Optimal Estimation Methods

(Lecture 9 – Total Least Squares)

Dr. John L. Crassidis

University at Buffalo – State University of New York
Department of Mechanical & Aerospace Engineering
Amherst, NY 14260-4400
johnc@buffalo.edu
http://www.buffalo.edu/~johnc

University at Buffalo The State University of New York

Motivation (i)

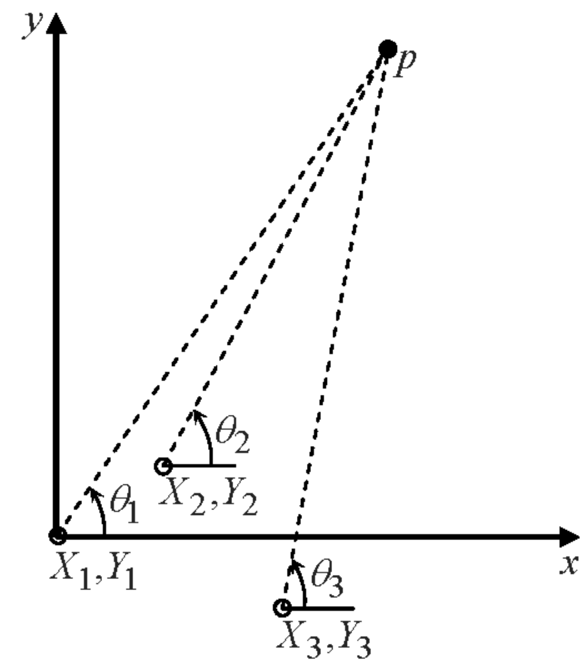
- Bearings-Only Point Estimation
  - The goal is to estimate the point p with coordinates (x, y) from bearings-only measurements
  - The baseline points, denoted by X_i and Y_i, are assumed to be imprecisely known
  - Bearing measurement model and baseline point models

$$\tilde{\theta}_i = \theta_i + \delta\theta_i$$

$$\tilde{X}_i = X_i + \delta X_i$$

$$\tilde{Y}_i = Y_i + \delta Y_i$$

where $\delta\theta_i$, δX_i and δY_i are zero-mean Gaussian processes with std $\sigma_{\theta i}$, σ_{Xi} and σ_{Yi}



- Observation model

$$\theta_i = \tan^{-1}\left(\frac{y - Y_i}{x - X_i}\right)$$

  - Nonlinear model, but we can convert to a linear one

Motivation (ii)

$\tan\theta_i = \frac{y - Y_i}{x - X_i} = \frac{\sin(\theta_i)}{\cos(\theta_i)} = ?$

- Take the tangent of both sides of the observation model

$-Y_i \cos\theta_i = -X_i \sin\theta_i$

$= -x_i \sin\theta_i + y_i \cos\theta_i$

$$y_i \equiv \mathbf{h}_i^T \mathbf{x} = -X_i \sin(\theta_i) + Y_i \cos(\theta_i)$$
$$\mathbf{h}_i = [-\sin(\theta_i) \quad \cos(\theta_i)]^T$$
$$\mathbf{x} = [x \quad y]^T$$

- This is now set up as a linear least squares model, but we have a slight problem…
- Replace the true values with the measured values and use the first-order approximations

$\sin(a+b) =$

$$\begin{aligned}
\sin(\theta_i + \delta\theta_i) &= \sin(\theta_i)\cos(\delta\theta_i) + \sin(\delta\theta_i)\cos(\theta_i) \\
&\approx \sin(\theta_i) + \delta\theta_i \cos(\theta_i) \\
\cos(\theta_i + \delta\theta_i) &= \cos(\theta_i)\cos(\delta\theta_i) - \sin(\delta\theta_i)\sin(\theta_i) \\
&\approx \cos(\theta_i) - \delta\theta_i \sin(\theta_i)
\end{aligned}$$

Motivation (iii)

- This leads to

$$\begin{aligned}
\tilde{y}_i =& -\tilde{X}_i \sin(\tilde{\theta}_i) + \tilde{Y}_i \cos(\tilde{\theta}_i) \\
=& -X_i \sin(\theta_i) + Y_i \cos(\theta_i) - \delta\theta_i X_i \cos(\theta_i) - \delta X_i \sin(\theta_i) - \underbrace{\delta\theta_i \delta X_i}_{\text{Assume uncorrelated}} \cos(\theta_i) \\
& - \delta\theta_i Y_i \sin(\theta_i) + \delta Y_i \cos(\theta_i) - \delta\theta_i \delta Y_i \sin(\theta_i)
\end{aligned}$$

$$\begin{aligned}
\tilde{\mathbf{h}}_i &= [-\sin(\tilde{\theta}_i) \quad \cos(\tilde{\theta}_i)]^T \\
&= [-\sin(\theta_i) - \delta\theta_i \cos(\theta_i) \quad \cos(\theta_i) - \delta\theta_i \sin(\theta_i)]^T
\end{aligned}$$

- Note that the basis functions now contain noise! : $\delta\theta_i$
- Take the expected values of both to give

$$E\{\tilde{y}_i\} = -X_i \sin(\theta_i) + Y_i \cos(\theta_i)$$

$$E\{\tilde{\mathbf{h}}_i\} = [-\sin(\theta_i) \quad \cos(\theta_i)]^T$$

- Define the covariances

$$\mathcal{R}_{yy_i} \equiv E\left\{(\tilde{y}_i - E\{\tilde{y}_i\})^2\right\}$$

$$\mathcal{R}_{hh_i} \equiv E\left\{(\mathbf{h}_i - E\{\mathbf{h}_i\})(\mathbf{h}_i - E\{\mathbf{h}_i\})^T\right\}$$

$$\mathcal{R}_{hy_i} \equiv E\left\{(\tilde{y}_i - E\{\tilde{y}_i\})(\mathbf{h}_i - E\{\mathbf{h}_i\})\right\}$$

Motivation (iv)

- Taking the expectations leads to

$$\begin{aligned}\mathcal{R}_{yy_i} &= \sigma^2_{\theta_i}\{[X_i\cos(\theta_i)+Y_i\sin(\theta_i)]^2+\sigma^2_{X_i}\cos^2(\theta_i)+\sigma^2_{Y_i}\sin^2(\theta_i)\}\\ &\quad+\sigma^2_{X_i}\sin^2(\theta_i)+\sigma^2_{Y_i}\cos^2(\theta_i)\end{aligned}$$

$$\mathcal{R}_{hh_i} = \sigma^2_{\theta_i}\begin{bmatrix}\cos^2(\theta_i) & \sin(\theta_i)\cos(\theta_i)\\ \sin(\theta_i)\cos(\theta_i) & \sin^2(\theta_i)\end{bmatrix}$$

$$\mathcal{R}_{hy_i} = \sigma^2_{\theta_i}[X_i\cos(\theta_i)+Y_i\sin(\theta_i)]\begin{bmatrix}\cos(\theta_i)\\ \sin(\theta_i)\end{bmatrix}$$

- Note that the covariance expressions are a function of the true values, which are unknown
  - Can replace them with the measured ones, which leads to only second-order errors
- The linear least squares model assumes no errors in the basis functions, so using it is not optimal
  - Total least squares accounts for this issue

Total Least Squares (i)

- The "total least squares" (TLS) model is given by

$$\tilde{\mathbf{y}} = \mathbf{y} + \Delta\mathbf{y}, \quad (m \times 1) \text{ vector}$$

$$\tilde{H} = H + \Delta H, \quad (m \times n) \text{ matrix}$$

$$\mathbf{y} = H\,\mathbf{x}$$

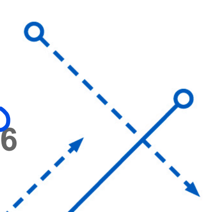
where $\Delta\mathbf{y}$ is the measurement noise vector and ΔH is the basis function noise matrix

- Define the following $m \times (n+1)$ matrix $\tilde{D} \equiv [\tilde{H} \quad \tilde{\mathbf{y}}]$
- The TLS problem seeks an optimal estimate of the $n \times 1$ vector $\mathbf{x}$, denoted by $\hat{\mathbf{x}}$ with $\hat{\mathbf{y}} = \hat{H}\hat{\mathbf{x}}$, where $\hat{\mathbf{y}}$ is the estimate of $\mathbf{y}$ and $\hat{H}$ is the estimate of H, which maximizes

$$p(\tilde{D}|D) = \frac{1}{(2\pi)^{m\times(n+1)/2}\,[\det(R)]^{1/2}} \exp\left\{-\frac{1}{2}\text{vec}^T(\tilde{D}^T - D^T)\,R^{-1}\text{vec}(\tilde{D}^T - D^T)\right\}$$

where $D \equiv [H \quad \mathbf{y}]$, which satisfies $D\,\mathbf{z} = \mathbf{0}$ with $\mathbf{z} \equiv [\mathbf{x}^T \quad -1]^T$, vec denotes a column vector formed by stacking the consecutive columns of the associated matrix, and R is the covariance matrix

