2.160 Identification, Estimation, and Learning

Part 1 Regression

Lecture 2

Least Squares Estimation for Deterministic Systems

H. Harry Asada
Department of Mechanical Engineering
MIT

Regression Analysis

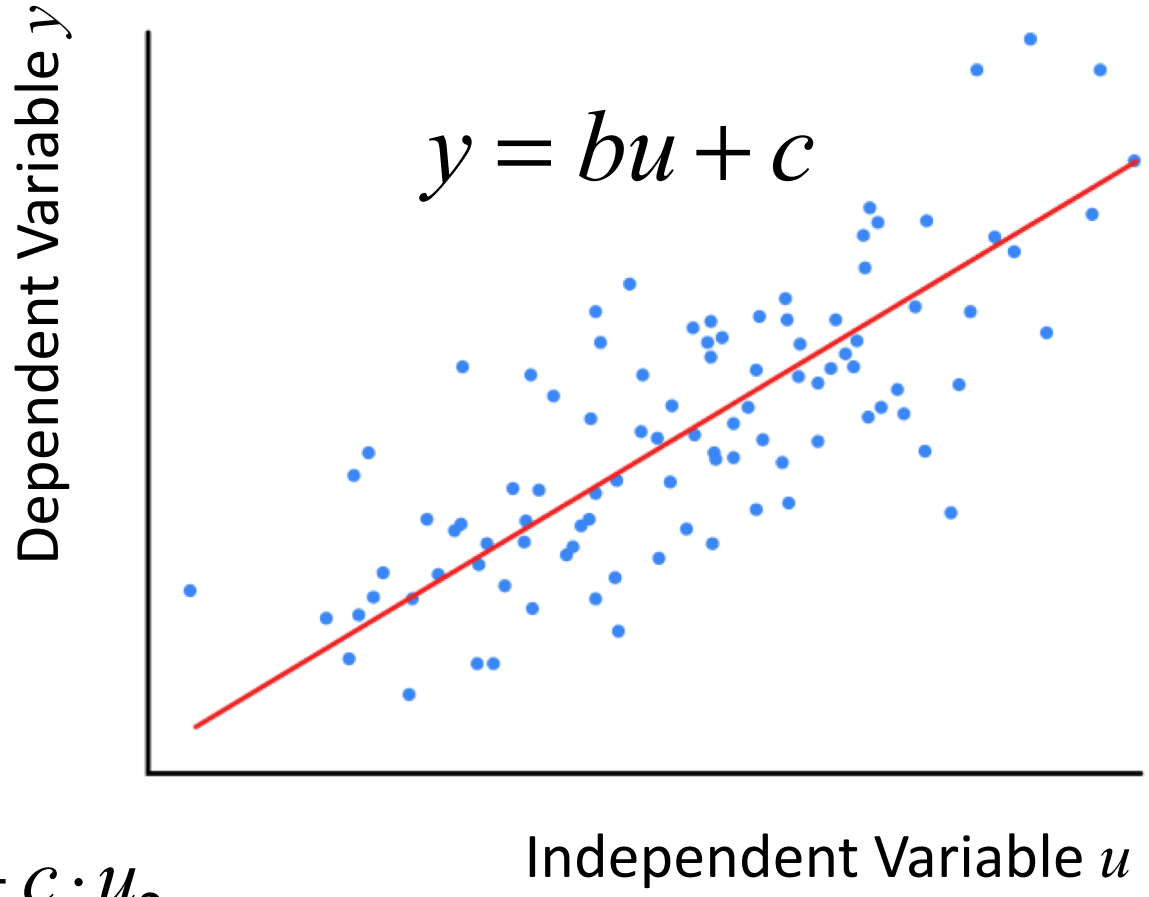
A statistical method for examining relationship between multiple variables. In particular, it is to analyze the influence of independent variables upon dependent variables.

