Lecture 3: Real-Time Parameter Estimation

- Least Squares and Recursive Computations
- Estimating Parameters in Dynamical Systems
- Experimental Conditions
- Examples

Preliminary Comments

In adaptive controllers the observations (data) are obtained **sequentially** in real-time.

Preliminary Comments

In adaptive controllers the observations (data) are obtained **sequentially** in real-time.

$$\Downarrow$$

This property puts **time constraint** on computation of estimates.

Preliminary Comments

In adaptive controllers the observations (data) are obtained **sequentially** in real-time.

$$\Downarrow$$

This property puts **time constraint** on computation of estimates.

To accommodate the constraint, one should

- Simplify algorithms, if possible :)
- Reorganize the computation in such a way that
  - updates of estimates are only done if new observations are obtained,
  - estimates from previous time steps are used as an input for the algorithm of computing updates.

Preliminary Comments

In adaptive controllers the observations (data) are obtained **sequentially** in real-time.

$\Downarrow$

This property puts **time constraint** on computation of estimates.

To accommodate the constraint, one should

- Simplify algorithms, if possible :)
- Reorganize the computation in such a way that
  - updates of estimates are only done if new observations are obtained,
  - estimates from previous time steps are used as an input for the algorithm of computing updates.

The last thoughts make an idea of **recursive computations** very attractive!

What We Expect from Recursive Computations?

Ideally, the procedure should be as follows:

What We Expect from Recursive Computations?

Ideally, the procedure should be as follows:

Time $\mathbf{0}$: An initial guess for the parameters $\hat{\theta}(0)$ is given.

What We Expect from Recursive Computations?

Ideally, the procedure should be as follows:

Time $\mathbf{0}$: An initial guess for the parameters $\hat{\theta}(0)$ is given.

$$\Downarrow$$

$$\dots$$

What We Expect from Recursive Computations?

Ideally, the procedure should be as follows:

Time $\mathbf{0}$: An initial guess for the parameters $\hat{\theta}(0)$ is given.

$$\Downarrow$$

$$\dots$$

Time t: $\hat{\theta}(t-1)$ is the estimate from the previous step.
New data $[y(t), \phi(t)]$ are obtained.

What We Expect from Recursive Computations?

Ideally, the procedure should be as follows:

Time $\mathbf{0}$: An initial guess for the parameters $\hat{\theta}(0)$ is given.

$$\Downarrow$$

$$\dots$$

Time t: $\hat{\theta}(t-1)$ is the estimate from the previous step.
New data $[y(t), \phi(t)]$ are obtained.

Compute an updated estimate:

$$\hat{\theta}(t) = \mathcal{F}\left(\hat{\theta}(t-1),\, y(t),\, \phi(t)\right).$$

What We Expect from Recursive Computations?

Ideally, the procedure should be as follows:

Time $\mathbf{0}$: An initial guess for the parameters $\hat{\theta}(0)$ is given.

$$\Downarrow$$

$$\ldots$$

Time t: $\hat{\theta}(t-1)$ is the estimate from the previous step.
New data $[y(t), \phi(t)]$ are obtained.

Compute an updated estimate:

$$\hat{\theta}(t) = \mathcal{F}\left(\hat{\theta}(t-1),\; y(t),\; \phi(t)\right).$$

$$\Downarrow$$

Time $(t+1)$: ...

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\hat{\theta}(t) \;=\; \left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \left(\sum_{i=1}^{t}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t}\phi(i)y(i)\right)$$

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\begin{aligned}
\hat{\theta}(t) &= \left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \left(\sum_{i=1}^{t}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t}\phi(i)y(i)\right) \\
&= \left(\sum_{i=1}^{t-1}\phi(i)\phi(i)^T+\phi(t)\phi(t)^T\right)^{-1}\left(\sum_{i=1}^{t-1}\phi(i)y(i)+\phi(t)y(t)\right)
\end{aligned}$$

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\hat{\theta}(t) = \left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \left(\sum_{i=1}^{t}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t}\phi(i)y(i)\right)$$

$$= \boxed{\left(\sum_{i=1}^{t-1}\phi(i)\phi(i)^T+\phi(t)\phi(t)^T\right)^{-1}}\left(\sum_{i=1}^{t-1}\phi(i)y(i)+\phi(t)y(t)\right)$$

The LS estimate at time $(t-1)$ was computed as

$$\hat{\theta}(t-1) = \left(\sum_{i=1}^{t-1}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)$$

$$= P(t-1)\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)$$

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\hat{\theta}(t) = \left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \left(\sum_{i=1}^{t}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t}\phi(i)y(i)\right)$$

$$= \left(P(t-1)^{-1} + \phi(t)\phi(t)^T\right)^{-1}\left(\sum_{i=1}^{t-1}\phi(i)y(i) + \phi(t)y(t)\right)$$

The LS estimate at time $(t-1)$ was computed as

$$\hat{\theta}(t-1) = \left(\sum_{i=1}^{t-1}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)$$

$$= P(t-1)\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)$$

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\hat{\theta}(t) = \left(\Phi^T \Phi\right)^{-1} \Phi^T Y = \left(\sum_{i=1}^{t} \phi(i)\phi(i)^T\right)^{-1} \left(\sum_{i=1}^{t} \phi(i)y(i)\right)$$

$$= \left(P(t-1)^{-1} + \phi(t)\phi(t)^T\right)^{-1} \left(P(t-1)^{-1}\hat{\theta}(t-1) + \phi(t)y(t)\right)$$

The LS estimate at time $(t-1)$ was computed as

$$\hat{\theta}(t-1) = \left(\sum_{i=1}^{t-1} \phi(i)\phi(i)^T\right)^{-1} \left(\sum_{i=1}^{t-1} \phi(i)y(i)\right)$$

$$= P(t-1)\left(\sum_{i=1}^{t-1} \phi(i)y(i)\right)$$

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\hat{\theta}(t) = \left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \left(\sum_{i=1}^{t}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t}\phi(i)y(i)\right)$$

$$= \left(P(t-1)^{-1} + \phi(t)\phi(t)^T\right)^{-1}\left(P(t-1)^{-1}\hat{\theta}(t-1) + \phi(t)y(t)\right)$$

The LS estimate at time $(t-1)$ was computed as

$$\hat{\theta}(t-1) = \left(\sum_{i=1}^{t-1}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)$$

$$= P(t-1)\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)$$

How to simplify computations? Especially for $P(t-1) \to P(t)$

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\hat{\theta}(t) = \left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \left(\sum_{i=1}^{t}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t}\phi(i)y(i)\right)$$

$$= \left(P(t-1)^{-1} + \phi(t)\phi(t)^T\right)^{-1}\left(P(t-1)^{-1}\hat{\theta}(t-1) + \phi(t)y(t)\right)$$

The LS estimate at time $(t-1)$ was computed as

$$\hat{\theta}(t-1) = \left(\sum_{i=1}^{t-1}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)$$

$$= P(t-1)\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)$$

$$P(t)^{-1} = P(t-1)^{-1} + \phi(t)\phi(t)^T$$

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\begin{aligned}\hat{\theta}(t) &= \left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \left(\sum_{i=1}^{t}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t}\phi(i)y(i)\right)\\ &= P(t)\left(P(t-1)^{-1}\,\hat{\theta}(t-1)+\phi(t)y(t)\right)\end{aligned}$$

The LS estimate at time $(t-1)$ was computed as

$$\begin{aligned}\hat{\theta}(t-1) &= \left(\sum_{i=1}^{t-1}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)\\ &= P(t-1)\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)\end{aligned}$$

$$P(t)^{-1} = P(t-1)^{-1} + \phi(t)\phi(t)^T$$

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\begin{aligned}\hat{\theta}(t) &= \left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \left(\sum_{i=1}^{t}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t}\phi(i)y(i)\right)\\ &= P(t)\left(\left\{P(t)^{-1} - \phi(t)\phi(t)^T\right\}\hat{\theta}(t-1) + \phi(t)y(t)\right)\end{aligned}$$

The LS estimate at time $(t-1)$ was computed as

$$\begin{aligned}\hat{\theta}(t-1) &= \left(\sum_{i=1}^{t-1}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)\\ &= P(t-1)\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)\end{aligned}$$

$$P(t)^{-1} = P(t-1)^{-1} + \phi(t)\phi(t)^T$$

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\hat{\theta}(t) = \left(\Phi^T \Phi\right)^{-1} \Phi^T Y = \left(\sum_{i=1}^{t} \phi(i)\phi(i)^T\right)^{-1} \left(\sum_{i=1}^{t} \phi(i)y(i)\right)$$

$$= \hat{\theta}(t-1) - \boxed{P(t)\phi(t)}\, \phi(t)^T \hat{\theta}(t-1) + \boxed{P(t)\phi(t)}\, y(t)$$