$[H \;\; y]\begin{bmatrix} x^T \\ -1 \end{bmatrix} = Hx^T - y = 0$



Total Least Squares (ii)

- Unfortunately because H now contains errors the constraint $\hat{\mathbf{y}} = \hat{H}\,\hat{\mathbf{x}}$ must also be added to the maximization problem
- The negative log-likelihood now leads to the following loss function

$\tilde{y} - \tilde{H}\hat{x} = 0$

$\parallel$

$$J(\hat{D}) = \frac{1}{2}\mathrm{vec}^T(\tilde{D}^T - \hat{D}^T)\,R^{-1}\mathrm{vec}(\tilde{D}^T - \hat{D}^T), \quad \text{s.t.} \quad \hat{D}\,\hat{\mathbf{z}} = \mathbf{0}$$

where $\hat{D} \equiv [\hat{H} \quad \hat{\mathbf{y}}]$ denotes the estimate of D and $\hat{\mathbf{z}} \equiv [\hat{\mathbf{x}}^T \quad -1]^T$

- For a unique solution it is required that the rank of $\hat{D}$ be n, which means $\hat{\mathbf{z}}$ spans the null space of $\hat{D}_{m \times (n+1)}$
  - Note that the dimension of $\hat{\mathbf{z}}$ is $n+1$
- This is a huge optimization problem to estimate the full $\hat{D}$
  - Possible to write loss function in terms of $\hat{\mathbf{x}}$ only though
- Can be simplified for a number of cases shown here
  - Assume that the covariance is identity matrix $R = I$
  - Element-wise uncorrelated and stationary Rows(D) have same covariance
  - Element-wise uncorrelated and non-stationary R is block diagonal

    Covariances can be different

- Assume that the covariance is identity matrix
  - The loss function for this case can be shown to be given by

$$J = \left|\left|\begin{bmatrix}\tilde{H} & \tilde{\mathbf{y}}\end{bmatrix} - \begin{bmatrix}\hat{H} & \hat{\mathbf{y}}\end{bmatrix}\right|\right|_F^2$$

  - The TLS estimate equation is given by

$$\hat{\mathbf{y}} = \hat{H}\hat{\mathbf{x}}_{\text{TLS}}$$

  - Define the following

$$\mathbf{e} \equiv \tilde{\mathbf{y}} - \hat{\mathbf{y}}$$

$$B \equiv \tilde{H} - \hat{H}$$

  - Then the estimate equation can be written as

$$(\tilde{H} - B)\hat{\mathbf{x}}_{\text{TLS}} = \tilde{\mathbf{y}} - \mathbf{e}$$

or

$$\hat{D}\begin{bmatrix}\hat{\mathbf{x}}_{\text{TLS}} \\ -1\end{bmatrix} = \mathbf{0}$$

This clearly shows that the matrix $\hat{D}$ must be rank deficient by one for a unique solution

where $\hat{D} \equiv \begin{bmatrix}(\tilde{H} - B) & (\tilde{\mathbf{y}} - \mathbf{e})\end{bmatrix}$

Total Least Squares (iv)

- Use a reduced-form SVD to obtain solution

$$\tilde{D} \equiv \begin{bmatrix} \tilde{H} & \tilde{\mathbf{y}} \end{bmatrix} = \underset{\substack{\uparrow \\ \text{skinny form}}}{USV^T} = \begin{bmatrix} U_{11} & \mathbf{u} \end{bmatrix} \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0}^T & s_{n+1} \end{bmatrix} \begin{bmatrix} V_{11} & \mathbf{v} \\ \mathbf{w}^T & v_{22} \end{bmatrix}^T \quad (1)$$

where U_{11} is an $m \times n$ matrix, $\mathbf{u}$ is an $m \times 1$ vector, V_{11} is an $m \times n$ matrix, $\mathbf{v}$ and $\mathbf{w}$ are $n \times 1$ vectors, and Σ is an $n \times n$ diagonal matrix given by $\Sigma = \mathrm{diag}[s_1 \quad s_2 \quad \cdots \quad s_n]$

- The goal is to make the estimate rank deficient by one
- Let's try the simplest approach to see if it's feasible; assume

$$\hat{D} \equiv \begin{bmatrix} (\tilde{H} - B) & (\tilde{\mathbf{y}} - \mathbf{e}) \end{bmatrix} = \begin{bmatrix} U_{11} & \mathbf{u} \end{bmatrix} \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} V_{11} & \mathbf{v} \\ \mathbf{w}^T & v_{22} \end{bmatrix}^T \quad (2)$$

Assume

- Clearly this meets our criterion due to the 0 in the middle matrix
- Note this approach does not imply that s_{n+1} is zero in general
  - Rather we are using most of the elements of the already computed U, V, and S matrices to ascertain whether or not a feasible solution exists for B and $\mathbf{e}$

Total Least Squares (v)

- Multiplying the matrices in Eq. (1) gives

$$\begin{aligned} \tilde{H} &= U_{11}\Sigma V_{11}^T + s_{n+1}\mathbf{u}\,\mathbf{v}^T \\ \tilde{\mathbf{y}} &= U_{11}\Sigma\mathbf{w} + s_{n+1}v_{22}\mathbf{u} \end{aligned} \qquad (3)$$

- Multiplying the matrices in Eq. (2) gives

$$\begin{aligned} \tilde{H} - B &= U_{11}\Sigma V_{11}^T \\ \tilde{\mathbf{y}} - \mathbf{e} &= U_{11}\Sigma\mathbf{w} \end{aligned} \qquad (4)$$

- Substituting Eq. (3) into (4) gives

$$\begin{aligned} B &= s_{n+1}\mathbf{u}\,\mathbf{v}^T \\ \mathbf{e} &= s_{n+1}v_{22}\mathbf{u} \end{aligned} \qquad (5)$$

- Thus solutions for B and $\mathbf{e}$ are indeed possible
- Substituting Eq. (4) into $(\tilde{H} - B)\hat{\mathbf{x}}_{\mathrm{TLS}} = \tilde{\mathbf{y}} - \mathbf{e}$ gives

$$U_{11}\Sigma V_{11}^T\hat{\mathbf{x}}_{\mathrm{TLS}} = U_{11}\Sigma\mathbf{w} \qquad (6)$$

Total Least Squares (vi)

- Multiply the partitions of $V\,V^T = I$, $V^T V = I$ and $U^T U = I$

$$V\,V^T = \begin{bmatrix} V_{11}V_{11}^T + \mathbf{v}\,\mathbf{v}^T & V_{11}\mathbf{w} + v_{22}\mathbf{v} \\ \mathbf{w}^T V_{11}^T + v_{22}\mathbf{v}^T & \mathbf{w}^T\mathbf{w} + v_{22}^2 \end{bmatrix} = \begin{bmatrix} I_{n\times n} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \qquad \text{(7a)}$$

$$V^T V = \begin{bmatrix} V_{11}^T V_{11} + \mathbf{w}\,\mathbf{w}^T & V_{11}^T\mathbf{v} + v_{22}\mathbf{w} \\ \mathbf{v}^T V_{11} + v_{22}\mathbf{w}^T & \mathbf{v}^T\mathbf{v} + v_{22}^2 \end{bmatrix} = \begin{bmatrix} I_{n\times n} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \qquad \text{(7b)}$$

$$U^T U = \begin{bmatrix} U_{11}^T U_{11} & U_{11}^T\mathbf{u} \\ \mathbf{u}^T U_{11} & \mathbf{u}^T\mathbf{u} \end{bmatrix} = \begin{bmatrix} I_{n\times n} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \qquad \text{(7c)}$$

- From Eq. (7c) we have $U_{11}^T U_{11} = I_{n\times n}$
- Left multiplying Eq. (6) by the transpose of U_{11} and using the above identity leads

$$V_{11}^T \hat{\mathbf{x}}_{\mathrm{TLS}} = \mathbf{w} \qquad \text{(8)}$$

Total Least Squares (vii)

- Left multiplying both sides of this equation by V_{11} and using $V_{11}V_{11}^T = I_{n\times n} - \mathbf{v}\,\mathbf{v}^T$ from Eq. (7a) gives

$$(I_{n\times n} - \mathbf{v}\,\mathbf{v}^T)\hat{\mathbf{x}}_{\text{TLS}} = V_{11}\mathbf{w} = -v_{22}\mathbf{v}$$

where the identity $V_{11}\mathbf{w} + v_{22}\mathbf{v} = \mathbf{0}$ from Eq. (7a) was used

