Lecture 2

Optimal Estimation Theory and Its Applications

Prof. Yan LIANG

liangyan@nwpu.edu.cn

School of Automation,

Northwestern Polytechnical University

Autumn,2019

Outline

Least Square Estimation

Least Squares Estimation

Least Weighted Squares Estimation

Recursive Least Squares Estimation

Iterated Least Squares Estimation

Least Square Estimation

Theory attracts practice as the magnet attracts iron.

Since all our measurements and observations are nothing more than approximations to the truth, the same must be true of all calculations resting upon them, and the highest aim of all computations made concerning concrete phenomenon must be to approximate, as nearly as practicable, to the truth.

Least Square Estimation

Here X is a n-dimensional vector to be estimated, the i-th measurement z_i is a m-dimensional vector and the linear function of X, v_i is the measurement error, and k is the measurement times.

$z_i = h_i X + v_i \qquad i = 1, 2, \cdots, k$

$$Z = HX + V \qquad Z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{bmatrix}_{km} \qquad H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \end{bmatrix}_{km \times n} \qquad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix}_{km}$$

Least Square Estimation

Consider $Z = HX + V$, our aim is to construct a function of the measurement, $\hat{X}$, as the best estimate of X so that the sum of squared error between Z and $H\hat{X}$ is minimum.

$$\hat{X} = \arg\left\{\min_X \{J(X)\}\right\}$$

with

$$J(X) = (Z - \hat{Z})^T (Z - \hat{Z}) = (Z - H\hat{X})^T (Z - H\hat{X})$$

Least Square Estimation

Application examples

- Identification of aircraft dynamic and static aerodynamic coefficients
- Orbit and attitude determination
- Position determination using triangulation

Least Square Estimation

Consider sensor measurement $Z =$col$\{$temperature z_1, flow rate $z_2\}$, we have the object function of LS is

$$J(X) = (Z - \hat{Z})^T (Z - \hat{Z}) = (\left[\begin{array}{c} z_1 \\ z_2 \end{array}\right] - \left[\begin{array}{c} h_1 \\ h_2 \end{array}\right] \hat{X})^T (\left[\begin{array}{c} z_1 \\ z_2 \end{array}\right] - \left[\begin{array}{c} h_1 \\ h_2 \end{array}\right] \hat{X})$$

$$= \left[\begin{array}{cc} z_1 - h_1 \hat{X} & z_2 - h_2 \hat{X} \end{array}\right] \left[\begin{array}{c} z_1 - h_1 \hat{X} \\ z_2 - h_2 \hat{X} \end{array}\right]^T$$

$$= (z_1 - h_1 \hat{X})^2 + (z_2 - h_2 \hat{X})^2$$

Least Square Estimation

$$\frac{\partial J}{\partial X}|_{X=\hat{X}} = -2H^T(Z - H\hat{X}) = 0$$

$$\Rightarrow \hat{X}_{LS}(Z) = (H^T H)^{-1} H^T Z$$

Does the LS always exist? the sufficient and necessary condition of the existence is $H^T H$ is invertible (positive definite, full-rank).

hint: Sensor deployment is constrained or even optimized according to the estimation requirement.

Least Square Estimation

Example 1 For a m-dimensional unknown vector X,its n-dimensional measurement $Z = HX + V$,where $m > n$, and V is the measurement error. Please judge the applicability of LS.

Definitely, H is a $n \times m$ matrix, $H^T H$ is a $m \times m$ matrix. The full-rank condition of $H^T H$ is rank$(H^T H) = m$, However, by the fact that

$$\operatorname{rank}\left(H^T H\right) \leqslant \operatorname{rank}(H) \leqslant \min(m, n) = n < m$$

we find that $H^T H$ is rank deficiency. The LS is not applicable.

Least Square Estimation

Example 2

Consider that the sensor obtain a measurement z_1 without satisfying the condition of the existence of the LS. If the measurement matrix is constant, i.e., $h_i = h \quad \forall i$, whether or not to obtain the condition of the existence of the LS through adding observation times?

