CERN–2008–003
14 March 2008

ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE
CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CAS CERN Accelerator School

Digital Signal Processing

Sigtuna, Sweden
31 May – 9 June 2007

Proceedings

Editor: D. Brandt

GENEVA
2008

Abstract

These proceedings present the lectures given at the twenty-first specialized course organized by the CERN Accelerator School (CAS), the topic being Digital Signal Processing. The course was held in Sigtuna, Sweden, from 31 May–9 June 2007. This is the first time this topic has been selected for a specialized course. Taking into account the number of related applications currently in use in accelerators around the world, it was recognized that such a topic should definitively be incorporated into the CAS series of specialized courses. The specific aim of the course was to introduce the participants to the use and programming of Digital Signal Processors (DSPs) and Field Programmable Gate Arrays (FPGAs) evaluation boards. The course consisted of lectures in the mornings covering fundamental background knowledge in mathematics, controls theory, design tools, programming hardware platforms, and implementation details. In the afternoons the students split into two groups with people working in pairs. One group worked on problem solving using a PC-based workstation connected to a DSP evaluation board while the second group performed the same exercise on PCs connected to a FPGA evaluation board. Half-way through the school, those working on DSPs moved to FPGAs and vice versa. The problems to be solved were of increasing difficulty and all of them were directly related to real and practical realizations currently in use in the accelerator field.

Foreword

The aim of the CERN Accelerator School (CAS) to collect, preserve and disseminate the knowledge accumulated in the world's accelerator laboratories applies not only to general accelerator physics, but also to related sub-systems, equipment, and technologies. This wider aim is achieved by means of specialized courses. For 2007, the topic of the course was Digital Signal Processing (DSP) and was held at the SigtunahÃűjden Hotel, Sigtuna, Sweden, from 31 May–9 June 2007. The course was organized in collaboration with the Uppsala University, Sweden.

This was the first time Digital Signal Processing was used as a topic for a CAS specialized course. However, when considering the number of related applications currently in use in the field of accelerators, it was felt that it was the right time to address this specific topic. The organization of such a course was a real challenge in itself, since it implied finding and transporting computers to Sweden, negotiating the availability of software licences for the duration of the course, negotiating the acquisition of DSP and FPGA evaluation boards, and preparing 50 computers to be compatible with the proposed exercises. It is a pleasure to express my thanks to CERN, the companies Synplicity, Mathworks and Xilinx, as well as the Texas Instruments Worldwide University Program, and the Xilinx University Program for their kind support, without which the course would not have been made possible.

To ensure that the required material was made available was certainly a mandatory pre-requisite, but the elaboration of the course and, in particular, the hands-on courses in the afternoon sessions were no less challenging. I would particularly like to acknowledge the outstanding contribution of our CERN colleagues Hermann Schmickler, Maria-Elena Angoletta and Javier Serrano for their contacts with the relevant industry and the high-quality afternoon courses they prepared and supervised, John Evans for his hard work in obtaining the software licences and Edwige Bournonville for the perfect preparation of the computers as well as their maintenance during the course. Similarly, I would like to thank Leif Gustafsson and Pawel Marciniewski (Uppsala University) for their highly appreciated contribution as tutors during the afternoon courses.

It is also important to thank the Local Organizing Committee (Professor Tord Ekelof, Dr. Volker Ziemann and Ms. Inger Ericson) for obtaining financial support from the Swedish Research Council (Vetenskapsraadet) to help CAS with the organization of the course.

As always, the backing of the CERN management, the guidance of the CAS Advisory and Programme Committees, the attention to detail of the Local Organizing Committee and the management and staff of the SigtunahÃűjden Hotel ensured that the school was held under optimum conditions.

Very special thanks must go to the lecturers for the enormous task of preparing, presenting, and writing up their topics.

Finally, the enthusiasm of the participants who came from more than 23 different countries around the world was convincing proof of the usefulness and success of the course.

It is my pleasure and privilege to thank most sincerely all those persons who helped me to make the course a success, including Barbara Strasser, Suzanne von Wartburg, and the team of the CERN Scientific Text Processing Service for their dedication and commitment to the production of this document.

Daniel Brandt
CERN Accelerator School

PROGRAMME
Digital Signal Processing (DSP)
31 May – 9 June 2007, Sigtuna, Sweden

Time	Friday 1 June	Saturday 2 June	Sunday 3 June	Monday 4 June	Tuesday 5 June	Wednesday 6 June	Thursday 7 June	Friday 8 June	Saturday 9 June
08:30 **09:30**	Types of Accelerators and Specific Needs I T. Shea	Math I M. Hoffmann	Math II M. Hoffmann	Math III M. Hoffmann	**E X C U R S I O N**	Math IV M. Hoffmann	Real Time Control of Beam Parameters I M. Dehler	Real Time Control of Beam Parameters II M. Dehler	**Buses to airport**
09:30 **10:30**	Types of Accelerators and Specific Needs II T. Shea	Control Theory I S. Simrock	Control Theory II S. Simrock	Control Theory III S. Simrock		RF Application I T. Schilcher	RF Application II T. Schilcher	Controls Integration II T. Shea	
	C O F F E E B R E A K					**C O F F E E B R E A K**			
11:00 **12:00**	High Level Modeling Tools I J. Evans	Introduction to DSP I M. Angoletta	Introduction to DSP II M. Angoletta	Introduction to DSP III M. Angoletta		Transverse Feedbacks M. Lonza	Controls Integration I T. Shea	BI Application II U. Raich	
12:00 **13:00**	High Level Modeling Tools II J. Evans	Introduction to FPGA I J. Serrano	Introduction to FPGA II J. Serrano	Introduction to FPGA III J. Serrano		Longitudinal Feedbacks M. Lonza	BI Application I U. Raich	Outlook or Seminar H. Schmickler	
	L U N C H					**L U N C H**			
14:30 **15:30**	Introduction to Afternoon Courses H. Schmickler	From Analog to Digital I J. Belleman	From Analog to Digital II J. Belleman	From Analog to Digital III J. Belleman		Parallel Lab course (DSP/FPGA) Group B/A	Parallel Lab course (DSP/FPGA) Group B/A	Parallel Lab course (DSP/FPGA) Group B/A	
15:30 **16:30**	Parallel Lab course (DSP/FPGA) Group A/B	Parallel Lab course (DSP/FPGA) Group A/B	Parallel Lab course (DSP/FPGA) Group A/B	Parallel Lab course (DSP/FPGA) Group A/B		Parallel Lab course (DSP/FPGA) Group B/A	Parallel Lab course (DSP/FPGA) Group B/A	Discussion Session D. Brandt	
	C O F F E E B R E A K					**C O F F E E B R E A K**			
17:00 **18:00**	Parallel Lab course (DSP/FPGA) Group A/B	Parallel Lab course (DSP/FPGA) Group A/B	Parallel Lab course (DSP/FPGA) Group A/B	Parallel Lab course (DSP/FPGA) Group B/A		Parallel Lab course (DSP/FPGA) Group B/A	Parallel Lab course (DSP/FPGA) Group B/A	Parallel Lab course (DSP/FPGA) Group B/A	
19:00	**Welcome Drink**	**DINNER**	**DINNER**	**DINNER**		**DINNER**	**Special Dinner**	**DINNER**	

Contents

Control theory

S. Simrock
DESY, Hamburg, Germany

Abstract

In engineering and mathematics, control theory deals with the behaviour of dynamical systems. The desired output of a system is called the reference. When one or more output variables of a system need to follow a certain reference over time, a controller manipulates the inputs to a system to obtain the desired effect on the output of the system. Rapid advances in digital system technology have radically altered the control design options. It has become routinely practicable to design very complicated digital controllers and to carry out the extensive calculations required for their design. These advances in implementation and design capability can be obtained at low cost because of the widespread availability of inexpensive and powerful digital processing platforms and high-speed analog IO devices.

1 Introduction

The emphasis of this tutorial on control theory is on the design of digital controls to achieve good dynamic response and small errors while using signals that are sampled in time and quantized in amplitude. Both transform (classical control) and state-space (modern control) methods are described and applied to illustrative examples. The transform methods emphasized are the root-locus method of Evans and frequency response. The state-space methods developed are the technique of pole assignment augmented by an estimator (observer) and optimal quadratic-loss control. The optimal control problems use the steady-state constant gain solution.

Other topics covered are system identification and non-linear control. System identification is a general term to describe mathematical tools and algorithms that build dynamical models from measured data. A dynamical model in this context is a mathematical description of the dynamic behaviour of a system or process.

Non-linear control is a sub-division of control engineering which deals with the control of non-linear systems. The behaviour of a non-linear system cannot be described as a linear function of the state of that system or the input variables to that system. There exist several well-developed techniques for analysing non-linear feedback systems. Control design techniques for non-linear systems also exist. These can be subdivided into techniques which attempt to treat the system as a linear system in a limited range of operation and use (well-known) linear design techniques for each region.

The main components of a control system (Fig. 1) are the plant and a sensor which is used to measure the variable to be controlled. The closed-loop system (Fig. 2) requires a connection from the system outputs to the inputs. For feedback control a compensator amplifies and filters the error signal (Fig. 3) to manipulate the input signals to the plant to minimize the errors of the variables.

Objective: Control theory is concerned with the analysis and design of a closed-loop control system.

Analysis: Closed-loop system is given to determine characteristic or behaviour.

Design: Desired system characteristics or behaviour are specified → configure or synthesize a closed-loop system.

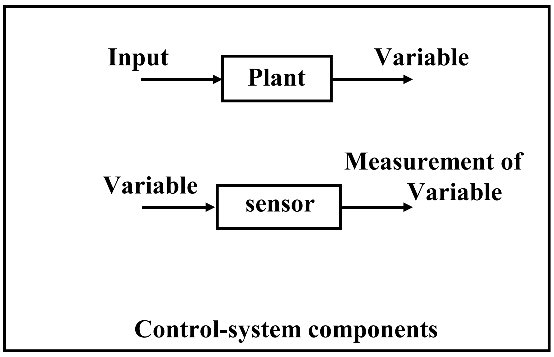


Fig. 1: Components of a control system

Definition: A closed-loop system is a system in which certain forces (we call these inputs) are determined, at least in part, by certain responses of the system (we call these outputs).

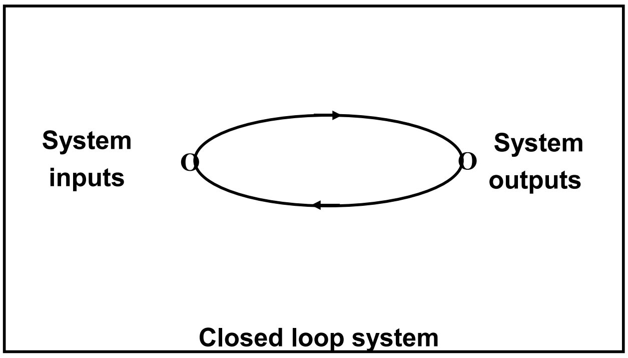


Fig. 2: Schematic of a closed-loop system

Definitions:

- The system for measurement of a variable (or signal) is called a sensor
- A **plant** of a control system is the part of the system to be controlled
- The **compensator** (or controller or simply filter) provides satisfactory characteristics for the total system.

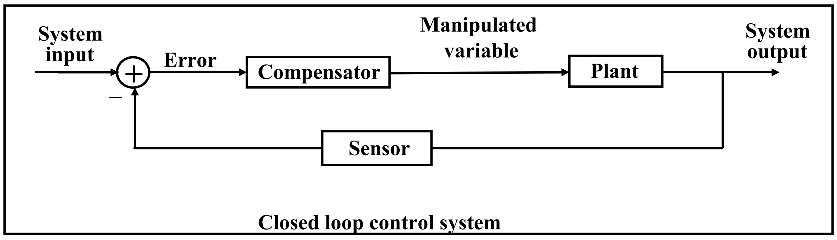


Fig. 3: Arrangement of controller and plant in a closed-loop system

Two types of control system:

- A **regulator** maintains a physical variable at some constant value in the presence of perturbances.
- A **servo mechanism** describes a control system in which a physical variable is required to follow or track some desired time function (originally applied in order to control a mechanical position or motion).

2 Examples of control systems

In the following we study two examples where feedback control is applied. The first example shows how to model an RF control system. The second and third examples show electronic circuits which can be used to design the compensator (or controller) filter.

2.1 RF control system

RF control systems are used to stabilize amplitude and phase of the accelerating fields. Amplitude and phase are measured, compared with the desired setpoints, and the amplified error signals drive actuators for amplitude and phase thereby forming a negative feedback loop.

Goal:
Maintain stable gradient and phase in presence of field perturbations.
Solution: Feedback for gradient amplitude and phase.

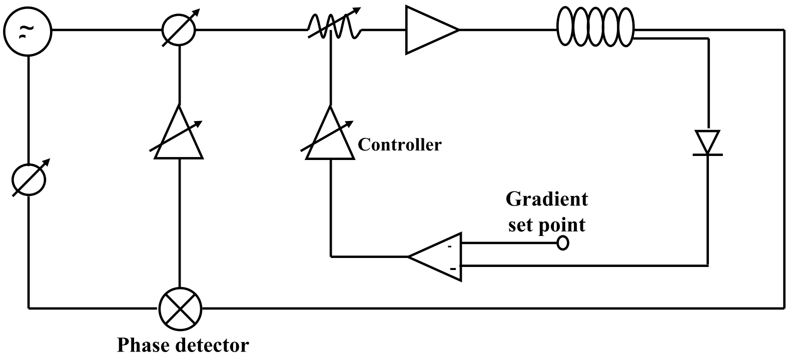


Fig. 4: RF control system with feedback control for amplitude and phase

Model: Mathematical description of input–output relation of components combined with block diagram.