y : Dependent Variable

u : Independent Variable

$$y = bu + c$$

Parameters to estimate



Homogeneous form

$$y = bu + c \cdot 1 \Rightarrow y = bu_1 + c \cdot u_2$$

A special case where $u_2 = 1$

Linear Regression

In general, consider a linear homogeneous equation

$$y = b_1 u_1 + b_2 u_2 + \cdots + b_m u_m \qquad (1)$$

where

$$\theta = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \in \Re^{m\times 1} \quad : \text{ Parameter vector}$$

$$\varphi = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} \in \Re^{m\times 1} \quad : \text{Regressor}$$

Problem:

Given a set of data

$$\varphi^{(1)}, y^{(1)}$$
$$\varphi^{(2)}, y^{(2)}$$
$$\vdots$$
$$\varphi^{(N)}, y^{(N)}$$

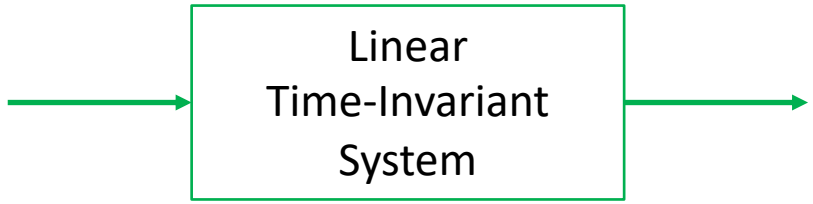
Find parameter vector θ

Dependent variable y is given as an inner product between θ and φ :

$$y = \theta^T \varphi \quad (= \varphi^T \theta) \quad : \text{ Linear Regression}$$

Remark

Eq. (1) looks a static relationship between input and output, but in fact it represents more general cases.



Example 1: Consider a discrete-time dynamical system:

$$y(t) = g_1 u(t-1) + g_2 u(t-2) + \cdots + g_m u(t-m)$$

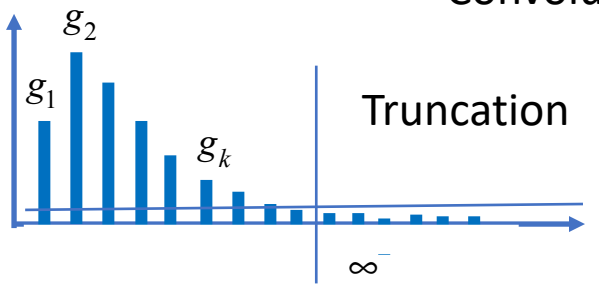
where

$$\theta = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \in \Re^{m\times 1} \qquad \varphi = \begin{pmatrix} u(t-1) \\ \vdots \\ u(t-m) \end{pmatrix} \in \Re^{m\times 1}$$

$$y(t) = \int_0^\infty g(\tau)u(t-\tau)\,d\tau$$

Convolution

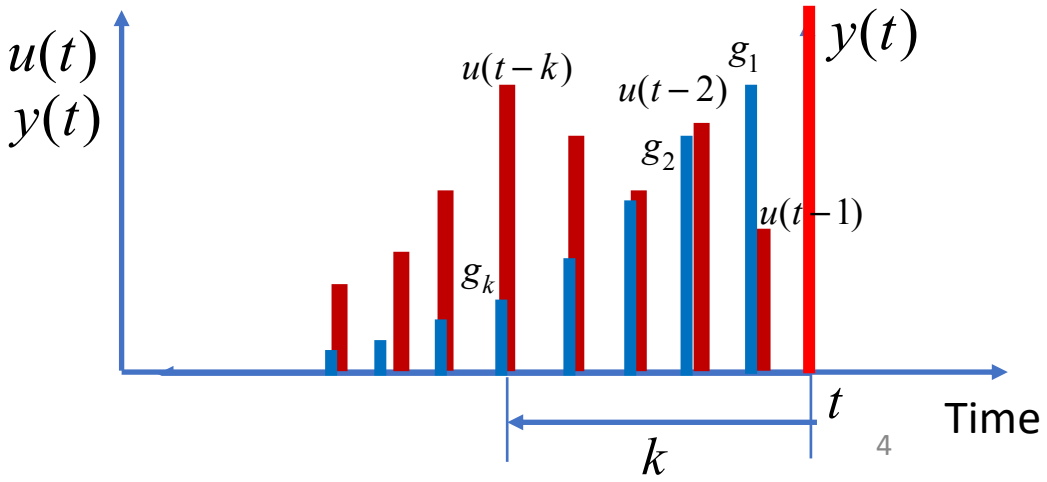
Impulse Response



$$y(t) = \sum_{k=0}^{\infty} g_k u(t-k)$$

This system is called Finite Impulse Response (FIR) System

$$y = \theta^T \varphi$$



"Given time-series input-output data, find the parameter values"

