System Identification and Robust Control

Lecture 1: Introduction

Eugenio Schuster

schuster@lehigh.edu
Mechanical Engineering and Mechanics
Lehigh University

SYSTEM IDENTIFICATION
Theory for the User (Second Edition)

Lennart Ljung
Linköping University, Sweden

Content

1. Time-invariant Linear Systems [Chapter 2]
2. Nonparametric Time and Frequency Domain Methods [Chapter 6]
3. Simulation and Prediction [Chapter 3]
4. Models for Linear Time-invariant Systems [Chapter 4]
5. Parameter Estimation Methods [Chapter 7]
6. Convergence and Consistency [Chapter 8]
7. Computing the Estimate [Chapter 9]
8. Recursive Estimation Methods [Chapter 10]

Another Books

- T. Söderström and P. Stoica, System Identification, Prentice Hall, 1989.

Book - Robust Control

MULTIVARIABLE FEEDBACK CONTROL
Analysis and design

Sigurd Skogestad
Norwegian University of Science and Technology
Ian Postlethwaite
University of Leicester

Content

1. Introduction [Chapter 1]
2. Classical feedback control [Chapter 2]
3. Performance limitations SISO [Chapter 5]
4. Uncertainty and robustness SISO [Chapter 7]
5. Elements of linear systems theory [Chapter 4]
6. Introduction to multivariable control [Chapter 3]
7. Performance limitations MIMO [Chapter 6]
8. Robust stability and performance MIMO [Chapter 8]
9. Controller design [Chapter 9]

Another Books

- C. Chen, Linear System Theory and Design, Oxford University Press, 1999.
- G. Dullerud and F. Paganini, A Course in Robust Control Theory, Springer, 2000.
- M. Green and D. Limebeer, Linear Robust Control, Prentice Hall, 1995.
- J. Helton and O. Merino, Classical Control Using H-infinity Methods, SIAM, 1998.
- J. Maciejowski, Multivariable Feedback Design, Addison Wesley, 1989.
- R. Sanchez-Pena and M. Sznaier, Robust Systems Theory and Applications, John Wiley & Sons, 1998.
- K. Zhou, Robust and Optimal Control, Prentice Hall, 1996.
- K. Zhou, Essentials of Robust Control, Prentice Hall, 1998.

Web Page

http://www.lehigh.edu/~eus204/teaching/ME450_SIRC/ME450_SIRC.html

The Process of Control System Design

1. Study the "plant" to be controlled and learn about control objectives
2. Model the system and simplify the model (if necessary)
3. Analyze the resulting model; determine its properties
4. Decide which variables are to be controlled (controlled outputs)
5. Decide on measurement (sensors)/manipulated (actuators) variables
6. Select the control configuration
7. Decide on the type of controller to be used
8. Decide on performance specifications (based on control objectives)
9. Design a controller
10. Analyze resulting controlled system (redesign if necessary)
11. Simulate the resulting controlled system
12. Repeat from Step 2, if necessary.
13. Choose hardware and software and implement the controller
14. Test and validate control system and tune on-line if necessary

The Process of Control System Design

Time will be spent on input-output "controllability analysis" of the plant/process.

Always keep in mind

- Power of control is limited.
- Control quality depends controller **AND** on plant/process.

The Control Problem

$$y = Gu + G_d d \tag{1}$$

y	:	output/controlled variable
u	:	input/manipulated variable
d	:	disturbance
r	:	reference/setpoint

Regulator problem	:	counteract d
Servo problem	:	let y follow r

Goal of control: make control error $e = y - r$ "small".

Note: To arrive at a good design for the controller K we need *a priori* information about expected disturbances/references and knowledge of plant model G and disturbance model G_d.

