Lu Sun, and many more.

A Notebook on Control System

To my family, friends and communities members who have been dedicating to the presentation of this notebook, and to all students, researchers and faculty members who might find this notebook helpful.

Contents

Foreword

If software or e-books can be made completely open-source, why not a notebook?

This brings me back to the summer of 2009 when I started my third year as a high school student in Harbin No. 3 High School. In around the end of August when the results of Gaokao (National College Entrance Examination of China, annually held in July) are released, people from photocopy shops would start selling notebooks photocopies that they claim to be from the top scorers of the exam. Much curious as I was about what these notebooks look like, never have I expected myself to actually learn anything from them, mainly for the following three reasons.

First of all, some (in fact many) of these notebooks were more difficult to understand than the textbooks. I guess we cannot blame the top scorers for being so smart that they sometimes make things extremely brief or overwhelmingly complicated.

Secondly, why would I want to adapt to notebooks of others when I had my own notebooks which in my opinion should be just as good as theirs.

And lastly, as a student in the top-tier high school myself, I knew that the top scorers of the coming year would probably be a schoolmate or a classmate. Why would I want to pay that much money to a complete stranger in a photocopy shop for my friend's notebook, rather than requesting a copy from him or her directly?

However, things had changed after my becoming an undergraduate student in 2010. There were so many modules and materials to learn in a university, and as an unfortunate result, students were often distracted from digging deeply into a module (For those who were still able to do so, you have my highest respect). The situation became even worse as I started pursuing my Ph.D. in 2014. As I had to focus on specific research areas entirely, I could hardly split much time on other irrelevant but still important and interesting contents.

This motivated me to start reading and taking notebooks for selected books and articles, just to force myself to spent time learning new subjects out of my comfort zone. I used to take hand-written notebooks. My very first notebook was on *Numerical Analysis*, an entrance level module for engineering background graduate students. Till today I still have on my hand dozens of these notebooks. Eventually, one day it suddenly came to me: why not digitalize them, and make them accessible online and open-source, and let everyone read and edit it?

—

As most of the open-source software, this notebook (and it applies to the other notebooks in this series as well) does not come with any "warranty" of any kind, meaning that there is no guarantee for the statement and knowledge in this notebook to be absolutely correct as it is not peer reviewed. **Do NOT cite this notebook in your academic research paper or book!** Of course, if you find anything helpful with your research, please trace back to the origin of the citation and double confirm it yourself, then on top of that determine whether or not to use it in your research.

This notebook is suitable as:

- a quick reference guide;
- a brief introduction for beginners of the module;
- a "cheat sheet" for students to prepare for the exam (Don't bring it to the exam unless it is allowed by your lecturer!) or for lecturers to prepare the teaching materials.

This notebook is NOT suitable as:

- a direct research reference;
- a replacement to the textbook;

because as explained the notebook is NOT peer reviewed and it is meant to be simple and easy to read. It is not necessary brief, but all the tedious explanation and derivation, if any, shall be "fold into appendix" and a reader can easily skip those things without any interruption to the reading experience.

—

Although this notebook is open-source, the reference materials of this notebook, including textbooks, journal papers, conference proceedings, etc., may not be open-source. Very likely many of these reference materials are licensed or copyrighted. Please legitimately access these materials and properly use them.

Some of the figures in this notebook is drawn using Excalidraw, a very interesting tool for machine to emulate hand-writing. The Excalidraw project can be found in GitHub, *excalidraw/excalidraw*.

Preface

Control System

List of Figures

List of Tables

Part I

Classic Control System

1

Classic Control System

CONTENTS

2

Multivariable Control System

CONTENTS

Part II

Modern Control System

3

Modern Control System

CONTENTS

4

Computer Controlled System

CONTENTS

Part III

Advanced Control System

5

Optimal and Robust Control System

CONTENTS

6

Model Predictive Control System

CONTENTS

7

Adaptive Control System

CONTENTS

Adaptive controller is generally referred to a controller with adjustable parameters and associated mechanism to adjust such parameters, to conform to new circumstances.

Comparing with most of the other control systems, an adaptive control system does not require a very precise modeling of the plant. More precisely, an adaptive control system can adapt the control gains to the unknown or time-varying plant and maintain the stability of the closed-loop system.

The main references of this chapter include:

- Åström, K.J. and Wittenmark, B., 2013. Adaptive control. Courier Corporation [1].