Least Square Estimation

Clearly, $rank\{h_1^T h_1\} < n$

for k observations, we have the measurement matrix

$H = \text{col}\{h_1, h_2, \cdots, h_k\}$

$$H^T H = \sum_{i=1}^{k} h_i^T h_i = k h^T h$$

So

$$rank\{H^T H\} = rank\{h^T h\} < n$$

Conclusion: Without changing the measurement matrix, it is impossible to obtain the existence condition of the LS just by adding observation times.

Least Square Estimation

Example 3

Consider the case that the sensor measurement is nonlinear

$$z_i = 5x_1^2 + 6x_1x_2 + 3x_3^5/(x_2^2 + 1) + v_i$$

where $X = \text{col}\{x_1, x_2, x_3\}$ is the vector to be estimated.

Try supplying the LS based estimation scheme .

Least Square Estimation

Let $y_1 = x_1^2 \quad y_2 = x_1 x_2 \quad y_3 = x_3^5/(x_2^2+1)$

then we have $z_i = 5y_1 + 6y_2 + 3y_3 + v_i$

Using the LS estimation scheme, we have $\hat{y_1}$, $\hat{y_2}$ and $\hat{y_3}$

Estiblish the following nonlinear transformation

$$\begin{cases} x_1^2 = \hat{y}_1 \\ x_1 x_2 = \hat{y}_2 \\ x_3^5/(x_2^2+1) = \hat{y}_3 \end{cases}$$

we can obtain $\hat{x_1}$, $\hat{x_2}$ and $\hat{x_3}$

Weighted Least Squares Estimation

To $Z = HX + V$, construct the function of the measurement $\hat{X}$ so that the square of the error of fitting measurement is minimum, i.e.,

$$\hat{X} = \min_{X}\{J_W(X)\}$$

with

$$J_W(X) = (Z - H\hat{X})^T W (Z - H\hat{X})$$

where W is a positive-definite weight matrix.

Weighted Least Squares Estimation

Consider the measurement $Z =$[temperature z_1 flow rate z_2], the object function of WLS is

$$J(X)$$

$$= (Z - \hat{Z})^T W (Z - \hat{Z}) = (\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \hat{X})^T W (\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \hat{X})$$

$$= \begin{bmatrix} z_1 - h_1\hat{X} & z_2 - h_2\hat{X} \end{bmatrix} W \begin{bmatrix} z_1 - h_1\hat{X} \\ z_2 - h_2\hat{X} \end{bmatrix}^T$$

$$= (z_1 - h_1\hat{X})^2 W_{11} + (z_2 - h_2\hat{X})^2 W_{22} + 2(z_1 - h_1\hat{X})(z_2 - h_2\hat{X}) W_{12}$$

Remark: What is the advantage of adding the weight matrix?

Weighted Least Squares Estimation

The advantage of the weight matrix

1. introduce free parameters and thus increase the method freedom. Through optimizing the introduced parameters, it is expected to improve the method performance.
2. eliminate the dimension effect
3. measurement elements may be different in fitting weight. Thus weight choice can accord with application requirements

Weighted Least Squares Estimation

$$\frac{\partial J_W}{\partial X}|_{X=\hat{X}} = -2H^T W(Z - H\hat{X}) = 0$$

$$\Rightarrow \hat{X}_{WLS}(Z) = (H^T WH)^{-1} H^T WZ$$

What is the sufficient and necessary conditions of the existence of WLS?

$H^T WH$ is invertible (positive definite,full-rank).

Example 4

Can adding the weight matrix improve the applicability of LWS, compared with LS?

For a m-dimensional unknown vector X, if the LS is not applicable, i.e., rank$(H^T H) < m$. For any a weight matrix W, we have rank$(WH) \leq$ rank(H), and thus have

$$rank(H^T WH) \ \leqslant \ rank(H^T H) < m$$

Thus the LWS is not applicable.

Conclusion: adding the weight matrix can not improve the applicability of WLS, compared with LS.

