8.8 Constrained LS

Why Constrain? Because sometimes we know (or believe!) certain values are not allowed for θ

For example: In emitter location you may know that the emitter's range can't exceed the "radio horizon"

You may also know that the emitter is on the left side of the aircraft (because you got a strong signal from the left-side antennas and a weak one from the right-side antennas)

Thus, when finding $\hat{\theta}_{LS}$ you want to constrain it to satisfy these conditions

Constrained LS Problem Statement

Say that S_c is the set of allowable θ values (due to constraints).

Then we seek $\hat{\theta}_{CLS} \in S_c$ such that

$$\left\| \mathbf{x} - \mathbf{H}\hat{\boldsymbol{\theta}}_{CLS} \right\|^2 = \min_{\boldsymbol{\theta} \in S_c} \left\| \mathbf{x} - \mathbf{H}\boldsymbol{\theta} \right\|^2$$

Types of Constraints

HARDER

1. Linear Equality $\mathbf{A}\theta = \mathbf{b}$
2. Nonlinear Equality $f(\theta) = \mathbf{b}$
3. Linear Inequality $\mathbf{A}\theta \geq \mathbf{b}$, $\mathbf{A}\theta \leq \mathbf{b}$
4. Nonlinear Inequality $f(\theta) \geq \mathbf{b}$, $f(\theta) \leq \mathbf{b}$

Constrained to a line, plane or hyperplane

Constrained to lie above/below a hyperplane

We'll Cover #1…. See Books on Optimization for Other Cases

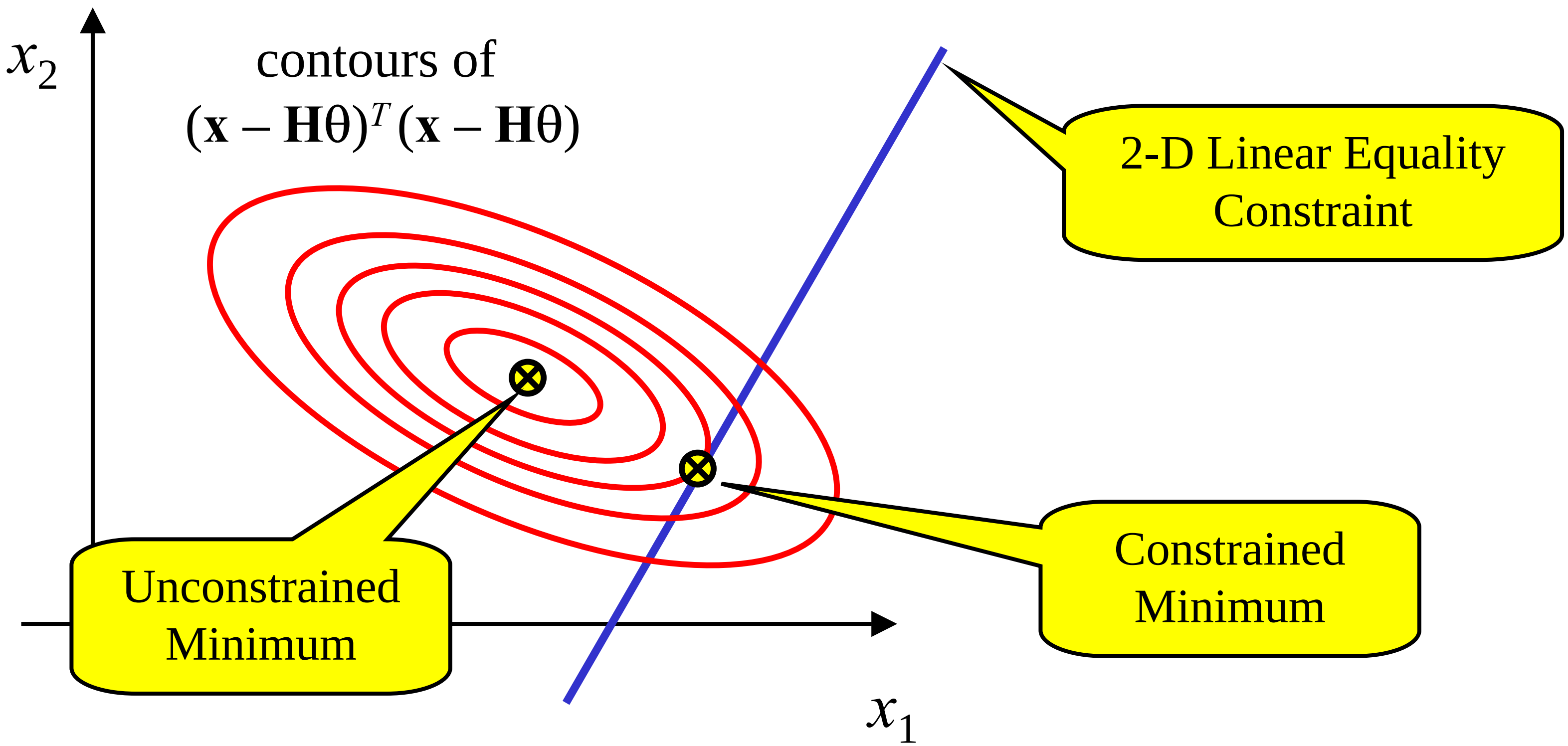
LS Cost with a Linear Equality Constraint