7.1 Introduction

Adaptive controller is generally referred to a controller with adjustable parameters and associated mechanism to adjust such parameters, to conform to new circumstances.

Due to the parameter adjustment mechanism, the controller becomes nonlinear. For example, while a conventional PID controller is linear, an adaptive PID controller is nonlinear since the $P(t)$, $I(t)$ and $D(t)$ are variables instead of constant gains.

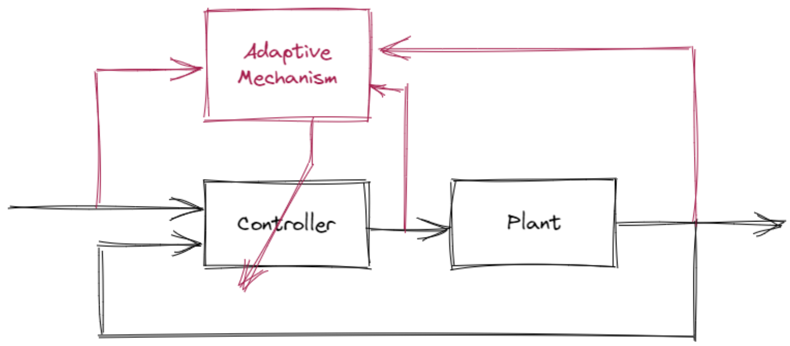

FIGURE 7.1
Adaptive control system general schema.

In practice, an adaptive controller is often a modified version of a conventional controller just like the adaptive PID controller example. There are often 2 types of feedback loops in an adaptive control system: the original loops that comes with the conventional controller, and the additional loops to adjust the parameters of the conventional controller. A demonstrative figure is given in Fig. 7.1.

7.1.1 A Brief History of Adaptive Control System

The first adaptive control system application was designed for autopilots of a high-performance aircraft, where the conventional control system was found to work for one condition but not for the other. Gain scheduling was adopted to solve this problem.

Later on more sophisticated adaptive control systems that utilizes state-space model and stability theory were introduced. Stochastic control theory and system identification techniques were developed side-by-side.

The proof of stability of the adaptive control system, especially universal stability of the system, started to draw attention. The proof can be done often only with strong restrictions. People started to investigate the connection and difference of robust control and system identification with adaptive control and system identification. Computer controlled system and artificial neural network started to emerge and people realized the connection between adaptive control and computer learning.

Adaptive control system commercialization started in 1980s, and are used in handling process dynamics and disturbances, and to provide automatic tuning of controller parameters.

7.1.2 Conventional Control System

The most conventional way of designing a control system is as follows.

1. Assume a linear model of the plant at the operating point.
2. Calculate or estimate the parameters of the plant model.
3. Design a closed-loop control system for the plant.

The control system designed following the above approach is usually intrinsically insensitive to modeling error and disturbance to some extend (due to the closed-loop implementation). But there are difficulties that could cause variation and troubles.

- **Nonlinear actuator** is one of the sources of modeling uncertainty, making the control system work only on/near the pre-determined operating point, but not universally.
- **Flow and speed variations** in the pipes of a system, if exist, can change from time to time. The different flow and speed can cause changes in the process dynamics, thus change the plant model.
- **Environmental dynamics and uncertainties** often draw significant concerns in control. For example, the aircraft flying at different height and speed suffers from different environmental conditions. The same applies to ship steering at different speed.

7.1.3 Adaptive Control Schema

Different adaptive control schemes have been designed to tackle the above issues. The commonly used ones are briefly introduced here.

Gain Scheduling

Measurement feedback may correlate well with changes in the process dynamics. The idea of gain scheduling is to map the process dynamic parameters with the control parameters, by either a function manner or a table lookup manner, i.e., schedule the control parameters to compensate the the process dynamics.

A demonstrative figure is given in Fig. 7.2.

Gain scheduling is one of the most intuitive adaptive control schema, and has the earliest use case among all.

Model-Reference Adaptive System (MRAS)

In this scenario, a reference model is built that generates the "ideal" output of the closed-loop system given the reference point. This is often a virtual model that does not suffer from disturbances and error.