Performance Analysis of LS

Theorem

If $E(V) = 0$, then $\hat{X}_{LS}$ and $\hat{X}_{WLS}$ are unbiased, i.e.,
$E(\hat{X}_{LS}) = E(\hat{X}_{WLS}) = E(X)$

Consideration:

1. Why consider the unbiasedness?
2. Is the unbiased estimate unique?
3. What is the requirement of the unbiasedness?

Performance Analysis of LS

Proof

$$\because \tilde{X}_{WLS} = X - \hat{X}_{WLS} = X - (H^T W H)^{-1} H^T W (HX + V)$$

$$= -\ (H^T W H)^{-1} H^T W V$$

$$E(\tilde{X}_{WLS}) = -(H^T W H)^{-1} H^T W E(V) = 0$$

$$\therefore E(\hat{X}_{WLS}) = E(X)$$

$$\hat{X}_{LS}(Z) = (H^T H)^{-1} H^T Z$$

$$Z = HX + V$$

$$\hat{X}_{WLS}(Z) = \left(H^T W H\right)^{-1} H^T W Z$$

Performance Analysis of LS

Theorem

If $R = E(VV^T)$, then the autocorrelation matrix of the WLS estimate error is

$$E(\tilde{X}_{WLS}\tilde{X}_{WLS}^T) = (H^T WH)^{-1} H^T WRWH(H^T WH)^{-1}$$

Proof

From $\tilde{X}_{WLS} = -(H^T WH)^{-1} H^T WV$ use the linearity of the expectation operator, we have

$$\begin{aligned} E(\tilde{X}_{WLS}\tilde{X}_{WLS}^T) &= (H^T WH)^{-1} H^T W \underline{E(V V^T)} W^T H (H^T W^T H)^{-1} \\ &= (H^T WH)^{-1} H^T WRWH(H^T WH)^{-1} \end{aligned}$$

Performance Analysis of LS

The mean square error of the WLS estimation is

$$E(\tilde{X}_{WLS}^T \tilde{X}_{WLS}) = \mathsf{trace}\{E(\tilde{X}_{WLS}\tilde{X}_{WLS}^T)\}$$

The autocorrelation matrix of the WLS estimate error is

$$\begin{aligned} E(\tilde{X}_{WLS}\tilde{X}_{WLS}^T) &= E((\tilde{X}_{WLS} - E\tilde{X}_{WLS}^T + E\tilde{X}_{WLS}^T)(\centerdot)^T) \\ &= E((\tilde{X}_{WLS} - E\tilde{X}_{WLS}^T)(\centerdot)^T) + (E\tilde{X}_{WLS})(\centerdot)^T) \end{aligned}$$

Performance Analysis of LS

Theorem

If $R = E(VV^T)$,then choosing $W = R^{-1}$, $E(\tilde{X}^T\tilde{X})$ is minimized.

or

$$E(\tilde{X}_{WLS}\tilde{X}_{WLS}^T) - E(\tilde{X}_{WLS}\tilde{X}_{WLS}^T)\Big|_{W=R^{-1}} \geqslant 0$$

Check whether the matrix is positive definite or positive semi-definite.

Performance Analysis of LS

Proof of Schwarz Inequality

$$\underline{\left[B - A^T (AA^T)^{-1} AB\right]^T} \left[B - A^T (AA^T)^{-1} AB\right] \geqslant 0$$

$$\Downarrow$$

$$\left[B^T - B^T A^T (AA^T)^{-1} A\right]$$

Expand the above inequality, we have

$$B^T B - 2B^T A^T (AA^T)^{-1} AB + B^T A^T (AA^T)^{-1} \underline{(AA^T)(AA^T)^{-1}} AB$$

$$= B^T B - (AB)^T (AA^T)^{-1} AB \geqslant 0$$

$$\boxed{B^T B \geqslant (AB)^T (AA^T)^{-1} AB} \quad --- \text{Schwarz Inequality}$$