Using Lagrange Multipliers… we need to minimize

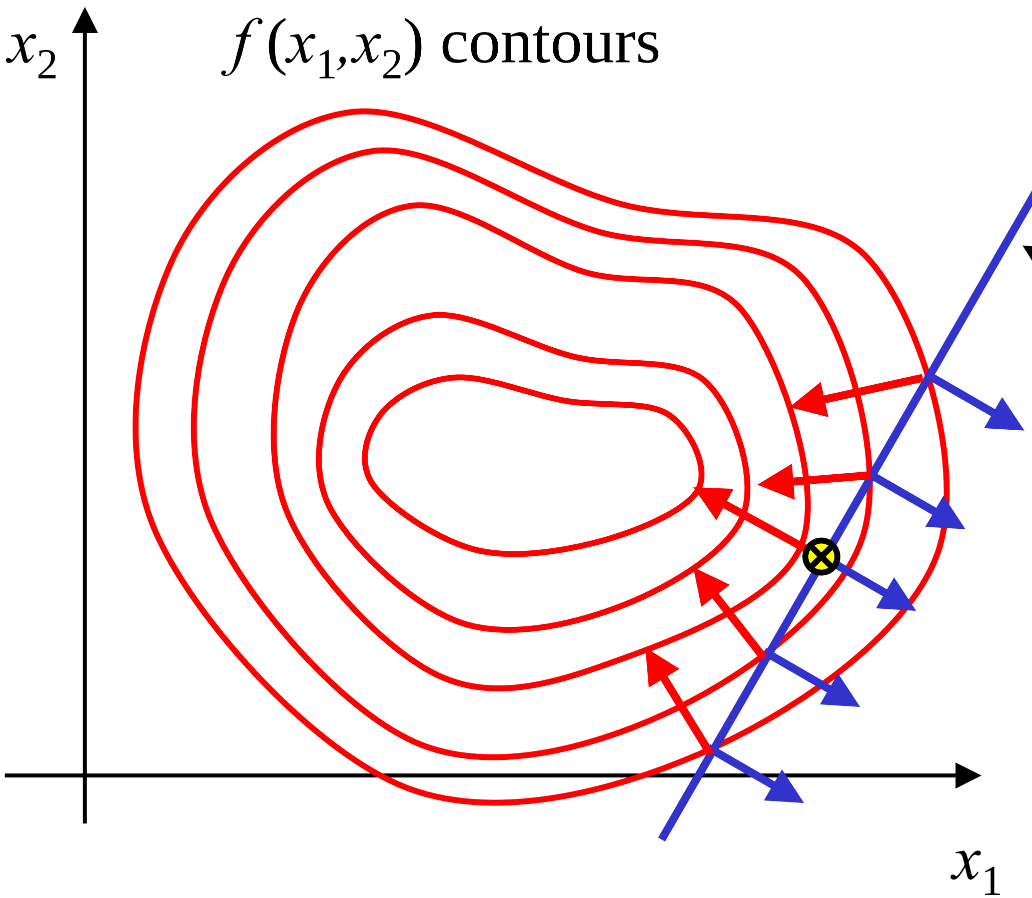
$$J_c(\boldsymbol{\theta}) = (\mathbf{x} - \mathbf{H}\boldsymbol{\theta})^T(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}) + \boldsymbol{\lambda}^T(\mathbf{A}\boldsymbol{\theta} - \mathbf{b})$$