→ This is a typical System Identification Problem

Linear Regressions

Example 2: Consider a nonlinear algebraic system:

$$y(t) = b_1 x_1 + b_2 x_1^3 + b_3 x_1 x_2 + b_4 e^{-3x_1}$$

where

$$\varphi = \begin{pmatrix} x_1 \\ x_1^3 \\ x_1 x_2 \\ e^{-3x_1} \end{pmatrix} \in \Re^{4\times 1} \qquad \theta = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \in \Re^{4\times 1}$$

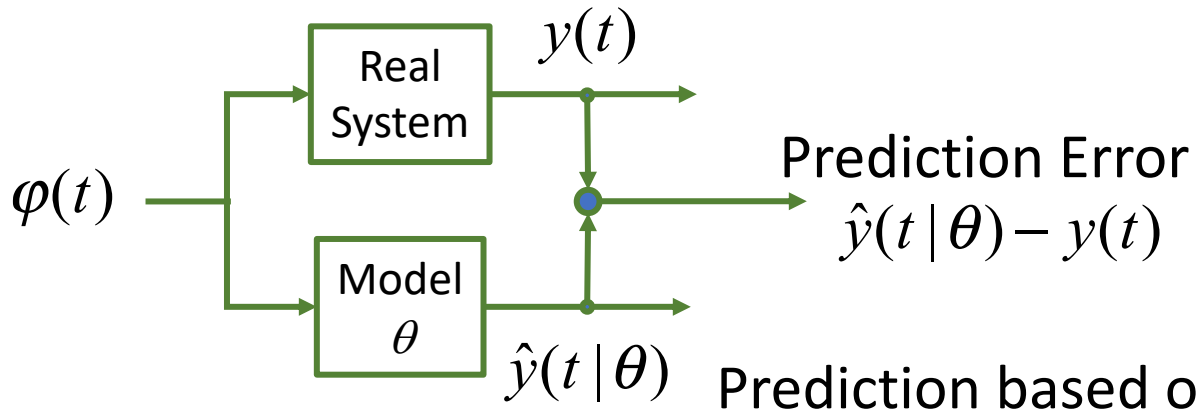
The nonlinear system can be represented as a linear regression model.

In PS #1, you will find other examples of nonlinear and/or dynamic systems, where unknown parameters are involved linearly in the governing equation. Obtaining the unknown parameters from data can be treated as a linear regression problem.

Least Squares Estimate (LSE)

For finding parameters from data, $y(t)$ and $\varphi(t)$

Prediction – Error Formalism



Mean Squared Error:

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}(\hat{y}(t|\theta) - y(t))^2$$

Least Squares Estimate (LSE) provides the parameter vector that minimizes the above Mean Squared Error:

$$\hat{\theta}^{LS} = \arg\min_{\theta} V_N(\theta)$$

Argument of function $V_N(\theta)$ that minimizes the value of the function.

Prediction Error Formalism

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}(\hat{y}(t\,|\,\theta) - y(t))^2$$

Broadly used in estimation, system identification, and machine learning

Solution The necessary conditions for V_N to take a minimum:

$$\frac{dV_N(\theta)}{d\theta} = 0$$: Differentiation of a scalar function V_N with respect to vector θ

$$\frac{\partial V_N(\theta)}{\partial \theta_i} = \frac{1}{N}\sum_{t=1}^{N}\frac{dX_t^2}{dX_t}\frac{\partial X_t}{\partial \theta_i} = \frac{2}{N}\sum_{t=1}^{N} X_t \frac{\partial \hat{y}(t\,|\,\theta)}{\partial \theta_i} \qquad V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} X_t^2 \quad \text{where} \quad X_t = \hat{y}(t\,|\,\theta) - y(t)$$

$$= \frac{2}{N}\sum_{t=1}^{N} X_t \varphi_i(t) \qquad : \quad \hat{y}(t\,|\,\theta) = \theta_1\varphi_1 + \cdots + \theta_i\varphi_i + \cdots + \theta_m\varphi_m$$