Performance Analysis of LS

Proof

Represent R as $S^T S$. (Why I can do it?)

we have $E(\tilde{X}\tilde{X}^T) = \underline{(H^T WH)^{-1} H^T W S^T}\,\underline{\underline{SWH(H^T WH)^{-1}}}$

Let $A = H^T S^{-1}, B = SWH(H^T WH)^{-1}$

then $AB = H^T S^{-1} SWH(H^T WH)^{-1} = I$

$$E(\tilde{X}\tilde{X}^T) = B^T B \geqslant (AB)^T (AA^T)^{-1} AB = (AA^T)^{-1}$$
$$= \left[H^T \underline{S^{-1}(S^{-T})} H\right]^{-1} = (H^T R^{-1} H)^{-1}$$

Thus $E(\tilde{X}\tilde{X}^T) \geqslant (H^T R^{-1} H)^{-1}$

Performance Analysis of LS

On the other hand, if choosing $W = R^{-1}$, we have

$$E(\tilde{X}\tilde{X}^T) = (H^T WH)^{-1} H^T WRWH(H^T WH)^{-1}$$
$$= (H^T R^{-1} H)^{-1} H^T R^{-1} R R^{-1} H(H^T R^{-1} H)^{-1}$$
$$= (H^T R^{-1} H)^{-1}$$

In the end, we prove that if choosing $W = R^{-1}$,
the resultant covariance matrix is the best.

$\hat{X}_{WLS}|_{W=R^{-1}} = (H^T R^{-1} H)^{-1} H^T R^{-1} Z \qquad ---Markov$ estimate

Recursive Least Square Estimation

All above methods are based on centralized process-batch process . Data saving and computation complexity increases with the measurement dimensions.

Recursive estimation: process measurements one-by-one, implement the estimate updating, and thus decrease the computation and saving burden.

Recursive estimate is not better than the batch-process estimate.

To LS and WLS, the accuracy of their recursive estimates are the same as the batch-process estimate.

Recursive Least Square Estimation

Recursive implementation of WLS

Based on the former k measurements, the WLS is

$$\hat{X}_k = (H_k^T \overline{W}_k H_k)^{-1} H_k^T \overline{W}_k Z_k$$
$$where\ P_k = (H_k^T \overline{W}_k H_k)^{-1}, so\ P_k^{-1} = H_k^T \overline{W}_k H_k$$
$$\hat{X}_k = P_k H_k^T \overline{W}_k Z_k$$

Here k is the recursive number

Recursive Least Square Estimation

Recursive implementation of WLS

$\hat{X}_{k+1} = P_{k+1} H_{k+1}^T \overline{W}_{k+1} Z_{k+1}$ (using the former $k+1$ measurements)

where

$$H_{k+1} = \begin{bmatrix} H_k \\ h_{k+1} \end{bmatrix} \qquad \overline{W}_{k+1} = \begin{bmatrix} \overline{W}_k & 0 \\ 0 & w_{k+1} \end{bmatrix} \qquad Z_{k+1} = \begin{bmatrix} Z_k \\ z_{k+1} \end{bmatrix}$$

h_{k+1}, ω_{k+1} and z_{k+1} are measurement mareix, weight matrix, and measurement vector related the $k+1$ measurement.

Recursive Least Square Estimation

Recursive implementation of WLS

$$\begin{cases} \hat{X}_{k+1} = P_{k+1} H_{k+1}^T \overline{W}_{k+1} Z_{k+1} \\ \qquad = P_{k+1} (H_k^T \overline{W}_k Z_k + h_{k+1}^T w_{k+1} z_{k+1}) \\ P_{k+1} = (H_{k+1}^T \overline{W}_{k+1} \, H_{k+1})^{-1} = (\underline{H_k^T \overline{W}_k \, H_k} + h_{k+1}^T w_{k+1} \, h_{k+1})^{-1} \end{cases}$$