- Multiplying both sides of this equation by v_{22} and using $v_{22}^2 = 1 - \mathbf{v}^T\mathbf{v}$ from Eq. (7b) yields

$$v_{22}(I_{n\times n} - \mathbf{v}\,\mathbf{v}^T)\hat{\mathbf{x}}_{\text{TLS}} = \mathbf{v}\,\mathbf{v}^T\mathbf{v} - \mathbf{v}$$

- The solution is given by (simply substitute it in above to prove it)

v - From partation of SVD(D)

$$\boxed{\hat{\mathbf{x}}_{\text{TLS}} = -\mathbf{v}/v_{22}}$$

- Hence only the vector $\mathbf{v}$ and scalar v_{22} are required to be computed for the solution
  - Saves on the computational cost over the form in Eq. (8)

Total Least Squares (viii)

- Element-wise uncorrelated and stationary case
  - Same essential steps as before, but need one more step
  - Let the matrix R be given by

$$R = \text{blkdiag}\begin{bmatrix}\mathcal{R} & \cdots & \mathcal{R}\end{bmatrix}$$

  where $\mathcal{R}$ is an $(n+1) \times (n+1)$ matrix
    - Note that the last diagonal element of the matrix $\mathcal{R}$ is the variance associated with the measurement errors
  - When the matrix $\tilde{H}$ involves functions of time, then this choice for R is equivalent to assuming a stationary process
    - This assumes that the rows of the matrix $\tilde{H}$ have errors with equal covariances
- Steps for the solution
  - First take the Cholesky decomposition of $\mathcal{R}$

$$\mathcal{R} = C^T C$$

  where C is defined as an upper block diagonal matrix

Total Least Squares (ix)

- Partition the inverse of C as

$$C^{-1} = \begin{bmatrix} C_{11} & \mathbf{c} \\ \mathbf{0}^T & c_{22} \end{bmatrix},$$

where C_{11} is an $n \times n$ matrix, $\mathbf{c}$ is an $n \times 1$ vector and c_{22} is a scalar

- Take the singular value decomposition of the following matrix

$$\tilde{D}C^{-1} = USV^T$$

  - Note that the "reduced" form SVD is used (S is a square matrix)

- Partition the matrix V as

$$V = \begin{bmatrix} V_{11} & \mathbf{v} \\ \mathbf{w}^T & v_{22} \end{bmatrix},$$

where V_{11} is an $n \times n$ matrix, $\mathbf{v}$ is an $n \times 1$ vector and v_{22} is a scalar

- The total least squares solution assuming $\mathcal{R} = \sigma^2 I$ is

$$\hat{\mathbf{x}}_{\text{ITLS}} = -\mathbf{v}/v_{22}$$

- The overall solution for general $\mathcal{R}$ is given by

$$\boxed{\hat{\mathbf{x}}_{\text{TLS}} = (C_{11}\hat{\mathbf{x}}_{\text{ITLS}} - \mathbf{c})/c_{22}}$$

Total Least Squares (x)

- The estimate is given by

$$\hat{D} = U_n S_n V_n^T C$$

  where U_n is the truncation of the matrix U to $m \times n$, S_n is the truncation of the matrix S to $n \times n$, and V_n is the truncation of the matrix V to $(n+1) \times n$

- Covariance derivation for the previous case is possible
  - Crassidis, J.L., and Cheng, Y., "Error-Covariance Analysis for the Total Least Squares Problem," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 4, July-Aug. 2014, pp. 1053-1063.
  - Define U_n by its rows

$$U_n = \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_m^T \end{bmatrix}$$

  where $\mathbf{u}_i^T$ is the i^{th} row of U_n

Total Least Squares (xi)

- Form the following $(n+1)\times(n+1)$ matrix

$$\Omega_i = \begin{bmatrix} \mathbf{v} \\ v_{22} \end{bmatrix}^T \otimes \begin{bmatrix} S_n^{-1}\mathbf{u}_i \\ 0 \end{bmatrix}$$

where $\otimes$ denotes the Kronecker product

- Next the following matrix is computed

$$B = V\left[\sum_{i=1}^{m} \Omega_i\, Z\, \Omega_i^T\right] V^T, \quad \text{where} \quad Z \equiv C^{-T}\mathcal{R}\,C^{-1}$$

- Partition the matrix B as

$$B = \begin{bmatrix} B_{11} & \mathbf{b} \\ \mathbf{b}^T & b_{22} \end{bmatrix},$$

where B_{11} is an $n \times n$ matrix, $\mathbf{b}$ is an $n \times 1$ vector and b_{22} is a scalar

- The covariance of the estimation errors is given by

$$\boxed{P_{\text{TLS}} = v_{22}^{-2} c_{22}^{-2} C_{11}\left[B_{11} + v_{22}^{-2} b_{22} \mathbf{v}\,\mathbf{v}^T - v_{22}^{-1}(\mathbf{b}\,\mathbf{v}^T + \mathbf{v}\,\mathbf{b}^T)\right] C_{11}^T}$$

TLS Example (i)

- The true H and $\mathbf{x}$ quantities are given by

$$H = \begin{bmatrix} 1 & \sin(t) & \cos(t) \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 1 \\ 0.5 \\ 0.3 \end{bmatrix}$$

- The matrix $\mathcal{R}$ is given by — uncorrelated and stationary

Covariance of H and y

$$\mathcal{R} = \begin{bmatrix} 1\times10^{-4} & 1\times10^{-6} & 1\times10^{-5} & 1\times10^{-9} \\ 1\times10^{-6} & 1\times10^{-2} & 1\times10^{-7} & 1\times10^{-6} \\ 1\times10^{-5} & 1\times10^{-7} & 1\times10^{-3} & 1\times10^{-6} \\ 1\times10^{-9} & 1\times10^{-6} & 1\times10^{-6} & 1\times10^{-4} \end{bmatrix} \leftarrow$$

check if positive definite

- Synthetic measurements are generated using a sampling interval of 0.01 seconds to a final time of 10 seconds
- One thousand Monte Carlo runs are executed
  - Three-sigma error bounds are derived from the computed error-covariance matrix

TLS Example (ii)

```
% True Output and Basis Functions
x_true=[1;0.5;0.3];
y=x_true(1)+x_true(2)*sin(t)+x_true(3)*cos(t);
h=[ones(m,1) sin(t) cos(t)];

% Measurement Covariance and Cholesky Decomposition
r=[1e-4 1e-6 1e-5 1e-9
   1e-6 1e-2 1e-7 1e-6
   1e-5 1e-7 1e-3 1e-6
   1e-9 1e-6 1e-6 1e-4];
c=chol(r);ci=inv(c);

% Number of Monte Carlo Runs and Storage of Variables
m_monte=1000;
x_itls_storage=zeros(m_monte,3);
x_tls_storage=zeros(m_monte,3);
```

TLS Example (iii)

```
% Main Loop
for i=1:m_monte

% Generate Noise
noise=correlated_noise(r,m);
ym=y+noise(:,4);hm=[h(:,1)+noise(:,1) h(:,2)+noise(:,2) h(:,3)+noise(:,3)];

% Cholesky Decomposition and SVD of Augmented Matrix
d=[hm ym];[m,n_d]=size(d);n=n_d-1;d_star=d*ci;[u,s,v]=svd(d_star,0);
```

skinny -form

```
% Isotropic Solution
x_itls=-v(1:n,n+1:n_d)*inv(v(n+1:n_d,n+1:n_d));

% Final Solution
x_tls=(ci(1:n,1:n)*x_itls-ci(1:n,n+1:n_d))*inv(ci(n+1:n_d,n+1:n_d));

% Store Solutions
x_itls_storage(i,:)=x_itls';x_tls_storage(i,:)=x_tls';

end
```

TLS Example (iv)

```
% Show Results
mean_err_itls=mean(x_itls_storage)-x_true'
mean_err_tls=mean(x_tls_storage)-x_true'

cov_itls=cov(x_itls_storage);
cov_tls=cov(x_tls_storage);

trace_mse_itls=trace(cov_itls+mean_err_itls'*mean_err_itls)'
trace_mse_tls=trace(cov_tls+mean_err_tls'*mean_err_tls)'

% Compute Covariance
rot_cov=ci'*r*ci;
s_bar_inv=inv(s(1:n,1:n));
v_last_col_tran=v(:,n_d)';
```