Repeating this partial derivative for all *i*'s yields:

$$\begin{pmatrix} \frac{\partial V_N(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial V_N(\theta)}{\partial \theta_i} \\ \vdots \\ \frac{\partial V_N(\theta)}{\partial \theta_m} \end{pmatrix} = \frac{2}{N}\sum_{t=1}^{N}(\hat{y}(t\,|\,\theta) - y(t)) \begin{pmatrix} \varphi_1(t) \\ \vdots \\ \varphi_i(t) \\ \vdots \\ \varphi_m(t) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} \Rightarrow \quad \therefore \sum_{t=1}^{N}(\hat{y}(t\,|\,\theta) - y(t))\varphi(t) = 0$$

Least Squares Estimate, Solution continued

$$\sum_{t=1}^{N}(\hat{y}(t\,|\,\theta)-y(t))\varphi(t)=0$$

Recall

$$\hat{y}(t\,|\,\theta)=\theta^T\cdot\varphi(t)=\varphi^T(t)\cdot\theta$$

$$\sum_{t=1}^{N}[\varphi^T(t)\cdot\theta]\varphi(t)=\sum_{t=1}^{N}y(t)\cdot\varphi(t)$$

$\varphi^T(t)\cdot\theta$ Is a scalar quantity. You can move it anywhere.

$$\left[\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right]\theta=\sum_{t=1}^{N}y(t)\cdot\varphi(t)$$

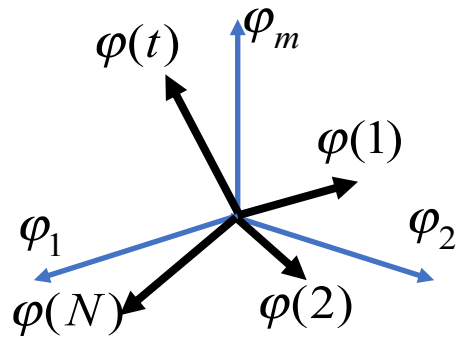
This is called Normal Equation.

Note $\sum\varphi(t)\varphi^T(t)$ is a m by m matrix. Assuming that it is non-singular,

$$\hat{\theta}^{LS}=\left[\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right]^{-1}\sum_{t=1}^{N}y(t)\cdot\varphi(t)$$

Remark 1

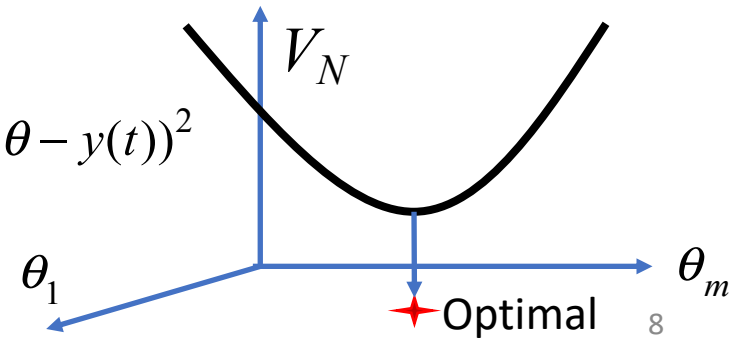
The regressor data $\varphi(1),\varphi(2),\cdots,\varphi(N)$ must contain a variety of vectors, spanning all the directions in the m-dimensional space.



Remark 2

When the m by m matrix is non-singular, then the squared error function V_N is convex, possessing a unique global minimum.

