ECEn 671: Mathematics of Signals and Systems

Randal W. Beard

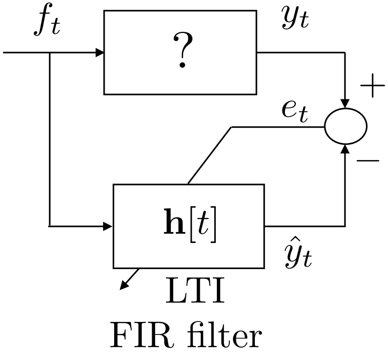
Brigham Young University

September 1, 2023

Section 1

Recursive Least Squares Filtering

Least Squares Filtering Problem



Problem Statement: Given the input data f_t and y_t, find the FIR filter coefficients $\mathbf{h}[t]$ that minimize the running least squared error e_t.

Least Squares Filtering Problem

Definition (Least Squares Filtering Problem)

Given the filter

$$\hat{y}_t = \sum_{i=1}^{m} h_i f_{t-i}$$

where the inputs f_t are known and we measure the actual outputs y_t, find the coefficients h_i such that the mean squared error

$$E = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

is minimized.

Batch Least Squares Filtering

If we assume $f_t = 0, t \leq 0$ we get

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} f_1 & 0 & \cdots & \cdots & 0 \\ f_2 & f_1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & \\ f_m & f_{m-1} & \cdots & \cdots & f_1 \\ f_{m+1} & f_m & f_{m-1} & \cdots & f_2 \\ \vdots & & & \ddots & \\ f_N & f_{N-1} & \cdots & \cdots & f_{N-m+1} \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_m \end{pmatrix}$$

Batch Least Squares Filtering, cont.

Define

$$\mathbf{q}_i = \begin{pmatrix} f_i & f_{i-1} & \dots f_{i-m+1} \end{pmatrix}^H$$

$$\mathbf{y}_N = \begin{pmatrix} \bar{y}_1 & \bar{y}_2 & \dots \bar{y}_N \end{pmatrix}^H$$

$$\mathbf{h}[N] = \begin{pmatrix} \bar{h}_1[N] & \bar{h}_2[N] & \dots \bar{h}_m[N] \end{pmatrix}^H$$

$$A_N = \begin{pmatrix} \mathbf{q}_1^H \\ \vdots \\ \mathbf{q}_m^H \end{pmatrix},$$

then the least squares problem reduces to

$$\mathbf{e}_N = \mathbf{y}_N - \underbrace{A_N \mathbf{h}[N]}_{\hat{\mathbf{y}}_N}$$

where $\mathbf{e}_N$ is the error to be minimized. From the projection theorem, $\|\mathbf{e}\|_2$ is minimized when

$$\underset{m\times 1}{\mathbf{h}[N]} = (\underset{m\times N}{A_N^H}\underset{N\times m}{A_N})^{-1}\underset{m\times N}{A_N^H}\underset{N\times 1}{\mathbf{y}_N}.$$

Batch Least Squares Filtering

- Note that the size of y_N and A_N grow linearly with time N.
- Therefore, each time step requires more computation than the last step. This is obviously problematic as $N \to \infty$.
- For some N, batch least squares is no longer a real-time algorithm.
- Note that at time $N+1$ the data include new samples, but includes all of the data available at time N.

??? Is is possible to design an algorithm with fixed computational cost at each time step, that produces the same least squares solution?

Recursive Least Squares Filtering

Define

$$\begin{aligned}
\mathbf{q}_t &= \begin{pmatrix} f_i & f_{i-1} & \dots f_{i-m+1} \end{pmatrix}^H \\
\mathbf{y}_t &= \begin{pmatrix} \bar{y}_1 & \bar{y}_2 & \dots \bar{y}_t \end{pmatrix}^H \\
\mathbf{h}[t] &= \begin{pmatrix} \bar{h}_1[t] & \bar{h}_2[t] & \dots \bar{h}_m[t] \end{pmatrix}^H \\
A_t &= \begin{pmatrix} \mathbf{q}_1^H \\ \vdots \\ \mathbf{q}_t^H \end{pmatrix}.
\end{aligned}$$

Then at time t we have $\mathbf{e}_t = \mathbf{y}_t - A_t\mathbf{h}[t]$. From the projection theorem, the error is minimized when

$$\mathbf{h}[t] = (A_t^H A_t)^{-1} A_t^H \mathbf{y}_t.$$

Recursive Least Squares Filtering, cont.

Let

$$R_{t-1} \triangleq A_{t-1}^H A_{t-1} = \begin{pmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_{t-1} \end{pmatrix} \begin{pmatrix} \mathbf{q}_1^H \\ \vdots \\ \mathbf{q}_{t-1}^H \end{pmatrix}$$

$$= \sum_{i=1}^{t-1} \mathbf{q}_i \mathbf{q}_i^H$$

be the associated Grammian when there are $t-1$ samples.
Suppose that we receive new data q_t and y_t at time t.
Then

$$\begin{aligned} R_t &= \sum_{i=1}^{t} \mathbf{q}_i \mathbf{q}_i^H \\ &= \sum_{i=1}^{t-1} \mathbf{q}_i \mathbf{q}_i^H + \mathbf{q}_t \mathbf{q}_t^H \\ &= R_{t-1} + \mathbf{q}_t \mathbf{q}_t^H. \end{aligned}$$

Recursive Least Squares Filtering, cont.