TLS Example (v)

```
d_cov=zeros(n_d,n_d);
for i = 1:m
 omega_mat=kron(v_last_col_tran,[s_bar_inv*u(i,1:n)';0]);
 d_cov=d_cov+omega_mat*rot_cov*omega_mat';
end
b_cov=v*d_cov*v';

cov_anal_itls=v(n_d,n_d)^(-2)*(b_cov(1:n,1:n)+v(n_d,n_d)^(-2)*v(1:n,n_d)...
          *v(1:n,n_d)'*b_cov(n_d,n_d)-v(n_d,n_d)^(-1)*(b_cov(1:n,n_d)...
          *v(1:n,n_d)'+v(1:n,n_d)*b_cov(1:n,n_d)'));
cov_anal_tls=ci(1:n,1:n)*cov_anal_itls*ci(1:n,1:n)'/ci(n_d,n_d)^2;

% Three Sigma Bounds
sig3=3*diag(cov_anal_tls).^(0.5);
```

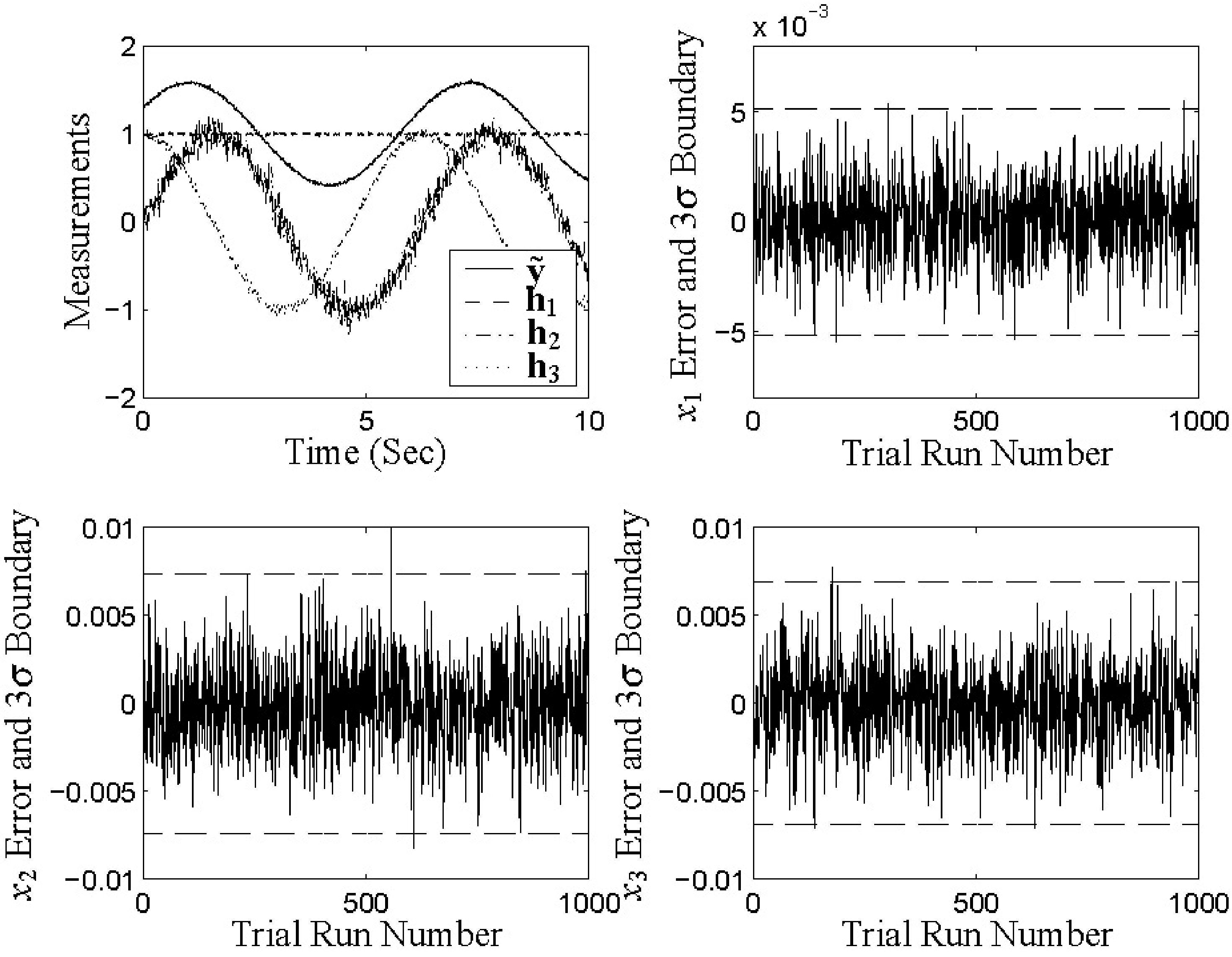
TLS Example (vi)

```
% Plot Results
subplot(221)
plot(t,ym,t,hm(:,1),'--',t,hm(:,2),'-.',t,hm(:,3),':')
set(gca,'fontsize',12)
legend('y','h1','h2','h3',4)
ylabel('Measurements')
xlabel('Time (Sec)')

subplot(222)
x_axis=[1:m_monte]';
plot(x_axis,sig3(1)*ones(m_monte,1),'r--',x_axis,x_tls_storage(:,1)-
x_true(1),'b',x_axis,-sig3(1)*ones(m_monte,1),'r--')
set(gca,'fontsize',12)
axis([0 m_monte -8e-3 8e-3])
ylabel('x1')
xlabel('Trial Run Number')
```

TLS Example (vii)

```
subplot(223)
plot(x_axis,sig3(2)*ones(m_monte,1),'r--',x_axis,x_tls_storage(:,2)-
x_true(2),'b',x_axis,-sig3(2)*ones(m_monte,1),'r--')
set(gca,'fontsize',12)
axis([0 m_monte -0.01 0.01])
ylabel('x2')
xlabel('Trial Run Number')

subplot(224)
plot(x_axis,sig3(3)*ones(m_monte,1),'r--',x_axis,x_tls_storage(:,3)-
x_true(3),'b',x_axis,-sig3(3)*ones(m_monte,1),'r--')
set(gca,'fontsize',12)
axis([0 m_monte -0.01 0.01])
ylabel('x3')
xlabel('Trial Run Number')
```

TLS Example (viii)



Total Least Squares (i)

- Element-Wise Uncorrelated and Non-Stationary Case
  - The covariance matrix is a block diagonal matrix

$$R = \mathrm{blkdiag}\begin{bmatrix}\mathcal{R}_1 & \cdots & \mathcal{R}_m\end{bmatrix}$$

with

$$\mathcal{R}_i = \begin{bmatrix}\mathcal{R}_{hh_i} & \mathcal{R}_{hy_i} \\ \mathcal{R}_{hy_i}^T & \mathcal{R}_{yy_i}\end{bmatrix}$$

where $\mathcal{R}_{hh_i}$ is an $n \times n$ matrix, $\mathcal{R}_{hy_i}$ is $n \times 1$ vector and $\mathcal{R}_{yy_i}$ is a scalar

  - Partition ΔH and the vector $\Delta \mathbf{y}$ by their rows

$$\Delta H = \begin{bmatrix}\boldsymbol{\delta}\mathbf{h}_1^T \\ \boldsymbol{\delta}\mathbf{h}_2^T \\ \vdots \\ \boldsymbol{\delta}\mathbf{h}_m^T\end{bmatrix}, \quad \Delta \mathbf{y} = \begin{bmatrix}\delta y_1 \\ \delta y_2 \\ \vdots \\ \delta y_m\end{bmatrix} \quad \Longrightarrow \quad \begin{aligned}\mathcal{R}_{hh_i} &= E\left\{\boldsymbol{\delta}\mathbf{h}_i \boldsymbol{\delta}\mathbf{h}_i^T\right\} \\ \mathcal{R}_{hy_i} &= E\left\{\delta y_i \boldsymbol{\delta}\mathbf{h}_i\right\} \\ \mathcal{R}_{yy_i} &= E\left\{\delta y_i^2\right\}\end{aligned}$$

Total Least Squares (ii)

- Partition $\tilde{D}$, $\hat{D}$, $\tilde{H}$ and the vector $\tilde{\mathbf{y}}$ by their rows

$$\tilde{D} = \begin{bmatrix} \tilde{\mathbf{d}}_1^T \\ \tilde{\mathbf{d}}_2^T \\ \vdots \\ \tilde{\mathbf{d}}_m^T \end{bmatrix}, \quad \hat{D} = \begin{bmatrix} \hat{\mathbf{d}}_1^T \\ \hat{\mathbf{d}}_2^T \\ \vdots \\ \hat{\mathbf{d}}_m^T \end{bmatrix}, \quad \tilde{H} = \begin{bmatrix} \tilde{\mathbf{h}}_1^T \\ \tilde{\mathbf{h}}_2^T \\ \vdots \\ \tilde{\mathbf{h}}_m^T \end{bmatrix}, \quad \tilde{\mathbf{y}} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_m \end{bmatrix}$$