The LS estimate at time $(t-1)$ was computed as

$$\hat{\theta}(t-1) = \left(\sum_{i=1}^{t-1} \phi(i)\phi(i)^T\right)^{-1} \left(\sum_{i=1}^{t-1} \phi(i)y(i)\right)$$

$$= P(t-1)\left(\sum_{i=1}^{t-1} \phi(i)y(i)\right)$$

$$P(t)^{-1} = P(t-1)^{-1} + \phi(t)\phi(t)^T$$

Basis for Recursive LS Computing

The LS estimate at time t is computed as

$$\begin{aligned}\hat{\theta}(t) &= \left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \left(\sum_{i=1}^{t}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t}\phi(i)y(i)\right)\\ &= \hat{\theta}(t-1) + P(t)\phi(t)\left(y(t)-\phi(t)^T\hat{\theta}(t-1)\right)\end{aligned}$$

The LS estimate at time $(t-1)$ was computed as

$$\begin{aligned}\hat{\theta}(t-1) &= \left(\sum_{i=1}^{t-1}\phi(i)\phi(i)^T\right)^{-1}\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)\\ &= P(t-1)\left(\sum_{i=1}^{t-1}\phi(i)y(i)\right)\end{aligned}$$

$$P(t)^{-1} = P(t-1)^{-1} + \phi(t)\phi(t)^T$$

Basis for Recursive LS Computing

To summarize, the update at time t can be computed as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \phi(t)^T\hat{\theta}(t-1)\right)$$

where

$$\begin{aligned} K(t) &= P(t)\phi(t) \\ P(t) &= \left(P(t-1)^{-1} + \phi(t)\phi(t)^T\right)^{-1} \end{aligned}$$

Basis for Recursive LS Computing

To summarize, the update at time t can be computed as

$$\hat{\theta}(t) \;=\; \hat{\theta}(t-1) + K(t)\Big(y(t) - \phi(t)^T\hat{\theta}(t-1)\Big)$$

where

$$\begin{aligned} K(t) &= P(t)\phi(t) \\ P(t) &= \Big(P(t-1)^{-1} + \phi(t)\phi(t)^T\Big)^{-1} \end{aligned}$$

Let us try now to simplify the last formula for $P(t)$ update

$$\Big(A + BD\Big)^{-1} = ??$$

Lemma:

Given matrices A, B, D of dimensions $n \times n$, $n \times m$ and $m \times n$ respectively, if the $n \times n$ and $m \times m$ matrices A and $(I_m + DB)$ are nonsigular, i.e.

$$\det A \neq 0, \quad \det(I_m + DB) \neq 0$$

Lemma:

Given matrices A, B, D of dimensions $n \times n$, $n \times m$ and $m \times n$ respectively, if the $n \times n$ and $m \times m$ matrices A and $(I_m + DB)$ are nonsingular, i.e.

$$\det A \neq 0, \quad \det(I_m + DB) \neq 0$$

---

then

- the $n \times n$ matrix $(A + BD)$ is nonsingular, i.e. $\det(A + BD) \neq 0$
- its inverse can be computed as follows

$$(A + BD)^{-1} = A^{-1} - A^{-1}B\left(I_m + DA^{-1}B\right)^{-1} DA^{-1}$$

Sketch of the Proof:

The direct computations show that

$$(A+BD)\times\left[A^{-1}-A^{-1}B\left(I_m+DA^{-1}B\right)^{-1}DA^{-1}\right] =$$

$$= \; I_n + BDA^{-1} - B\left(I_m+DA^{-1}B\right)^{-1}DA^{-1} -$$

$$-BDA^{-1}B\left(I_m+DA^{-1}B\right)^{-1}DA^{-1}$$

Sketch of the Proof:

The direct computations show that

$$(A+BD)\times\left[A^{-1}-A^{-1}B\left(I_m+DA^{-1}B\right)^{-1}DA^{-1}\right]=$$

$$= \; I_n+BDA^{-1}-B\boxed{\left(I_m+DA^{-1}B\right)^{-1}DA^{-1}}-$$

$$-BDA^{-1}B\boxed{\left(I_m+DA^{-1}B\right)^{-1}DA^{-1}}$$

Sketch of the Proof:

The direct computations show that

$$(A+BD)\times\left[A^{-1} - A^{-1}B\left(I_m + DA^{-1}B\right)^{-1}DA^{-1}\right] =$$

$$= \quad I_n + BDA^{-1} - B\left(I_m + DA^{-1}B\right)^{-1}DA^{-1} -$$

$$-BDA^{-1}B\left(I_m + DA^{-1}B\right)^{-1}DA^{-1}$$

$$= \quad I_n + BDA^{-1} -$$

$$-B\left(I_m + DA^{-1}B\right)\left(I_m + DA^{-1}B\right)^{-1}DA^{-1}$$

Sketch of the Proof:

The direct computations show that

$$(A+BD)\times\left[A^{-1}-A^{-1}B\left(I_m+DA^{-1}B\right)^{-1}DA^{-1}\right] =$$

$$= I_n + BDA^{-1} - B\left(I_m+DA^{-1}B\right)^{-1}DA^{-1} -$$

$$-BDA^{-1}B\left(I_m+DA^{-1}B\right)^{-1}DA^{-1}$$

$$= I_n + BDA^{-1} -$$

$$-B\left(I_m+DA^{-1}B\right)\left(I_m+DA^{-1}B\right)^{-1}DA^{-1}$$

$$= I_n + BDA^{-1} - BDA^{-1} = I_n$$

The formula

$$\left(A + BD\right)^{-1} = A^{-1} - A^{-1}B\left(I_m + DA^{-1}B\right)^{-1}DA^{-1}$$

should be applied to the expression

$$P(t) = \left(P(t-1)^{-1} + \phi(t)\phi(t)^{T}\right)^{-1}$$

The formula

$$\left(A + BD\right)^{-1} = A^{-1} - A^{-1}B\left(I_m + DA^{-1}B\right)^{-1}DA^{-1}$$

should be applied to the expression

$$P(t) = \left(P(t-1)^{-1} + \phi(t)\phi(t)^T\right)^{-1}$$

We obtain (with $\phi \equiv \phi(t)$)

$$P(t) = P(t-1) - P(t-1)\phi\left(1 + \phi^T P(t-1)\phi\right)^{-1}\phi^T P(t-1)$$

The formula

$$\left(A+BD\right)^{-1} = A^{-1} - A^{-1}B\left(I_m + DA^{-1}B\right)^{-1}DA^{-1}$$

should be applied to the expression

$$P(t) = \left(P(t-1)^{-1} + \phi(t)\phi(t)^T\right)^{-1}$$

We obtain (with $\phi \equiv \phi(t)$)

$$P(t) = P(t-1) - P(t-1)\phi\left(1 + \phi^T P(t-1)\phi\right)^{-1}\phi^T P(t-1)$$

We can simplify computation of the gain $K(t) = P(t)\phi(t)$

$$K(t) = P(t-1)\phi\left[1 - \left(1 + \phi^T P(t-1)\phi\right)^{-1}\phi^T P(t-1)\phi\right]$$

The formula

$$\left(A+BD\right)^{-1}=A^{-1}-A^{-1}B\left(I_m+DA^{-1}B\right)^{-1}DA^{-1}$$

should be applied to the expression

$$P(t)=\left(P(t-1)^{-1}+\phi(t)\phi(t)^T\right)^{-1}$$

We obtain (with $\phi\equiv\phi(t)$)

$$P(t)=P(t-1)-P(t-1)\phi\left(1+\phi^T P(t-1)\phi\right)^{-1}\phi^T P(t-1)$$

We can simplify computation of the gain $K(t)$

$$K(t)=P(t)\phi(t)=P(t-1)\phi\left(1+\phi^T P(t-1)\phi\right)^{-1}$$

and then

$$P(t)=\left(I_m-K(t)\,\phi(t)^T\right)\,P(t-1).$$

Theorem (Recursive Least Squares):

Assume that for all $t \geq t_0$ the excitation condition is valid, i.e.