w.r.t. $\boldsymbol{\theta}$ *and* $\boldsymbol{\lambda}$

Linear Equality Constraint



Constrained Optimization: Lagrange Multiplier



Constraint: $g(x_1,x_2) = C$

$g(x_1,x_2) - C = h(x_1,x_2) = 0$

Ex. $ax_1 + bx_2 - c = 0$

$\Rightarrow x_2 = (-a/b)x_1 + c/b$

A Linear Constraint

$$\nabla h(x_1,x_2) = \begin{bmatrix} \dfrac{\partial h(x_1,x_2)}{\partial x_1} \\ \dfrac{\partial h(x_1,x_2)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

Ex. The grad vector has "slope" of $b/a \Rightarrow$ orthogonal to constraint line

Constrained Max occurs when:

$$\nabla f(x_1,x_2) = -\lambda \nabla h(x_1,x_2)$$

$$\Rightarrow \quad \nabla f(x_1,x_2) + \lambda \nabla h(x_1,x_2) = 0$$

$$\boxed{\nabla\left[f(x_1,x_2) + \lambda\left(g(x_1,x_2) - C\right)\right] = 0}$$

LS Solution with a Linear Equality Constraint

Follow the usual steps for Lagrange Multiplier Solution:

1. Set $\dfrac{\partial J_c}{\partial \boldsymbol{\theta}} = \mathbf{0} \;\Rightarrow\; \hat{\boldsymbol{\theta}}_{CLS}$ as a function of λ $\;\hat{\boldsymbol{\theta}}_{CLS}(\lambda)$

$$-2\mathbf{H}^T\mathbf{x} + 2\mathbf{H}^T\mathbf{H}\boldsymbol{\theta} + \mathbf{A}^T\boldsymbol{\lambda} = \mathbf{0} \;\Rightarrow\; \hat{\boldsymbol{\theta}}_c(\boldsymbol{\lambda}) = \underbrace{\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x}}_{\hat{\boldsymbol{\theta}}_{uc}} - \frac{1}{2}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{A}^T\boldsymbol{\lambda}$$

Unconstrained Estimate

2. Solve for λ to make $\hat{\theta}_{CLS}$ satisfy the constraint: $\underbrace{\mathbf{A}\hat{\boldsymbol{\theta}}_c(\boldsymbol{\lambda}) = \mathbf{b}}_{solve\ for\ \boldsymbol{\lambda} \Rightarrow \boldsymbol{\lambda}_c}$

$$\mathbf{A}\left[\hat{\boldsymbol{\theta}}_{uc} - \frac{1}{2}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{A}^T\boldsymbol{\lambda}\right] = \mathbf{b} \;\Rightarrow\; \boldsymbol{\lambda}_c = 2\left[\mathbf{A}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{A}^T\right]^{-1}\left(\mathbf{A}\hat{\boldsymbol{\theta}}_{uc} - \mathbf{b}\right)$$