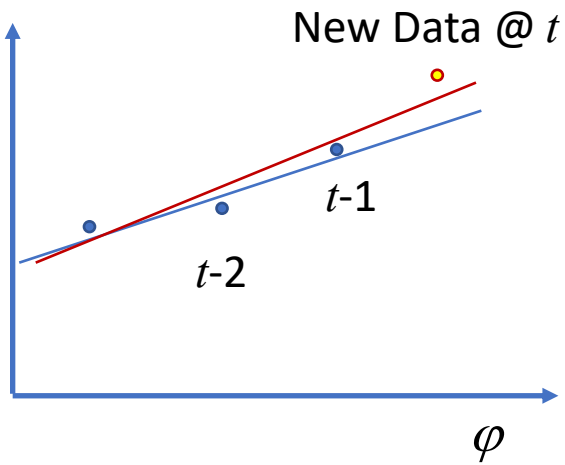
$$V_N(\theta)=\frac{1}{N}\sum_{t=1}^{N}(\varphi(t)^T\theta-y(t))^2$$



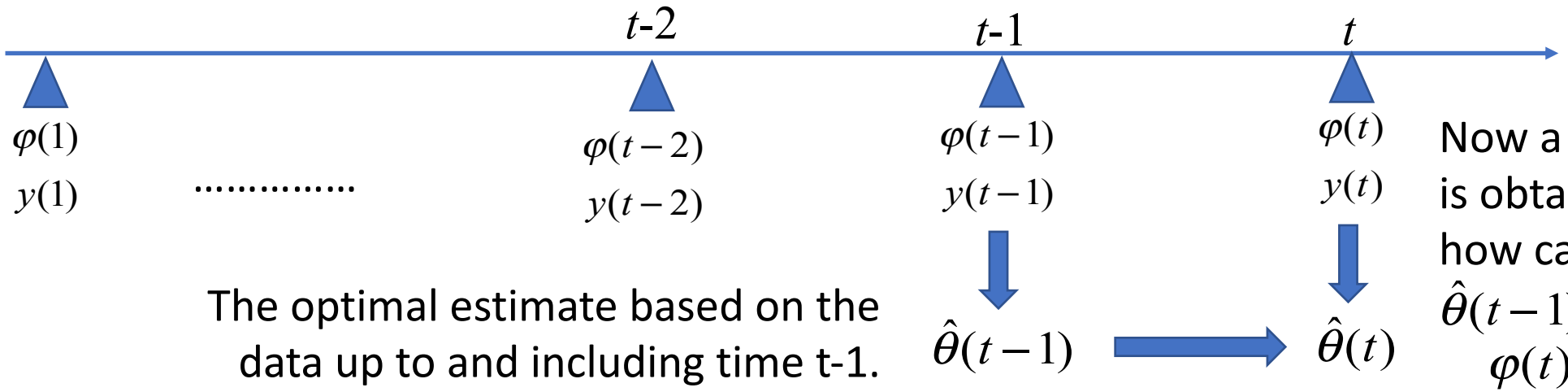
2.2 The Recursive Least Squares Algorithm

- The Least Squares Estimate we have obtained is an off-line batch processing algorithm after collecting all data.

$$\hat{\theta}^{LS} = \left[\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\right]^{-1} \sum_{t=1}^{N} y(t)\cdot\varphi(t)$$



- Suppose that data are observed in sequence, like Carl Friedrich Gauss did in his planetary observation every night.
- Rather than waiting until all the data are obtained, you want to obtain the optimal estimate based on the data you have obtained so far.



The optimal estimate based on the data up to and including time t-1.

Now a new observation is obtained at time t, how can we update $\hat{\theta}(t-1)$ by using $\varphi(t)$ and $y(t)$?

If the optimal estimate $\hat{\theta}(t)$ depends only on $\hat{\theta}(t-1)$ and new data $\varphi(t)$, $y(t)$, we can forget old data.