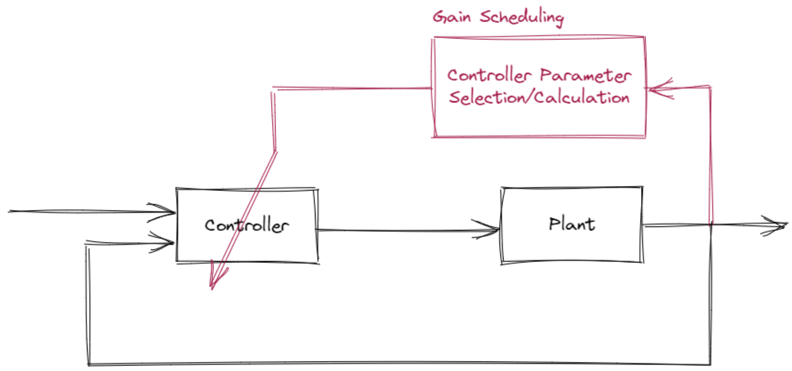

FIGURE 7.2
Gain scheduling schema.

The MRAS then uses the output of both the actual plant and the reference model to calculate the controller parameter adjustment mechanism to minimize the difference between the outputs of the two models.

A demonstrative figure is given in Fig. 7.3.

The MRAS is somewhat like a learning system. Usually, there is a "adaptation rate" (just like the learning rate) that controls the adapting speed.

Self-Tuning Regulator (STR)

The idea of STR is to estimate the system process and design and change the whole control system in real-time. Unlike the gain scheduling approach and MRAS where parameter adjustment mechanism is calculated in real-time, in STR approach, the controller designing problem is solved in real-time. From this sense, STR can be taken as the "high-end" gain scheduling problem.

A demonstrative figure is given in Fig. 7.4.

STR can be made very flexible, because it is essentially a new control system design from scratch each time the environment changes.

Dual Control

The aforementioned adaptive control schemas do not taken into account the stochastics and statistics of the measurement and estimation uncertainty.

To analyze the performance of the system under estimation uncertainty and to develop control algorithm that can optimize the system performance under such uncertainty, nonlinear stochastic control theory needs to be used, which leads to the notion of dual control.

A demonstrative figure is given in Fig. 7.5.

This control system is essentially a nonlinear controller with a robust/adaptive filter that estimates the process dynamics and the states of the plant, together with statistics information of the estimates.

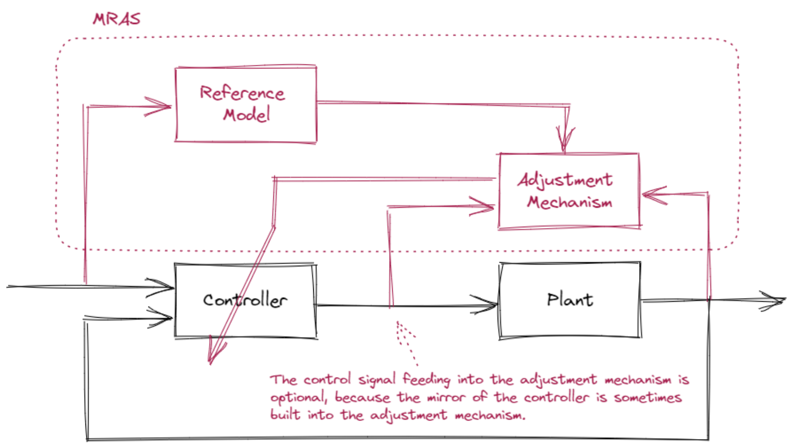

FIGURE 7.3
MARS schema.

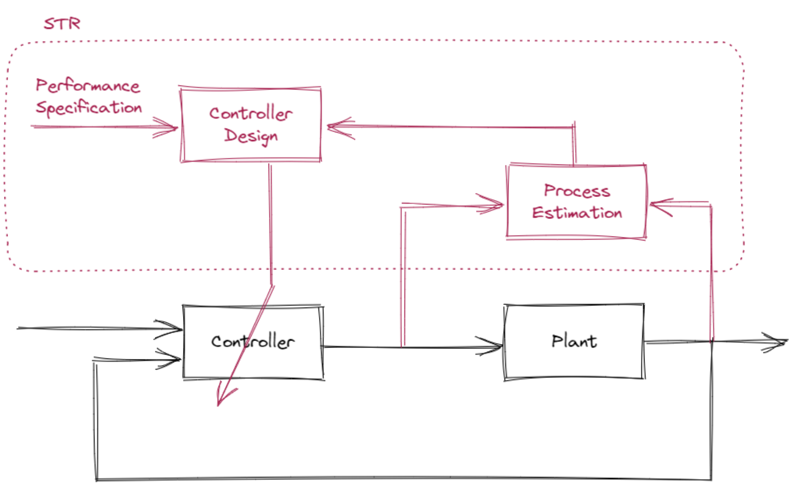

FIGURE 7.4
STR schema.