or

$$P_{k+1}^{-1} = P_k^{-1} + h_{k+1}^T w_{k+1} h_{k+1}$$

Recursive Least Square Estimation

Recursive implementation of WLS

$$\begin{aligned}
\hat{X}_{k+1} &= P_{k+1}(H_k^T \overline{W}_k Z_k + h_{k+1}^T w_{k+1} z_{k+1}) \\
&= P_{k+1}(P_k^{-1} P_k H_k^T \overline{W}_k Z_k + h_{k+1}^T w_{k+1} z_{k+1}) \\
&= P_{k+1}\left[(P_{k+1}^{-1} - h_{k+1}^T w_{k+1} h_{k+1})\hat{X}_k + h_{k+1}^T w_{k+1} z_{k+1}\right] \\
&= \hat{X}_k + P_{k+1} h_{k+1}^T w_{k+1}(z_{k+1} - h_{k+1}\hat{X}_k)
\end{aligned}$$

Let $K_{k+1} = P_{k+1} h_{k+1}^T w_{k+1}$

we have $\hat{X}_{k+1} = \hat{X}_k + K_{k+1}(z_{k+1} - h_{k+1}\hat{X}_k)$

Recursive Least Square Estimation

Recursive implementation of WLS

$$P_{k+1}^{-1} = P_k^{-1} + h_{k+1}^T w_{k+1} h_{k+1}$$

Based on the following matrix-inversion formula

$$(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} = A_{11}^{-1} + A_{11}^{-1}A_{12}(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}A_{21}A_{11}^{-1}$$

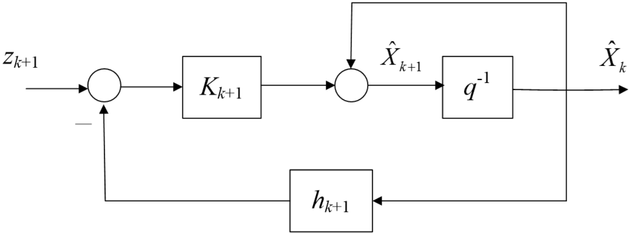
$$P_{k+1} = \Big[P_k^{-1} - (-h_{k+1}^{-1})w_{k+1}h_{k+1}\Big]^{-1}$$

$$= P_k + P_k(-h_{k+1}^{-1})\Big[w_{k+1}^{-1} - h_{k+1}P_k(-h_{k+1}^T)\Big]^{-1}h_{k+1}P_k$$

Thus $P_{k+1} = P_k - P_k h_{k+1}^T\Big[h_{k+1}P_k h_{k+1}^T + w_{k+1}^{-1}\Big]^{-1}h_{k+1}P_k$

The other expression of K_{k+1}: $K_{k+1} = P_k h_{k+1}^T(h_{k+1}P_k h_{k+1}^T + w_{k+1}^{-1})^{-1}$

Recursive Least Square Estimation



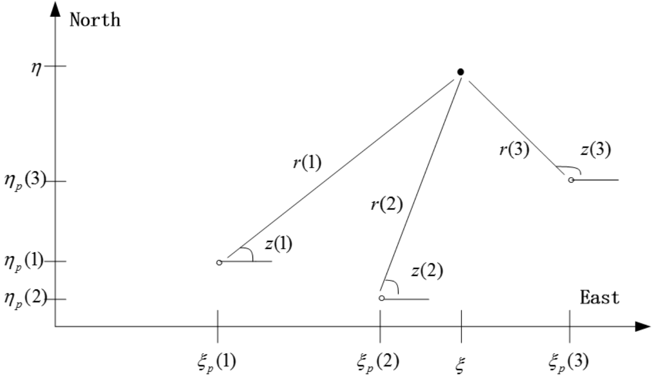
Recursive implementation of WLS

Without a priori information, the initial value

$$\hat{X}_0 = 0, \quad P_0 = cI, \quad c \to \infty$$

Recursive Least Square Estimation

LS for nonlinear systems



Multi-station angle-only target localization

Iterative Least Square Estimation

Problem Formulation

Target location $x = \text{col}\{\xi, \eta\}$

Sensor location $x_p(i) = \text{col}\{\xi_p(i), \eta_p(i)\} \quad i = 1, 2, 3$

Senor measurement $z(i) = \tan^{-1}(\frac{\eta - \eta_p(i)}{\xi - \xi_p(i)}) + v(i) = h(x, x_p(i)) + v(i)$

measurement error $v(i)$ is zero-mean Gaussian white noises with covariance σ^2

Object

Given sensor location and sensor measurements, obtain the target location so that the error of fitting is minimum.

