$

if the process $\{\mathbf{u}_n\}$ is a sliding window on a scalar time series. Please note that this is not the general case at all.

Suggestions about the rate of convergence are also available from (30). Specifically, the slowest eigenmode of $\mathbf{w}_n$ to converge will clearly have rate

$$\rho = \max\{|1 - 2\mu\lambda_{min}|, |1 - 2\mu\lambda_{max}|\} \tag{36}$$

meaning that the error in this mode will converge to zero at rate ρ^n. Fastest (min-max) convergence is obtained when μ is selected such that the two are equal:

$$1 - 2\mu\lambda_{min} = -(1 - 2\mu\lambda_{max}) \tag{37}$$

or

$$\mu = \frac{1}{\lambda_{max} + \lambda_{min}} \tag{38}$$

whereat we would find the slowest rate to be

$$\rho = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}} \tag{39}$$

Note that we want ρ to be as close to zero as possible for quick convergence. If $\lambda_{min} = 0$ this means that we have no convergence at all; but it is hard to interpret that fact since for this zero eigenvalue $\mathbf{q}_{min}$ we have necessarily

$$\mathbf{q}_{min}^H \mathbf{u_n} = 0 \tag{40}$$

In general errors in $\mathbf{w}_n$ in the subspace coined by the smaller eigenvalues of $\mathbf{R}$ may be large; but they may also have little effect on the filter's performance. See the next subsection.

3.3 Steady-State Error

The aim is to approximate the effect of a "jumpy" $\mathbf{w}_n$ on the error. We need second moments, and we have to do things indirectly. We have of course

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu\mathbf{u}_n e[n]^* \tag{41}$$

We take the variance:

$$
\begin{aligned}
\mathcal{E}\{|\mathbf{w}_n|^2\} &= \mathcal{E}\{|\mathbf{w}_{n-1} + \mu\mathbf{u}_n e[n]^*|^2\} \qquad (42)\\
&= \mathcal{E}\{|\mathbf{w}_{n-1}|^2]\} + \mu\mathcal{E}\{\mathbf{w}_{n-1}^H\mathbf{u}_n e[n]^*\} \\
&\quad + \mu\mathcal{E}\{\mathbf{u}_n^H\mathbf{w}_{n-1}e[n]\} + \mu^2\mathcal{E}\{\mathbf{u}_n^H\mathbf{u}_n|e[n]|^2\} \qquad (43)
\end{aligned}
$$

Now, we make the assumption that the filter is converged (no longer transient) so we can assume that

$$
\mathcal{E}\{|\mathbf{w}_n|^2\} = \mathcal{E}\{|\mathbf{w}_{n-1}|^2\} \qquad (44)
$$

and both using this in (43) and expanding for $e[n]$ we have

$$
\begin{aligned}
0 &= \mu\mathcal{E}\{\mathbf{w}_{n-1}^H\mathbf{u}_n(d[n]^* - \mathbf{u}_n^H\mathbf{w}_{n-1})\} \\
&\quad + \mu\mathcal{E}\{(d[n] - \mathbf{w}_{n-1}^H\mathbf{u}_n)\mathbf{u}_n^H\mathbf{w}_{n-1}\} + \mu^2\mathcal{E}\{\mathbf{u}_n^H\mathbf{u}_n|e[n]|^2\} \qquad (45)
\end{aligned}
$$

where we've also taken advantage of the fact that we can re-order products of things that are scalar. Now, using our previous assumption that $\mathbf{w}_{n-1}$ and $\mathbf{u}_n$ are independent, we have

$$
\begin{aligned}
0 &\approx \mu\mathcal{E}\{\mathbf{w}_{n-1}\}^H\mathbf{p} - \mathcal{E}\{\mathbf{w}_{n-1}^H\mathbf{R}\mathbf{w}_{n-1}\} \\
&\quad + \mu\mathbf{p}^H\mathcal{E}\{\mathbf{w}_{n-1}\} - \mathcal{E}\{\mathbf{w}_{n-1}^H\mathbf{R}\mathbf{w}_{n-1}\} + \mu^2\mathcal{E}\{\mathbf{u}_n^H\mathbf{u}_n|e[n]|^2\} \qquad (46)
\end{aligned}
$$