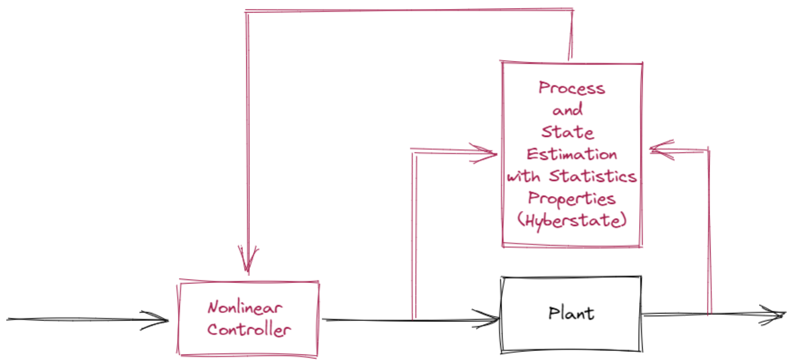

FIGURE 7.5
Dual control schema.

7.1.4 A Typical Adaptive Control Problem Formulation

For simplicity, consider the following LTI plant realizations.

Continuous time domain state space model:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

Continuous time domain transfer function model:

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \ldots + b_m}{s^n + a_1 s^{n-1} + \ldots + a_n}$$

Continuous time domain input-output model (with p the differential operator, $p = \frac{d}{dt}$):

$$y(t) = G(p)u(t)$$

Discrete time domain state-space model:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned}$$

Discrete time domain transfer function:

$$G(z) = \frac{B(z)}{A(z)} = \frac{b_0 z^m + b_q z^{m-1} + \ldots + b_m}{z^n + a_1 z^{n-1} + \ldots + a_n}$$

Discrete time domain input-output model (with q the forward shift operator, $qy(t) = y(t+1)$):

$$y(k) = H(q)u(k)$$

The above models may be describing the same plant, with different level of details. For example, when dealing with transfer function and input-output models, the initial condition of the system is often ignored. When dealing with discrete time domain plant model, it is open a sampled system derived from the original system.

If the parameters in the models are known, it is straight forward to design a closed-loop controller for them. This is called the underlying design problem. The adaptive control problem, on the other hand, is to design a controller without knowing the parameters. In response, the controller comes with adjustable control gains and associated adjustment mechanism.

A typical adaptive controller design shall include the following procedures:

1. Characterize the desired closed-loop behavior.
2. Determine a suitable control law with adjustable parameters.
3. Design adjustment mechanism.
4. Implement the control law with the parameters adjusted using the adjustment mechanism.

Some successful use cases for adaptive control systems include

- Automatic tuning controller, such as the PID automatic tuning controller. In practice, all conventional control systems can benefit from automatic parameter tuning, if done correctly.
- Gain scheduling, which is the standard technique used in flight control system for high-performance aircraft, and is becoming increasingly popular in industrial process control.
- Continuous adaptation control for continuously varying system.
- "Human-in-the-loop" control system.

Since adaptive control system is often more costly and complicated than a conventional constant-gain controller, it should be used wisely and not abused. If the dynamics of the system are manageable, consider using parametric model based control algorithm rather than the adaptive control system, the later of which is usually more complicated.

7.2 Parameter Estimation

Parameter estimation (or more generally, system identification) is widely used, either directly or indirectly, in many adaptive control schemas. An obvious example is STR, where the process dynamics estimation is built into the controller. Parameter estimation also happens implicitly in MRAS.

The key factors in system identification include the following.

- Selection of model structure.
- Experiment design.
- Parameter and state estimation.
- Validation.

One thing to notice is that in adaptive control scope, the plant is assumed to change continuously, thus the parameter estimation needs to be done in a recursive manner.

7.2.1 Least Squares Estimation

Karl Friedrich Gauss defines LS estimation problem as follows.

LS estimation:

The parameters in the model should be chose such that the sum of the squares of the differences between the observed and the computed values, multiplied by numbers that measure the degree of precision, is a minimum.

Notice that from the above definition, if all observations share the same degree of precision, the multipliers to all observations shall be the same, otherwise they should be different. Conventionally, the later case is referred as "Weighted LS (WLS) estimation" where each multiplier is called the "weight" associated with the observation. The more precise an observation, the higher its associated weight. On the other hand, the former case where all observations share the same precision is referred as LS estimation.