Amplitude loop (general form):

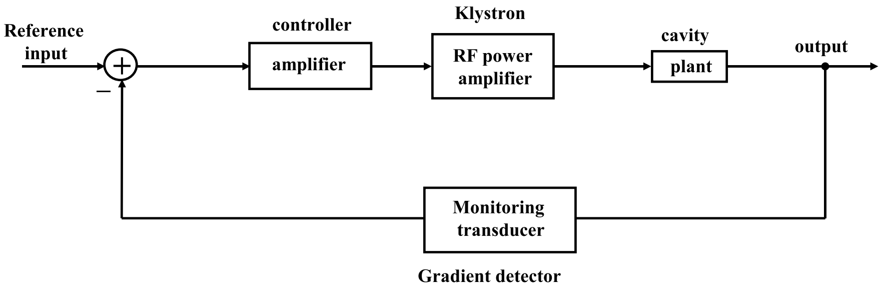


Fig. 5: General form feedback loop for amplitude control with blocks for the system components

RF control model using *‘transfer functions’*

A transfer function of a *linear* system is defined as the ratio of the Laplace transform of the output and the Laplace transform of the input with ICs = zero.

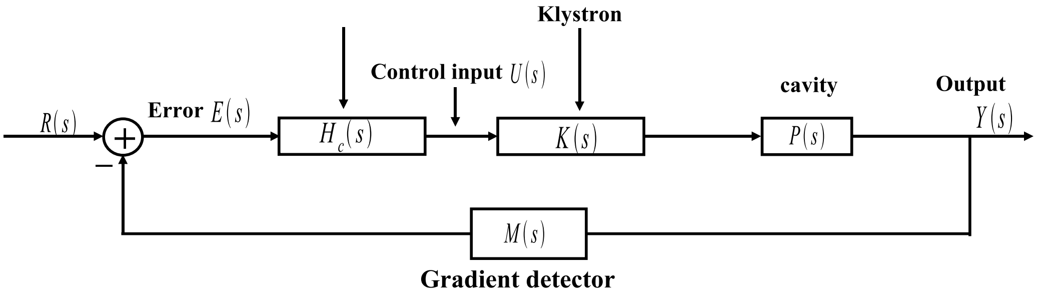


Fig. 6: Description of system components in the amplitude control loop by transfer functions

Input–output relations

$Input$	$Output$	$Transfer\ function$
$U(s)$	$Y(s)$	$G(s) = P(s)K(s)$
$E(s)$	$Y(s)$	$L(s) = G(s)H_c(s)$
$R(s)$	$Y(s)$	$T(s) = (1 + L(S)M(s))^{-1}L(s)$

2.2 Passive network circuit

For high frequencies the network shown in (Fig. 7) forms a voltage divider determined by the ratio of the resistors, while at low frequencies the input voltage is unchanged at the output.

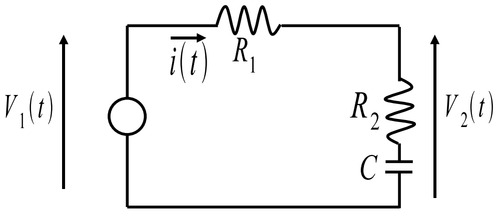


Fig. 7: Passive network for a frequency-dependent voltage divider

Differential equations:

$$R_1 i(t) + R_2 i(t) + \frac{1}{C}\int_0^t i(\tau)d\tau = v_1(t)$$

$$R_2 i(t) + \frac{1}{C}\int_0^t i(\tau)d\tau = v_2(t)$$

Laplace transform:

$$R_1 I(s) + R_2 I(s) + \frac{1}{sC}I(s) = V_1(s)$$

$$R_2 I(s) + \frac{1}{s.C}I(s) = V_2(s)$$

Laplace transform:

$$G(s) = \frac{V_2(s)}{V_1(s)} = \frac{R_2Cs+1}{(R_1+R_2)Cs+1}$$

Input V_1, Output V_2 .

2.3 Operational amplifier circuit

The transfer function of the circuit shown in (Fig. 8) represents an amplifier with very high gain at low frequencies and a gain determined by the ratio of the resistors at high frequencies. The more general case is shown in (Fig. 9) and allows the realization of a wide range of transfer functions.

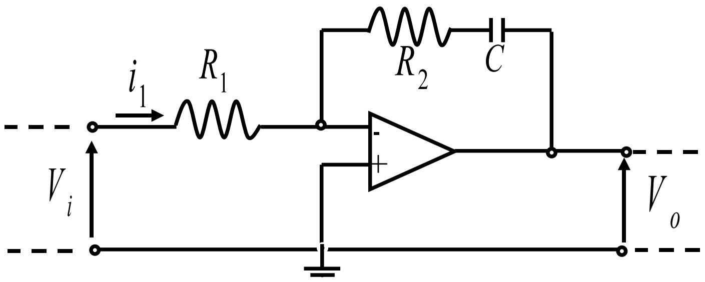


Fig. 8: Active circuit with operational amplifier

$V_i(s) = R_1 I_1(s)$ and $V_o(s) = -(R_2 + \frac{1}{sC}) I_1(s)$

$G(s) = \frac{V_0(s)}{V_i(s)} = -\frac{R_2 Cs+1}{R_1 Cs}$

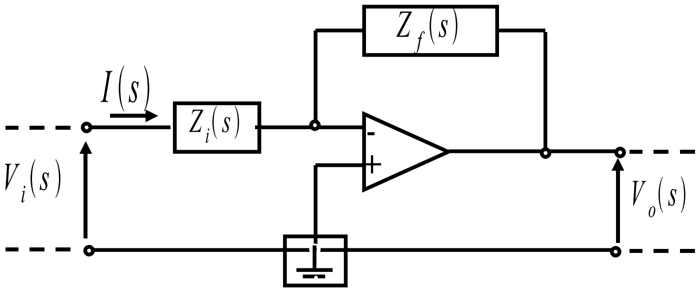


Fig. 9: Generalized version of active circuit with operational amplifier using impedances in the input and feedback section

It is convenient to derive a transfer function for a circuit with a single operational amplifier that contains input and feedback impedance:

$V_i(s) = Z_i(s) I(s)$ and $V_0(s) = -Z_f(s) I(s) \rightarrow G(s) = \frac{V_0(s)}{V_i(s)} = -\frac{Z_f(s)}{Z_i(s)}$

3 Model of dynamic systems

Models of dynamic systems can be described by differential equations. A system of order n can be reduced to a set of n first-order equations. These can be written in a matrix formalism called state-space representation.

We shall study the following dynamic system:

Parameters: k: spring constant, γ: damping constant, $u(t)$: force
Quantity of interest: $y(t)$: displacement from equilibrium

Differential equation: Newton's third law (m = 1)

$\ddot{y}(t) = \sum F_{ext} = -ky(t) - \gamma\dot{y}(t) + u(t)$

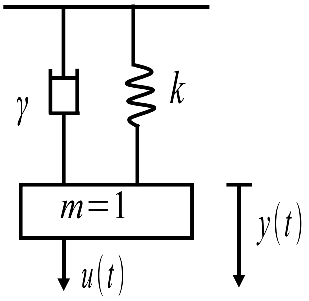


Fig. 10: Harmonic oscillator with mass m, spring constant k, and damping term γ

$$\dot{y} + \gamma\dot{y}(t) + ky(t) = u(t)$$

$$y(0) = y_0, \dot{y}(0) = \dot{y}_0$$

1. Equation is linear (i.e., no y^2 like terms).
2. Ordinary (as opposed to partial e.g., $\frac{\partial}{\partial x}\frac{\partial}{\partial t}f(x,t) = 0$).
3. All coefficients constant: $k(t) = k, \gamma(t) = \gamma$ for all t.

Stop calculating, let us paint. Picture to visualize differential equation.

1. Express highest-order term (put it to one side)

   $$\ddot{y}(t) = -ky(t) - \gamma\dot{y}(t) + u(t)$$

2. Put adder in front.

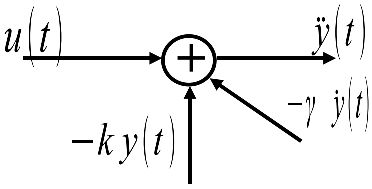


Fig. 11: Visualization of the differential equation for the harmonics oscillator

3. Synthesize all other terms using integrators.

3.1 Linear Ordinary Differential Equation (LODE)

General form of LODE:

$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + ... + a_1\dot{y}(t) + a_0y(t) = b_m u^{(m)}(t) + ... + b_1\dot{u}(t) + b_0u(t)$
m, n positive integers, $m \leq n$; coefficients $a_0, a_1, ..., a_{n-1}, b_0, ...b_m$ real numbers.

Mathematical solution: I hope you know it

Solution of LODE: $y(t) = y_h(t) + y_p(t)$.
The solution is the sum of the homogeneous solution $y_h(t)$ (natural response) solving

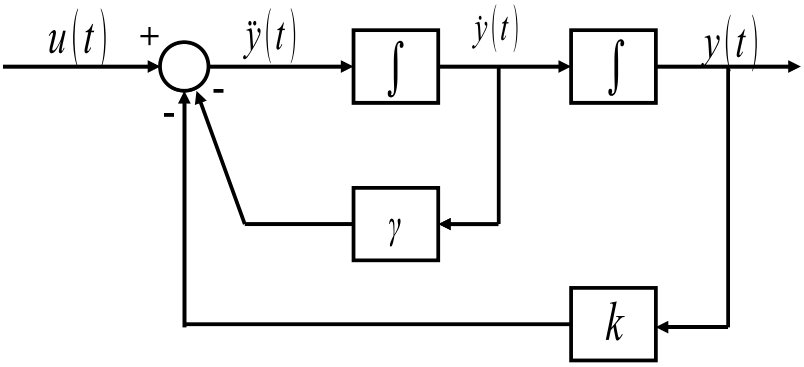


Fig. 12: Visualization of the second-order equation for the simple harmonic oscillator. The diagram can be used to program an analog computer.

$$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + ... + a_1\dot{y}(t) + a_0y(t) = 0$$

and the particular solution $y_p(t)$.

How to get natural response $y_h(t)$? Characteristic polynomial

$$x(\lambda) = \lambda_n + a_{n-1}\lambda_{n-1} + a_1\lambda + a_0 = 0$$

$$(\lambda - \lambda_1)^r.(\lambda - \lambda_{r+1}...(\lambda - \lambda_n) = 0$$

$$y_h(t) = (c_1 + c_2t + ... + c_rt^{r-1})e^{\lambda_1 t} + c_{r+1}e^{\lambda_{r+1}t}... + c_ne^{\lambda_n t}$$

The determination of $y_p(t)$ is relatively simple, if the input $u(t)$ yields only a finite number of independent derivates, e.g., $u(t) \cong e^{xit}, \beta_r t^r$.

Most important for control system/feedback design:
$y_{(n-1)}(t) + a_{n-1}y^{(n-1)}(t) + ... + a_1\dot{y}(t) + a_0y(t) = b_mu^{(m)}(t) + ... + b_1\dot{u}(t) + b_0u(t)$.

In general: Any linear time-invariant system described by a LODE can be realized, simulated, and easily visualized in a block diagram $(n = 2, m = 2)$

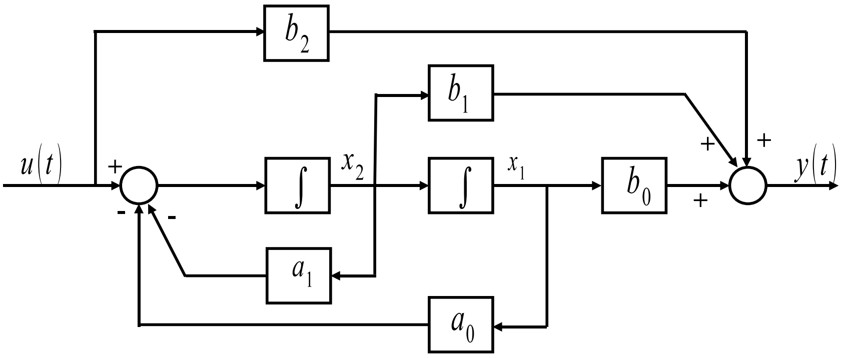


Fig. 13: Control-canonical form of a second-order differential equation

It is very useful to visualize *interaction* between the variables.

What are x_1 and x_2?

More explanations later, for now: Please simply accept it.

3.2 State-space equation

Any system which can be represented by a LODE can be represented in *state space form* (matrix differential equation).