In the solution $\mathbf{h}_t = (A_t^H A_t)^{-1} A_t^H \mathbf{y}_t$, we need $R_t^{-1} \triangleq (A_t^H A_t)^{-1}$. Note that

$$R_t^{-1} = (\underbrace{R_{t-1}}_{A} + \underbrace{q_t}_{X} \underbrace{}_{R=1} \underbrace{q_t^H}_{Y})^{-1}$$

and recall the matrix inversion lemma:

$$(A + XRY)^{-1} = A^{-1} - A^{-1}X(R^{-1} + YA^{-1}X)^{-1}YA^{-1}$$

Therefore

$$R_t^{-1} = R_{t-1}^{-1} - R_{t-1}^{-1}\mathbf{q}_t(1 + \mathbf{q}_t^H R_{t-1}^{-1}\mathbf{q}_t)^{-1}\mathbf{q}_t^H R_{t-1}^{-1}.$$

Recursive Least Squares Filtering, cont.

Defining $P_t = R_t^{-1}$ gives

$$P_t = P_{t-1} - \frac{P_{t-1}\mathbf{q}_t\mathbf{q}_t^H P_{t-1}}{1 + \mathbf{q}_t^H P_{t-1}\mathbf{q}_t}.$$

Define the (Kalman) gain as

$$\mathbf{k}_t = \frac{P_{t-1}\mathbf{q}_t}{1 + \mathbf{q}_t^H P_{t-1}\mathbf{q}_t}$$

Then

$$P_t = P_{t-1} - \mathbf{k}_t\mathbf{q}_t^H P_{t-1}.$$

Note that we have found a fixed computational scheme to update

$$P_t = (A_t^H A_t)^{-1}$$

using old data P_{t-1} and new data $\mathbf{q}_t$.

Recursive Least Squares Filtering, cont.

In the solution $\mathbf{h}[t] = (A_t^H A_t)^{-1} A_t^H \mathbf{y}_t$, we have found a clever way to update $P_t = (A_t^H A_t)^{-1}$ recursively. Define

$$\mathbf{z}_t \triangleq A_t^H \mathbf{y}_t.$$

We need a recursive update for $\mathbf{z}_t$.
Toward that end note that

$$\begin{aligned}
\mathbf{z}_t &= A_t^H \mathbf{y}_t \\
&= \sum_{i=1}^{t} \mathbf{q}_i y_i \\
&= \sum_{i=1}^{t-1} \mathbf{q}_i y_i + \mathbf{q}_t y_t \\
&= \mathbf{z}_{t-1} + \mathbf{q}_t y_t
\end{aligned}$$

Recursive Least Squares Filtering, cont.

Therefore

$$
\begin{aligned}
\mathbf{h}_t &= (A_t^H A_t)^{-1} A_t^H \mathbf{y}_t \\
&= P_t \mathbf{z}_t \\
&= (P_{t-1} - \mathbf{k}_t \mathbf{q}_t^H P_{t-1})(\mathbf{z}_{t-1} + \mathbf{q}_t y_t) \\
&= P_{t-1}\mathbf{z}_{t-1} - \mathbf{k}_t \mathbf{q}_t^H P_{t-1}\mathbf{z}_{t-1} + P_{t-1}\mathbf{q}_t y_t - \mathbf{k}_t \mathbf{q}_t^H P_{t-1}\mathbf{q}_t y_t \\
&= \mathbf{h}_{t-1} - \mathbf{k}_t \mathbf{q}_t^H \mathbf{h}_{t-1} + \underbrace{\left(P_{t-1} - \mathbf{k}_t \mathbf{q}_t^H P_{t-1}\right)}_{P_t} \mathbf{q}_t y_t \\
&= \mathbf{h}_{t-1} + \mathbf{k}_t (y_t - \mathbf{q}_t^H \mathbf{h}_{t-1}) \\
\implies \mathbf{h}_t &= \mathbf{h}_{t-1} + \mathbf{k}_t (y_t - \hat{y}),
\end{aligned}
$$

where we have used the fact that $P_t q_t = \mathbf{k}_t$.
Note that $\hat{y}_t = \mathbf{q}_t^H \mathbf{h}_{t-1}$ is the predicted output, and $e_t = y_t - \hat{y}_t$ is the quantity that is being minimized.

Summary: Recursive Least Squares Filtering

At time $t = 0$ initialize algorithm with

$$\begin{aligned} P_0 &= \alpha I, \text{ where } \alpha > 0 \text{ is a large number} \\ \mathbf{h}_0 &= 0. \end{aligned}$$

At time t, get y_t, f_t, and compute $\mathbf{q}_t$ from f_t. Update the least squares estimate using

$$\begin{aligned} \mathbf{k}_t &= \frac{P_{t-1}\mathbf{q}_t}{1 + \mathbf{q}_t^H P_{t-1}\mathbf{q}_t} \\ P_t &= P_{t-1} - \mathbf{k}_t \mathbf{q}_t^H P_{t-1} \\ \mathbf{h}_t &= \mathbf{h}_{t-1} + \mathbf{k}_t(y_t - \mathbf{q}_t^H \mathbf{h}_{t-1}). \end{aligned}$$

This is equivalent to a discrete time Kalman filter with stationary dynamics.