- The loss function reduces down to

$$J(\hat{\mathbf{x}}) = \frac{1}{2}\sum_{i=1}^{m}(\tilde{\mathbf{d}}_i - \hat{\mathbf{d}}_i)^T \mathcal{R}_i^{-1}(\tilde{\mathbf{d}}_i - \hat{\mathbf{d}}_i), \quad \text{s.t.} \quad \hat{\mathbf{d}}_j^T \hat{\mathbf{z}} = 0, \quad j = 1, 2, \ldots, m$$

- Use Lagrange multiplier approach to convert problem to an unconstrained one

$$J'(\hat{\mathbf{d}}_i) = \lambda_1 \hat{\mathbf{d}}_1^T \hat{\mathbf{z}} + \lambda_2 \hat{\mathbf{d}}_2^T \hat{\mathbf{z}} + \cdots + \lambda_m \hat{\mathbf{d}}_m^T \hat{\mathbf{z}} + \frac{1}{2}\sum_{i=1}^{m}(\tilde{\mathbf{d}}_i - \hat{\mathbf{d}}_i)^T \mathcal{R}_i^{-1}(\tilde{\mathbf{d}}_i - \hat{\mathbf{d}}_i) \quad (1)$$

where λ_i is a Lagrange multiplier for each constraint

Total Least Squares (iii)

- Taking the partial of Eq. (1) w.r.t. each $\hat{\mathbf{d}}_i$ leads to the following m necessary conditions

$$\mathcal{R}_i^{-1}\hat{\mathbf{d}}_i - \mathcal{R}_i^{-1}\tilde{\mathbf{d}}_i + \lambda_i\hat{\mathbf{z}} = \mathbf{0}, \quad i = 1, 2, \ldots, m \qquad (2)$$

- Left multiplying by $\hat{\mathbf{z}}^T\mathcal{R}_i$ and using the constraint $\hat{\mathbf{d}}_i^T\hat{\mathbf{z}} = 0$ gives

$$\lambda_i = \frac{\hat{\mathbf{z}}^T\tilde{\mathbf{d}}_i}{\hat{\mathbf{z}}^T\mathcal{R}_i\,\hat{\mathbf{z}}}$$

- Substituting this equation back into Eq. (2) gives

$$\hat{\mathbf{d}}_i = \left[I_{(n+1)\times(n+1)} - \frac{\mathcal{R}_i\hat{\mathbf{z}}\,\hat{\mathbf{z}}^T}{\hat{\mathbf{z}}^T\mathcal{R}_i\,\hat{\mathbf{z}}}\right]\tilde{\mathbf{d}}_i \qquad (3)$$

where $I_{(n+1)\times(n+1)}$ is an $(n+1)\times(n+1)$ identity matrix

- Substituting Eq. (3) into Eq. (1) gives

$$J(\hat{\mathbf{x}}) = \frac{1}{2}\sum_{i=1}^{m}\frac{(\tilde{\mathbf{d}}_i^T\hat{\mathbf{z}})^2}{\hat{\mathbf{z}}^T\mathcal{R}_i\,\hat{\mathbf{z}}}$$

- Unfortunately represents a non-convex optimization problem

Total Least Squares (iv)

- The necessary conditions for optimality gives

$$\frac{\partial J(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} = \sum_{i=1}^{m} \frac{e_i\, \tilde{\mathbf{h}}_i}{\hat{\mathbf{x}}^T \mathcal{R}_{hh_i} \hat{\mathbf{x}} - 2\mathcal{R}_{hy_i}^T \hat{\mathbf{x}} + \mathcal{R}_{yy_i}} - \frac{e_i^2 (\mathcal{R}_{hh_i} \hat{\mathbf{x}} - \mathcal{R}_{hy_i})}{(\hat{\mathbf{x}}^T \mathcal{R}_{hh_i} \hat{\mathbf{x}} - 2\mathcal{R}_{hy_i}^T \hat{\mathbf{x}} + \mathcal{R}_{yy_i})^2} = \mathbf{0}$$

where $e_i \equiv \tilde{\mathbf{h}}_i^T \hat{\mathbf{x}} - \tilde{y}_i$

- A closed-form solution is not possible unfortunately
- An iteration approach is provided by

$$\hat{\mathbf{x}}^{(j+1)} = \left[\sum_{i=1}^{m} \frac{\tilde{\mathbf{h}}_i \tilde{\mathbf{h}}_i^T}{\gamma_i(\hat{\mathbf{x}}^{(j)})} - \frac{e_i^2(\hat{\mathbf{x}}^{(j)}) \mathcal{R}_{hh_i}}{\gamma_i^2(\hat{\mathbf{x}}^{(j)})} \right]^{-1} \left[\sum_{i=1}^{m} \frac{\tilde{y}_i \tilde{\mathbf{h}}_i}{\gamma_i(\hat{\mathbf{x}}^{(j)})} - \frac{e_i^2(\hat{\mathbf{x}}^{(j)}) \mathcal{R}_{hy_i}}{\gamma_i^2(\hat{\mathbf{x}}^{(j)})} \right]$$

$$\gamma_i(\hat{\mathbf{x}}^{(j)}) \triangleq \hat{\mathbf{x}}^{(j)T} \mathcal{R}_{hh_i} \hat{\mathbf{x}}^{(j)} - 2\mathcal{R}_{hy_i}^T \hat{\mathbf{x}}^{(j)} + \mathcal{R}_{yy_i}$$

$$e_i(\hat{\mathbf{x}}^{(j)}) \triangleq \tilde{\mathbf{h}}_i^T \hat{\mathbf{x}}^{(j)} - \tilde{y}_i$$

where $\hat{\mathbf{x}}^{(j)}$ denotes the estimate at the j^{th} iteration

TLS – FIM (i)

- Derivation of the Fisher Information matrix (FIM)
  - Note that the solution solves the MLE problem and that the estimate is unbiased (not quite true, but the bias is small)
  - Thus the covariance is given by the inverse of the FIM
  - The likelihood is treated as a function of $\mathbf{x}$ and H

$$p(\tilde{D}|\mathbf{x}, H) = \frac{1}{(2\pi)^{m/2}\left[\det\left(R\right)\right]^{1/2}} \exp\left\{-\frac{1}{2}\mathrm{vec}^T\left(\tilde{D}^T - D^T(\mathbf{x},\, H)\right) R^{-1}\mathrm{vec}\left(\tilde{D}^T - D^T(\mathbf{x},\, H)\right)\right\}$$

with $D(\mathbf{x},\, H) \equiv [H \quad H\mathbf{x}]$

  - For this case because $\tilde{\mathbf{d}}_i$ and $\tilde{\mathbf{d}}_j$, $i \neq j$, are independent of each other, the likelihood function reduces to

$$\begin{aligned} p(\tilde{D}|\mathbf{x}, H) &= \frac{1}{\prod_{i=1}^m \left[\det\left(2\pi\mathcal{R}_i\right)\right]^{1/2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^m \left(\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{x}_i,\, \mathbf{h}_i)\right)^T \mathcal{R}_i^{-1}\left(\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{x}_i,\, \mathbf{h}_i)\right)\right\} \\ &= \prod_{i=1}^m p\left(\tilde{\mathbf{d}}_i|\mathbf{x}, \mathbf{h}_i\right) \end{aligned}$$

with

$$\mathbf{d}_i(\mathbf{x}_i,\, \mathbf{h}_i) \equiv [\mathbf{h}_i^T \ \ \mathbf{h}_i^T\mathbf{x}]^T$$

$$p(\tilde{\mathbf{d}}_i|\mathbf{x}, \mathbf{h}_i) \equiv \frac{1}{\left[\det\left(2\pi\mathcal{R}_i\right)\right]^{1/2}} \exp\left\{-\frac{1}{2}\left(\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{x}_i,\, \mathbf{h}_i)\right)^T \mathcal{R}_i^{-1}\left(\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{x}_i,\, \mathbf{h}_i)\right)\right\}$$

TLS – FIM (ii)

- We now derive the FIM for $p(\tilde{\mathbf{d}}_i|\mathbf{x},\mathbf{h}_i)$; define

$$\mathbf{a}_i \equiv \begin{bmatrix} \mathbf{x} \\ \mathbf{h}_i \end{bmatrix}, \quad p(\tilde{\mathbf{d}}_i|\mathbf{a}_i) \equiv p(\tilde{\mathbf{d}}_i|\mathbf{x},\mathbf{h}_i), \quad \mathbf{d}_i(\mathbf{a}_i) \equiv \mathbf{d}_i(\mathbf{x},\mathbf{h}_i)$$