Let us go back to our first example (Newton's law):

$\ddot{y}(t) + \gamma\dot{y}(t) + ky(t) = u(t)$

Step 1: Deduce set of first-order differential equation in variables

$x_j(t)$ (so-called states of system)

$x_1(t) \cong Position : y(t)$

$x_2(t) \cong Velocity : \dot{y}(t) :$

$\dot{x_1}(t) = \dot{y}(t) = x_2(t)$

$\dot{x_2}(t) = \ddot{y}(t) = -ky(t) - \gamma\dot{y}(t) + u(t) = -kx_1(t) - \gamma x_2(t) + u(t)$

One LODE of order n transformed into n LODEs of order 1

Step 2: Put everything together in a matrix differential equation:

$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & -\gamma \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$\dot{x}(t) = Ax(t) + Bu(t)$ State equation

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

$y(t) = Cx(t) + Du(t)$ Measurement equation

Definition: The *system state* x of a system at any time t_0 is the 'amount of information' that, together with all inputs for $t \geq t_0$, uniquely determines the behaviour of the system for all $t \geq t_0$.

The linear time-invariant (LTI) analog system is described by the standard form of the state-space equation $\dot{x}(t) = Ax(t) + Bu(t)$ state equation

$y(t) = Cx(t) + Du(t)$ state equation,

where $\dot{x}(t)$ is the time derivative of the vector $x(t) = \begin{bmatrix} x_1(t) \\ ... \\ x_2(t) \end{bmatrix}$.

The system is completely described by the state-space matrices A, B, C, D (in most cases $D = 0$).

Declaration of variables

Variable	**Dimension**	**Name**
$X(t)$	$nx1$	State vector
A	nXn	System matrix
B	nXr	Input matrix
$u(t)$	$rX1$	Input vector
$y(t)$	$pX1$	Output vector
C	pXn	Output matrix
D	pXr	Matrix representing direct coupling between input and output

Why all this work with state-space equation? Why bother with it?

Because, given any system of the LODE form

$$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + ... + a_1\dot{y}(T) + a_0y(t) = b_mu^{(m)}(t) + ... + b_1\dot{u}(t) + b_0u(t)$$

the system can be represented as

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

With, for example, **control-canonical form** (case $n = 3$, $m = 3$):

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},\ C = \begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix},\ D = b_3$$

or **observer-canonical form:**

$$A = \begin{bmatrix} 0 & 0 & -a_0 \\ 1 & 0 & -a_1 \\ 0 & 1 & -a_2 \end{bmatrix},\ B = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix},\ C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},\ D = b_3$$

1. Notation is very compact, but not unique.
2. Computers love the state-space equation. (Trust us!)
3. Modern control (1960$-$now) uses the state-space equation
4. General (vector) block diagram for easy visualization

3.2.1 Block diagrams:

Now: solution of the state-space equation in the time domain:

$$x(t) = \phi(t)x(0) + \int_0^t \phi(\tau)Bu(t - \tau)d\tau\ .$$

Natural response + particular solution

$$y(t) = Cx(t) + Du(t)$$

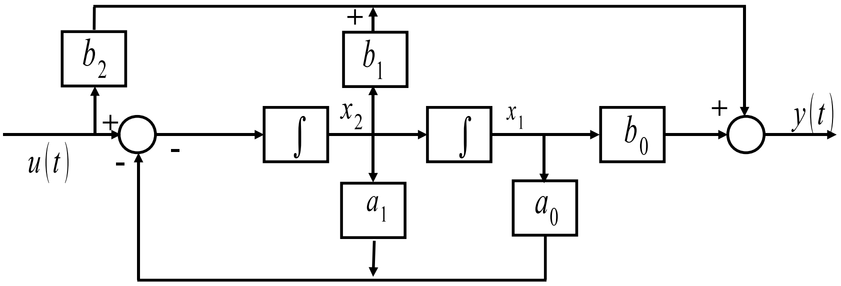


Fig. 14: Control-canonical form of a differential equation of second order

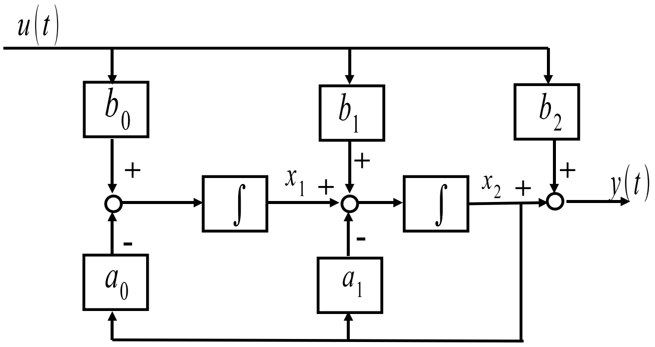


Fig. 15: Observer-canonical form of a differential equation of second order

$= C\phi(t)x(0) + C\int_0^t \phi(\tau)Bu(t-\tau)d\tau + Du(t)$.

With the **state transition matrix**

$\phi(t) = I + At + \frac{A^2}{2!}t^2 + \frac{A^3}{3!}t^3 + ... = e^{At}$,

which is an exponential series in the matrix A (time evaluation operator).
Properties of $\phi(t)$ (state transition matrix):

1. $\frac{d\phi(t)}{dt} = A\phi(t)$
2. $\phi(0) = I$
3. $\phi(t_1 + t_2) = \phi(t_1).\phi(t_2)$
4. $\phi^{-1} = \phi(-t)$

Example:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \Longrightarrow A^2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \phi(t) = I + At = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} = e^{At}$$

Matrix A is a nilpotent matrix.

3.2.2 Example

We have the following differential equation:

$\frac{d^2}{dt^2}y(t) + 4\frac{d}{dt}y(t) + 3u(t) = 2u(t)$.

1. State equations of the differential equation:

   Let $x_1(t) = y(t)$ and $x_2(t) = \dot{y}(t)$ then

   $\dot{x_1}(t) = \dot{y}(t) = x_2(t)$

   $\dot{x_2}(t) + 4x_2(t) + 3x_1(t) = 2u(t)$

   $\dot{x_2} = -3x_1(t) - 4x_2(t) + 2u(t)$

2. Write the state equations in matrix form:

   Define system state $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u(t)$
   $y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$.

3.2.3 Cavity model

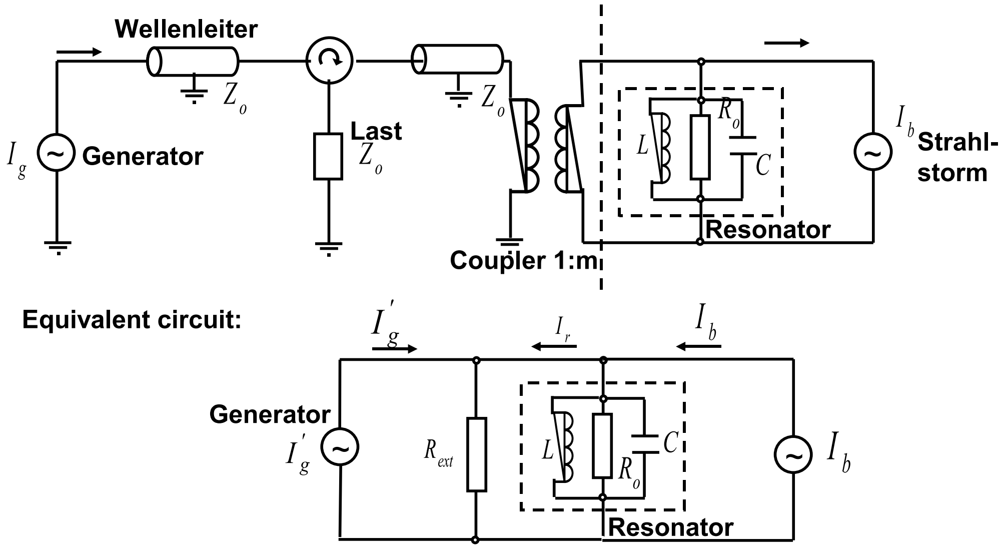


Fig. 16:

$$C.\ddot{U} + \frac{1}{R_L}.\dot{U} + \frac{1}{L}.U = \dot{I_g'} + \dot{I_b}$$

$$\omega_{\frac{1}{2}} := \frac{1}{2R_LC} = \frac{\omega_0}{2Q_L}$$

$$\ddot{U} + 2\omega_{\frac{1}{2}}.\dot{U} + \omega_0^2.U = 2R_L\omega_{\frac{1}{2}}.(\frac{2}{m}\dot{I}_g + \dot{I}_b)$$

Only the envelope of RF (real and imaginary part) is of interest:

$$U(t) = (U_r(t) + iU_i(t)).\text{exp}(i\omega_{HF}(t)$$

$$I_g(t) = (I_gr(t) + iI_{gi}(t)).\text{exp}(i\omega_{HF}(t))$$

$$I_b(t) = (I_{b\omega r}(t) + iI_{b\omega i}(t)).\text{exp}(i\omega_{HF}t) = 2(I_{bor}(t) + iI_{boi}(t)).\text{exp}(i\omega_{HF}t)$$

Neglect small terms in derivatives for U and I.

$\ddot{U}_r + i\ddot{U}_i(t) << \omega_{HF}^2(U_r(t) + iU_i(t))$

$2\omega_{\frac{1}{2}}(\dot{U}_r + i\dot{U}_r(t)) << \omega_{HF}^2(U_r(t) + iU_i(t))$

$\int_{t_1}^{t_2}(\dot{I}_r(t) + i\dot{I}_i(t))dt << \int_{t_1}^{t_2}\omega_{HF}(I_r(t) + iI_i(t))dt$

Envelope equations for real and imaginary component.

$\dot{U}_r(t) + \omega_{\frac{1}{2}}.U_r + \Delta\omega.U_i = \omega_{HF}\frac{r}{Q}.(\frac{1}{m}I_{gr} + I_{bor})$

$\dot{U}_i(t) + \omega_{\frac{1}{2}}.U_i - \Delta\omega.U_r = \omega_{HF}(\frac{r}{Q}).(\frac{1}{m}I_{gi} + I_{boi})$

Matrix equations:

$$\begin{bmatrix} \dot{U}_r(t) \\ \dot{U}_i(t) \end{bmatrix} = \begin{bmatrix} -\omega_{\frac{1}{2}} & -\Delta\omega \\ \Delta\omega & -\omega_{\frac{1}{2}} \end{bmatrix} . \begin{bmatrix} U_r(t) \\ U_i(t) \end{bmatrix} + \omega_{HF}(\frac{r}{Q}). \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} . \begin{bmatrix} \frac{1}{m}I_{gr}(t) + I_{bor}(t) \\ \frac{1}{m}I_{gi}(t) + I_{boi}(t) \end{bmatrix}$$

With system Matrices: $A = \begin{bmatrix} -\omega_{\frac{1}{2}} & -\Delta\omega \\ \Delta\omega & -\omega_{\frac{1}{2}} \end{bmatrix}$ $\qquad B = \omega_{HF}(\frac{r}{Q}). \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$\overrightarrow{x}(t) = \begin{bmatrix} U_r(t) \\ U_i(t) \end{bmatrix}$ $\qquad \overrightarrow{u}(t) = \begin{bmatrix} \frac{1}{m}I_{gr}(t) + I_{bor}(t) \\ \frac{1}{m}I_{gi}(t) + I_{boi}(t) \end{bmatrix}$

General form: $\dot{\overrightarrow{x}} = A.\vec{x}(t) + B.\overrightarrow{u}(t)$.

Solution:

$\overrightarrow{x}(t) = \phi(t).\overrightarrow{x}(0) + \int_0^t \phi(t - t').B.\overrightarrow{u}(t')dt'$

$\phi(t) = e^{-\omega_{\frac{1}{2}}t} \begin{bmatrix} \cos(\Delta\omega t) & -\sin(\Delta\omega t) \\ \sin(\Delta\omega t) & \cos(\Delta\omega t) \end{bmatrix}$

Special case:

$\overrightarrow{u}(t) = \begin{bmatrix} \frac{1}{m}I_{gr}(t) + I_{bor}(t) \\ \frac{1}{m}I_{gi}(t) + I_{boi}(t) \end{bmatrix} =: \begin{bmatrix} I_r \\ I_i \end{bmatrix}$

$$\begin{bmatrix} U_r(t) \\ U_i(t) \end{bmatrix} = \frac{\omega_{HF}(\frac{r}{Q})}{\omega_{\frac{1}{2}}^2 + \Delta\omega^2}. \begin{bmatrix} \omega_{\frac{1}{2}} & -\Delta\omega \\ \Delta\omega & \omega_{\frac{1}{2}} \end{bmatrix} .\{1 - \begin{bmatrix} \cos(\Delta\omega t) & -\sin(\Delta\omega t) \\ \sin(\Delta\omega t) & \cos(\Delta\omega t) \end{bmatrix} e^{-\omega_{\frac{1}{2}}t}\}. \begin{bmatrix} I_r \\ I_i \end{bmatrix}$$

3.3 Mason's rule

Mason's rule is a simple formula for reducing block diagrams. It works for continuous and discrete systems. In its most general form it is messy, except for the special case when all paths touch:

$H(s) = \frac{\Sigma(\text{forward path gains})}{1-\Sigma(\text{loop path gains})}$

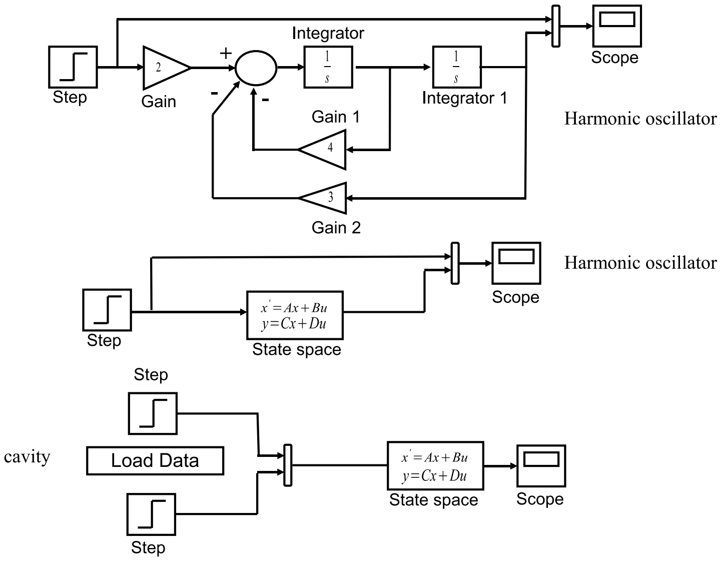


Fig. 17: Simulink diagram of the harmonic oscillator as analog computer and in state-space form. The state-space form of the cavity model follows the same structure.