The Control Problem

Major difficulties: Model (G, G_d) inaccurate $\Rightarrow$ RealPlant: $G_p = G + E$;

$E =$ "uncertainty" or "perturbation" (unknown but bounded)

- **Nominal stability (NS)** : system is stable with no model uncertainty
- **Nominal Performance (NP)** : system satisfies performance specifications with no model uncertainty
- **Robust stability (RS)** : system stable for "all" perturbed plants
- **Robust performance (RP)** : system satisfies performance specifications for all perturbed plants

Transfer Functions

The course will be make extensive use of the *transfer function* representation, i.e., we will work on the frequency domain.

Given a state-space representation

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned}$$

the transfer-function representation is given by (assuming SISO system)

$$\frac{y(s)}{u(s)} = G(s) = C(sI - A)^{-1}B + D, \quad G(s) = \left[\begin{array}{cc} A & B \\ C & D \end{array} \right].$$

For a MIMO system we write $Y(s) = G(s)U(s)$.

Transfer Functions

$$G(s) = \frac{\beta_{n_z} s^{n_z} + \cdots + \beta_1 s + \beta_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \tag{2}$$

For multivariable systems, $G(s)$ is a matrix of transfer functions.
$n =$ order of denominator (or pole polynomial) or *order of the system*
$n_z =$ order of numerator (or zero polynomial)
$n - n_z =$ pole excess or *relative order*.

Definition:

- A system $G(s)$ is *strictly proper* if $G(s) \to 0$ as $s \to \infty$.
- A system $G(s)$ is *semi-proper* or *bi-proper* if $G(s) \to D \neq 0$ as $s \to \infty$.
- A system $G(s)$ which is strictly proper or semi-proper is *proper*.
- A system $G(s)$ is *improper* if $G(s) \to \infty$ as $s \to \infty$.

Remark:
All practical systems have zero gain at sufficiently high frequency, and are therefore strictly proper. It is often convenient, however, to model high-frequency effects by a non-zero D-term, and hence semi-proper models are frequently used.

Transfer Functions

- Invaluable insights are obtained from simple frequency-dependent plots.
- Important concepts for feedback such as bandwidth and peaks of closed-loop transfer functions may be defined.
- $G(j\omega)$ gives the response to a sinusoidal input of frequency ω.
- A series of interconnected systems corresponds in the frequency domain to the multiplication of individual transfer functions, whereas in the time domain, the evaluation of convolution operations is required.
- Poles and zeros appear explicitly in factorized transfer functions.
- Uncertainty is more easily handled in frequency domain.

Scaling

Proper scaling simplifies controller design and performance analysis.

SISO :

unscaled:

$$\widehat{y} = \widehat{G}\widehat{u} + \widehat{G}_d\widehat{d}; \quad \widehat{e} = \widehat{y} - \widehat{r} \tag{3}$$

scaled:

$$d = \widehat{d}/\widehat{d}_{\max}, \quad u = \widehat{u}/\widehat{u}_{\max} \tag{4}$$

where:

- $\widehat{d}_{\max}$ — largest expected change in disturbance
- $\widehat{u}_{\max}$ — largest allowed input change

Scaling

Scale $\widehat{y}$, $\widehat{e}$ and $\widehat{r}$ by:

- $\widehat{e}_{\max}$ — largest allowed control error, or
- $\widehat{r}_{\max}$ — largest expected change in reference value

Usually:

$$y = \widehat{y}/\widehat{e}_{\max}, \quad r = \widehat{r}/\widehat{e}_{\max}, \quad e = \widehat{e}/\widehat{e}_{\max} \tag{5}$$

MIMO :

$$d = D_d^{-1}\widehat{d}, \quad u = D_u^{-1}\widehat{u}, \quad y = D_e^{-1}\widehat{y} \tag{6}$$

$$e = D_e^{-1}\widehat{e}, \quad r = D_e^{-1}\widehat{r} \tag{7}$$

where $D_e = \widehat{e}_{\max}$, $D_u = \widehat{u}_{\max}$, $D_d = \widehat{d}_{\max}$ and $D_r = \widehat{r}_{\max}$ are diagonal scaling *matrices*