Iterative Least Square Estimation

Augment the measurements

$$z = \begin{bmatrix} z(1) \\ z(2) \\ z(3) \end{bmatrix} = h(x, x_p) + \upsilon$$

with $$h(x, x_p) = \begin{bmatrix} h(x, x_p(1)) \\ h(x, x_p(2)) \\ h(x, x_p(3)) \end{bmatrix} \quad \upsilon = \begin{bmatrix} \upsilon(1) \\ \upsilon(2) \\ \upsilon(3) \end{bmatrix}$$

$$\upsilon \sim N(0, R = diag\{\sigma^2, \sigma^2, \sigma^2\})$$

Iterative Least Square Estimation

$$\hat{x}_{j+1}^{ILS} = \hat{x}_j^{ILS} + (J_j^T R^{-1} J_j)^{-1} J_j^T R^{-1}[z - h(\hat{x}_j^{ILS}, x_p)]$$

$\hat{x}_j^{ILS}$ The estimation after the j-th iteration is

$$J_j = \left.\frac{\partial h(x, x_p)}{\partial x}\right|_{x=\hat{x}_j^{ILS}} = \left.\begin{bmatrix} \frac{\partial h(x,x_p(1))}{\partial \xi} & \frac{\partial h(x,x_p(1))}{\partial \eta} \\ \frac{\partial h(x,x_p(2))}{\partial \xi} & \frac{\partial h(x,x_p(2))}{\partial \eta} \\ \frac{\partial h(x,x_p(3))}{\partial \xi} & \frac{\partial h(x,x_p(3))}{\partial \eta} \end{bmatrix}\right|_{x=\hat{x}_j^{ILS}}$$

Jacobian matrix after the j-th iteration.

Iterative Least Square Estimation

Condition of Iteration stopping

1. The iteration number reaches the maximum
2. The distance between the neighbored estimates is lower than the threshold.

Initial condition of iteration

From any two measurements, utilize triangulate localization, and thus obtain the initial iterative value

The mean square error of the j-th iteration is $tr\{(J_j^T R^{-1} J_j)^{-1}\}$

The unbiasedness of iterative estimate

May be unbiased

Model-matching Test

In the LS, measurement matrix H is known. H represent the relationship between sensor and environment. If H is unknown, we can consider

- for a time function, determine the parameters of polynomial approximation
- for a non-analytic function, based on input and output data, implement function approximation via base function

1. If parameter number is chosen too small, then the error of fitting will be very large.(under-fitting)
2. If parameter number is chosen too large, the estimate in some parameters is error-dominant.(over-fitting)

It is needed to measure the goodness of fit via hypothesis test.

Model-matching Test

The goodness of fit is

$$J^* = (Z - H\hat{X})^T P_J^{-1} (Z - H\hat{X})$$

where

$$\begin{aligned}
& P_J \triangleq E\{(Z - H\hat{X})(Z - H\hat{X})^T\} \\
& = E\{(H\tilde{X} + V)(\bullet)^T\} \\
& = (I - (H^T R^{-1} H)^{-1} H^T R^{-1}) E\{V V^T\} (\bullet)^T \\
& = (I - (H^T R^{-1} H)^{-1} H^T R^{-1}) R (\bullet)^T \\
& = R - H(H^T R^{-1} H)^{-1} H^T < \ \ R
\end{aligned}$$

To $Z_k = HX + V_k$, if use the current measurement as the prediction of the measurement at the next time $\hat{z}_{k+1} = z_k$ the estimate variance 2R. If the noise V is zero-mean Gaussian, then the error of fitting obeys x^2 distribution with the freedom being $kn_z - n_x$