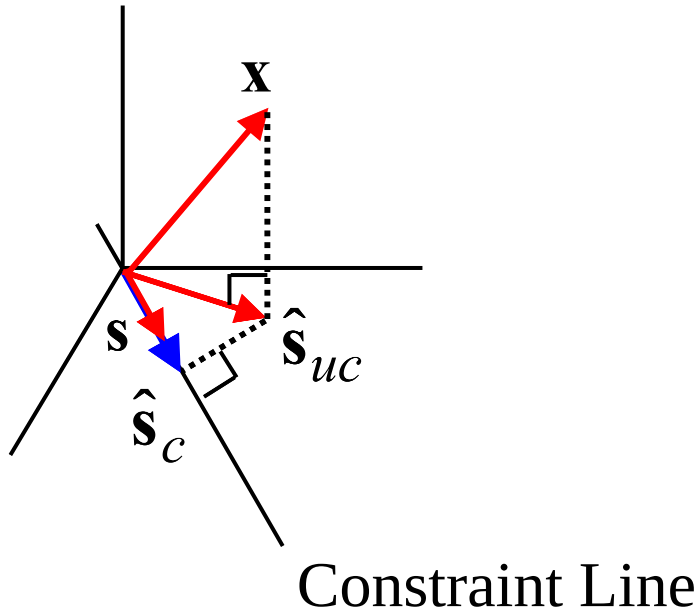
3. Plug in to get the constrained solution: $\hat{\boldsymbol{\theta}}_c = \hat{\boldsymbol{\theta}}_c(\boldsymbol{\lambda}_c)$

$$\boxed{\hat{\boldsymbol{\theta}}_c = \hat{\boldsymbol{\theta}}_{uc} - \underbrace{\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{A}^T\left[\mathbf{A}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{A}^T\right]^{-1}\left(\mathbf{A}\hat{\boldsymbol{\theta}}_{uc} - \mathbf{b}\right)}_{\text{"Correction Term"}}}$$

Amount of Constraint Deviation

Geometry of Constrained Linear LS

The above result can be interpreted geometrically:



Constrained Estimate of the Signal is the Projection of the Unconstrained Estimate onto the Linear Constraint Subspace

8.9 Nonlinear LS

Everything we've done up to now has assumed a linear observation model… but we've already seen that many applications have nonlinear observation models: $\mathbf{s}(\theta) \neq \mathbf{H}\theta$

Recall: For linear case – closed-form solution

< Not so for nonlinear case!! >

Must use numerical, iterative methods to minimize the LS cost given by:

$$J(\theta) = [\mathbf{x} - \mathbf{s}(\theta)]^T [\mathbf{x} - \mathbf{s}(\theta)]$$

But first… Two Tricks!!!

<u>Two Tricks for Nonlinear LS</u>

Sometimes it is possible to:

1. Transform into a <u>Linear</u> Problem
2. Separate out any Linear Parameters

Sometimes Possible to Do Both Tricks Together

<u>Trick #1:</u> Seek an invertible function $\left\{ \begin{matrix} g(\boldsymbol{\theta}) = \boldsymbol{\alpha} \\ \boldsymbol{\theta} = g^{-1}(\boldsymbol{\alpha}) \end{matrix} \right\}$

such that

$\mathbf{s}(\theta(\alpha)) = \mathbf{H}\alpha$, which can be easily solved for $\hat{\alpha}_{LS}$

and then find $\hat{\theta}_{LS} = g^{-1}(\hat{\alpha}_{LS})$

<u>Trick #2:</u> See if <u>some</u> of the parameters *are* <u>linear:</u>

Try to decompose $\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}$ to get $\mathbf{s}(\boldsymbol{\theta}) = \mathbf{H}(\boldsymbol{\alpha})\boldsymbol{\beta}$

Linear in β!!!

Nonlinear in α

Example of Linearization Trick

Consider estimation of a sinusoid's amplitude and phase (with a known frequency):

$$s[n] = A\cos(2\pi f_o n + \phi) \qquad \boldsymbol{\theta} = \begin{bmatrix} A \\ \phi \end{bmatrix}$$

But we can re-write this model as:

$$s[n] = \underbrace{A\cos(\phi)}_{\alpha_1}\cos(2\pi f_o n) - \underbrace{A\sin(\phi)}_{\alpha_2}\sin(2\pi f_o n)$$

which is linear in $\alpha = [\alpha_1\ \alpha_2]^T$ so: $\hat{\boldsymbol{\alpha}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x}$

Then map this estimate back using

$$\hat{\boldsymbol{\theta}} = g^{-1}(\hat{\boldsymbol{\alpha}}) = \begin{bmatrix} \sqrt{\hat{\alpha}_1^2 + \hat{\alpha}_2^2} \\ \tan^{-1}\left(\dfrac{-\hat{\alpha}_2}{\hat{\alpha}_1}\right) \end{bmatrix}$$

Note that for this example this is merely exploiting polar-to-rectangular ideas!!!