Scaling

Substituting (6) and (7) into (3):

$$D_e y = \widehat{G} D_u u + \widehat{G}_d D_d d; \quad D_e e = D_e y - D_e r$$

and introducing the scaled transfer functions

$$G = D_e^{-1} \widehat{G} D_u, \quad G_d = D_e^{-1} \widehat{G}_d D_d \tag{8}$$

Model in terms of scaled variables:

$$y = Gu + G_d d; \quad e = y - r \tag{9}$$

Often also:

$$\tilde{r} = \widehat{r} / \widehat{r}_{\max} = D_r^{-1} \widehat{r} \tag{10}$$

so that:

$$r = R\tilde{r} \quad \text{where} \quad R \triangleq D_e^{-1} D_r = \widehat{r}_{\max} / \widehat{e}_{\max} \tag{11}$$

Scaling

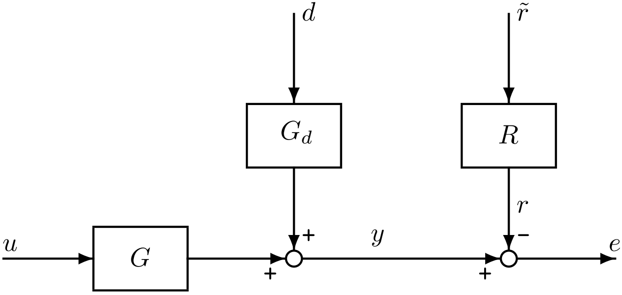


Figure: Model in terms of scaled variables

Objective:
for $|d(t)| \leq 1$ and $|\tilde{r}(t)| \leq 1$,
manipulate u with $|u(t)| \leq 1$
such that $|e(t)| = |y(t) - r(t)| \leq 1$.

Linearization

Given the general nonlinear model

$$\dot{x} = f(x, u) \tag{12}$$

where (x^*, u^*) is an equilibrium, i.e., $f(x^*, u^*) = 0$.

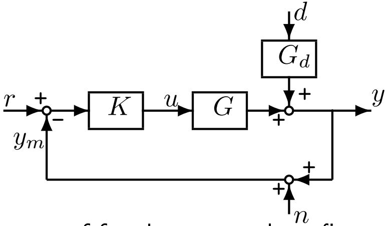
The linearization around this equilibrium is given by

$$\dot{\tilde{x}} = \left(\frac{\partial f}{\partial x}\right)^* \tilde{x} + \left(\frac{\partial f}{\partial u}\right)^* \tilde{u} = A\tilde{x} + B\tilde{u} \tag{13}$$

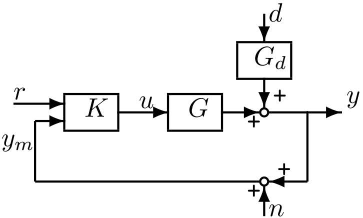
where $\tilde{x} = x - x^*$ and $\tilde{u} = u - u^*$. Therefore,

$$\tilde{x}(s) = (sI - A)^{-1} B\tilde{u} \tag{14}$$

Notation

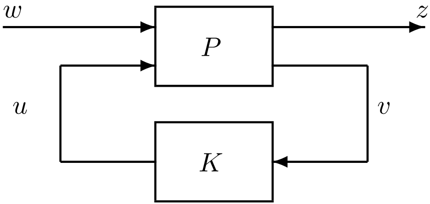


(a) One degree-of-freedom control configuration



(b) Two degrees-of-freedom control configuration

Notation



(c) General control configuration

Figure: Control configurations

Notation

Table: Nomenclature

K	controller, in whatever configuration. Sometimes broken down into parts. For example, in Figure 2(b), $K = [K_r \quad K_y]$ where K_r is a prefilter and K_y is the feedback controller.
Conventional configurations (Fig 2(a), 2(b)):	
G	plant model
G_d	disturbance model
r	reference inputs (commands, setpoints)
d	disturbances (process noise)
n	measurement noise
y	plant outputs. (include the variables to be controlled ("primary" outputs with reference values r) and possibly additional "secondary" measurements to improve control)
y_m	measured y
u	control signals (manipulated plant inputs)