$$\Phi(t)^T \Phi(t) > 0.$$

Theorem (Recursive Least Squares):

Assume that for all $t \geq t_0$ the excitation condition is valid, i.e.

$$\Phi(t)^T \Phi(t) > 0.$$

Given $\hat{\theta}(t_0)$ and $P(t_0) = (\Phi(t_0)^T \Phi(t_0))^{-1}$, the LS estimate satisfies the recursive equations

$$\boxed{\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \phi(t)^T \hat{\theta}(t-1)\right)}$$

$$\boxed{K(t) = P(t-1)\phi(t) / \left(1 + \phi(t)^T P(t-1)\phi(t)\right)}$$

$$\boxed{P(t) = \left(I_m - K(t)\,\phi(t)^T\right)\, P(t-1)}$$

Comments:

- The equation

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \phi(t)^T\hat{\theta}(t-1)\right)$$

can be seen as a procedure to change the value of estimate if the current value cannot predict the output

$$y(t) \neq \phi(t)^T\hat{\theta}(t-1)$$

Comments:

- The equation

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \phi(t)^T\hat{\theta}(t-1)\right)$$

can be seen as a procedure to change the value of estimate if the current value cannot predict the output

$$y(t) \neq \phi(t)^T\hat{\theta}(t-1)$$

- The excitation condition

$$\Phi(t)^T\Phi(t) > 0, \quad \forall t \geq t_0$$

implies that one needs to wait a number of time steps in order to initialize in proper way the recursive computations. In this case the initial conditions are

$$P(t_0) = \left(\Phi(t_0)^T\Phi(t_0)\right)^{-1} \qquad \hat{\theta}(t_0) = P(t_0)\Phi(t_0)^T Y(t_0)$$

Comments:

- The equation

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \phi(t)^T\hat{\theta}(t-1)\right)$$

can be seen as a procedure to change the value of estimate if the current value cannot predict the output

$$y(t) \neq \phi(t)^T\hat{\theta}(t-1)$$

- The excitation condition

$$\Phi(t)^T\Phi(t) > 0, \quad \forall t \geq t_0$$

implies that one needs to wait a number of time steps in order to initialize in proper way the recursive computations. In this case the initial conditions are

$$P(t_0) = \left(\Phi(t_0)^T\Phi(t_0)\right)^{-1} \qquad \hat{\theta}(t_0) = P(t_0)\Phi(t_0)^TY(t_0)$$

What to do if you like to start recursive computations at $t = 0$?

Modification for the start-up

Can we start with

$$\boldsymbol{P(0) = P_0 > 0} \qquad \text{and} \qquad \boldsymbol{\hat{\theta}(0) = \theta_0} \;\; ?$$

Modification for the start-up

Can we start with

$$P(0) = P_0 > 0 \qquad \text{and} \qquad \hat{\theta}(0) = \theta_0 \ ?$$

Consider the modified loss-function to be minimized

$$V_N(\theta) = \frac{1}{2}\sum_{i=1}^{N}\Big(y(i) - \phi(i)^T\theta\Big)^2 + \frac{1}{2}(\theta - \theta_0)^T P_0^{-1}(\theta - \theta_0)$$

where

- θ_0 is the initial guess and
- P_0^{-1} is the measure of our confidence in this guess.

Modification for the start-up

Can we start with

$$P(0) = P_0 > 0 \qquad \text{and} \qquad \hat{\theta}(0) = \theta_0 \ ?$$

Consider the modified loss-function to be minimized

$$V_N(\theta) = \frac{1}{2}\sum_{i=1}^{N}\Big(y(i) - \phi(i)^T\theta\Big)^2 + \frac{1}{2}(\theta - \theta_0)^T P_0^{-1}(\theta - \theta_0)$$

where

- θ_0 is the initial guess and
- P_0^{-1} is the measure of our confidence in this guess.

Following the computations done for the case $P_0^{-1} = 0$:

$$\boxed{\hat{\theta}(N) = \Big(\Phi(N)^T\Phi(N) + P_0^{-1}\Big)^{-1}\Big(\Phi(N)^T Y(N) + P_0^{-1}\theta_0\Big)}$$

Modification for the start-up (cont'd)

Introducing the notation

$$P(t) = \left(\Phi(t)^T \Phi(t) + P_0^{-1} \right)^{-1} = \left(\sum_{i=0}^{t} \phi(i)\, \phi(i)^T + P_0^{-1} \right)^{-1}$$

Modification for the start-up (cont'd)

Introducing the notation

$$P(t) = \left(\Phi(t)^T\Phi(t) + P_0^{-1}\right)^{-1} = \left(\sum_{i=0}^{t} \phi(i)\,\phi(i)^T + P_0^{-1}\right)^{-1}$$

we have $P(t) = \left(P(t-1)^{-1} + \phi(t)\,\phi(t)^T\right)^{-1}$ and

$$\begin{aligned}\hat{\theta}(t) &= P(t)\left(\sum_{i=0}^{t} \phi(i)\,y(i) + P_0^{-1}\theta_0\right) \\ &= P(t)\left(\sum_{i=0}^{t-1} \phi(i)\,y(i) + P_0^{-1}\theta_0 + \phi(t)\,y(t)\right) \\ &= P(t)\left(P(t-1)^{-1}\,\hat{\theta}(t-1) + \phi(t)\,y(t)\right).\end{aligned}$$

Modification for the start-up (cont'd)

Introducing the notation

$$P(t) = \left(\Phi(t)^T \Phi(t) + P_0^{-1}\right)^{-1} = \left(\sum_{i=0}^{t} \phi(i)\,\phi(i)^T + P_0^{-1}\right)^{-1}$$

we have $\quad P(t) = \left(P(t-1)^{-1} + \phi(t)\,\phi(t)^T\right)^{-1} \quad$ and

$$\begin{aligned}
\hat{\theta}(t) &= P(t)\left(\sum_{i=0}^{t} \phi(i)\,y(i) + P_0^{-1}\theta_0\right) \\
&= P(t)\left(\sum_{i=0}^{t-1} \phi(i)\,y(i) + P_0^{-1}\theta_0 + \phi(t)\,y(t)\right) \\
&= P(t)\left(P(t-1)^{-1}\,\hat{\theta}(t-1) + \phi(t)\,y(t)\right).
\end{aligned}$$

This is the same as for the usual Recursive Least Square!
Hence, the only modification is the initial values.

Designing Kalman Filter:

Consider the dynamical system

$$\theta_k = \theta_{k-1}, \quad y_k = C_k^T \theta_k + e_k, \quad k = 1, 2, 3, \ldots$$

Here (with $y_k = y(i)$, $C_k = \phi(i)$, $e_k = e(i)$)

- θ_k is the state vector,
- y_k is the vector of measurements,
- e_k is the noise with

$$E\, e_k = 0, \quad E\, e_k^2 = Q, \quad E\,(e_k\, e_m) = 0 \text{ for } k \neq m$$

- the initial condition θ_0 is independent with $e_k\ \forall\, k$ and

$$E\,\theta_0 = \theta^0, \qquad E\,(\theta_0 - \theta^0)(\theta_0 - \theta^0)^T = P_0 > 0.$$

Designing Kalman Filter:

Consider the dynamical system

$$\theta_k = \theta_{k-1}, \quad y_k = C_k^T \theta_k + e_k, \quad k = 1, 2, 3, \ldots$$

Here (with $y_k = y(i)$, $C_k = \phi(i)$, $e_k = e(i)$)

- θ_k is the state vector,
- y_k is the vector of measurements,
- e_k is the noise with

$$E\, e_k = 0, \quad E\, e_k^2 = Q, \quad E\,(e_k\, e_m) = 0 \text{ for } k \neq m$$

- the initial condition θ_0 is independent with $e_k\ \forall\, k$ and

$$E\, \theta_0 = \theta^0, \qquad E\,(\theta_0 - \theta^0)(\theta_0 - \theta^0)^T = P_0 > 0.$$

Let us determine the minimum variance recursive estimator (Kalman filter) for this system

Designing Kalman Filter (Cont'd):

Predicting Step:

$$\hat{\boldsymbol{\theta}}_{k|k-1} = \hat{\boldsymbol{\theta}}_{k-1|k-1} \qquad \Big\{\text{the copy of dynamics: } \boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1}\Big\}$$

Designing Kalman Filter (Cont'd):

Predicting Step:

$$\hat{\theta}_{k|k-1} = \hat{\theta}_{k-1|k-1} \qquad \left\{ \text{the copy of dynamics: } \theta_k = \theta_{k-1} \right\}$$

Updating Step:

Given new data $[y_k,\ C_k]$, we can improve $\hat{\theta}_{k|k-1}$ by

$$\hat{\theta}_{k|k} = \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right)$$

Here L_k is a matrix parameter to be defined

Designing Kalman Filter (Cont'd):

Predicting Step:

$$\hat{\theta}_{k|k-1} = \hat{\theta}_{k-1|k-1} \qquad \left\{\text{the copy of dynamics: } \theta_k = \theta_{k-1}\right\}$$

Updating Step:

Given new data $[y_k,\ C_k]$, we can improve $\hat{\theta}_{k|k-1}$ by

$$\hat{\theta}_{k|k} = \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right)$$

Here L_k is a matrix parameter to be defined

In Kalman filter L_k is chosen so that the covariance of estimate

$$P_{k|k} := E \left(\theta_k - \hat{\theta}_{k|k} \right) \left(\theta_k - \hat{\theta}_{k|k} \right)^T$$

is minimal in some sense.