Example of Separation Trick

Consider a signal model of three exponentials:

$$s[n] = A_1 r^n + A_2 r^{2n} + A_3 r^{3n} \qquad 0 < r < 1$$

$$\boldsymbol{\theta} = [\underbrace{A_1 \quad A_2 \quad A_3}_{\boldsymbol{\beta}^T} \quad \underbrace{r}_{\alpha}]^T$$

Then we can write:

$$\mathbf{H}(r) = \begin{bmatrix} 1 & 1 & 1 \\ r & r^2 & r^3 \\ \vdots & \vdots & \vdots \\ r^{N-1} & r^{2(N-1)} & r^{3(N-1)} \end{bmatrix}$$

$$\mathbf{s}(\boldsymbol{\theta}) = \mathbf{H}(r)\boldsymbol{\beta}$$

$$\hat{\boldsymbol{\beta}}(r) = [\mathbf{H}^T(r)\mathbf{H}(r)]^{-1}\mathbf{H}^T(r)\mathbf{x}$$

Then we need to minimize :

$$J(r) = \left[\mathbf{x} - \mathbf{H}(r)\hat{\boldsymbol{\beta}}(r)\right]^T \left[\mathbf{x} - \mathbf{H}(r)\hat{\boldsymbol{\beta}}(r)\right]$$

$$= \left[\mathbf{x} - \mathbf{H}(r)[\mathbf{H}^T(r)\mathbf{H}(r)]^{-1}\mathbf{H}^T(r)\mathbf{x}\right]^T \left[\mathbf{x} - \mathbf{H}(r)[\mathbf{H}^T(r)\mathbf{H}(r)]^{-1}\mathbf{H}^T(r)\mathbf{x}\right]$$

Depends on only <u>one</u> variable… so might conceivably just compute on a grid and find minimum

Iterative Methods for Solving Nonlinear LS

Goal: Find θ value that minimizes $J(\theta) = [\mathbf{x}-\mathbf{s}(\theta)]^T [\mathbf{x}-\mathbf{s}(\theta)]$ without computing it over a p-dimensional grid

Two most common approaches:

1. **Newton-Raphson**
   a. Analytically find $\partial J(\theta)/\partial\theta$
   b. Apply Newton-Raphson to find a zero of $\partial J(\theta)/\partial\theta$
      (i.e. linearize $\partial J(\theta)/\partial\theta$ about the current estimate)
   c. Iteratively Repeat
2. **Gauss-Newton**
   a. Linearize signal model $\mathbf{s}(\theta)$ about the current estimate
   b. Solve resulting linear problem
   c. Iteratively Repeat

Both involve:

- **Linearization (but they each linearize something different!)**
- **Solve linear problem**
- **Iteratively improve result**

Newton-Raphson Solution to Nonlinear LS

To find minimum of $J(\theta)$: set $\underbrace{\dfrac{\partial J}{\partial \boldsymbol{\theta}}}_{\triangleq\, g(\boldsymbol{\theta})} = 0$

Need to find $\dfrac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \dfrac{\partial J(\boldsymbol{\theta})}{\partial \theta_1} \\ \vdots \\ \dfrac{\partial J(\boldsymbol{\theta})}{\partial \theta_p} \end{bmatrix}$ for $J(\boldsymbol{\theta}) = \sum_{i=0}^{N-1} (x[i] - s_{\boldsymbol{\theta}}[i])^2$

Taking these partials gives:

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_j} = \underbrace{-2}_{\substack{can\\ ignore\\ Why?}} \sum_{i=0}^{N-1} \underbrace{(x[i] - s_{\boldsymbol{\theta}}[i])}_{\triangleq\, r_i} \underbrace{\frac{\partial s_{\boldsymbol{\theta}}[i]}{\partial \theta_j}}_{\triangleq\, h_{ij}}$$

Now set to zero: $\underbrace{\sum_{i=0}^{N-1} r_i h_{ij}}_{\substack{Matrix \\ \times Vector}} = 0 \quad for\ j = 1, \ldots, p \qquad \Rightarrow g(\boldsymbol{\theta}) = \mathbf{H}_{\boldsymbol{\theta}}^T \mathbf{r}_{\boldsymbol{\theta}} = \mathbf{0}$

Depend nonlinearly on θ

$$\mathbf{H}_{\boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial s_{\boldsymbol{\theta}}[0]}{\partial \theta_1} & \frac{\partial s_{\boldsymbol{\theta}}[0]}{\partial \theta_2} & \cdots & \frac{\partial s_{\boldsymbol{\theta}}[0]}{\partial \theta_p} \\ \frac{\partial s_{\boldsymbol{\theta}}[1]}{\partial \theta_1} & \frac{\partial s_{\boldsymbol{\theta}}[1]}{\partial \theta_2} & \cdots & \frac{\partial s_{\boldsymbol{\theta}}[1]}{\partial \theta_p} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial s_{\boldsymbol{\theta}}[N-1]}{\partial \theta_1} & \frac{\partial s_{\boldsymbol{\theta}}[N-1]}{\partial \theta_2} & \cdots & \frac{\partial s_{\boldsymbol{\theta}}[N-1]}{\partial \theta_p} \end{bmatrix} \qquad \mathbf{r}_{\boldsymbol{\theta}} = \begin{bmatrix} (x[0] - s_{\boldsymbol{\theta}}[0]) \\ \vdots \\ (x[N-1] - s_{\boldsymbol{\theta}}[N-1]) \end{bmatrix}$$

Define the i^{th} <u>row</u> of $\mathbf{H}_{\boldsymbol{\theta}}$: $\mathbf{h}_i^T(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial s_{\boldsymbol{\theta}}[i]}{\partial \theta_1} & \frac{\partial s_{\boldsymbol{\theta}}[i]}{\partial \theta_2} & \cdots & \frac{\partial s_{\boldsymbol{\theta}}[i]}{\partial \theta_P} \end{bmatrix}$

Then the equation to solve is: $g(\boldsymbol{\theta}) = \mathbf{H}_{\boldsymbol{\theta}}^T \mathbf{r}_{\boldsymbol{\theta}} = \sum_{n=0}^{N-1} r_{\boldsymbol{\theta}}[n] \mathbf{h}_i(\boldsymbol{\theta}) = \mathbf{0}$

For Newton-Raphson we linearize g(θ) around our current estimate and iterate:

Need this

$$\hat{\boldsymbol\theta}_{k+1} = \hat{\boldsymbol\theta}_k - \left[\left[\frac{\partial g(\boldsymbol\theta)}{\partial \boldsymbol\theta}\right]^{-1} g(\boldsymbol\theta)\right]\Bigg|_{\boldsymbol\theta=\hat{\boldsymbol\theta}_k} = \hat{\boldsymbol\theta}_k - \left[\left[\frac{\partial \mathbf{H}_{\boldsymbol\theta}^T \mathbf{r}_{\boldsymbol\theta}}{\partial \boldsymbol\theta}\right]^{-1} \mathbf{H}_{\boldsymbol\theta}^T \mathbf{r}_{\boldsymbol\theta}\right]\Bigg|_{\boldsymbol\theta=\hat{\boldsymbol\theta}_k}$$