  - The FIM for $\mathbf{a}_i$ is

$$F_i^a = E\left\{ \left(\frac{\partial}{\partial \mathbf{a}_i} \ln[p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)] \right) \left(\frac{\partial}{\partial \mathbf{a}_i} \ln[p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)] \right)^T \right\}$$

  - The natural logarithm of $p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)$ is

$$\ln[p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)] = -\frac{1}{2}\left(\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{a}_i)\right)^T \mathcal{R}_i^{-1} \left(\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{a}_i)\right) - \frac{1}{2}\ln\det(2\pi R_i)$$

  - Taking partials gives

$$\frac{\partial}{\partial \mathbf{a}_i} \ln[p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)] = \begin{bmatrix} 0_{n\times n} & \mathbf{h}_i \\ I_{n\times n} & \mathbf{x} \end{bmatrix} \mathcal{R}_i^{-1} \left(\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{a}_i)\right)$$

TLS – FIM (iii)

- Because $E\{\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{a}_i)\} = \mathbf{0}$, then

$$E\left\{\frac{\partial}{\partial \mathbf{a}_i}\ln[p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)]\right\} = \begin{bmatrix} 0_{n\times n} & \mathbf{h}_i \\ I_{n\times n} & \mathbf{x} \end{bmatrix} \mathcal{R}_i^{-1} E\left\{\left(\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{a}_i)\right)\right\} = \mathbf{0}$$

- This means that the following regularity condition is met

$$E\left\{\frac{\partial}{\partial \mathbf{a}_i}\ln[p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)]\right\} \equiv \int \frac{\partial}{\partial \mathbf{a}_i}\ln[p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)]\, p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)\, d\tilde{\mathbf{d}}_i = \int \left[\frac{\partial p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)}{\partial \mathbf{a}_i}\right] d\tilde{\mathbf{d}}_i = \mathbf{0}$$

- Post-multiplying $\partial \ln[p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)]/\partial \mathbf{a}_i$ by its transpose, taking the expectation and using

$$E\left\{\left(\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{a}_i)\right)\left(\tilde{\mathbf{d}}_i - \mathbf{d}_i(\mathbf{a}_i)\right)^T\right\} = \mathcal{R}_i$$

gives

$$F_i^a = E\left\{\left(\frac{\partial}{\partial \mathbf{a}_i}\ln[p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)]\right)\left(\frac{\partial}{\partial \mathbf{a}_i}\ln[p(\tilde{\mathbf{d}}_i|\mathbf{a}_i)]\right)^T\right\} = \begin{bmatrix} 0_{n\times n} & \mathbf{h}_i \\ I_{n\times n} & \mathbf{x} \end{bmatrix} \mathcal{R}_i^{-1} \begin{bmatrix} 0_{n\times n} & \mathbf{h}_i \\ I_{n\times n} & \mathbf{x} \end{bmatrix}^T$$

TLS – FIM (iv)

- The next step is to derive the FIM for $\hat{\mathbf{x}}$
  - The total Fisher information for $\hat{\mathbf{x}}$ will be denoted by F and the Fisher information corresponding to a single measurement $\tilde{\mathbf{d}}_i$ will be denoted by F_i (note that all the F_i are rank-one)
  - Because $\tilde{\mathbf{d}}_i$ and $\tilde{\mathbf{d}}_j$ are independent of each other, and $\mathbf{h}_i$ and $\mathbf{h}_j$ are different for $i \neq j$, then $F = \sum_{i=1}^{m} F_i$
  - Partition the inverse of the matrix $\mathcal{R}$ as

$$\mathcal{R}_i^{-1} \triangleq \begin{bmatrix} \Gamma_i & \boldsymbol{\beta}_i \\ \boldsymbol{\beta}_i^T & \vartheta_i \end{bmatrix}$$

    where Γ_i is an $n \times n$ matrix, $\boldsymbol{\beta}_i$ is an $n \times 1$ vector and ϑ_i is a scalar
  - The elements of $\mathcal{R}$ can be written as

$$\mathcal{R}_{hh_i} = \left(\Gamma_i - \boldsymbol{\beta}_i\boldsymbol{\beta}_i^T/\vartheta_i\right)^{-1} \qquad (1)$$

$$\mathcal{R}_{hy_i} = -\mathcal{R}_{hh_i}\boldsymbol{\beta}_i/\vartheta_i$$

$$\mathcal{R}_{yy_i} = \frac{1}{\vartheta_i} + \frac{\boldsymbol{\beta}_i^T\mathcal{R}_{hh_i}\boldsymbol{\beta}_i}{\vartheta_i^2}$$

TLS – FIM (v)

- Because F_i is rank 1, the following are equivalent

$$\mathbf{h}_i^T F_i \mathbf{h}_i = \frac{h_i^4}{\mathbf{z}^T \mathcal{R}_i\, \mathbf{z}}$$

$$\eta_i^{-1} \equiv \left(\mathbf{h}_i^T F_{xx_i} \mathbf{h}_i - \mathbf{h}_i^T F_{xh_i} F_{hh_i}^{-1} F_{xh_i}^T \mathbf{h}_i\right)^{-1} = \frac{\mathbf{z}^T \mathcal{R}_i\, \mathbf{z}}{h_i^4} \qquad (2)$$

$$F_i = \frac{\mathbf{h}_i \mathbf{h}_i^T}{\mathbf{z}^T \mathcal{R}_i\, \mathbf{z}} \qquad (3)$$

where $h_i^2 \triangleq \mathbf{h}_i^T \mathbf{h}_i$

- Let's prove these now; define the following variables

$$\mathcal{A}_i \equiv \mathbf{h}_i^T F_{xx_i} \mathbf{h}_i,\ \mathcal{B}_i \equiv \mathbf{h}_i^T F_{xh_i},\ \mathcal{C}_i \equiv F_{xh_i}^T \mathbf{h}_i = \mathcal{B}^T \text{ and } \mathcal{D}_i \equiv F_{hh_i}$$

- Explicitly computing $\mathcal{A}_i$ gives

$$\mathcal{A}_i = \mathbf{h}_i^T \begin{bmatrix} 0_{n\times n} & \mathbf{h}_i \end{bmatrix} \begin{bmatrix} \Gamma_i & \boldsymbol{\beta}_i \\ \boldsymbol{\beta}_i^T & \vartheta_i \end{bmatrix} \begin{bmatrix} 0_{n\times n} \\ \mathbf{h}_i^T \end{bmatrix} \mathbf{h}_i = h_i^4\, \vartheta_i$$

TLS – FIM (vi)

- Explicitly computing $\mathcal{B}_i$ gives

$$\mathcal{B}_i = \mathbf{h}_i^T \begin{bmatrix} 0_{n\times n} & \mathbf{h}_i \end{bmatrix} \begin{bmatrix} \Gamma_i & \boldsymbol{\beta}_i \\ \boldsymbol{\beta}_i^T & \vartheta_i \end{bmatrix} \begin{bmatrix} I_{n\times n} \\ \mathbf{x}^T \end{bmatrix} = h_i^2(\boldsymbol{\beta}_i^T + \vartheta_i \mathbf{x}^T)$$

- Explicitly computing $\mathcal{D}_i$ gives

$$\mathcal{D}_i = \begin{bmatrix} I_{n\times n} & \mathbf{x} \end{bmatrix} \begin{bmatrix} \Gamma_i & \boldsymbol{\beta}_i \\ \boldsymbol{\beta}_i^T & \vartheta_i \end{bmatrix} \begin{bmatrix} I_{n\times n} \\ \mathbf{x}^T \end{bmatrix} = \Gamma_i + \mathbf{x}\,\boldsymbol{\beta}_i^T + \boldsymbol{\beta}_i \mathbf{x} + \vartheta_i \mathbf{x}\,\mathbf{x}^T$$

- By the matrix inversion lemma we have

$$\left(\mathcal{A}_i - \mathcal{B}_i \mathcal{D}_i^{-1} \mathcal{C}_i\right)^{-1} = \mathcal{A}_i^{-1} + \mathcal{A}_i^{-1} \mathcal{B}_i \left(\mathcal{D}_i - \mathcal{C}_i \mathcal{A}_i^{-1} \mathcal{B}_i\right)^{-1} \mathcal{C}_i \mathcal{A}_i^{-1}$$