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$\begin{aligned}\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k\left(y_k - C_k^T\hat{\theta}_{k|k-1}\right)\\ &= \hat{\theta}_{k|k-1} + L_k\left(\left\{C_k^T\theta_k + e_k\right\} - C_k^T\hat{\theta}_{k|k-1}\right)\\ &= \left(I - L_kC_k^T\right)\hat{\theta}_{k|k-1} + L_kC_k^T\theta_k + L_ke_k\end{aligned}$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$
\begin{aligned}
\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right) \\
&= \hat{\theta}_{k|k-1} + L_k \left(\left\{ C_k^T \theta_k + e_k \right\} - C_k^T \hat{\theta}_{k|k-1} \right) \\
&= \left(I - L_k C_k^T \right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k
\end{aligned}
$$

Then for any matrix L_k we have

$$
P_{k|k} = E \left(\theta_k - \hat{\theta}_{k|k} \right) \left(\theta_k - \hat{\theta}_{k|k} \right)^T = E \left(z_k - L_k e_k \right) \left(z_k - L_k e_k \right)^T
$$

$$
z_k = \left(I - L_k C_k^T \right) \left(\theta_k - \hat{\theta}_{k|k-1} \right)
$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$
\begin{aligned}
\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right) \\
&= \hat{\theta}_{k|k-1} + L_k \left(\left\{ C_k^T \theta_k + e_k \right\} - C_k^T \hat{\theta}_{k|k-1} \right) \\
&= \left(I - L_k C_k^T \right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k
\end{aligned}
$$

Then for any matrix L_k we have

$$
\begin{aligned}
P_{k|k} &= E \left(\theta_k - \hat{\theta}_{k|k} \right) \left(\theta_k - \hat{\theta}_{k|k} \right)^T = E \left(z_k - L_k e_k \right) \left(z_k - L_k e_k \right)^T \\
&= E z_k z_k^T + E L_k e_k (L_k e_k)^T = E z_k z_k^T + L_k \, E e_k e_k^T \, L_k{}^T
\end{aligned}
$$

$$
z_k = \left(I - L_k C_k^T \right) \left(\theta_k - \hat{\theta}_{k|k-1} \right)
$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$\begin{aligned}\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \hat{\theta}_{k|k-1} + L_k \left(\left\{ C_k^T \theta_k + e_k \right\} - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \left(I - L_k C_k^T \right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k\end{aligned}$$

Then for any matrix L_k we have

$$\begin{aligned}P_{k|k} &= E \left(\theta_k - \hat{\theta}_{k|k} \right) \left(\theta_k - \hat{\theta}_{k|k} \right)^T = E \left(z_k - L_k e_k \right) \left(z_k - L_k e_k \right)^T \\ &= E z_k z_k^T + E L_k e_k (L_k e_k)^T = E z_k z_k^T + L_k Q L_k{}^T\end{aligned}$$

$$z_k = \left(I - L_k C_k^T \right) \left(\theta_k - \hat{\theta}_{k|k-1} \right)$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$\begin{aligned}\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \hat{\theta}_{k|k-1} + L_k \left(\left\{ C_k^T \theta_k + e_k \right\} - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \left(I - L_k C_k^T \right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k\end{aligned}$$

Then for any matrix L_k we have

$$\begin{aligned}P_{k|k} &= E\left(\theta_k - \hat{\theta}_{k|k} \right)\left(\theta_k - \hat{\theta}_{k|k} \right)^T = E\left(z_k - L_k e_k \right)\left(z_k - L_k e_k \right)^T \\ &= E z_k z_k^T + E L_k e_k (L_k e_k)^T = E z_k z_k^T + L_k Q L_k{}^T \\ &= \left(I - L_k C_k^T \right) \boxed{E(\theta_k - \hat{\theta}_{k|k-1})(\theta_k - \hat{\theta}_{k|k-1})^T} \left(I - L_k C_k^T \right)^T + \\ &\qquad + L_k Q L_k{}^T\end{aligned}$$

$$z_k = \left(I - L_k C_k^T \right)\left(\theta_k - \hat{\theta}_{k|k-1} \right)$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$\begin{aligned}
\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right) \\
&= \hat{\theta}_{k|k-1} + L_k \left(\left\{ C_k^T \theta_k + e_k \right\} - C_k^T \hat{\theta}_{k|k-1} \right) \\
&= \left(I - L_k C_k^T \right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k
\end{aligned}$$

Then for any matrix L_k we have

$$\begin{aligned}
P_{k|k} &= E \left(\theta_k - \hat{\theta}_{k|k} \right) \left(\theta_k - \hat{\theta}_{k|k} \right)^T = E \left(z_k - L_k e_k \right) \left(z_k - L_k e_k \right)^T \\
&= E z_k z_k^T + E L_k e_k (L_k e_k)^T = E z_k z_k^T + L_k Q {L_k}^T \\
&= \left(I - L_k C_k^T \right) P_{k|k-1} \left(I - L_k C_k^T \right)^T + L_k Q {L_k}^T
\end{aligned}$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$\begin{aligned}
\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right) \\
&= \hat{\theta}_{k|k-1} + L_k \left(\left\{ C_k^T \theta_k + e_k \right\} - C_k^T \hat{\theta}_{k|k-1} \right) \\
&= \left(I - L_k C_k^T \right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k
\end{aligned}$$

Then for any matrix L_k we have

$$\begin{aligned}
P_{k|k} &= E \left(\theta_k - \hat{\theta}_{k|k} \right) \left(\theta_k - \hat{\theta}_{k|k} \right)^T = E \left(z_k - L_k e_k \right) \left(z_k - L_k e_k \right)^T \\
&= E z_k z_k^T + E L_k e_k (L_k e_k)^T = E z_k z_k^T + L_k Q L_k{}^T \\
&= \left(I - L_k C_k^T \right) P_{k|k-1} \left(I - L_k C_k^T \right)^T + L_k Q L_k{}^T
\end{aligned}$$

$$= P_{k|k-1} - L_k C_k^T P_{k|k-1} - P_{k|k-1} C_k L_k{}^T + L_k \left(Q + C_k^T P_{k|k-1} C_k \right) L_k{}^T$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$\begin{aligned}\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \hat{\theta}_{k|k-1} + L_k \left(\left\{ C_k^T \theta_k + e_k \right\} - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \left(I - L_k C_k^T \right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k\end{aligned}$$

Then for any matrix L_k we have

$$\begin{aligned}P_{k|k} &= E \left(\theta_k - \hat{\theta}_{k|k} \right) \left(\theta_k - \hat{\theta}_{k|k} \right)^T = E \left(z_k - L_k e_k \right) \left(z_k - L_k e_k \right)^T \\ &= E z_k z_k^T + E L_k e_k (L_k e_k)^T = E z_k z_k^T + L_k Q L_k{}^T \\ &= \left(I - L_k C_k^T \right) P_{k|k-1} \left(I - L_k C_k^T \right)^T + L_k Q L_k{}^T\end{aligned}$$

$$= P_{k|k-1} - L_k C_k^T P_{k|k-1} - P_{k|k-1} C_k L_k{}^T + L_k \left(Q + C_k^T P_{k|k-1} C_k \right) L_k{}^T$$

$$= W_0 + L_k W_1 + W_1^T L_k{}^T + L_k W_2 L_k{}^T$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$\begin{aligned}\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1}\right)\\ &= \hat{\theta}_{k|k-1} + L_k \left(\left\{C_k^T \theta_k + e_k\right\} - C_k^T \hat{\theta}_{k|k-1}\right)\\ &= \left(I - L_k C_k^T\right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k\end{aligned}$$