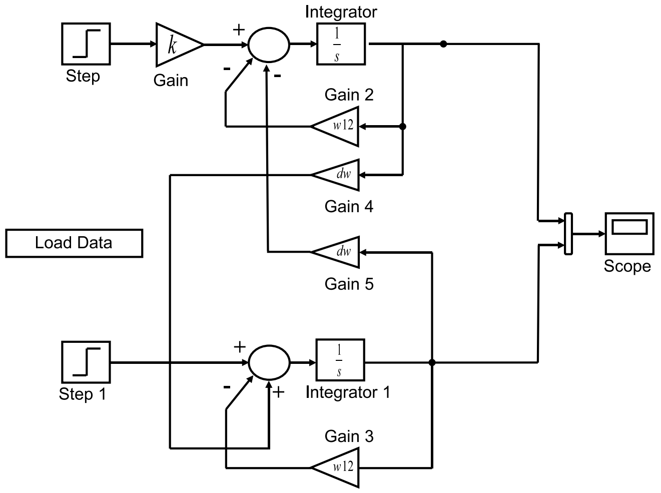


Fig. 18: Simulink diagram of the cavity model with two inputs and output for real and imaginary parts of the drive and field vectors

Two paths are said to *touch* if they have a component in common, e.g., an adder.

Forward path: F_1:$1 - 10 - 11 - 5 - 6$ and F_2:$1 - 2 - 3 - 4 - 5 - 6$

Loop path: I_1:$3 - 4 - 5 - 8 - 9$ and I_2:$5 - 6 - 7$

$G(f_1) = H_5 H_3, G(f_2) = H_1 H_2 H_3, G(I_1) = H_2 H_4, G(I_2) = H_3$

Check: All paths touch (contain adder between 4 and 5)

$\Longrightarrow$ By Mason's rule: $H = \frac{G(f_1)+G(f_2)}{1-G(l_1)-G(l_2)} = \frac{H_5 H_3 + H_1 H_2 H_3}{1-H_2 H_4 - H_3} = \frac{H_3(H_5+H_1 H_2)}{1-H_2 H_4 - H_3}$.

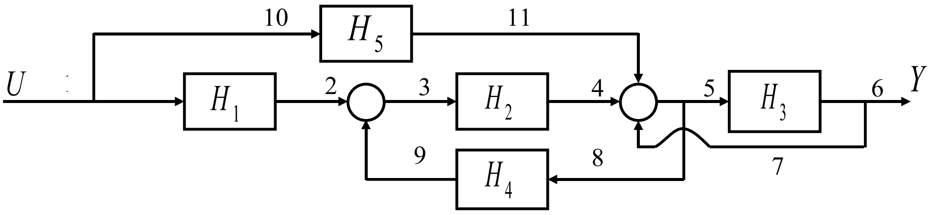


Fig. 19: Example for Mason's rule. All connections in the forward and loop paths are shown.

4 Transfer function $G(s)$

The transfer function is commonly used in the analysis of single-input single-output electronic filters, for instance. It is mainly used in signal processing, communication theory, and control theory. The term is often used exclusively to refer to linear, time-invariant systems (LTI), as covered in this article. Most real systems have non-linear input/output characteristics, but many systems, when operated within nominal parameters (not 'over-driven') have behaviour that is close enough to linear that LTI system theory is an acceptable representation of the input/output behaviour.

Continuous-time space model:

$\dot{x}(t) = Ax(t) + Bu(t)$ State equation

$y(t) = Cx(t) + Du(t)$ Measurement equation

The transfer function describes the input–output relation of the system.

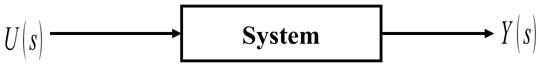


Fig. 20: System with the Laplace transforms of input and output signals. The ratio is the transfer function.

$sX(s) - x(0) = AX(s) + BU(s)$

$X(s) = (sI - A)^{-1}x(0) + (sI - A)^{-1}BU(s)$

$= \phi(s)x(0) + \phi(s)BU(s)$

$Y(s) = CX(s) + DU(s)$

$= C[(sI - A)^{-1}]x(0) + [c(sI - A)^{-1}B + D]U(s)$

$= C\phi(s)x(0) + C\phi(s)BU(s) + DU(s)$

Transfer function $G(s)$ (pxr) (case:$X(0) = 0$):

$G(s) = C(sI - A)^{-1}B + D = C\phi(s)B + D$

Transfer function of TESLA cavity including $\frac{8}{9} - \pi$ mode

$$H_{cont}(s) \approx H_{cav}(s) = H_{\Pi}(s) + H_{\frac{8\pi}{9}}(s)$$

$$\pi-\text{ mode } H_{\Pi}(s) = \frac{(\omega_{\frac{1}{2}})\pi}{\Delta\omega_{\pi}^{2}+(s+(\omega_{\frac{1}{2}})_{\pi})^{2}} \begin{bmatrix} s+(\omega_{\frac{1}{2}})_{\pi} & -\Delta\omega_{\pi} \\ -\Delta\omega_{\pi} & s+(\omega_{\frac{1}{2}})_{\pi} \end{bmatrix}$$

$$\frac{8}{9}\pi-\text{ mode } H_{\frac{8\pi}{9}}(s) = -\frac{(\omega_{\frac{1}{2}})_{\frac{8\pi}{9}}}{\Delta\omega_{\frac{8\pi}{9}}^{2}+(s+(\omega_{\frac{1}{2}})_{\frac{8\pi}{9}})^{2}} \begin{bmatrix} s+(\omega_{\frac{1}{2}})^{\frac{8\pi}{9}} & -\Delta\omega_{\frac{8\pi}{9}} \\ \Delta\omega_{\frac{8\pi}{9}} & s+(\omega_{\frac{1}{2}})_{\frac{8\pi}{9}} \end{bmatrix}$$

4.1 Transfer function of a closed-loop system

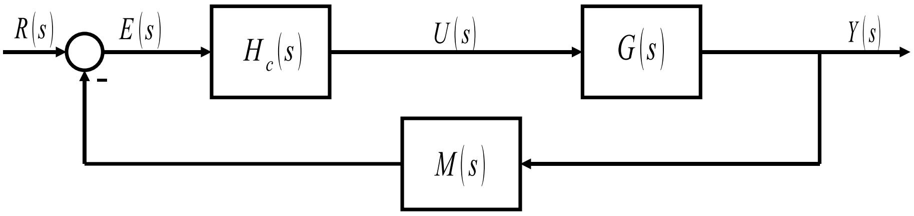


Fig. 21: Diagram of a closed-loop system. Mason's rule can be applied to determine the closed-loop transfer function.

We can deduce for the output of the system:

$$Y(s) = G(s)U(s) = G(s)H_c(s)E(s)$$

$$= G(s)H_c(s)[R(s) - M(s)Y(s)$$

$$= L(s)R(s) - L(s)M(s)Y(s)$$

With $L(s)$ the transfer function of the open-loop system (controller and plant)

$$(1 + L(s)M(s))Y(s) = L(s)R(s)$$

$$Y(s) = (I + L(s)M(s))^{-1}L(s)R(s) = T(S)R(s),$$

where $T(s)$ is called the reference transfer function.

4.2 Sensitivity

The ratio of change of the transfer function $T(s)$ by the parameter b can be defined as

System characteristics change with system parameter variations $S = \frac{\Delta T(s)}{T(s)} \cdot \frac{b}{\Delta b}$.

The sensitivity function is defined as: $S_b^T = \lim_{\Delta b \longrightarrow 0} \frac{\Delta T(s)}{\Delta b} \cdot \frac{b}{T(s)} = \frac{T(s)}{b} \cdot \frac{b}{T(s)}$.

Or, in general, sensitivity function of a characteristic W with respect to the parameter b:

$S_b^W = \frac{W}{b} \cdot \frac{b}{W}$.

Example: Plant with proportional feedback given by $G_c(s) = K_p$ and $G_p(s) = \frac{K}{s+0.1}$

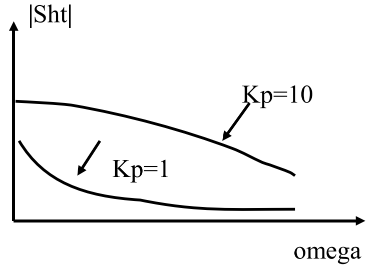


Fig. 22: Sensitivity as function of frequency for different feedback gains

Plant transfer function $T(s)$: $T(s)\frac{K_pG_p(s)}{1+K_pG_p(s)H_k}$

$S_H^T(j\omega) = \frac{-K_pG_p(j\omega)H_k}{1+K_pG_p(j\omega)H_k} = \frac{-0.25K_p}{0.1+0.25K_p+j\omega}$

Increase of H results in decrease of T. System cannot be insensitive to both H, T.

4.3 Disturbance rejection

Disturbances are system influences we do not control and whose impact on the system we want to minimize.

$C(s) = \frac{G_c(s).G_p(s)}{1+G_c(s).G_p(s).H(s)}.R(s) + \frac{G_d(s)}{1+G_c(s).G_p(s).H(s)}$

$= T(s).R(s). + T_d(s).D(s).$

To reject disturbances, make $T.d(s).D(s)$ small!

1. Use frequency response approach to investigate disturbance rejection.
2. In general $T_d(j\omega)$ cannot be small for all $-\omega$. Design $T_d(j\omega)$ small for significant portion of system bandwidth.
3. Reduce the gain $G_d(j\omega)$ between disturbance input and output.
4. Increase the loop gain $G_cG_p(j\omega)$ without increasing the gain $G_d(j\omega)$. Usually accomplished by the choice of $G_c(j\omega)$ compensator.

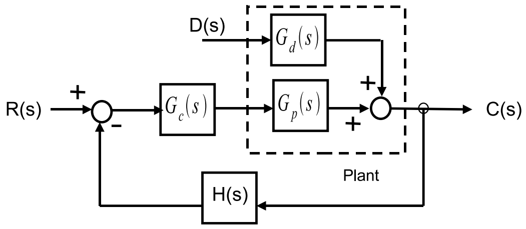


Fig. 23: Diagram with all transfer functions and signals used to define the disturbance rejection

5. Reduce the disturbance magnitude $d(t)$: should always be attempted if reasonable.
6. Use feed forward compensation, if disturbance can be measured.

5 Stability

In electrical engineering, specifically signal processing and control theory, BIBO stability is a form of stability for signals and systems. BIBO stands for Bounded-Input Bounded-Output. If a system is BIBO stable then the output will be bounded for every input to the system that is bounded.

Now we have learnt so far that
the impulse response tells us everything about the system response to arbitrary input signal $u(t)$.

What we have not learnt:
If we know the transfer function $G(s)$, how can we deduce the system behaviour?

What can we say about the system stability?

Definition: A linear time invariant system is **BIBO** stable (Bounded-Input-Bounded-Output) for all bounded inputs $|u(t)| \leq M_1$ (for all t) exists a boundary for the output signal M_2, so that $|y(t)| \leq M_2$ (for all t) with M_1 and $|M_2|$ are positive real numbers.

If input never exceeds M_1 and output never exceeds M_2 then we have BIBO stability!

Note: It has to be valid for ALL bounded input signals!

Example: $Y(s) = G(s)U(s)$, integrator $G(s) = \frac{1}{s}$

1. Case $u(t) = \delta(t), U(s) = 1$

   $|y(t)| = |L^{-1}[Y(s)]| = |L^{-1}[\frac{1}{s}]| = 1$
2. Case $u(t) = 1, U(s) = \frac{1}{s}$

   $|y(t)| = |L^{-1}[Y(s)]| = |L^{-1}[\frac{1}{s^2}]| = t$

BIBO stability has to be proven for any input. It is not sufficient to show its validity for a single input signal!

Condition for BIBO stability:

We start from the input–output relation

$Y(s) = G(s)U(s).$

By means of the convolution theorem we get

$|y(t)| = |\int_0^t g(\tau)u(t-\tau)d\tau| \leq \int_0^t |g(\tau)||u(t-\tau)|d\tau \leq M_1 \int_0^\infty |g(\tau)|d\tau \leq M_2.$

Therefore it follows immediately that if the impulse response is absolutely integrable

$\int_0^\infty |g(t)|dt < \infty,$

then the system is BIBO stable.