$$\frac{\partial \mathbf{H}_{\boldsymbol\theta}^T \mathbf{r}_{\boldsymbol\theta}}{\partial \boldsymbol\theta} = \frac{\partial}{\partial \boldsymbol\theta}\sum_{n=0}^{N-1} \mathbf{h}_n(\boldsymbol\theta) r_{\boldsymbol\theta}[n] = \sum_{n=0}^{N-1} \frac{\partial r_{\boldsymbol\theta}[n]\mathbf{h}_n(\boldsymbol\theta)}{\partial \boldsymbol\theta} = \sum_{n=0}^{N-1} \underbrace{\frac{\partial \mathbf{h}_n(\boldsymbol\theta)}{\partial \boldsymbol\theta}}_{\triangleq \mathbf{G}_n(\boldsymbol\theta)} r_{\boldsymbol\theta}[n] + \underbrace{\sum_{n=0}^{N-1} \mathbf{h}_n(\boldsymbol\theta) \frac{\partial r_{\boldsymbol\theta}[n]}{\partial \boldsymbol\theta}}_{-\mathbf{H}_{\boldsymbol\theta}^T \mathbf{H}_{\boldsymbol\theta}}$$

Derivative of Product Rule

$$\left[\mathbf{G}_n(\boldsymbol\theta)\right]_{ij} = \frac{\partial^2 s_{\boldsymbol\theta}[n]}{\partial \theta_i \partial \theta_j} \quad i,j = 1,2,\ldots,p$$

$$\frac{\partial r_{\boldsymbol\theta}[n]}{\partial \boldsymbol\theta} = \frac{\partial (x[n]-s_{\boldsymbol\theta}[n])}{\partial \boldsymbol\theta} = -\begin{bmatrix} \dfrac{\partial s_{\boldsymbol\theta}[n]}{\partial \theta_1} \\ \dfrac{\partial s_{\boldsymbol\theta}[n]}{\partial \theta_2} \\ \vdots \\ \dfrac{\partial s_{\boldsymbol\theta}[n]}{\partial \theta_p} \end{bmatrix}$$

$$\boxed{\frac{\partial \mathbf{H}_{\boldsymbol\theta}^T \mathbf{r}_{\boldsymbol\theta}}{\partial \boldsymbol\theta} = \sum_{n=0}^{N-1} \mathbf{G}_n(\boldsymbol\theta)\big(x[n]-s_{\boldsymbol\theta}[n]\big) - \mathbf{H}_{\boldsymbol\theta}^T \mathbf{H}_{\boldsymbol\theta}}$$

So the Newton-Raphson method becomes:

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - \left[\left[\frac{\partial \mathbf{H}_{\boldsymbol{\theta}}^T \mathbf{r}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}\right]^{-1} \mathbf{H}_{\boldsymbol{\theta}}^T \mathbf{r}_{\boldsymbol{\theta}}\right]\Bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_k}$$

$$= \hat{\boldsymbol{\theta}}_k + \left[\mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \mathbf{H}_{\hat{\boldsymbol{\theta}}_k} - \sum_{n=0}^{N-1} \mathbf{G}_n(\hat{\boldsymbol{\theta}}_k)\left(x[n] - s_{\hat{\boldsymbol{\theta}}_k}[n]\right)\right]^{-1} \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \left(\mathbf{x} - \mathbf{s}_{\hat{\boldsymbol{\theta}}_k}\right)$$

1st partials of signal w.r.t. parameters

2nd partials of $s[n]$ w.r.t. parameters

<u>Note</u>: if the signal <u>*is* linear</u> in parameters… this collapses to the non-iterative result we found for the linear case!!!

Newton-Raphson LS Iteration Steps:

1. Start with an initial estimate
2. Iterate the above equation until change is "small"

Gauss-Newton Solution to Nonlinear LS

First we linearize the model around our current estimate by using a Taylor series and keeping only the linear terms:

$$\mathbf{s}_{\boldsymbol{\theta}} \approx \mathbf{s}_{\hat{\boldsymbol{\theta}}_k} + \underbrace{\left[\left. \frac{\partial \mathbf{s}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_k} \right]}_{\triangleq \mathbf{H}(\hat{\boldsymbol{\theta}}_k)} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_k)$$