Model-matching Test

Under-fitting test

If the hypothesis is statistically significant, i.e.,

$$J^* > c = \chi^2_{kn_z - n_x}(1 - \alpha)$$

then it means under-fitting. Through adding the number of model parameters until the following inequality is satisfied

$$J^* \leqslant c = \chi^2_{kn_z - n_x}(1 - \alpha)$$

Model-matching Test

over-fitting test

Consider the measurement noise is the zero-mean Gaussian noise with known variance, its i-th element estimate is

$$\hat{x}_i(k) \sim N[x_i, P_{ii}(k)]$$

If the parameter is over-fitted, then $x_i = 0$

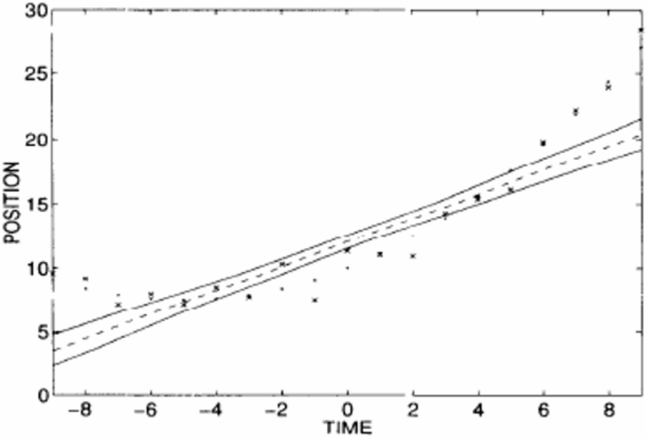
$$H_0 : x_i = 0 \quad H_1 : x_i \neq 0 \;\; \text{satisfy} \;\; P\{\text{accept } H_1 | H_0 \text{ is ture}\} = \alpha$$

If and only if

$$\frac{|\hat{x}_i(k)|}{[P_{ii}(k)]^{1/2}} > c' = \zeta(1 - \frac{\alpha}{2})$$

we accept H_1. It means that this parameter estimate is nonzero, not over-fitting.

Model-matching Test



$\cdots$ true trajectory;

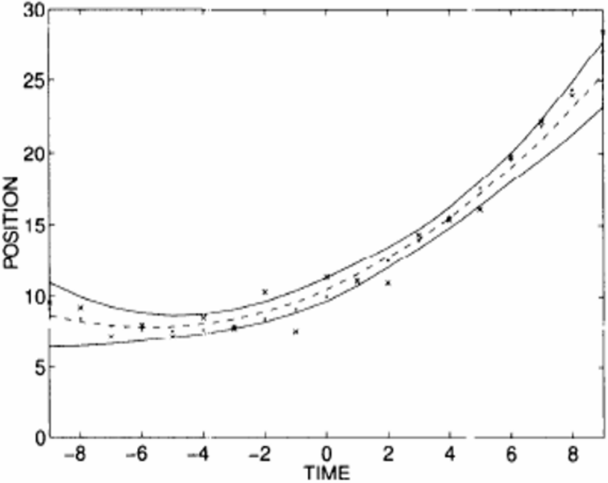
$\times$ measurements;

$--$estimated trajectory;

— 95% confidence region

(a) $n_x = 2$

Model-matching Test



(b) $n_x = 3$

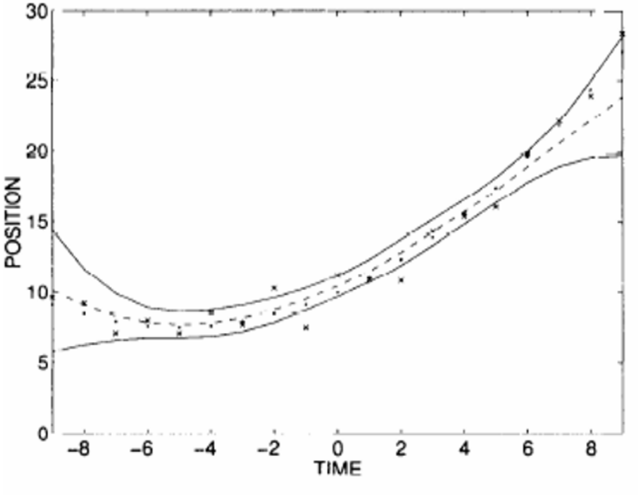
$\cdots$ true trajectory;