Then for any matrix L_k we have

$$P_{k|k} = E\left(\theta_k - \hat{\theta}_{k|k}\right)\left(\theta_k - \hat{\theta}_{k|k}\right)^T$$

$$= P_{k|k-1} - L_k C_k^T P_{k|k-1} - P_{k|k-1} C_k L_k{}^T + L_k \left(Q + C_k^T P_{k|k-1} C_k\right) L_k{}^T$$

$$= W_0 + L_k W_1 + W_1^T L_k{}^T + L_k W_2 L_k{}^T$$

$$= W_0 + \left(L_k X + Y\right)\left(L_k X + Y\right)^T - Y Y^T$$

$$L_k{}^{opt} = -Y X^{-1}, \quad P_{k|k}(L_k{}^{opt}) = W_0 - Y Y^T$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$\begin{aligned}\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \hat{\theta}_{k|k-1} + L_k \left(\left\{ C_k^T \theta_k + e_k \right\} - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \left(I - L_k C_k^T \right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k\end{aligned}$$

Then for any matrix L_k we have

$$P_{k|k} = E \left(\theta_k - \hat{\theta}_{k|k} \right) \left(\theta_k - \hat{\theta}_{k|k} \right)^T$$

$$= P_{k|k-1} - L_k C_k^T P_{k|k-1} - P_{k|k-1} C_k L_k{}^T + L_k \left(Q + C_k^T P_{k|k-1} C_k \right) L_k{}^T$$

$$= W_0 + \boxed{L_k W_1 + W_1^T L_k{}^T + L_k W_2 L_k{}^T}$$

$$= W_0 + L_k X Y^T + Y X^T L_k{}^T + L_k X X^T L_k{}^T$$

$$W_2 = X X^T, \; W_1 = X Y^T \quad \Rightarrow \quad Y^T = X^{-1} W_1 = W_2^{-\frac{1}{2}} W_1$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$\begin{aligned}\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \hat{\theta}_{k|k-1} + L_k \left(\left\{ C_k^T \theta_k + e_k \right\} - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \left(I - L_k C_k^T \right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k\end{aligned}$$

Then for any matrix L_k we have

$$P_{k|k} = E \left(\theta_k - \hat{\theta}_{k|k} \right) \left(\theta_k - \hat{\theta}_{k|k} \right)^T$$

$$= P_{k|k-1} - L_k C_k^T P_{k|k-1} - P_{k|k-1} C_k L_k{}^T + L_k \left(Q + C_k^T P_{k|k-1} C_k \right) L_k{}^T$$

$$= W_0 + L_k W_1 + W_1^T L_k{}^T + L_k W_2 L_k{}^T$$

$$W_2 = X X^T, \quad Y^T = X^{-1} W_1 = W_2^{-\frac{1}{2}} W_1$$

$$\Downarrow$$

$$L_k{}^{opt} = -Y X^{-1}, \quad P_{k|k}^{opt} = W_0 - Y Y^T$$

Designing Kalman Filter (Cont'd):

The estimate $\hat{\theta}_{k|k}$ can be expressed as follows

$$\begin{aligned}\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + L_k \left(y_k - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \hat{\theta}_{k|k-1} + L_k \left(\left\{ C_k^T \theta_k + e_k \right\} - C_k^T \hat{\theta}_{k|k-1} \right) \\ &= \left(I - L_k C_k^T \right) \hat{\theta}_{k|k-1} + L_k C_k^T \theta_k + L_k e_k\end{aligned}$$

Then for any matrix L_k we have

$$P_{k|k} = E \left(\theta_k - \hat{\theta}_{k|k} \right) \left(\theta_k - \hat{\theta}_{k|k} \right)^T$$

$$= P_{k|k-1} - L_k C_k^T P_{k|k-1} - P_{k|k-1} C_k {L_k}^T + L_k \left(Q + C_k^T P_{k|k-1} C_k \right) {L_k}^T$$

$$= W_0 + L_k W_1 + W_1^T {L_k}^T + L_k W_2 {L_k}^T$$

$$W_2 = X X^T, \quad Y^T = X^{-1} W_1 = W_2^{-\frac{1}{2}} W_1$$

$$\Downarrow$$

$${L_k}^{opt} = -W_1^T W_2^{-1}, \quad P_{k|k}^{opt} = W_0 - W_1^T W_2^{-1} W_1$$

Designing Kalman Filter (summary):

The optimal gain $\boldsymbol{L_k}$ that ensures the minimal variance $\boldsymbol{P_{k|k}}$ of updated estimate $\boldsymbol{\hat{\theta}_{k|k}}$ is

$$
\begin{aligned}
\boldsymbol{L_k}^{opt} &= -\boldsymbol{W_1^T W_2^{-1}} \\
&= \boldsymbol{P_{k|k-1} C_k \left(Q + C_k^T P_{k|k-1} C_k\right)^{-1}}
\end{aligned}
$$

Designing Kalman Filter (summary):

The optimal gain L_k that ensures the minimal variance $P_{k|k}$ of updated estimate $\hat{\theta}_{k|k}$ is

$$\begin{aligned} {L_k}^{opt} &= -W_1^T W_2^{-1} \\ &= P_{k|k-1} C_k \left(Q + C_k^T P_{k|k-1} C_k \right)^{-1} \end{aligned}$$

The corresponding variance is

$$\begin{aligned} P_{k|k}^{opt} &= W_0 - W_1^T W_2^{-1} W_1 \\ &= P_{k|k-1} - P_{k|k-1} C_k \left(Q + C_k^T P_{k|k-1} C_k \right)^{-1} C_k^T P_{k|k-1} \end{aligned}$$

Designing Kalman Filter (summary):

The optimal gain L_k that ensures the minimal variance $P_{k|k}$ of updated estimate $\hat{\theta}_{k|k}$ is

$$\begin{aligned} {L_k}^{opt} &= -W_1^T W_2^{-1} \\ &= P_{k|k-1} C_k \left(Q + C_k^T P_{k|k-1} C_k \right)^{-1} \end{aligned}$$

The corresponding variance is

$$\begin{aligned} P_{k|k}^{opt} &= W_0 - W_1^T W_2^{-1} W_1 \\ &= P_{k|k-1} - P_{k|k-1} C_k \left(Q + C_k^T P_{k|k-1} C_k \right)^{-1} C_k^T P_{k|k-1} \end{aligned}$$

These expressions coincide with the modified Recursive Computations of the Least Square Estimate for the case when the variance of noise Q equals 1.

Recursive Least Squares with exponential forgetting

Consider the modified loss-function to be minimized

$$V_t(\theta) = \frac{1}{2}\sum_{i=1}^{t} \lambda^{t-i}\Big(y(i) - \phi(i)^T\theta\Big)^2 + \frac{\lambda^t}{2}(\theta - \theta_0)^T P_0^{-1}(\theta - \theta_0)$$

with $\quad 0 < \lambda \leq 1.$

Recursive Least Squares with exponential forgetting

Consider the modified loss-function to be minimized

$$V_t(\theta) = \frac{1}{2}\sum_{i=1}^{t}\lambda^{t-i}\Big(y(i)-\phi(i)^T\theta\Big)^2+\frac{\lambda^t}{2}(\theta-\theta_0)^T P_0^{-1}(\theta-\theta_0)$$

with $0 < \lambda \leq 1$. One can obtain the following

$$\boxed{\hat\theta(t) = \hat\theta(t-1) + K(t)\left(y(t)-\phi(t)^T\hat\theta(t-1)\right)}$$

$$\boxed{K(t) = \frac{P(t-1)\phi(t)}{\lambda + \phi^T(t)P(t-1)\phi(t)}}$$

$$\boxed{P(t) = \left(I_m - K(t)\,\phi(t)^T\right)\,P(t-1)/\lambda}$$

Projection algorithm

Given the model $y(t) = \phi(t)^T \theta$, an estimate $\hat{\theta}(t-1)$ for θ and the value of $y(t)$, find

$$\hat{\theta}(t) = \arg\min \left\{ \|\hat{\theta}(t) - \hat{\theta}(t-1)\| : \quad y(t) = \phi(t)^T \hat{\theta}(t) \right\}.$$