- Substitute the $\mathcal{A}_i$, $\mathcal{B}_i$ and $\mathcal{D}_i$ expressions into $\left(\mathcal{D}_i - \mathcal{C}_i \mathcal{A}_i^{-1} \mathcal{B}_i\right)$

$$\mathcal{D}_i - \mathcal{C}_i \mathcal{A}_i^{-1} \mathcal{B}_i = \Gamma_i - \boldsymbol{\beta}_i \boldsymbol{\beta}_i^T / \vartheta_i$$

- Using Eq. (1) gives

$$\left(\mathcal{D}_i - \mathcal{C}_i \mathcal{A}_i^{-1} \mathcal{B}_i\right)^{-1} = \mathcal{R}_{hh_i}$$

TLS – FIM (vii)

- So η_i^{-1} in Eq. (2) is explicitly given by

$$\begin{aligned}
\eta_i^{-1} &= \frac{1}{h_i^4\,\vartheta_i} + \frac{1}{\vartheta_i^2}\left(\boldsymbol{\beta}_i^T + \vartheta_i \mathbf{x}^T\right)\mathcal{R}_{hh_i}\left(\boldsymbol{\beta}_i + \vartheta_i \mathbf{x}\right) \\
&= \frac{1}{h_i^4}\left(\mathbf{x}^T\mathcal{R}_{hh_i}\mathbf{x} + \frac{2}{\vartheta_i}\mathbf{x}^T\mathcal{R}_{hh_i}\boldsymbol{\beta}_i + \frac{1}{\vartheta_i} + \frac{1}{\vartheta_i^2}\boldsymbol{\beta}_i^T\mathcal{R}_{hh_i}\boldsymbol{\beta}_i\right) \\
&= \frac{1}{h_i^4}\left(\mathbf{x}^T\mathcal{R}_{hh_i}\mathbf{x} - 2\mathbf{x}^T\mathcal{R}_{hy_i} + \mathcal{R}_{yy_i}\right) \\
&= \frac{\mathbf{z}^T\mathcal{R}_i\,\mathbf{z}}{h_i^4}
\end{aligned}$$

- Therefore, from Eq. (3) the FIM is given by

$$\boxed{F = \sum_{i=1}^{m} \frac{\mathbf{h}_i\mathbf{h}_i^T}{\mathbf{z}^T\mathcal{R}_i\,\mathbf{z}}}$$

- Note that if there are no errors in the basis functions, then

$$F = \sum_{i=1}^{m} \mathcal{R}_{yy_i}^{-1}\mathbf{h}_i\mathbf{h}_i^T$$

= FIM for the standard least squares problem

Example (i)

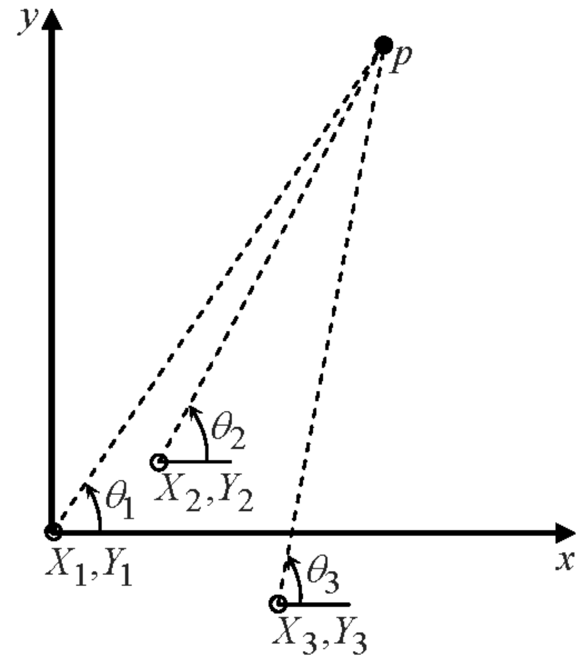
- Bearings-Only Point Estimation
  - The goal is to estimate the point p with coordinates (x, y) from bearings-only measurements
  - The baseline points, denoted by X_i and Y_i, are assumed to be imprecisely known
  - Bearing measurement model and baseline point models

$$\tilde{\theta}_i = \theta_i + \delta\theta_i$$

$$\tilde{X}_i = X_i + \delta X_i$$

$$\tilde{Y}_i = Y_i + \delta Y_i$$

where $\delta\theta_i$, δX_i and δY_i are zero-mean Gaussian processes with std $\sigma_{\theta i}$, σ_{Xi} and σ_{Yi}



- Observation model

$$\theta_i = \tan^{-1}\left(\frac{y - Y_i}{x - X_i}\right)$$

  - Nonlinear model, but we can convert to a linear one

Example (ii)

- Derived covariance

$$\mathcal{R}_{yy_i} = \sigma_{\theta_i}^2 \{[X_i \cos(\theta_i) + Y_i \sin(\theta_i)]^2 + \sigma_{X_i}^2 \cos^2(\theta_i) + \sigma_{Y_i}^2 \sin^2(\theta_i)\}$$
$$+ \sigma_{X_i}^2 \sin^2(\theta_i) + \sigma_{Y_i}^2 \cos^2(\theta_i)$$

$$\mathcal{R}_{hh_i} = \sigma_{\theta_i}^2 \begin{bmatrix} \cos^2(\theta_i) & \sin(\theta_i)\cos(\theta_i) \\ \sin(\theta_i)\cos(\theta_i) & \sin^2(\theta_i) \end{bmatrix}$$

$$\mathcal{R}_{hy_i} = \sigma_{\theta_i}^2 [X_i \cos(\theta_i) + Y_i \sin(\theta_i)] \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix}$$

- Note that the covariance matrix does not contain the true locations x and y, unlike other approaches
  - Stansfield, R. G., “Statistical Theory of D.F. Fixing,” *Journal of the Institution of Electrical Engineers – Part IIIA: Radiocommunication*, Vol. 94, No. 15, March-April 1947, pp. 762–770.
  - Gavish, M. and Miss, A. J., “Performance Analysis of Bearing-Only Target Location Algorithm,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 28, No. 3, July 1992, pp. 817–828.

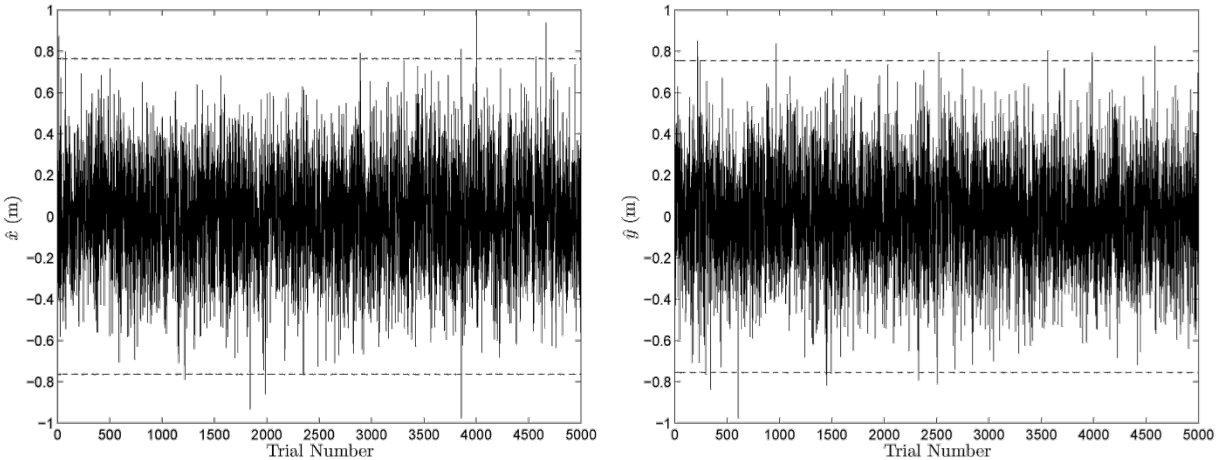
Example (iii)

- Simulation
  - The location of the point p is given at $(100,\,200)$ meters
  - The baseline points are time varying with $X_i = 500\sin(0.01\,t_i)$ and $Y_i = 300\cos(0.2\,t_i)$
  - The variances are given by $\sigma^2_{\theta_i} = (1\pi/180)^2$ rad^2 and $\sigma^2_{X_i} = \sigma^2_{Y_i} = 25$ m^2 for all i points
  - The final time of the simulation run is 10 seconds and measurements of are taken at 0.01 second intervals
  - Five thousand Monte Carlo runs are executed in order to compare the actual errors with the computed 3σ bounds