$\times$ measurements;

$--$estimated trajectory;

— 95% confidence region

Model-matching Test



⋯ true trajectory;

× measurements;

−−estimated trajectory;

(c) $n_x = 4$

— 95% confidence region

Homework

Homework 1

The movement parameter of a aircraft evolves with time according to $y_t = at^2 + bt + c$. Sample in each 1s period, and thus obtain 20 data pair $(1, y_1), \cdots, (20, y_{20})$. Obtain the LS estimate of a, b and c.

Data of y: 2.9828 4.525 6.1155 7.234 8.4329 9.1259 10.1800 10.8600 10.9300 11.1410 10.6090 10.4800 10.3830 9.5808 8.4611 7.4678 6.3942 4.1592 3.0029 0.5503

Homework 1(cont.)

Requirement

- prepare matlab programs for batch and recursive methods
- submit the technical report including the estimates of a, b and c; plot the t-vs-estimate curve of a,b and c, analyze the relationship between data length and estimate accuracy; plot measurements and their estimates based on the parameter estimates.

Homework

Homework 2

Consider the actual but unknown vector $X = [3, 2, 0]^T$, the measurement matrix H is a 3×3 unite matrix, measurement noise is zero-mean Gaussian with covariance diag$\{$1,9,2$\}$.

- bring out 30 independent measurements, and then plot the RLS estimate of each element in X vs observation times. Also plot state and measurements.
- plot sampled error variance of each element in X based on 100 simulations, and compare with its theoretical variance

Remark:require simulation explanation and analysis

Appendix—-Matrix Inversion

$$1. \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} A_{11}^{-1} & -A_{11}^{-1} A_{12} A_{22}^{-1} \\ 0 & A_{22}^{-1} \end{bmatrix}$$

$$2. \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} I & A_{12} A_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{11} - A_{12} A_{22}^{-1} A_{21} & 0 \\ A_{21} & A_{22} \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ A_{21} A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} - A_{21} A_{11}^{-1} A_{12} \end{bmatrix}$$

Appendix—-Matrix Inversion

Considering that J is a scale variable, X and y are vectors, and W and A are matrices of proper dimensions, we have

$$\frac{\partial J}{\partial x} = \begin{bmatrix} \frac{\partial J}{\partial x_1} \\ \vdots \\ \frac{\partial J}{\partial x_n} \end{bmatrix} \qquad \frac{\partial J}{\partial x} = \left[\frac{\partial y}{\partial x^T}\right]^T \frac{\partial J}{\partial y} \neq \frac{\partial J}{\partial y}\frac{\partial y}{\partial x}$$

For $J = x^T Wy$, we have

$$\begin{cases} \frac{\partial J}{\partial x} = Wy \\ \frac{\partial J}{\partial y} = \left(x^T W\right)^T = W^T x \end{cases}$$

Appendix—-Matrix Inversion

for $J = x^T W x$, we have

$$\frac{\partial J}{\partial x} = Wx + W^T x = (W + W^T)x = 2Wx$$

$$(Ax)_i = \sum_i a_{ij} x_j \qquad \frac{\partial (Ax)_i}{\partial x_j} = \frac{\partial \sum_i a_{ij} x_j}{\partial x_j} = a_{ij}$$

$$\Delta J(x, y) = \frac{\partial J}{\partial x}\Delta x + \frac{\partial J}{\partial y}\Delta y \qquad \frac{\partial Ax}{\partial x^T} = A$$