Projection algorithm

Given the model $y(t) = \phi(t)^T \theta$, an estimate $\hat{\theta}(t-1)$ for θ and the value of $y(t)$, find

$$\hat{\theta}(t) = \arg\min \left\{ \|\hat{\theta}(t) - \hat{\theta}(t-1)\| : \quad y(t) = \phi(t)^T \hat{\theta}(t) \right\}.$$

To solve the problem, let us minimize

$$V(\theta, \lambda) = \frac{1}{2}\Big(\theta - \hat{\theta}(t-1)\Big)^T \Big(\theta - \hat{\theta}(t-1)\Big) + \lambda \Big(y(t) - \phi(t)^T \theta\Big).$$

Projection algorithm

Given the model $y(t) = \phi(t)^T \theta$, an estimate $\hat{\theta}(t-1)$ for θ and the value of $y(t)$, find

$$\hat{\theta}(t) = \arg\min \left\{ \|\hat{\theta}(t) - \hat{\theta}(t-1)\| : \quad y(t) = \phi(t)^T \hat{\theta}(t) \right\}.$$

To solve the problem, let us minimize

$$V(\theta, \lambda) = \frac{1}{2}\Big(\theta - \hat{\theta}(t-1)\Big)^T \Big(\theta - \hat{\theta}(t-1)\Big) + \lambda \Big(y(t) - \phi(t)^T \theta\Big).$$

The conditions for the minimum are

$$\mathrm{grad}_{\theta} V(\theta, \lambda) = \theta - \hat{\theta}(t-1) - \lambda\, \phi(t) = 0 \qquad \text{for} \quad \theta = \hat{\theta}(t)$$

$$\frac{\partial V(\theta, \lambda)}{\partial \lambda} = y(t) - \phi(t)^T \theta = 0 \qquad \text{for} \quad \theta = \hat{\theta}(t).$$

Projection algorithm

Given the model $y(t) = \phi(t)^T \theta$, an estimate $\hat{\theta}(t-1)$ for θ and the value of $y(t)$, find

$$\hat{\theta}(t) = \arg\min \left\{ \|\hat{\theta}(t) - \hat{\theta}(t-1)\| : \quad y(t) = \phi(t)^T \hat{\theta}(t) \right\}.$$

To solve the problem, let us minimize

$$V(\theta, \lambda) = \frac{1}{2}\Big(\theta - \hat{\theta}(t-1)\Big)^T \Big(\theta - \hat{\theta}(t-1)\Big) + \lambda\Big(y(t) - \phi(t)^T \theta\Big).$$

The conditions for the minimum are

$$\hat{\theta}(t) - \hat{\theta}(t-1) - \lambda\,\phi(t) = 0$$

$$y(t) - \phi(t)^T \hat{\theta}(t) = 0.$$

Projection algorithm (cont'd) / Gradient algorithm

Substituting

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \lambda\,\phi(t)$$

into $y(t) = \phi(t)^T\hat{\theta}(t)$ and solving for λ:

$$\lambda = \Big(y(t) - \phi(t)^T\hat{\theta}(t-1)\Big)/\Big(\phi(t)^T\phi(t)\Big)$$

and substituting back we have the Projection algorithm

$$\boxed{\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\phi(t)}{\phi(t)^T\phi(t)}\Big(y(t) - \phi(t)^T\hat{\theta}(t-1)\Big)}$$

Projection algorithm (cont'd) / Gradient algorithm

Substituting

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \lambda\, \phi(t)$$

into $y(t) = \phi(t)^T \hat{\theta}(t)$ and solving for λ:

$$\lambda = \Big(y(t) - \phi(t)^T \hat{\theta}(t-1)\Big) / \Big(\phi(t)^T \phi(t)\Big)$$

and substituting back we have the Projection algorithm

$$\boxed{\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\phi(t)}{\phi(t)^T \phi(t)} \Big(y(t) - \phi(t)^T \hat{\theta}(t-1)\Big)}$$

Modifying the algorithm to avoid possible devisions by zero, we obtain the Gradient algorithm

$$\boxed{\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\gamma\, \phi(t)}{\alpha + \phi(t)^T \phi(t)} \Big(y(t) - \phi(t)^T \hat{\theta}(t-1)\Big)}$$

with $2 > \gamma > 0$ and $\alpha > 0$.

Continuous-time Models

Consider the regression model

$$y(\tau) = \phi(\tau)^T \theta^0$$

with τ defined on $[0, t]$.

Continuous-time Models

Consider the regression model

$$y(\tau) = \phi(\tau)^T \theta^0$$

with τ defined on $[0, t]$.

To compute the estimate $\hat{\theta}(t)$ of θ^0, we minimize the function

$$V_t(\theta) = \int_0^t \frac{e^{-\alpha(t-\tau)}}{2} \Big(\underbrace{y(\tau) - \phi(\tau)^T \theta}_{\text{prediction error}} \Big)^2 d\tau + \frac{e^{-\alpha t}}{2} (\theta - \theta_0)^T P_0^{-1} (\theta - \theta_0)$$

where the inverse of $P_0 = P_0^T > 0$ defines how much we trust in the initial guess θ_0, and $\alpha \geq 0$ is the forgetting factor.

Continuous-time Models

Consider the regression model

$$y(\tau) = \phi(\tau)^T \theta^0$$

with τ defined on $[0, t]$.

To compute the estimate $\hat{\theta}(t)$ of θ^0, we minimize the function

$$V_t(\theta) = \int_0^t \frac{e^{-\alpha\,(t-\tau)}}{2} \Big(\underbrace{y(\tau) - \phi(\tau)^T \theta}_{\text{prediction error}} \Big)^2 d\tau + \frac{e^{-\alpha\, t}}{2} (\theta - \theta_0)^T P_0^{-1} (\theta - \theta_0)$$

where the inverse of $P_0 = P_0^T > 0$ defines how much we trust in the initial guess θ_0, and $\alpha \geq 0$ is the forgetting factor.

The condition for minimum at $\theta = \hat{\theta}(t)$ is

$$0 = \text{grad}_{\theta} V_t(\theta)$$

Continuous-time Models

Consider the regression model

$$y(\tau) = \phi(\tau)^T \theta^0$$

with τ defined on $[0, t]$.

To compute the estimate $\hat{\theta}(t)$ of θ^0, we minimize the function

$$V_t(\theta) = \int_0^t \frac{e^{-\alpha(t-\tau)}}{2} \Big(\underbrace{y(\tau) - \phi(\tau)^T \theta}_{\text{prediction error}} \Big)^2 d\tau + \frac{e^{-\alpha t}}{2} (\theta - \theta_0)^T P_0^{-1} (\theta - \theta_0)$$

where the inverse of $P_0 = P_0^T > 0$ defines how much we trust in the initial guess θ_0, and $\alpha \geq 0$ is the forgetting factor.

The condition for minimum at $\theta = \hat{\theta}(t)$ is

$$0 = \int_0^t e^{-\alpha(t-\tau)} \Big(-\phi(\tau)\, y(\tau) + \phi(\tau)\, \phi(\tau)^T \theta \Big) d\tau + e^{-\alpha t} P_0^{-1} (\theta - \theta_0)$$

Continuous-time Models

Consider the regression model

$$y(\tau) = \phi(\tau)^T \theta^0$$

with τ defined on $[0, t]$.

To compute the estimate $\hat{\theta}(t)$ of θ^0, we minimize the function

$$V_t(\theta) = \int_0^t \frac{e^{-\alpha(t-\tau)}}{2} \Big(\underbrace{y(\tau) - \phi(\tau)^T \theta}_{\text{prediction error}} \Big)^2 d\tau + \frac{e^{-\alpha t}}{2} (\theta - \theta_0)^T P_0^{-1} (\theta - \theta_0)$$

where the inverse of $P_0 = P_0^T > 0$ defines how much we trust in the initial guess θ_0, and $\alpha \geq 0$ is the forgetting factor.

The condition for minimum is

$$0 = \int_0^t e^{\alpha \tau} \Big(-\phi(\tau)\, y(\tau) + \phi(\tau)\, \phi(\tau)^T \hat{\theta}(t) \Big) d\tau + P_0^{-1} \Big(\hat{\theta}(t) - \theta_0 \Big)$$

Continuous-time Models

Consider the regression model

$$y(\tau) = \phi(\tau)^T \theta^0$$

with τ defined on $[0, t]$.