To reduce confusion, from now on LS estimation and WLS estimation are used to refer to the former and later cases, respectively.

Formulate LS estimation problem on a LTI system as follows. Let $\theta_i, i = 1, ..., n$ be n unknown parameters in the model and $y(j), j = 1, ..., m$ be m measurements collected from the mode. It is required that $m \geq n$. Variable $\phi_i(j)$ is called a regression variable (or a regressor) that links the measurements with the model parameters, so that

$$y(j) \quad = \quad \phi_1(j)\theta_1 + \phi_2(j)\theta_2 + \ldots + \phi_n(j)\theta_n + \epsilon(j) \tag{7.1}$$

where $\epsilon(j)$ is assumed to be independent measurement noise associated with

$y(j)$. Equation (7.1) is called a regression model. Putting (7.1) into matrix form gives

$$y = \Phi\theta + \epsilon \tag{7.2}$$

where

$$\begin{aligned} y &= \begin{bmatrix} y(1) & \dots & y(m) \end{bmatrix}^T \\ \Phi &= \begin{bmatrix} \phi_1(1) & \dots & \phi_n(1) \\ \vdots & \ddots & \vdots \\ \phi_1(m) & \dots & \phi_n(m) \end{bmatrix} \\ \theta &= \begin{bmatrix} \theta_1 & \dots & \theta_n \end{bmatrix}^T \\ \epsilon &= \begin{bmatrix} \epsilon(1) & \dots & \epsilon(m) \end{bmatrix}^T \end{aligned}$$

with y the measurement vector, Φ the regression model matrix, and θ the unknown system parameter vector.

LS estimation of θ in (7.2) is formulated as follows. Select estimation $\hat{\theta} = [\hat{\theta}_1, ..., \hat{\theta}_n]^T$ so that the following cost function $V(\hat{\theta})$ is minimized.

$$V(\hat{\theta}) = \frac{1}{2}\sum_{i=1}^{m} e(i)^2 \tag{7.3}$$

where

$$\begin{aligned} e(i) &= y(i) - \Big(\phi_1(i)\hat{\theta}_1 + \phi_2(i)\hat{\theta}_2 + \dots + \phi_n(i)\hat{\theta}_n\Big) \\ &= y(i) - \phi(i)^T\hat{\theta} \end{aligned} \tag{7.4}$$

with

$$\phi(i)^T = \begin{bmatrix} \phi_1(i) & \dots & \phi_n(i) \end{bmatrix}^T$$

which is the ith row of Φ. Variable $e(i)$ is called the residual.

Put everything into matrix format. Equation (7.3) becomes

$$\begin{aligned} V(\hat{\theta}) &= \frac{1}{2}\sum_{i=1}^{m}\Big(y(i) - \phi(i)^T\hat{\theta}\Big)^2 \\ &= \frac{1}{2}(y - \Phi\hat{\theta})^T(y - \Phi\hat{\theta}) \end{aligned} \tag{7.5}$$

Denote $e = [e(1), ..., e(m)]^T$. From (7.4),

$$e = y - \Phi\hat{\theta} \tag{7.6}$$

and (7.5) becomes

$$V(\hat{\theta}) = \frac{1}{2}e^Te \tag{7.7}$$

The quadratic optimization problem given in (7.7) has the following unique analytical solution. Differentiating (7.7) w.r.t. θ gives (using numerator-layout notation)

$$\frac{dV(\hat{\theta})}{d\theta} = \frac{dV(\hat{\theta})}{de}\frac{de}{d\theta} \tag{7.8}$$

$$= -e^T\Phi \tag{7.9}$$

Equating (7.9) to zero gives the minimum of (7.7) as follows.

$$-e^T\Phi = 0 \tag{7.10}$$

Substituting (7.6) into (7.10) gives

$$\begin{aligned} -\left(y - \Phi\hat{\theta}\right)^T \Phi &= 0 \\ -\Phi^T \left(y - \Phi\hat{\theta}\right) &= 0 \\ \Phi^T\Phi\hat{\theta} &= \Phi^T y \\ \hat{\theta} &= \left(\Phi^T\Phi\right)^{-1}\Phi^T y \end{aligned} \tag{7.11}$$

provided that $\Phi^T\Phi$ is nonsingular, which is known as the excitation condition. Equation (7.11) is the solution to the LS estimation problem in (7.3).