Find a recursive formula: $\hat{\theta}^{LS}(t) = \hat{\theta}^{LS}(t-1) + [Correction\ based\ on\ \varphi(t), y(t)]$

Recursive Least Squares (Continued)

$$\hat{\theta}^{LS}(t) = \hat{\theta}^{LS}(t-1) + [Correction\ based\ on\ \varphi(t), y(t)]$$

This must be the same as the optimal solution $\hat{\theta}^{LS} = \left[\sum_{i=1}^{t} \varphi(i)\varphi^T(i)\right]^{-1} \sum_{i=1}^{t} y(i)\cdot\varphi(i)$

Let us derive the recursive formula to update the estimate with new observation $\varphi(t)$ and $y(t)$

Step 1 We assume that enough data are initially available so that the matrix $\sum_{i=1}^{t} \varphi(i)\varphi^T(i)$ is nonsingular.

Define $P_t = \left[\sum_{i=1}^{t} \varphi(i)\varphi^T(i)\right]^{-1}$ and $B_t = \sum_{i=1}^{t} y(i)\cdot\varphi(i)$ So that $\hat{\theta}^{LS} = P_t B_t$

Split each of these into the new observation and the one from the previous step.

$$B_t = \sum_{i=1}^{t-1} y(i)\cdot\varphi(i) + y(t)\cdot\varphi(t) = B_{t-1} + y(t)\cdot\varphi(t)$$

$$P_t^{-1} = \sum_{i=1}^{t} \varphi(i)\varphi^T(i) = \sum_{i=1}^{t-1} \varphi(i)\varphi^T(i) + \varphi(t)\varphi^T(t) = P_{t-1}^{-1} + \varphi(t)\varphi^T(t) \quad (12)$$

Update of B_t is straightforward, but P_t is different. The inverse of the m by m matrix must be computed. Do we need to take matrix inversion in every recursive step of computation?

$$P_t^{-1} = P_{t-1}^{-1} + \varphi(t)\varphi^T(t)$$

Step 2. The Matrix Inversion Lemma

Can we compute P_t recursively without taking matrix inversion? Yes, we can.

Pre-multiply P_t and post-multiply P_{t-1} to eq.(12):

$$P_t P_t^{-1} P_{t-1} = P_t P_{t-1}^{-1} P_{t-1} + P_t \varphi(t)\varphi^T(t) P_{t-1} \qquad (13)$$

Further post-multiply $\varphi(t)$

$$P_{t-1}\varphi(t) = P_t\varphi(t) + P_t \varphi(t)\varphi^T(t) P_{t-1}\varphi(t) = P_t\varphi(t)(1+\varphi^T(t) P_{t-1}\varphi(t))$$

Note that $1+\varphi^T(t) P_{t-1}\varphi(t)$ is a scalar quantity. Divide both sides by $1+\varphi^T(t) P_{t-1}\varphi(t) \neq 0$

$$P_t\varphi(t) = \frac{P_{t-1}\varphi(t)}{1+\varphi^T(t) P_{t-1}\varphi(t)}$$

Post-multiplying $\varphi^T(t) P_{t-1}$ to left-hand side and using (13) yield $P_t \varphi(t)\varphi^T(t) P_{t-1} = P_{t-1} - P_t$

$$\therefore \quad P_t = P_{t-1} - \frac{P_{t-1}\varphi(t)\varphi^T(t) P_{t-1}}{1+\varphi^T(t) P_{t-1}\varphi(t)} \qquad (14)$$

The right-had side is computed with P_{t-1} and $\varphi(t)$ alone.

$P_t = \left[\sum_{i=1}^{t} \varphi(i)\varphi^T(i)\right]^{-1}$ the inverse of a m by m matrix is computed without taking matrix inversion.

Division is only with a scalar quantity.
Much faster to compute than the standard matrix inversion.

$$P_t = P_{t-1} - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{1+\varphi^T(t)P_{t-1}\varphi(t)} \qquad (14)$$

This is a special case of the Matrix Inversion Lemma.
Max Woodbury (1950) has extended it to a general case.

General Matrix Inversion Lemma:
$A, B, C,$ and D are arbitrary matrices with consistent dimensions.

$$[A+BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B+C^{-1}]^{-1}DA^{-1}$$

$A \to P_{t-1}^{-1}$, $B \to \varphi(t)$, $C \to 1$, $D \to \varphi^T(t)$, $A^{-1} \to P_{t-1}$

This agrees with (14).
Check it on your own.