To compute the estimate $\hat{\theta}(t)$ of θ^0, we minimize the function

$$V_t(\theta) = \int_0^t \frac{e^{-\alpha(t-\tau)}}{2} \Big(\underbrace{y(\tau) - \phi(\tau)^T \theta}_{\text{prediction error}} \Big)^2 d\tau + \frac{e^{-\alpha t}}{2} (\theta - \theta_0)^T P_0^{-1} (\theta - \theta_0)$$

where the inverse of $P_0 = P_0^T > 0$ defines how much we trust in the initial guess θ_0, and $\alpha \geq 0$ is the forgetting factor.

The condition for minimum is

$$\left[\int_0^t e^{\alpha \tau} \phi(\tau)\, \phi(\tau)^T d\tau + e^{-\alpha t} P_0^{-1} \right] \hat{\theta}(t) = \int_0^t e^{\alpha \tau} \phi(\tau)\, y(\tau) d\tau + P_0^{-1} \theta_0$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$R(t) = \int_0^t e^{\alpha\,\tau} \phi(\tau)\,\phi(\tau)^T d\tau + e^{-\alpha\,t} P_0^{-1}.$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$\boldsymbol{R}(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,\phi(\tau)^{\tau} d\tau + e^{-\alpha\,t}P_0^{-1}.$$

Then

$$\boldsymbol{R}(t)\,\hat{\theta}(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,y(\tau)d\tau + P_0^{-1}\theta_0$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$R(t) = \int_0^t e^{\alpha\,\tau} \phi(\tau)\, \phi(\tau)^T d\tau + e^{-\alpha\, t} P_0^{-1}.$$

Then

$$R(t)\, \hat{\theta}(t) = \int_0^t e^{\alpha\,\tau} \phi(\tau)\, y(\tau) d\tau + P_0^{-1} \theta_0$$

Differentiate with respect to t, we obtain the updating law:

$$\left(\frac{d}{dt} R(t)\right) \hat{\theta}(t) + R(t)\, \frac{d}{dt} \hat{\theta}(t) = e^{\alpha\, t} \phi(t)\, y(t)$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$R(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,\phi(\tau)^T d\tau + e^{-\alpha\,t}P_0^{-1}.$$

Then

$$R(t)\,\hat{\theta}(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,y(\tau)d\tau + P_0^{-1}\theta_0$$

Solving for $\frac{d}{dt}\hat{\theta}(t)$

$$R(t)\,\frac{d}{dt}\hat{\theta}(t) = e^{\alpha\,t}\phi(t)\,y(t) - \left(\frac{d}{dt}R(t)\right)\hat{\theta}(t)$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$R(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,\phi(\tau)^T d\tau + e^{-\alpha\,t}P_0^{-1}.$$

Then

$$R(t)\,\hat{\theta}(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,y(\tau)d\tau + P_0^{-1}\theta_0$$

Solving for $\frac{d}{dt}\hat{\theta}(t)$

$$\frac{d}{dt}\hat{\theta}(t) = R(t)^{-1}\left[e^{\alpha\,t}\phi(t)\,y(t) - \left(\frac{d}{dt}R(t)\right)\hat{\theta}(t)\right]$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$\boldsymbol{R(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,\phi(\tau)^T d\tau + e^{-\alpha\,t}P_0^{-1}.}$$

Then

$$\boldsymbol{R(t)\,\hat{\theta}(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,y(\tau)d\tau + P_0^{-1}\theta_0}$$

Substituting $\boldsymbol{\frac{d}{dt}R(t)}$

$$\boldsymbol{\frac{d}{dt}\hat{\theta}(t) = R(t)^{-1}\left[e^{\alpha\,t}\phi(t)\,y(t) - \left(e^{\alpha\,t}\phi(t)\,\phi(t)^T\right)\hat{\theta}(t)\right]}$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$R(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,\phi(\tau)^T d\tau + e^{-\alpha\,t}P_0^{-1}.$$

Then

$$R(t)\,\hat{\theta}(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,y(\tau)d\tau + P_0^{-1}\theta_0$$

Introducing $\quad P(t) = e^{\alpha\,t}\,R(t)^{-1}$

$$\boxed{\dot{\hat{\theta}}(t) = P(t)\,\phi(t)\left[y(t) - \phi(t)^T\hat{\theta}(t)\right]; \qquad \hat{\theta}(0) = \theta_0}$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$R(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,\phi(\tau)^T d\tau + e^{-\alpha\,t}P_0^{-1}.$$

Then

$$R(t)\,\hat{\theta}(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,y(\tau)d\tau + P_0^{-1}\theta_0$$

Introducing $\quad P(t) = e^{\alpha\,t}\,R(t)^{-1}$

$$\boxed{\dot{\hat{\theta}}(t) = P(t)\,\phi(t)\left[y(t) - \phi(t)^T\hat{\theta}(t)\right]; \qquad \hat{\theta}(0) = \theta_0}$$

Differentiating $\quad P(t)\,R(t) = e^{\alpha\,t}\,I_m$

$$\left(\frac{d}{dt}P(t)\right)R(t) + P(t)\left(\frac{d}{dt}R(t)\right) = \alpha\,e^{\alpha\,t}\,I_m$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$R(t) = \int_0^t e^{\alpha\,\tau} \phi(\tau)\,\phi(\tau)^T d\tau + e^{-\alpha\,t} P_0^{-1}.$$

Then

$$R(t)\,\hat{\theta}(t) = \int_0^t e^{\alpha\,\tau} \phi(\tau)\,y(\tau) d\tau + P_0^{-1}\theta_0$$

Introducing $\quad P(t) = e^{\alpha\,t}\,R(t)^{-1}$

$$\boxed{\dot{\hat{\theta}}(t) = P(t)\,\phi(t)\left[y(t) - \phi(t)^T\hat{\theta}(t)\right]; \qquad \hat{\theta}(0) = \theta_0}$$

Substituting $\quad \frac{d}{dt}R(t)$

$$\left(\frac{d}{dt}P(t)\right)R(t) + P(t)\left(e^{\alpha\,t}\phi(t)\,\phi(t)^T\right) = \alpha\,e^{\alpha\,t}\,I_m$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$R(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,\phi(\tau)^T d\tau + e^{-\alpha\,t} P_0^{-1}.$$

Then

$$R(t)\,\hat{\theta}(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,y(\tau)d\tau + P_0^{-1}\theta_0$$

Introducing $\quad P(t) = e^{\alpha\,t}\,R(t)^{-1}$

$$\boxed{\dot{\hat{\theta}}(t) = P(t)\,\phi(t)\left[y(t) - \phi(t)^T\hat{\theta}(t)\right]; \qquad \hat{\theta}(0) = \theta_0}$$

Using $\quad R(t) = P(t)^{-1}\,e^{\alpha\,t}$

$$\left(\frac{d}{dt}P(t)\right)P(t)^{-1}\,e^{\alpha\,t} + P(t)\left(e^{\alpha\,t}\phi(t)\,\phi(t)^T\right) = \alpha\,e^{\alpha\,t}\,I_m$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$R(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,\phi(\tau)^T d\tau + e^{-\alpha\,t}P_0^{-1}.$$

Then

$$R(t)\,\hat{\theta}(t) = \int_0^t e^{\alpha\,\tau}\phi(\tau)\,y(\tau)d\tau + P_0^{-1}\theta_0$$

Introducing $\quad P(t) = e^{\alpha\,t}\,R(t)^{-1}$

$$\boxed{\dot{\hat{\theta}}(t) = P(t)\,\phi(t)\left[y(t) - \phi(t)^T\hat{\theta}(t)\right]; \qquad \hat{\theta}(0) = \theta_0}$$

Canceling $\quad e^{\alpha\,t} \neq 0$

$$\left(\frac{d}{dt}P(t)\right)P(t)^{-1} + P(t)\Big(\phi(t)\,\phi(t)^T\Big) = \alpha\,I_m$$

Least Squares Algorithm for continuous-time

Introduce the following notation

$$R(t) = \int_0^t e^{\alpha\,\tau} \phi(\tau)\,\phi(\tau)^T d\tau + e^{-\alpha\,t} P_0^{-1}.$$

Then

$$R(t)\,\hat{\theta}(t) = \int_0^t e^{\alpha\,\tau} \phi(\tau)\,y(\tau) d\tau + P_0^{-1}\theta_0$$

Introducing $\quad P(t) = e^{\alpha\,t}\,R(t)^{-1}$

$$\boxed{\dot{\hat{\theta}}(t) = P(t)\,\phi(t)\left[y(t) - \phi(t)^T\hat{\theta}(t)\right]; \qquad \hat{\theta}(0) = \theta_0}$$

Finally

$$\boxed{\dot{P}(t) = \alpha\,P(t) - P(t)\left(\phi(t)\,\phi(t)^T\right)P(t); \qquad P(0) = P_0}$$

Continuous-time dynamical systems

Consider the input-output model $y(t) \longleftarrow u(t)$:

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n\, y = b_1 \frac{d^{m-1} y}{dt^{m-1}} + \cdots + b_m\, u$$

Continuous-time dynamical systems

Consider the input-output model $y(t) \longleftarrow u(t)$:

$$\underbrace{\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n\, y}_{A(p)\, y(t)} = \underbrace{b_1 \frac{d^{m-1} y}{dt^{m-1}} + \cdots + b_m\, u}_{B(p)\, u(t)}$$

$n \geq m$ and p is the differentiation operator: $p\, y(t) = \frac{dy}{dt}$.