A Quick Note on Matrix Differentiation

In numerator-layout notion, the scalar $V(\hat{\theta})$ differentiated w.r.t. the vector e in (7.8) is given by

$$\begin{aligned} \frac{dV(\hat{\theta})}{de} &= \begin{bmatrix} \frac{dV(\hat{\theta})}{de(1)} & \dots & \frac{dV(\hat{\theta})}{de(m)} \end{bmatrix} \\ &= \begin{bmatrix} e(1) & \dots & e(m) \end{bmatrix} \\ &= e^T \end{aligned}$$

The vector e differentiated w.r.t. the vector θ is given by

$$\begin{aligned} \frac{de}{d\theta} &= \begin{bmatrix} \frac{de(1)}{d\theta_1} & \dots & \frac{de(m)}{d\theta_n} \\ \vdots & \ddots & \vdots \\ \frac{de(m)}{d\theta_1} & \dots & \frac{de(m)}{d\theta_n} \end{bmatrix} \\ &= \begin{bmatrix} -\phi_1(1) & \dots & -\phi_n(1) \\ \vdots & \ddots & \vdots \\ -\phi_1(m) & \dots & -\phi_n(m) \end{bmatrix} \\ &= -\Phi \end{aligned}$$

where

$$\frac{de(i)}{d\theta_j} = -\phi_j(i)$$

because of (7.4).

Therefore,

$$\frac{dV(\hat{\theta})}{d\theta} = \frac{dV(\hat{\theta})}{de}\frac{de}{d\theta} = -e^T\Phi$$

which is (7.9).

A WLS estimation problem can be formulated similarly, by changing (7.3) with

$$V(\hat{\theta}) = \frac{1}{2}\sum_{i=1}^{m} w(i)e(i)^2$$

The solution to a WLS estimation can be similarly derived and the result is

$$\hat{\theta} = \left(\Phi^T W \Phi\right)^{-1} \Phi^T W y$$

where $W = \text{diag}(w(1), ..., w(m))$.

7.2.2 Statistics Properties of LS Estimation

In the discussions below, We will assume that all measurement noise $\epsilon(i), i = 1, ..., m$ are i.i.d. noise, whose mean and variance are zero and σ^2 respectively.

Substituting (7.2) into (7.11) gives

$$\begin{aligned} \hat{\theta} &= \left(\Phi^T\Phi\right)^{-1} \Phi^T \left(\Phi\theta + \epsilon\right) \\ &= \theta + \left(\Phi^T\Phi\right)^{-1} \Phi^T \epsilon \end{aligned} \tag{7.12}$$

Using (7.12), the following statistics properties of LS estimation can be drawn.

$$\begin{aligned} E\left[\hat{\theta}\right] &= \theta \\ \text{Cov}\left[\hat{\theta}\right] &= E\left[\hat{\theta}\hat{\theta}^T\right] \\ &= \sigma^2\left(\Phi^T\Phi\right)^{-1} \end{aligned}$$

which suggests that the estimator is zero mean, and the variance of the noise σ^2 can potentially be estimated from the state estimate $\hat{\theta}$. Details are given as follows. Substituting (7.2), (7.11) into (7.6) gives

$$e = \left(I - \Phi\left(\Phi^T\Phi\right)^{-1}\Phi^T\right)\epsilon \tag{7.13}$$

Using (7.13),

$$E\left[ee^T\right] = \sigma^2\left(I-\Phi\left(\Phi^T\Phi\right)^{-1}\Phi^T\right)\left(I-\Phi\left(\Phi^T\Phi\right)^{-1}\Phi^T\right)^T \quad (7.14)$$

Taking the trace of matrix from both sides of (7.14) gives

$$E\left[e^Te\right] = (m-n)\sigma^2 \quad (7.15)$$

From (7.7) and (7.15),

$$\sigma^2 = \frac{2}{m-n}E\left[V(\hat{\theta})\right]$$

can be used to estimate the variance of the noise.

Part IV

System Identification

8

Classic System Identification and State Estimation

CONTENTS

9

Optimal and Robust State Estimation

CONTENTS

10

Adaptive State Estimation

CONTENTS

Part V

Artificial Intelligence

11

Artificial Neural Network

CONTENTS

Part VI

Advanced Topics

12

Fuzzy Control System

CONTENTS

13

Multi-Agent Control System

CONTENTS

Bibliography

[1] Karl J Åström and Björn Wittenmark. *Adaptive control*. Courier Corporation, 2013.