Step 3 Reduce $\hat{\theta}^{LS} = P_t B_t$ to a recursive formula

We can show that the recursive formula is given by the following form

$$\hat{\theta}^{LS}(t) = \hat{\theta}^{LS}(t-1) + K_t[\, y(t) - \hat{y}(t \mid \hat{\theta}^{LS}(t-1))]$$

Prediction Error: Negative feedback

An optimal gain for correcting the estimate

Goal:

1) Show that $\hat{\theta}^{LS} = P_t B_t$ can be written in the above recursive form; and
2) Find the optimal gain K_t .

By definition

$$\hat{\theta}^{LS}(t) - \hat{\theta}^{LS}(t-1) = P_t B_t - P_{t-1} B_{t-1}$$

From Step 1 and Step 2

$$= \left(P_{t-1} - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{1+\varphi^T(t)P_{t-1}\varphi(t)} \right)(B_{t-1} + y(t)\cdot\varphi(t)) - P_{t-1}B_{t-1}$$

$P_{t-1}B_{t-1}$ cancels

$$= P_{t-1}y\varphi - \frac{P_{t-1}\varphi\varphi^T P_{t-1}}{1+\varphi^T P_{t-1}\varphi}(B_{t-1} + y\cdot\varphi)$$

Omitting (t)

$$\hat{\theta}^{LS}(t)-\hat{\theta}^{LS}(t-1)=P_{t-1}y\varphi-\frac{P_{t-1}\varphi\varphi^T P_{t-1}}{1+\varphi^T P_{t-1}\varphi}(B_{t-1}+y\cdot\varphi)$$

Scalar y can be moved

$$=\frac{P_{t-1}y\varphi+P_{t-1}y\varphi\varphi^T P_{t-1}\varphi-P_{t-1}\varphi\varphi^T P_{t-1}B_{t-1}-P_{t-1}\varphi\varphi^T P_{t-1}y\cdot\varphi}{1+\varphi^T P_{t-1}\varphi}$$

Factoring out $P_{t-1}\varphi$

$$=\frac{P_{t-1}\varphi}{1+\varphi^T P_{t-1}\varphi}[\,y(t)-\varphi^T P_{t-1}B_{t-1}]$$

$P_{t-1}B_{t-1}=\hat{\theta}^{LS}(t-1)$

$$=\frac{P_{t-1}\varphi(t)}{1+\varphi^T(t)P_{t-1}\varphi(t)}[\,y(t)-\varphi^T(t)\hat{\theta}^{LS}(t-1)]$$

We have obtained the Recursive Least Squares Algorithm

$y(t)-\varphi^T(t)\hat{\theta}^{LS}(t-1)$ Prediction error

$$\hat{\theta}^{LS}(t)=\hat{\theta}^{LS}(t-1)+K_t[\,y(t)-\hat{y}(t\,|\,\hat{\theta}^{LS}(t-1))]$$

where

$$K_t=\frac{P_{t-1}\varphi(t)}{1+\varphi^T(t)P_{t-1}\varphi(t)}$$

$$P_t=P_{t-1}-\frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{1+\varphi^T(t)P_{t-1}\varphi(t)}$$

The Recursive Least Squares Algorithm

$$\hat{\theta}^{LS}(t) = \hat{\theta}^{LS}(t-1) + K_t[\, y(t) - \hat{y}(t \mid \hat{\theta}^{LS}(t-1))]$$

where $$K_t = \frac{P_{t-1}\varphi(t)}{1+\varphi^T(t)P_{t-1}\varphi(t)} \qquad t = 1,2,\cdots$$

$$P_t = P_{t-1} - \frac{P_{t-1}\varphi(t)\varphi^T(t)P_{t-1}}{1+\varphi^T(t)P_{t-1}\varphi(t)}$$

Initial conditions:

$\hat{\theta}^{LS}(0) = \hat{\theta}_0 =$ Arbitrary, e.g. $\hat{\theta}_0 = 0$

$P_0 =$ Positive Definite Matrix, e.g. $P_0 = I$ Identity Matrix

Carl Friedrich Gauss discovered the Recursive Least Squares Algorithm in 1821.