Continuous-time dynamical systems

Consider the input-output model $y(t) \longleftarrow u(t)$:

$$\underbrace{\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n\, y}_{A(p)\, y(t)} = \underbrace{b_1 \frac{d^{m-1} y}{dt^{m-1}} + \cdots + b_m\, u}_{B(p)\, u(t)}$$

$n \geq m$ and p is the differentiation operator: $\quad p\, y(t) = \frac{dy}{dt}$.

How do we estimate the vector of parameters

$$\theta^0 = \left[a_1, \quad \ldots, \quad a_n, \quad b_1, \quad \ldots, \quad b_m \right]^T$$

from the measured data $\quad \{ [y(\tau), u(\tau)] : \quad \tau \in [0, t] \} \quad$?

Continuous-time dynamical systems

Consider the input-output model $y(t) \longleftarrow u(t)$:

$$\underbrace{\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n\, y}_{A(p)\, y(t)} = \underbrace{b_1 \frac{d^{m-1} y}{dt^{m-1}} + \cdots + b_m\, u}_{B(p)\, u(t)}$$

$n \geq m$ and p is the differentiation operator: $\quad p\, y(t) = \frac{dy}{dt}$.

How do we estimate the vector of parameters

$$\theta^0 = \Big[a_1, \quad \ldots, \quad a_n, \quad b_1, \quad \ldots, \quad b_m\Big]^T$$

from the measured data $\quad \big\{[y(\tau), u(\tau)] : \quad \tau \in [0, t]\big\} \quad$?

It would be easy to rewrite the model in the standard form, but there is one problem: differentiation can not be realized as a proper transfer function.

Regression for continuous-time dynamical systems

Trick: introduce a filter $\boxed{H_f(p)}$

$$A(p)\, y(t) = B(p)\, u(t) \;\Rightarrow\; \boxed{H_f(p)}\, A(p)\, y(t) = \boxed{H_f(p)}\, B(p)\, u(t)$$

Regression for continuous-time dynamical systems

Trick: introduce a filter $\boxed{H_f(p)}$

$$A(p)\,y(t) = B(p)\,u(t) \;\Rightarrow\; A(p)\,\underbrace{\boxed{H_f(p)\,y(t)}}_{y_f(t)} = B(p)\,\underbrace{\boxed{H_f(p)\,u(t)}}_{u_f(t)}$$

Regression for continuous-time dynamical systems

Trick: introduce a filter $H_f(p)$

$$A(p)\,y(t) = B(p)\,u(t) \;\Rightarrow\; A(p)\,\underbrace{H_f(p)\,y(t)}_{y_f(t)} = B(p)\,\underbrace{H_f(p)\,u(t)}_{u_f(t)}$$

With a stable minimum-phase filter with relative degree n or higher, such as $H_f(s) = \dfrac{1}{s^n + \lambda_1\, s^{n-1} + \cdots + \lambda_n}$, we have

$$A(p)\,y(t) = B(p)\,u(t) \;\Leftrightarrow\; \boxed{A(p)\,y_f(t) = B(p)\,u_f(t)}$$

where derivatives of order $1, \ldots, n$ for

$$\boxed{y_f(t) = H_f(p)\,y(t)} \quad \text{and} \quad \boxed{u_f(t) = H_f(p)\,u(t)}$$

can be realized as proper transfer functions.

Now, since

$$p^n\,y_f(t) = -a_1\,p^{n-1}\,y_f(t) - \cdots - a_n\,y_f(t) + b_1\,p^{m-1}\,u_f(t) + \cdots + b_n\,u_f(t)$$

Now, since

$$p^n\, y_f(t) = -a_1\, p^{n-1}\, y_f(t) - \cdots - a_n\, y_f(t) + b_1\, p^{m-1}\, u_f(t) + \cdots + b_n\, u_f(t)$$

The regression model is

$$p^n\, y_f(t) = \phi(t)^T\, \theta^0, \qquad \phi(t)^T = \Big[-p^{n-1}\, y_f(t),\ \ldots,\ p^{m-1}\, u_f(t)\Big]$$

Now, since

$$p^n\, y_f(t) = -a_1\, p^{n-1}\, y_f(t) - \cdots - a_n\, y_f(t) + b_1\, p^{m-1}\, u_f(t) + \cdots + b_n\, u_f(t)$$

The regression model is

$$p^n\, y_f(t) = \phi(t)^T\, \theta^0, \qquad \phi(t)^T = \left[-p^{n-1}\, y_f(t),\ \ldots,\ p^{m-1}\, u_f(t) \right]$$

and the estimation scheme is

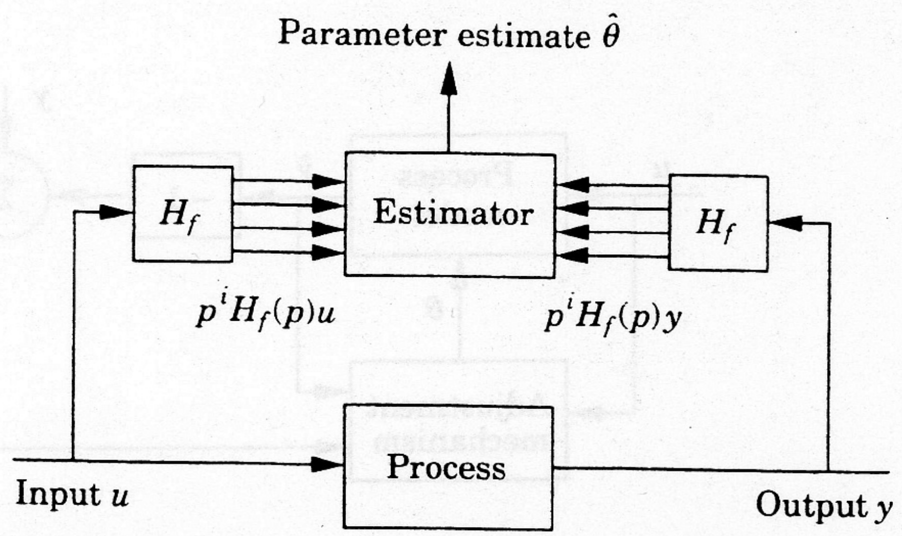

Figure 2.5 Block diagram of estimator with filters H_f.

Next Lecture / Assignments:

- Next lecture (April 14, 10:00-12:00, in A206Tekn): Convergence and Persistent Excitation.

Next Lecture / Assignments:

- Next lecture (April 14, 10:00-12:00, in A206Tekn): Convergence and Persistent Excitation.
- Derive the formulae for recursive least squares algorithm with exponential forgetting.

Another homework problem:

Consider the discrete-time system $y(t) = H(q)\,u(t)$

represented by the transfer function $H(z) = \dfrac{b_1\,z + b_2}{z^2 + a_1\,z + a_2}$.

- Write a recursive least squares algorithm with exponential forgetting to estimate the parameters $\{a_1, a_2, b_1, b_2\}$.
- Simulate your algorithm with the true parameters of the system $a_1 = a_2 = 0.5$, $b_1 = 0$, $b_2 = 1$. Study performance of the algorithm for
  - different initial conditions for the parameter estimates (try 0 initial conditions and at least one other choice),
  - different values of the forgetting factor (at least three values including $\lambda = 1$),
  - a unit step input and a square wave of unit amplitude and a period of 10 samples.
- Discuss the simulation results.