Notation

General configuration (Fig 2(c)):

P generalized plant model. Includes G and G_d and the interconnection structure between the plant and the controller.
May also include weighting functions.

w exogenous inputs: commands, disturbances and noise

z exogenous outputs; "error" signals to be minimized, e.g. $y - r$

v controller inputs for the general configuration, e.g. commands, measured plant outputs, measured disturbances, etc. For the special case of a one degree-of-freedom controller with perfect measurements we have $v = r - y$.

u control signals

How to Obtain Linear Models?

Linear models my be obtained from:

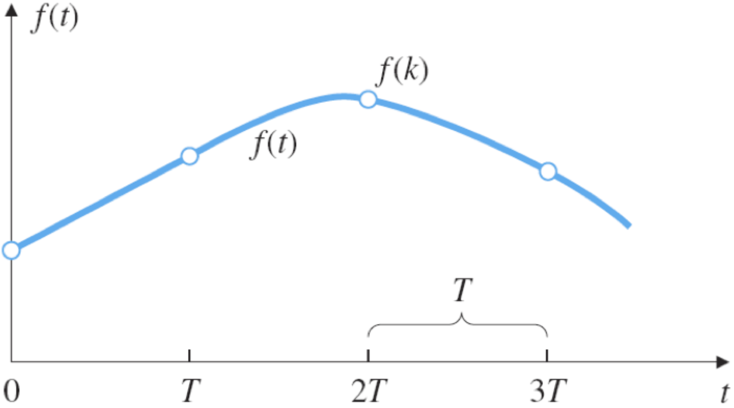
- "First-principles" modeling. Linearization may be needed.
- "Data-driven" modeling. Model identified from input-output data.
- Combination of these two approaches.

Note that regardless of how the model is obtained, the model (G, G_d) will always be inaccurate:

RealPlant: $G_p = G + E$;

$E =$ "uncertainty" or "perturbation" (unknown but bounded)

System Identification



T: Sampling period; $f = 1/T$: Sampling rate/frequency (Hz)

1. Identification of Discrete-time Model

Difference Equation:

$$y[k] + a_1 y[k-1] + \cdots + a_n y[k-n] = b_o u[k] + \cdots + b_m u[k-m]$$

Transfer Function (Z-Transform):

$$\left(1 + a_1 z + \cdots + a_n z^{-n}\right) Y(z) = \left(b_o + \cdots + b_m z^{-m}\right) U(z)$$

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_o + \cdots + b_m z^{-m}}{1 + a_1 z + \cdots + a_n z^{-n}}$$

2. Discrete-time $\rightarrow$ Continuous-time Model

Difference Equation $\rightarrow$ Differential Equation:

$$y^{(n)} + \widehat{a}_1 y^{(n-1)} + \cdots + \widehat{a}_n y = \widehat{b}_o u^{(m)} + \widehat{b}_1 u^{(m-1)} + \cdots + \widehat{b}_m u$$

Transfer Function (Z-Transform) $\rightarrow$ Transfer Function (s-Transform):

$$\left(s^n + \widehat{a}_1 s^{n-1} + \cdots + \widehat{a}_n\right) Y(s) = \left(\widehat{b}_o s^m + \widehat{b}_1 s^{m-1} + \cdots + \widehat{b}_m\right) U(s)$$

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\widehat{b}_o s^m + \widehat{b}_1 s^{m-1} + \cdots + \widehat{b}_m}{s^n + \widehat{a}_1 s^{n-1} + \cdots + \widehat{a}_n}$$

READY FOR ROBUST CONTROL DESIGN!!!