5.1 Poles and zeros

Can stability be determined if we know the TF of a system?

$$G(s) = C\phi(s)B + D = C\frac{[sI-A]_{adj}}{\chi(s)}B + D$$

Coefficients of transfer function $G(s)$ are rational functions in the complex variable s

$$g_{ij}(s) = a.\frac{\Pi_{k=1}^{m}(s-z_k)}{\pi_{l=1}^{n}(s-p_l)} = \frac{N_{ij}(s)}{D_{ij}(s)}$$

z_k zeros, p_l poles, α real constant, and it is $m \leq n$ (we assume common factors have already been cancelled!).

What do we know about the zeros and the poles?

Since numerator $N(s)$ and denominator $D(s)$ are polynomials with real coefficients, poles and zeros must be real numbers or must arise as complex conjugated pairs!

5.2 Stability directly from state-space

Recall that: $H(s) = C(sI - A)^{-1}B + D$.

Assuming $D = 0$ (D could change zeros but not poles),

$$H(s) = \frac{C_{adj}(sI-A)B}{\text{det}(sI-A)} = \frac{b(s)}{a(s)}.$$

Assuming there are no common factors between the poly $C_{adj}(sI - A)B$ and det$(sI - A)$ i.e., no pole-zero cancellations (usually true, system called 'minimal') then we can identify
and

$$b(s) = C_{adj}(sI - A)B$$

$$a(s) = \text{det}(sI - A)$$

i.e., poles are root of det$(sI - A)$.

Let λ_i be the i-th eigen value of A if $\mathbf{Re}(\lambda_i) \leq 0$ for all $i \Longrightarrow$ system stable.

So with a computer, with eigenvalue solver, one can determine system stability directly from coupling Matrix A.
A system is BIBO stable if, for every bounded input, the output remains bounded with increasing time.

For a LTI system, this definition requires that all poles of the closed-loop transfer-function (all roots of the system characteristic equation) lie in the left half of the complex plane.

5.3 Several methods are available for stability analysis:

1. Routh Hurwitz criterion
2. Calculation of exact locations of roots

  - root locus technique
  - Nyquist criterion
  - Bode plot

3. Simulation (only general procedures for nonlinear systems)

While the first criterion proofs whether a feedback system is stable or unstable, the second method also provides information about the setting time (damping term). Pole locations tell us about impulse response i.e., also stability:

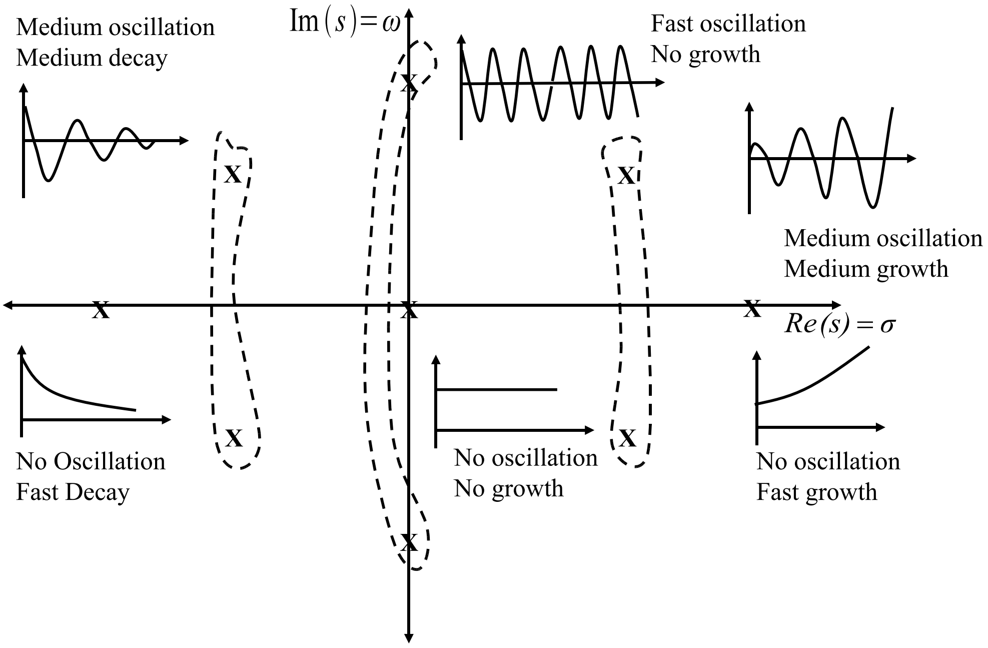


Fig. 24: Stability diagram in the complex plane. The system is stable if the poles are located in the left half of the plane. Oscillations occur for conjugate complex poles.

Furthermore: Keep in mind the following picture and facts!

1. Complex pole pair: Oscillation with growth to decay
2. Real pole: exponential growth or decay
3. Poles are the eigenvalues of the matrix A.
4. Position of zeros goes into the size of c_j....

In general a complex root must have a corresponding conjugate root $(N(s), D(s))$ polynomials with real coefficients.

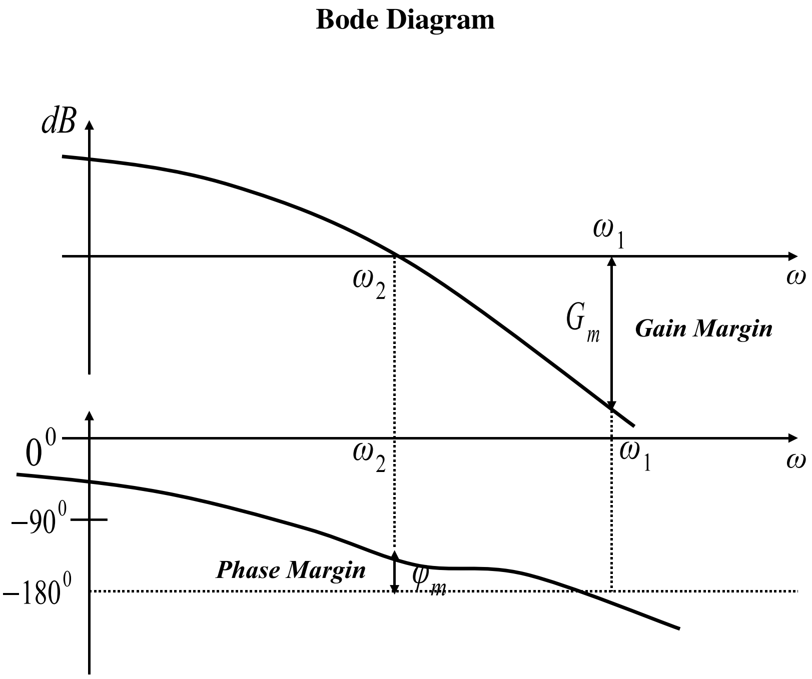


Fig. 25: Bode diagram showing phase and gain margins for stable systems

The closed loop is stable if the phase of the unity crossover frequency of the *open loop* is larger than -180 degrees.

5.4 Root locus analysis

Definition: A root locus of a system is a plot of the roots of the system characteristic equation (the poles of the closed-loop transfer function) while some parameter of the system (usually the feedback gain) is varied. $KH(s) = \frac{K}{(s-p_1)(s-p_2)(s-p_3)}$.

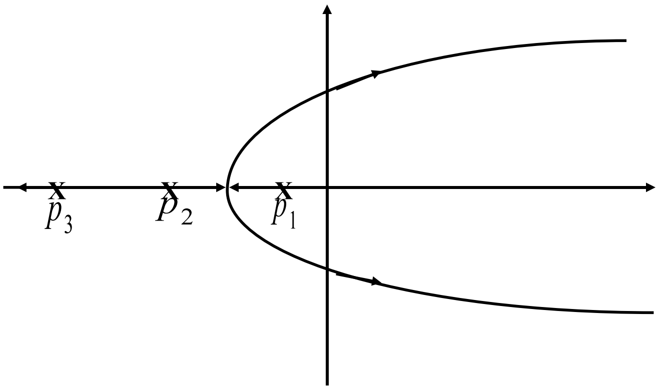


Fig. 26: Concept of root locus method

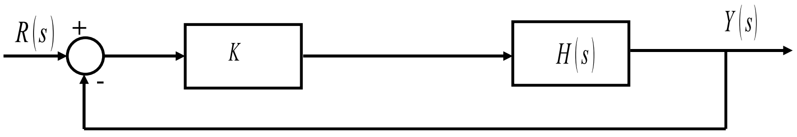


Fig. 27: Transfer function model of the closed-loop system used for the root locus method

$G_{cl}(s) = \frac{KH(s)}{1+KH(s)}$ roots at $1 + KH(s) = 0$.

How do we move the poles by varying the constant gain K?

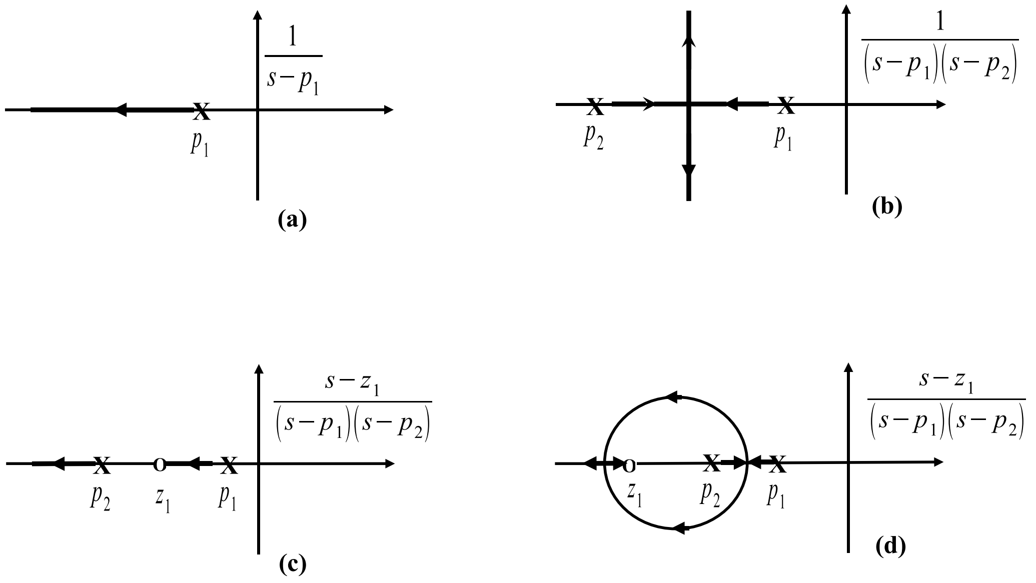

$\frac{1}{s-p_1}$
p_1
(a)
$\frac{1}{(s-p_1)(s-p_2)}$
p_2
p_1
(b)
$\frac{s-z_1}{(s-p_1)(s-p_2)}$
p_2
z_1
p_1
(c)
$\frac{s-z_1}{(s-p_1)(s-p_2)}$
z_1
p_2
p_1
(d)

Fig. 28: Root locus curves: some typical transfer functions

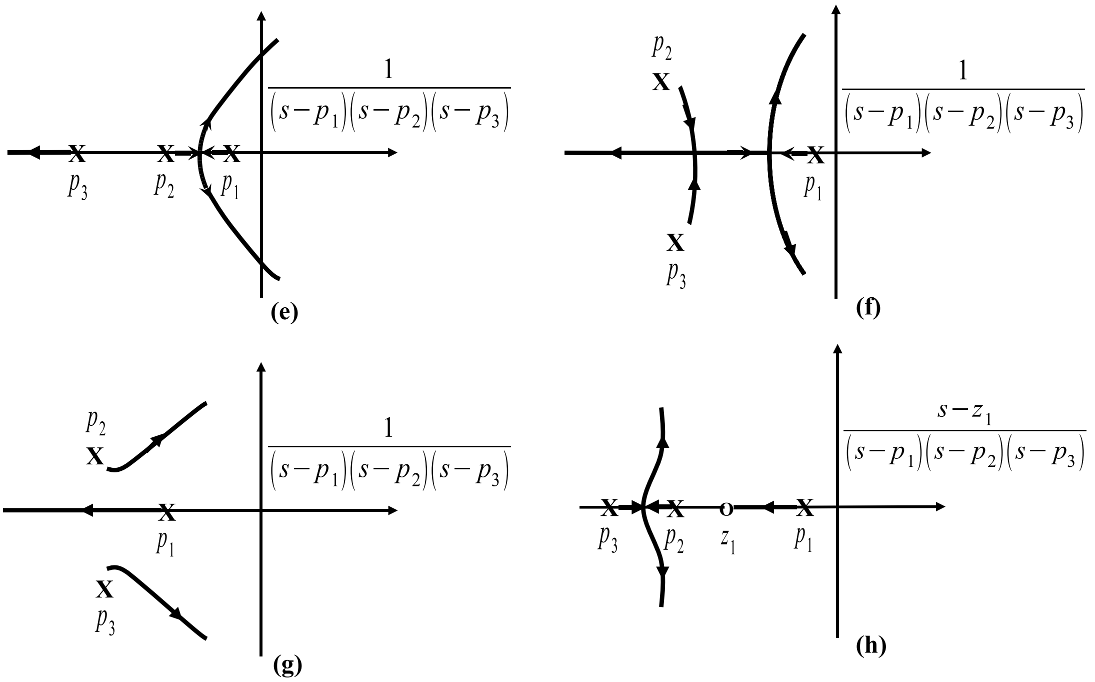

$\frac{1}{(s-p_1)(s-p_2)(s-p_3)}$
p_3
p_2
p_1
(e)
p_2
$\frac{1}{(s-p_1)(s-p_2)(s-p_3)}$
p_1
p_3
(f)
p_2
$\frac{1}{(s-p_1)(s-p_2)(s-p_3)}$
p_1
p_3
(g)
$\frac{s-z_1}{(s-p_1)(s-p_2)(s-p_3)}$
p_3
p_2
z_1
p_1
(h)

Fig. 29: More root locus curves: some typical transfer functions

6 Feedback

Feedback is both a mechanism, process, and signal that is looped back to control a system within itself. This loop is called the feedback loop. A control system usually has input and output to the system; when the output of the system is fed back into the system as part of its input, it is called the ‘feedback’. Feedback and regulation are self-related. The negative feedback helps to maintain stability in a system in spite of external changes.