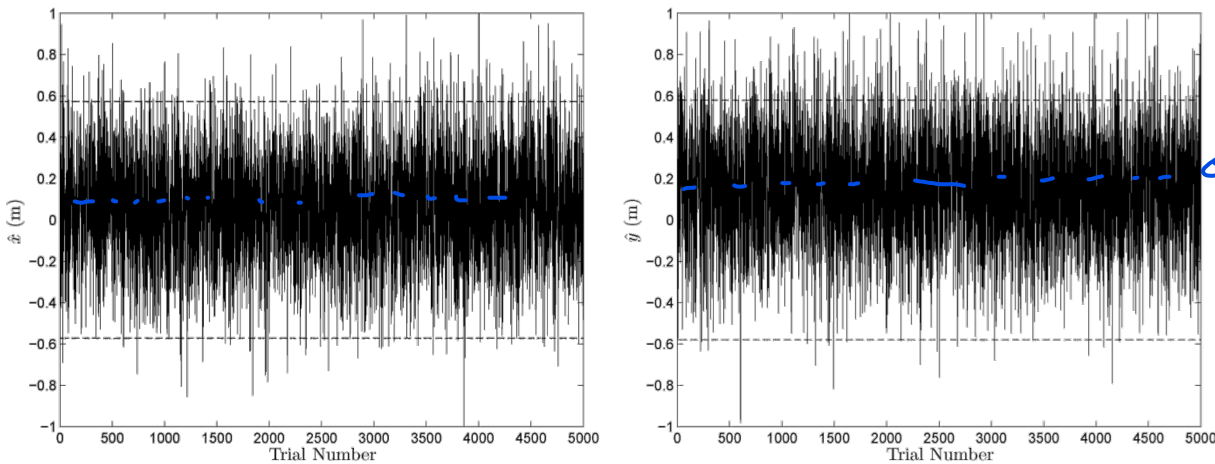
Example (iv)

TLS estimate errors and bounds

mean ≈ 0



What happens if we use standard least squares? Under estimated errors



bias is visible

Linear least squares solution is not optimal and even biased

Example (v)

```
% Time
t=[0:0.01:10]';m=length(t);

% Object Location and Sensor Locations
x_obj=[100;200];x_sensor=[500*sin(0.01*t) 300*cos(0.2*t)];

% True Angle
theta=atan((x_obj(2)-x_sensor(:,2))./(x_obj(1)-x_sensor(:,1)));

% True H and y Matrices
h=[-sin(theta) cos(theta)];y=h*x_obj;

% Measurement Standard Deviation
sig_theta=1*pi/180;
sig_x=5;sig_y=5;

% Monte Carlo Runs
m_monte=5000;x_tls_monte=zeros(m_monte,2);x_ls_monte=zeros(m_monte,2);
sig3_tls=zeros(m_monte,2);sig3_ls=zeros(m_monte,2);
```

Example (vi)

```
% Estimates for H and y
hest=zeros(m,2);yest=zeros(m,1);sig3_h=zeros(m,2);sig3_y=zeros(m,1);

for j = 1:m_monte

% Generate Measurements
thetam=theta+sig_theta*randn(m,1);
hm=[-sin(thetam) cos(thetam)];
x_sensorm=[x_sensor(:,1)+sig_x*randn(m,1) x_sensor(:,2)+sig_y*randn(m,1)];
ym=hm(:,1).*x_sensorm(:,1)+hm(:,2).*x_sensorm(:,2);

% Least Squares Solution
meas_var_ls=sig_theta^2*(cos(thetam).*x_sensorm(:,1)...
   +sin(thetam).*x_sensorm(:,2).^2+(sig_x^2*cos(thetam).^2)...
   +sig_y^2*sin(thetam).^2)+(sig_x^2*sin(thetam).^2)...
   +sig_y^2*cos(thetam).^2;
p_ls=inv(([hm(:,1)./meas_var_ls hm(:,2)./meas_var_ls])'*hm);
x_ls=p_ls*([hm(:,1)./meas_var_ls hm(:,2)./meas_var_ls])'*ym;
```

Example (vii)

```
% Total Least Sqaures Solution
x_tls=x_ls;j_count=1;max_it=100;stop_crit=10;
while stop_crit > 1e-5;
 x_tls_old=x_tls; z=[x_tls_old;-1]; g=zeros(2); pvec=zeros(2,1);
 for i=1:m
  sincos=cos(thetam(i))*x_sensorm(i,1)+sin(thetam(i))*x_sensorm(i,2);
  ryy=sig_theta^2*(sincos^2+(sig_x^2*cos(thetam(i))^2)...
  +sig_y^2*sin(thetam(i))^2)+(sig_x^2*sin(thetam(i))^2)...
  +sig_y^2*cos(thetam(i))^2;
  rhh=sig_theta^2*[cos(thetam(i))^2 cos(thetam(i))*sin(thetam(i))
      cos(thetam(i))*sin(thetam(i)) sin(thetam(i))^2];
  rhy=sig_theta^2*sincos*[cos(thetam(i));sin(thetam(i))];
  rd=[rhh rhy;rhy' ryy];gam=z'*rd*z;e_res=hm(i,:)*x_tls_old-ym(i);
  g=g+hm(i,:)'*hm(i,:)/gam-rhh*e_res^2/gam^2;
  pvec=pvec+hm(i,:)'*ym(i)/gam-rhy*e_res^2/gam^2;
 end
gi=inv(g);x_tls=gi*pvec;
stop_crit=norm(x_tls_old-x_tls);
j_count=j_count+1;if j_count > max_it, break, disp('Maximum Iterations Achieved'), end
end
```

Example (viii)

```
% Estimates and 3-sigma Bounds
x_tls_monte(j,:)=x_tls';
sig3_tls(j,:)=diag(gi)'.^(0.5)*3;
x_ls_monte(j,:)=x_ls';
sig3_ls(j,:)=diag(p_ls)'.^(0.5)*3;

end

% Covariance for H and y
z=[x_tls;-1];
for i=1:m
 sincos=cos(thetam(i))*x_sensorm(i,1)+sin(thetam(i))*x_sensorm(i,2);
 ryy=sig_theta^2*(sincos^2+(sig_x^2*cos(thetam(i))^2)...
     +sig_y^2*sin(thetam(i))^2)+(sig_x^2*sin(thetam(i))^2)...
     +sig_y^2*cos(thetam(i))^2;
 rhh=sig_theta^2*[cos(thetam(i))^2 cos(thetam(i))*sin(thetam(i))
     cos(thetam(i))*sin(thetam(i)) sin(thetam(i))^2];
 rhy=sig_theta^2*sincos*[cos(thetam(i));sin(thetam(i))];
 rd=[rhh rhy;rhy' ryy];gam=z'*rd*z; e_res=hm(i,:)*x_tls-ym(i);
 hest(i,:)=hm(i,:)-(rhh*x_tls-rhy)'*e_res/gam;yest(i,:)=ym(i)-(rhy'*x_tls-ryy)*e_res/gam;
```

Example (ix)

```
b=rd(1:2,1:2)*x_tls-rd(1:2,3);
mh=[eye(2)-b*x_tls'/gam b/gam];
nh=-b*hest(i,:)/gam;
sig3_h(i,:)=diag(mh*rd*mh'+nh*gi*nh')'.^(0.5)*3;

beta=rd(1:2,3)'*x_tls-rd(3,3);
my=[-beta*x_tls'/gam 1+beta/gam];
ny=-beta*hest(i,:)/gam;
sig3_y(i,:)=diag(my*rd*my'+ny*gi*ny')'.^(0.5)*3;
end
```

Example (x)

```
% Plot TLS Results

x_monte=[1:m_monte]';
plot(x_monte,-sig3_tls(:,1),'--',x_monte,x_tls_monte(:,1)-x_obj(1),x_monte,sig3_tls(:,1),'--')
set(gca,'fontsize',12)
axis([0 5000 -1 1])
ylabel('x1')
xlabel('Trial Number')

pause

plot(x_monte,-sig3_tls(:,2),'--',x_monte,x_tls_monte(:,2)-x_obj(2),x_monte,sig3_tls(:,2),'--')
set(gca,'fontsize',12)
axis([0 5000 -1 1])
ylabel('x2')
xlabel('Trial Number')

pause
```

Example (xi)

```
% Plot Standard LS Results

plot(x_monte,-sig3_ls(:,1),'--',x_monte,x_ls_monte(:,1)-x_obj(1),x_monte,sig3_ls(:,1),'--')
set(gca,'fontsize',12)
axis([0 5000 -1 1])
ylabel('x1')
xlabel('Trial Number')

pause

plot(x_monte,-sig3_ls(:,2),'--',x_monte,x_ls_monte(:,2)-x_obj(2),x_monte,sig3_ls(:,2),'--')
set(gca,'fontsize',12)
axis([0 5000 -1 1])
ylabel('x2')
xlabel('Trial Number')
```