Then we use this linearized model in the LS cost:

$$\begin{aligned} J(\boldsymbol{\theta}) &= [\mathbf{x} - \mathbf{s}_{\boldsymbol{\theta}}]^T [\mathbf{x} - \mathbf{s}_{\boldsymbol{\theta}}] \\ &\approx \left[\mathbf{x} - \left\{ \mathbf{s}_{\hat{\boldsymbol{\theta}}_k} + \mathbf{H}_{\hat{\boldsymbol{\theta}}_k} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_k) \right\} \right]^T \left[\mathbf{x} - \left\{ \mathbf{s}_{\hat{\boldsymbol{\theta}}_k} + \mathbf{H}_{\hat{\boldsymbol{\theta}}_k} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_k) \right\} \right] \\ &= \Big[\underbrace{\mathbf{x} - \mathbf{s}_{\hat{\boldsymbol{\theta}}_k} + \mathbf{H}_{\hat{\boldsymbol{\theta}}_k} \hat{\boldsymbol{\theta}}_k}_{\triangleq \mathbf{y}} - \mathbf{H}_{\hat{\boldsymbol{\theta}}_k} \boldsymbol{\theta} \Big]^T \Big[\underbrace{\mathbf{x} - \mathbf{s}_{\hat{\boldsymbol{\theta}}_k} + \mathbf{H}_{\hat{\boldsymbol{\theta}}_k} \hat{\boldsymbol{\theta}}_k}_{\triangleq \mathbf{y}} - \mathbf{H}_{\hat{\boldsymbol{\theta}}_k} \boldsymbol{\theta} \Big] \end{aligned}$$

All Known Things

This gives a form for the LS cost that looks like a linear problem!!

$$J(\boldsymbol{\theta}) = \left[\mathbf{y} - \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}\boldsymbol{\theta}\right]^T \left[\mathbf{y} - \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}\boldsymbol{\theta}\right]$$

We know the LS solution to that problem is

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{k+1} &= \left[\mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}\right]^{-1} \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \mathbf{y} \\
&= \left[\mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}\right]^{-1} \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \left(\mathbf{x} - \mathbf{s}_{\hat{\boldsymbol{\theta}}_k} + \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}\hat{\boldsymbol{\theta}}_k\right) \\
&= \underbrace{\left[\mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}\right]^{-1} \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}}_{=\mathbf{I}} \hat{\boldsymbol{\theta}}_k + \left[\mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}\right]^{-1} \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \left(\mathbf{x} - \mathbf{s}_{\hat{\boldsymbol{\theta}}_k}\right)
\end{aligned}$$

Gauss-Newton LS Iteration:

$$\boxed{\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + \left[\mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}\right]^{-1} \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \left(\mathbf{x} - \mathbf{s}_{\hat{\boldsymbol{\theta}}_k}\right)}$$

Gauss-Newton LS Iteration Steps:

1. Start with an initial estimate
2. Iterate the above equation until change is “small”

Newton-Raphson vs. Gauss-Newton

How do these two methods compare?

G-N:
$$\boxed{\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + \left[\mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}\right]^{-1} \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \left(\mathbf{x} - \mathbf{s}_{\hat{\boldsymbol{\theta}}_k}\right)}$$

N-R:
$$\boxed{\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + \left[\mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \mathbf{H}_{\hat{\boldsymbol{\theta}}_k} - \underbrace{\sum_{n=0}^{N-1} \mathbf{G}_n(\hat{\boldsymbol{\theta}}_k)\left(x[n] - s_{\hat{\boldsymbol{\theta}}_k}[n]\right)}_{\text{The term of 2nd partials is missing in the Gauss-Newton Equation}}\right]^{-1} \mathbf{H}_{\hat{\boldsymbol{\theta}}_k}^T \left(\mathbf{x} - \mathbf{s}_{\hat{\boldsymbol{\theta}}_k}\right)}$$

The term of 2nd partials is missing in the Gauss-Newton Equation

Which is better?

Typically I prefer Gauss-Newton:

See p. 683 of *Numerical Recipes* book

- $\mathbf{G}_n$ matrices are often small enough to be negligible
- … or the error term is small enough to make the sum term negligible
- Inclusion of the sum term can sometimes de-stablize the iteration