At convergence it is easy to see that

$$
\mathcal{E}\{\mathbf{w}_{n-1}\} = \mathbf{p}^H\mathbf{R}^{-1}\mathbf{p} \qquad (47)
$$

which rewrites (46) as

$$
0 = 2\mu\mathbf{p}^H\mathbf{R}^{-1}\mathbf{p} - 2\mathcal{E}\{\mathbf{w}_{n-1}^H\mathbf{R}\mathbf{w}_{n-1}\} + \mu^2\mathcal{E}\{\mathbf{u}_n^H\mathbf{u}_n|e[n]|^2\} \qquad (48)
$$

The tricky step here is to remember the *p.o.o.*: we assume that at convergence $\mathcal{E}\{\mathbf{u}_n e[n]\} = 0$, and hence claim that this implies $\mathbf{u}_n$ and $e[n]$ are independent. As such (48) becomes

$$
0 \approx 2\mu\mathbf{p}^H\mathbf{R}^{-1}\mathbf{p} - 2\mathcal{E}\{\mathbf{w}_{n-1}^H\mathbf{R}\mathbf{w}_{n-1}\} + \mu^2 Tr(\mathbf{R})\mathcal{E}\{J(\mathbf{w}_{n-1})\} \qquad (49)
$$

or

$$
\mathcal{E}\{\mathbf{w}_{n-1}^H\mathbf{R}\mathbf{w}_{n-1}\} = \mathbf{p}^H\mathbf{R}^{-1}\mathbf{p} + \frac{1}{2}\mu Tr(\mathbf{R})\mathcal{E}\{J(\mathbf{w}_{n-1})\} \qquad (50)
$$

where we have used $J(\mathbf{w}_{n-1}) \equiv |e[n]|^2$.

Now (50) isn't especially illuminating, but it is useful – remember that we said this was indirect. So let us examine the steady-state error directly:

$$\begin{aligned}
\mathcal{E}\{J(\mathbf{w}_{n-1})\} &= \mathcal{E}\{|e[n]|^2\} && (51)\\
&= \mathcal{E}\{|d[n] - \mathbf{w}_{n-1}^H \mathbf{u}_n|^2\} && (52)\\
&\approx \sigma_d^2 \; - \; 2\mathbf{p}^H \mathbf{R}^{-1}\mathbf{p} + \mathcal{E}\{\mathbf{w}_{n-1}^H \mathbf{R} \mathbf{w}_{n-1}\} && (53)\\
&= \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1}\mathbf{p} \; + \; \frac{1}{2}\mu Tr(\mathbf{R})\mathcal{E}\{J(\mathbf{w}_{n-1})\} && (54)\\
&= \frac{J_{min}}{1 - \frac{1}{2}\mu Tr(\mathbf{R})} && (55)
\end{aligned}$$

where $J_{min} \equiv J(\mathbf{R}^{-1}\mathbf{p}) = \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1}\mathbf{p}$ is the optimal Wiener filter error. Note that (53) requires the usual approximation that that $\mathbf{w}_{n-1}$ and $\mathbf{u}_n$ are independent and (54) results from insertion of (50). It is interesting to compare (55) to (32): apparently the upper bound on μ causes divergent steady-state error. It is perhaps not surprising to find that the smaller μ the better then steady-state performance.

4 Subspace Tracking

The use of the LMS algorithm in a problem that has easily expressible Wiener terms ($d[n]$, $\mathbf{u}_n$, etc.) is straightforward. This section discusses an especially famous LMS application that is both non-standard and confusingly-named. I find the textbook obscurantist about subspace tracking, hence I'll call out my own understanding of it.

Suppose we wish to design an LMS algorithm to minimize

$$J(\mathbf{w}) \; \equiv \; \frac{1}{2}\mathcal{E}\{|\mathbf{w}^H \mathbf{u}_n|^2\} \tag{56}$$

subject to a constraint

$$\mathbf{w}^H \mathbf{q} \; = \; 1 \tag{57}$$

and the factor $\frac{1}{2}$ in (56) is irrelevant but will make our lives simpler. There is no $d[n]$ here and the presence of a constraint is new; let us see what happens. We will later see this is the MVDR problem in beamforming or spectral estimation; but here we wish to solve it adaptively, whereas later we will use block averages. The idea is that we seek a filter $\mathbf{w}$ that has minimum output power subject to the stricture that it "listens" to a frequency (or direction) $\mathbf{q}$. Put another way, we wish to place nulls (zeros) of the filter where they can do the most good, but not suppress any desired signal at all. It's worth mentioning we could replace (57) by

$$\mathbf{w}^H\mathbf{A} \;=\; \mathbf{b} \tag{58}$$

for some $\mathbf{b}$ (might be all 1's) if that we want to "listen" to several directions or frequencies at the same time. More on that later, for now we'll stay with (57).