The idea:
Suppose we have a system or ‘plant’

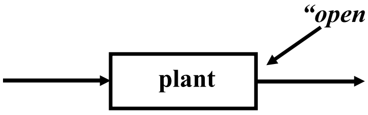


Fig. 30: Plant with input and output signals. Open-loop diagram.

We want to improve some aspect of the plant’s performance by observing the output and applying an appropriate ‘correction’ signal. **This is done by the feedback**.

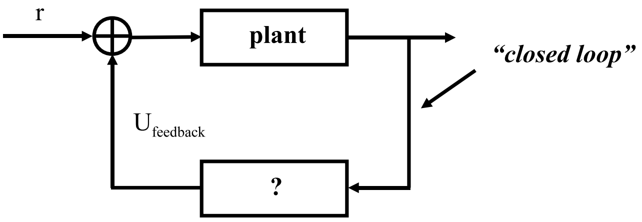


Fig. 31: Closed-loop system with reference input and plant output

Question: What should this be?

Open-loop gain: $G^{O.L}(s) = G(s) = (\frac{u}{y})^{-1}$

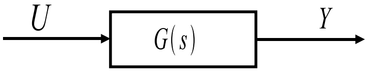


Fig. 32: Transfer functions and signals in the open-loop system

Closed-loop gain: $G^{C.L}(s) = \frac{G(s)}{1+G(s)H(s)}$

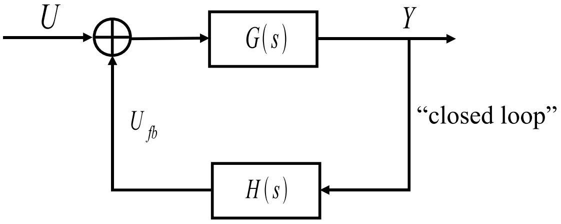


Fig. 33: Transfer functions and signals in the closed-loop system

Proof: $y = G(u - u_{fb}) = Gu - Gu_{fb} \Longrightarrow y + GH_y = Gu$

$= Gu - GH_y \implies \frac{y}{u} = \frac{G}{(1+GH)}$.

6.1 Simple harmonic oscillator

Consider a simple harmonic oscillator with feedback proportional to x i.e.:

where

$\ddot{x} + \gamma\dot{x} + \omega_n^2 x = u + u_{fb}$

$u_{fb}(t) = -\alpha x(t)$.

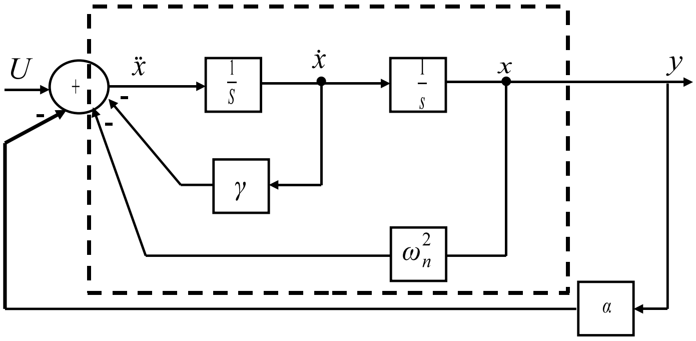


Fig. 34: Simple harmonic oscillator with proportional feedback

Then

$\ddot{x} + \gamma\dot{x} + \omega_n^2 x = u - \alpha x$

$\implies \ddot{x} + \gamma\dot{x} + (\omega_n^2 + \alpha)x = u$.

Same as before, except that 'natural' frequency $\omega_n^2 + \alpha$

now the closed-loop transfer function is $G^{C.L.}(s) = \frac{1}{s^2+\gamma s+(\omega_n^2+\alpha)}$.

So the effect of the proportional feedback in this case is *to increase the bandwidth of the system* (and reduce gain slightly, but this can easily be compensated for by adding a constant gain in front).

6.1.1 Simple harmonic oscillator with integral feedback

Suppose we use *integral feedback* in a simple harmonic oscillator

$u_{fb}(t) = -\alpha \int_0^t x(\tau)d\tau$

i.e. $\ddot{x} + \gamma\dot{x} + \omega_n^2 x = u - \alpha \int_0^t x(\tau)d\tau$

Differentiating once more yields $\dddot{x} + \gamma\ddot{x} + \omega_n^2\dot{x} + \alpha x = \dot{u}$.

No longer just simple SHO add another state $G^{C.L.}(s) = \frac{\frac{1}{s^2+\gamma s+\omega_n^2}}{1+(\frac{\alpha}{s})(\frac{1}{s^2+\gamma s+(\omega_n^2+\alpha)})} = \frac{s}{s(s^2+\gamma s+\omega_n^2)+\alpha}$

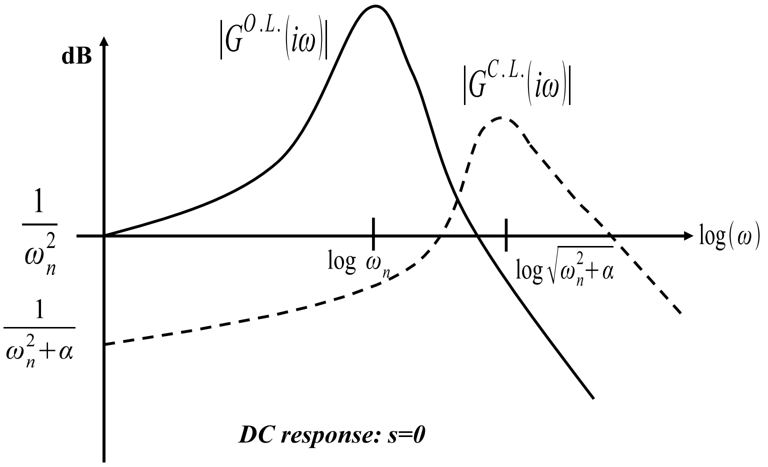


Fig. 35: Bode plot of SHO with proportional feedback. Open and closed-loop transfer function are shown.

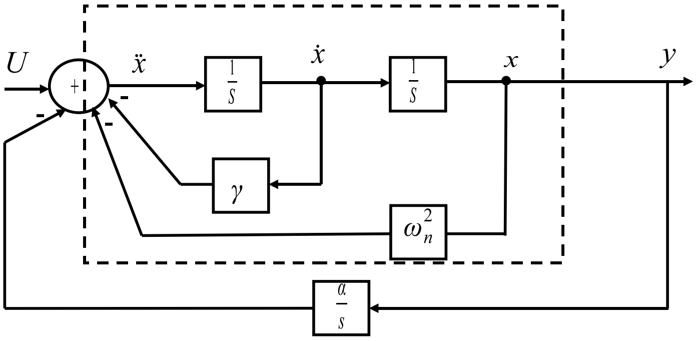


Fig. 36: Simple harmonic oscillator with integral feedback

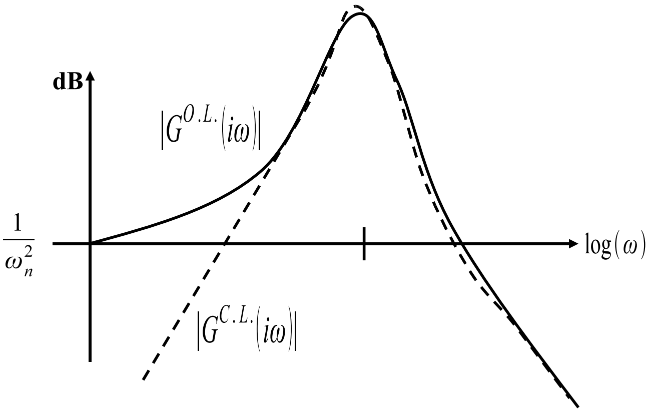


Fig. 37: Bode plot of SHO with integral feedback. Open and closed-loop transfer function are shown.

Observe that

1. $G^{C.L}(0 = 0)$.
2. For large s (and hence for large ω)

$G^{C.L}(s) \approx \frac{1}{(s^2+\gamma s+\omega_n^2} \approx G^{O.L.}(s)$.

So integral feedback has killed the DC gain i.e., the system *rejects constant* disturbances.

6.1.2 Simple harmonic oscillator with differential feedback

Suppose we now apply *differential feedback* in a simple harmonic oscillator i.e.

$u_{fb}(t) = -\alpha \dot{x}(t)$

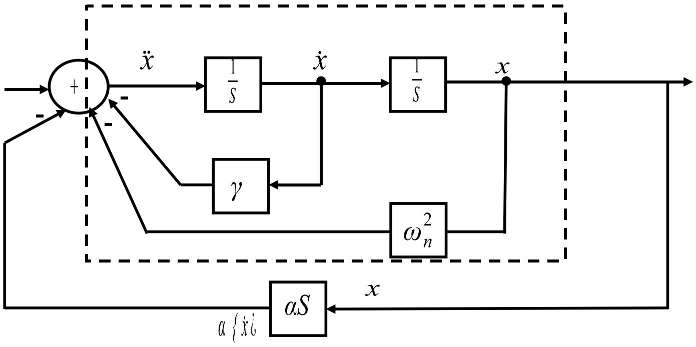


Fig. 38: Simple harmonic oscillator with differential feedback

Now we have $\ddot{x} + (\gamma + \alpha)\dot{x} + \omega_n^2 x = u$.

So the effect of differential feedback is to increase damping.

Now $G^{C.L}(s) = \frac{1}{s^2+(\gamma+\alpha)s+\omega_n^2}$.

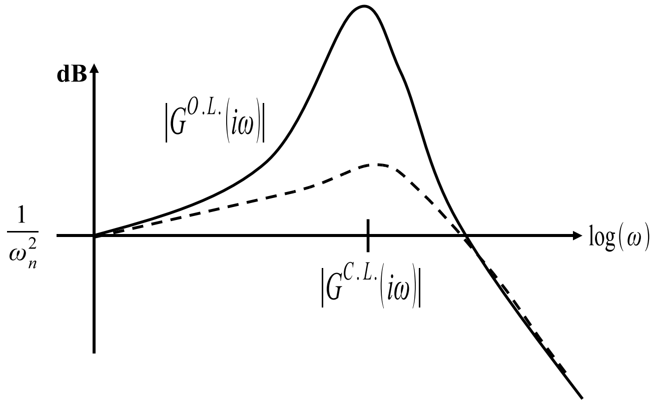


Fig. 39: Bode plot of SHO with differential feedback. Open and closed-loop transfer function are shown.

So the effect of differential feedback here is to 'flatten the resonance' i.e. **damping is increased**.

Note: Differentiators can never be built exactly, only approximately.

6.2 PID controller

1. The last three examples of feedback can all be combined to form a PID controller (proportional-integral-differential). $u_{fb} = u_p + u_d + u_l$

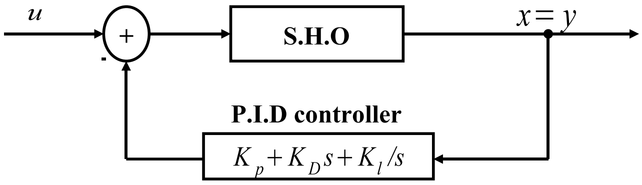


Fig. 40: SHO with PID controller

2. In the example above SHO was a very simple system and it was clear what the physical interpretation of P. or I. or D. did. But for *large complex systems* it is not obvious.

   $\Longrightarrow x$ **Require arbitrary 'tweaking'**

   That is what we are trying to avoid.

For example, if you are so clever let us see you do this with your PID controller:

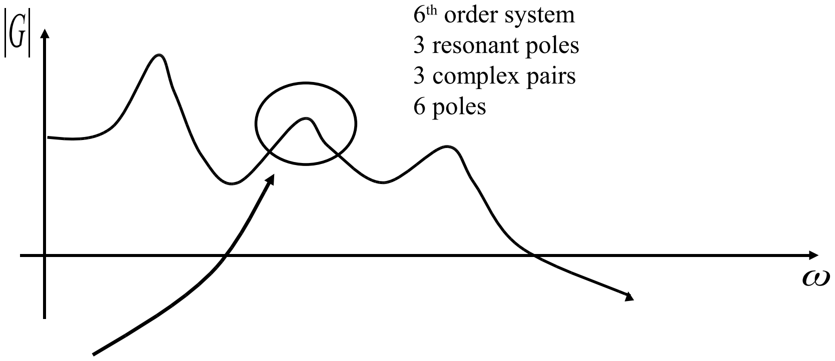


Fig. 41: PID controller adjustment problem

Damp this mode, but leave the other two modes undamped, just as they are.