We adopt a Wiener approach, and pose this as a Lagrange multiplier optimization:

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{2}\mathbf{w}^H\mathbf{R}\mathbf{w} \;-\; \lambda(\mathbf{w}^H\mathbf{q} - 1) && (59)\\ \nabla J(\mathbf{w}) &= \mathbf{R}\mathbf{w} \;-\; \lambda\mathbf{q} && (60) \end{aligned}$$

Using the LMS idea we have the update

$$\begin{aligned} \mathbf{w}_n &= \mathbf{w}_{n-1} \;-\; \mu\nabla J_{\mathbf{w}}(\mathbf{w}_{n-1}) && (61)\\ &= \mathbf{w}_{n-1} \;-\; \mu\nabla(\mathbf{R}\mathbf{w} \;-\; \lambda\mathbf{q}) && (62) \end{aligned}$$

The LMS idea is to estimate

$$\hat{\mathbf{R}} \;=\; \mathbf{u}_n\mathbf{u}_n^H \tag{63}$$

so we get

$$\mathbf{w}_n \;=\; \mathbf{w}_{n-1} \;-\; \mu(\mathbf{u}_n\mathbf{u}_n^H\mathbf{w}_{n-1} \;-\; \lambda\mathbf{q}) \tag{64}$$

The *subspace-tracking* idea is to select λ to satisfy (57) at all times n. We get

$$\begin{aligned} (\mathbf{w}_{n-1} \;-\; \mu(\mathbf{u}_n\mathbf{u}_n^H\mathbf{w}_{n-1} \;-\; \lambda\mathbf{q}))^H\mathbf{q} &= 1 && (65)\\ (\mathbf{u}_n\mathbf{u}_n^H\mathbf{w}_{n-1})^H\mathbf{q} &= \lambda^*\mathbf{q}^H\mathbf{q} && (66)\\ (\mathbf{w}_{n-1}^H\mathbf{u}_n)(\mathbf{u}_n^H\mathbf{q}) &= \lambda^*\mathbf{q}^H\mathbf{q} && (67)\\ \lambda &= \frac{\mathbf{q}^H\mathbf{u}_n\mathbf{u}_n^H\mathbf{w}_{n-1}}{\mathbf{q}^H\mathbf{q}} && (68) \end{aligned}$$

where (66) follows because the constraint is assumed satisfied for time $n-1$ and (67) because we can re-order products of scalars.

So now (64) becomes

$$\begin{aligned} \mathbf{w}_n &= \mathbf{w}_{n-1} \;-\; \mu(\mathbf{u}_n\mathbf{u}_n^H\mathbf{w}_{n-1} \;-\; \mathbf{q}\lambda) && (69)\\ &= \mathbf{w}_{n-1} \;-\; \mu(\mathbf{u}_n\mathbf{u}_n^H\mathbf{w}_{n-1} \;-\; \mathbf{q}\frac{\mathbf{q}^H\mathbf{u}_n\mathbf{u}_n^H\mathbf{w}_{n-1}}{\mathbf{q}^H\mathbf{q}}) && (70)\\ &= \mathbf{w}_{n-1} \;-\; \mu\left(\mathbf{I} \;-\; \frac{\mathbf{q}\mathbf{q}^H}{\mathbf{q}^H\mathbf{q}}\right)\mathbf{u}_n\mathbf{u}_n^H\mathbf{w}_{n-1} && (71)\\ &= \mathbf{w}_{n-1} \;-\; \mu\mathbf{P}\mathbf{u}_n e[n]^* && (72) \end{aligned}$$

where the error is all the remaining "noise"

$$e[n] \equiv \mathbf{w}_{n-1}^H \mathbf{u}_n \tag{73}$$

and we have

$$\mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{q}\mathbf{q}^H}{\mathbf{q}^H\mathbf{q}} \tag{74}$$

It's easy to see that $\mathbf{P}$ is a *projection* matrix: $\mathbf{P}\mathbf{u}_n$ removes all the $\mathbf{u}_n$ that is parallel to $\mathbf{q}$ (where according to the constraint you want there to be no updating) but leaves leaves all the remaining subspace unchanged. The projection (74) becomes

$$\mathbf{P} \equiv \mathbf{I} - \mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H \tag{75}$$

if (58) is used in place of (57).