This could turn out to be a tweaking nightmare. It will get you nowhere fast!

We shall see how this problem can be solved easily.

6.3 Full state feedback

Suppose we have a system $\dot{x}(t) = Ax(t) + Bu(t)$, $y(t) = Cx(t)$.

Since the state vector $x(t)$ contains all current information about the system the most general feedback makes use of **all** the state information.

$u - -K_1x_1 - ... - K_nx_n = -Kx$

where $k = [k_1.....k_n]$ (Row matrix)

Where example: In SHO examples

Proportional feedback: $u_p = -k_p x = -\left[\begin{array}{cc} K_p & 0 \end{array}\right]$

Differential feedback: $u_D = -k_D \dot{x} = -\left[\begin{array}{cc} 0 & k_D \end{array}\right]$

Theorem:

If there are no pole cancellations in $G_{O.L}(s) = \frac{b(s)}{a(s)} = C(sI - A)^{-1}B$,

then we can move the eigen values of $A - BK$ anywhere we want using full state feedback.

Proof: Given **any** system as L.O.D.E or state space it can be written as

$$\begin{bmatrix} x_1 \\ ... \\ ... \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & ... & 0 \\ 0 & ... & ... & ... \\ 0 & ... & ... & 1 \\ -a_0 & ... & ... & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ ... \\ ... \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ .. \\ 1 \end{bmatrix} u$$

$$y = \left[\begin{array}{cccc} b_0 & ... & ... & b_{n-1} \end{array}\right] \begin{bmatrix} x_1 \\ ... \\ ... \\ x_n \end{bmatrix}$$

where $G^{O.L.} = C(sI - A)^{-1}B = \frac{b_{n-1}s^{n-1} + ... + b_0}{s^n + a_{n-1}s^{n-1} + ... + a_0}$ i.e. the first row of $A^{O.L}$ gives the coefficients of the denominator

$a^{O.L}(s) = \det(sI - A^{O.L.}) = s^n + a_{n-1}s^{n-1} + ... + a_0.$

$$\text{Now } A^{C.L.} = A^{O.L.} - BK = \begin{bmatrix} 0 & 1 & ... & 0 \\ 0 & ... & ... & 1 \\ 0 & ... & ... & 1 \\ -a_0 & ... & ... & -a_{n-1} \end{bmatrix} -$$

$$\begin{bmatrix} 0 \\ 0 \\ ... \\ 1 \end{bmatrix} \left[\begin{array}{cccc} k_0 & ... & ... & K_{n-1} \end{array}\right] \begin{bmatrix} 0 & 1 & ... & 1 \\ 0 & ... & ... & ... \\ 0 & ... & ... & 1 \\ -(a_0 + k_0) & ... & ... & -(a_{n-1} + k_{n-1}) \end{bmatrix}$$

So the closed-loop denominator

$a^{C.L.}(s) = \det(sI - A^{C.L}) = s^n + (a_0 + k_0)s^{n-1} + ... + (a_{n-1} + k_{n-1}).$

Using $u = -Kx$ we have direct control over every closed-loop denominator coefficient. We can place the root anywhere we want in the s plane.

Example: Detailed block diagram of SHO with full state feedback.

Of course this **assumes** we have access to the $\dot{x}$ state, which we actually do not in practice.

However, let us ignore that 'minor' practical detail for now. (The Kalman filter will show us how to get $\dot{x}$ form x.)
With full state feedback we have (assume D = 0).

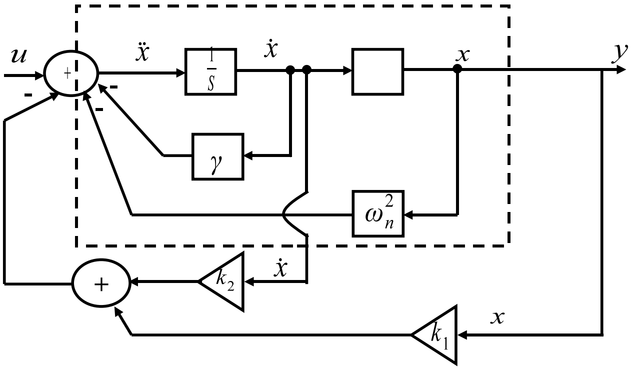


Fig. 42: SHO with full state feedback

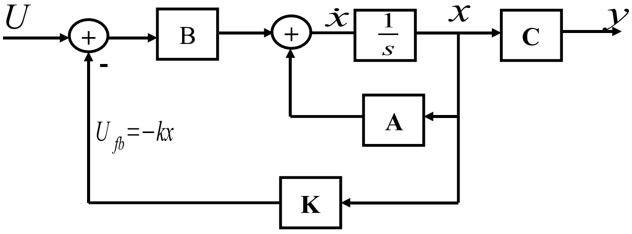


Fig. 43: Diagram of state space system with full state feedback

So

$\dot{x} = Ax + B[u + u_{fb}] = Ax + Bu + BKu_{fb}$

$\dot{x} = (A - BK)x + Bu$

$u_{fb} = -Kx$

$y = Cx$

With full state feedback, we get a new closed loop matrix.

$A^{C.L} = (A^{O.L} - BK)$

Now all stability information is given by the eigen values of the new A matrix.
The linear time-invariant system $\dot{x} = Ax + Bu$ $y = Cx$ is said to be controllable if it is possible to find some input $u(t)$ that will transfer the initial state $x(0)$ to the origin of state-space, $x(t_0) = 0$, with t_0 finite.

The solution of the state equation is

$x(t) = \phi(t)x(0) + \int_0^t \phi(\tau)Bu(t - \tau)d\tau.$

For the system to be controllable, a function $u(t)$ must exist that satisfies the equation

$0 = \phi(t_0)x(0) + \int_0^{t_0} \phi\tau Bu(t_0 - \tau)d\tau.$

With t_0 finite, it can be shown that this condition is satisfied if the controllability matrix

$C_M = [B\ AB\ A^2B\ ....A^{n-1}B]$ was inverse. This is equivalent to the matrix C_M having full rank (rank n for an n^{th} order differential equation).
Observable:

- The linear time-invarient system is said to be observable if the initial conditions $x(0)$ can be determined from the output function $y(t), 0 \leq t \leq t_1$ where t_1 is finite with $y(t) = Cx = C\phi(t)x_0 + C\int_0^t \phi(\tau)Bu(t-\tau)d\tau$.
- The system is observable if this equation can be solved for $x(0)$. It can be shown that the system is observable if the matrix: $O_M = \begin{bmatrix} C \\ CA \\ ... \\ CA^{n-1} \end{bmatrix}$.
- Was inverse. This is equivalent to the matrix C_M having full rank (rank n for an n-th order differential equation).

7 Discrete systems

Digital control is a branch of control theory that uses digital computers to act as a system. Depending on the requirements, a digital control system can take the form of a microcontroller to an ASIC to a standard desktop computer. Since a digital computer is a discrete system the Laplace transform is replaced with the Z-transform. Also since a digital computer has finite precision (see Quantization) extra care is needed to ensure the error in coefficients, A/D conversion, D/A conversion, etc. is not producing undesired or unplanned effects.

A discrete system or discrete-time system, as opposed to a continuous-time system, is one in which the signals are sampled periodically. It is usually used to connote an analog sampled system, rather than a digital sampled system, which uses quantized values.

Where do discrete systems arise?

Typical control engineering example: Assume the DAC+ADC are clocked at sampling period T.
Then $u(t)$ is given by: $u(k) \equiv u_c(t); kT \leq t < (k+1)T$

$y(k) \equiv y_c(kT); k = 0, 1, 2, ...$

Suppose: time continuous system is given by state-space $\dot{x_c}(t) = Ax_c(t) + Bu_c(t); x_c(0) = x_0$

$y_c(t) = Cx_c(t) + Du_c(t)$.

7.1 State space description

Can we obtain a direct relationship between $u(k)$ and $y(k)$? We want an equivalent discrete system:
Yes! we can obtain an equivalent discrete system.
Recall $\quad x_x(t) = e^{At}x_c(0) + \int_0^t e^{A\tau}.Bu_c(t-\tau)d\tau$.

From this $\quad x_c(kT+T) = e^{AT}x_c(kT) + \int_0^T e^{A\tau}.Bu_c(kT-\tau)d\tau$.

Observe that $\quad u(kT+T-\tau) = u(kT)$ for $\tau\epsilon[0,T]$ i.e. $u(kT+T-\tau)$ is constant $u(kT)$ over

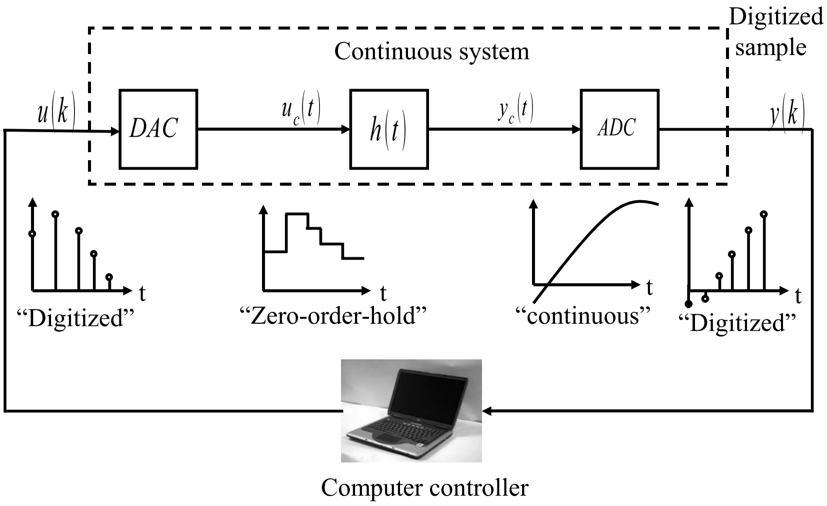


Fig. 44: Schematic of digital controller with analog-to-digital and digital-to-analog converters. The continuous and discretized signals are shown in the feedback loop.

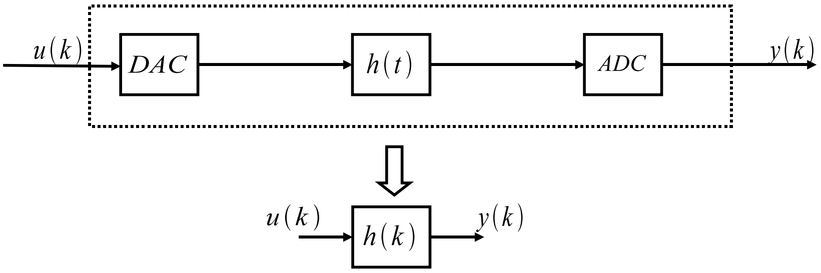


Fig. 45: Conversion from continuous system to discrete transfer function

$\tau\epsilon[0,T]$

can pull out of integral

$\Longrightarrow x_c(kT+T) = e^{At}x_c(kT) + (\int_0^t e^{A\tau}.Bd\tau)u_c(kT)$

$x(k+1) = A_dx(k) + B_du(k)$

$y(k) = C_dx(k) + D_du(k)$

$x(0) = x_c(0)$.

So $A_d = e^{AT}$,$B_d = \int_0^T e^{A\tau}.Bd\tau BD_d = C$, $D_d = D$.

So we have an exact (note: $x(k+1) = x(k) + \dot{x}(k)T + O(.)$) discrete time equivalent to the time continuous system at sample times $t = kT-$ no numerical approximation! A linear ordinary difference equation looks similar to a LODE

$y(k+n)+a_{n-1}y(k+n-1)+...+a_1y(k+1)+a_0y(k) = b_mu(k+m)+...+b_1u(k+1)+b_0u(k)n \geq m$; assumes initial values $yy(n-1),...y(1),y(0) = 0$.

7.2 Z transform

Z transform of the LODE yields (linearity of Z transform):

$$z_nY(z) + z^{n-1}a_{n-1}Y(z) + ... + za_1y(z) + a_0Y(z) = z_mb_mU(z) + .. + zb_1U(z) + b_0U(z)$$

it follows the input–output relation:

$$(z^n + z^{n-1}a_{n-1} + ... + za_1 + a_0)Y(z) = (z^mb_m + ... + zb_1 + b_0)U(z)$$

$$Y(z) = \frac{z^mb_m+...+zb_1+b_0}{z^n+...+za_1+a_0}U(z)$$

$$Y(z) = G(z)U(z).$$

Once again: if $U(z) = 1, (u(k) = \delta(k))$, then $Y(z) = G(z)$.

Transfer function of system is the Z transform of its pulse response!
$x(k + 1) = A_dx(k) + B_du(k)$

$y(k) = Cx(k) + Du(k).$

Applying the Z transform on first equation:

$$z.X(z) - zx(0) = A_dX(z) + B_dU(z)$$

$$(zI - A_d)X(z) = zx(0) + Bu(z)$$

$X(z) = (zI - A)^{-1}zx(0) + (zI - A_d)^{-1}BU(z).$ Homogeneous solution.

NOW: Particular solution:

$$Y(z) = CX(z) + DU(z) = C(zI - A_d)^{-1}zx(0) + (C(zI - A_d)^{-1}B + D)U(z)$$

if $x(0) = 0$ then we get the input–output relation:

$Y(z) = G(z)U(z)$ with $G(z) = C(zI - A_d)^{-1}B + D.$

Exactly as for the continuous systems.

To analyse discrete-time systems: the Z transform (analog to Laplace transform for time-continuous system).
It converts linear ordinary difference equation into algebraic equations: easier to find a solution of the system!
It gives the frequency response for free!
Z transform == generalized discrete-time Fourier transform

Given any sequence $f(k)$ the discrete-time Fourier tranform is $F(\tilde{\omega}) = \Sigma_{k=-\infty}^{\infty}f(k)e^{-i\omega k}$

$\omega = 2\Pi f, f = \frac{1}{T}$. The sampling frequency in Hz, T : difference/time between two samples.

In the same spirit: $F(z) = Z[f(k)] = \Sigma_{k=0}^{\infty}f(k)z^{-k}$

with z a complex variable. Note: if $f(k) = 0$ for $k = -1, -2, ...$ then $\tilde{F}(\omega) = F(z = e^{j\omega})$.

7.3 Stability

A discrete LTI system is BIBO stable if $|u(k)| < M, \forall k = |y(k)|; \forall k$

$|y(k)| = |\Sigma_0^k u(k-i)h(i)| \leq \Sigma_0^k |u(k-i)||h(i)| \leq M\Sigma_0^k|h(i)| \leq M\Sigma_0^\infty|h(i)|$.
For LODE state space system: $H(z) = \alpha\frac{\pi_{i=1}(z-z_i)}{\pi_{i=1}^n(z-p_i)} = \Sigma_{i=1}^k \beta_i T_i(z)$.

With partial fraction of the rational function, once again pole locations tell us a lot about the shape of pulse response.
Zeros determine the size of β_i. In general a complex pair → oscillatory growth/damping.

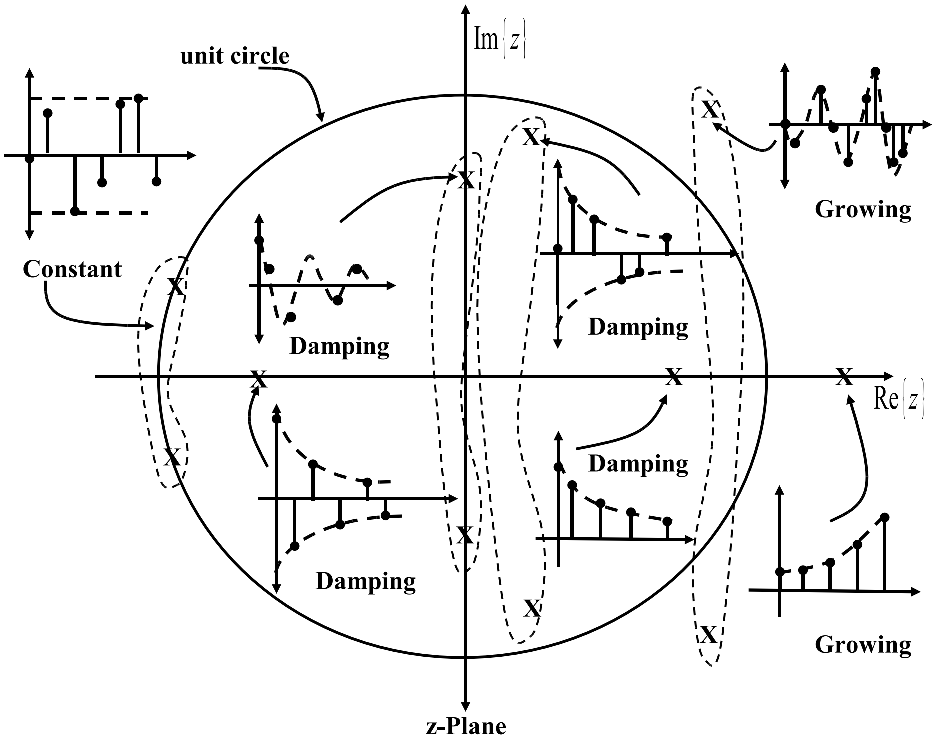


Fig. 46: Stability diagram for discrete system. The system is stable if the poles are located inside the unity circle.

A real pole → exponential growth/decay but may be oscillatory too (e.g. $r^n 1(n)$ where $r < 0$).

The farther inside the unit circle → the faster the damping → the higher the stability, i.e. $|p_i| \leq 1$ → system stable. Stability directly from state space: exactly as for cts systems, assuming no pole-zero cancellations and $D = 0$

$$H(z) = \frac{b(z)}{a(z)} = C(zI - A_d)^{-1}B_d = \frac{C_{adj}(zI - A_d)B_d}{det(zI - A_d)}$$

$$b(z) = C_{adj}(zI - A_d)B_d \quad a(z) = det(zI - A_d).$$

The poles are eigenvalues of A_d. So check stability, use an eigenvalue solver to get eigenvalues of the matrix A_d, then if $|\lambda_i| < 1$ for all i then the system is stable. Where $|\lambda_i|$ is the i^{th} e-value of A_d.

7.4 Cavity model

Converting the transfer function from the continuous cavity model to the discrete model:

$$H(s) = \tfrac{\omega_{12}}{\Delta\omega^2+(s+\omega_{12})^2}\begin{bmatrix} s+\omega_{12} & -\Delta\omega \\ \Delta\omega & s+\omega_{12} \end{bmatrix}.$$

The discretization of the model is represented by the z transform:

$$H(z) = (1-\tfrac{1}{z})z(\tfrac{H(s)}{s}) = \tfrac{z-1}{z}.zL^{-1}\tfrac{H(s)}{s}|_{t=kT_s}$$

$$H(z) = \frac{\omega_{12}}{\Delta\omega^2+\omega_{12}^2}.\begin{bmatrix} \omega_{12} & -\Delta\omega \\ \Delta\omega & \omega_{12} \end{bmatrix} - (\frac{\omega_{12}}{\Delta\omega^2+\omega_{12}^2}.\frac{z-1}{z^2-2ze^{\omega_{12}T_s}.cos(\Delta\omega T)_s+e^{2\omega_{12}T_s}})$$
$$.((z-e^{\omega_{12}T_s}.cos(\Delta\omega T_s)).\begin{bmatrix} \omega_{12} & -\Delta\omega \\ \Delta\omega & \omega_{12} \end{bmatrix}) - e^{\omega_{12}T_s}.sin(\Delta\omega T_s).\begin{bmatrix} \Delta\omega & \omega_{12} \\ -\omega_{12} & \Delta\omega \end{bmatrix}.$$

Given: $x(k+1) = Ax(k) + Bu(k)$

$z(k) = Cx(k)$. Assume $D = 0$ for simplicity.

8 Optimal control:

Optimal control deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. A control problem includes a cost functional that is a function of state and control variables. An optimal control is a set of differential equations describing the paths of the control variables that minimize the cost functional.

Suppose the system is unstable or almost unstable. We want to find $u_{fb}(k)$ which will bring $x(k)$ to zero, quickly, from any initial condition.

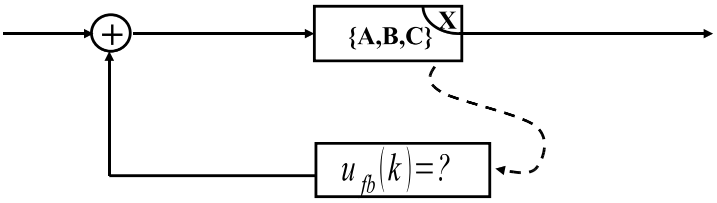


Fig. 47: Formulation of the problem of the optimal controller

A quadratic form is a quadratic function of the components of a vector:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad f(x) = f(x_1, x_2) = ax_1^2 + bx_1x_2 + cx_1 + dx_2^2$$

$$[x_1, x_2]\begin{bmatrix} a & \frac{1}{2}b \\ \frac{1}{2}b & d \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$f(x) = x^TQx$, quadratic part, $+P^Tx$, linear part, $+e$ constant.

What do we mean by ‘bad’ damping and ‘cheap’ control? We now define precisely what we mean. Consider:

$$J \equiv \Sigma_{i=0}^{\infty} x_i^TQx_i + u_i^TRu_i.$$

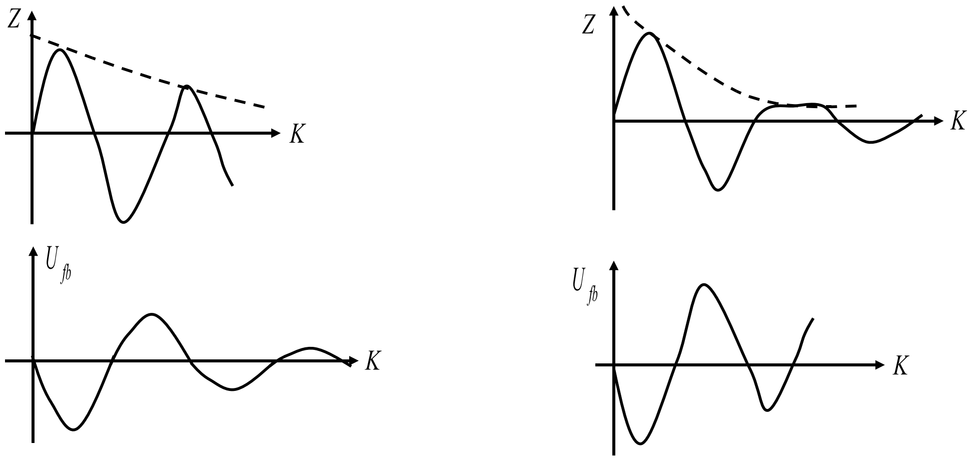


Fig. 48: Tradeoff between bad damping and cheap control (left side) and good damping and expensive control (right)

The first term penalizes large state executions, the second penalizes large control. $Q \geq 0, R > 0$

Can there be a tradeoff between state excursions and control by varying Q and R?

Large $Q \longrightarrow$ 'good' damping important

Large $R \longrightarrow$ actuator effort 'expensive'

(Linear quadratic regulator)

$x_{i+1} = Ax_i + Bu_i; \quad x_0$

Find control sequence $u_0, u_1, u_2, ...$ such that $J = \Sigma_{i=0}^{\infty} x_i^T Q x_i + u_i^T R u_i$ = minimum.

Answer: The optimal control sequence is a state feedback sequence $u_{i_0^{\infty}}$

$u_i = -K_{opt} x_i$

$K_{opt} = (R + B^T SB)^{-1} B^T SA$

$S = A^T SA + Q - A^T AB(R + B^T SB)^{-1} B^T SA$

Algebraic Riccati Equation (ARE) for discrete-time systems.

Note: Since $u_i =$ state feedback, it works for any initial state x_0.
Remarks:

1. So optimal control, $u_i = K_{opt} x_i$ is state feedback! This is why we are interested in state feedback.
2. Equation ARE is a matrix quadratic equation. Looks pretty intimidating but computer can solve in a second.
3. No tweaking! Just specify A, B, C, D and Q and R, press return button, LQR routine spits out $K_{opt}-$ done (of course guaranteed optimal in the sense that it minimizes).
4. Design is guaranteed optimal in the sense that it minimizes.

$J_{lqr}(x_0, u_i{}_0^{\infty} x_i^T Q x_i + u_i^T R u_i$

(Of course that does not mean it is the 'best' in the absolute sense.)
As we vary the Q/R ratio we get whole family of k_{lqr}'s, i.e., can trade off between state excursion (damping) vs actuator effort (control)

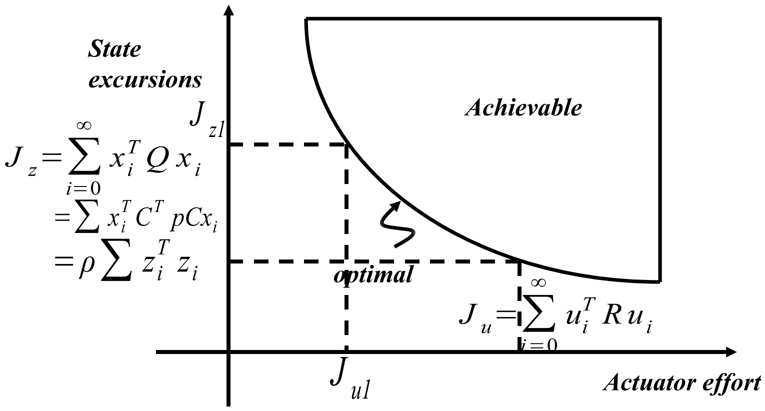


Fig. 49: Tradeoff between control effort and rms state excursions

Our optimal control has the form $u_{opt}(k) = -K(k)x_{opt}(k)$.

This assumes that we have complete state information $x_{opt}(k)$. This is not actually the case, e.g., in SHO, we might have only a position sensor but not a velocity sensor.

8.1 The Kalman filter

How can we obtain 'good' estimates of the velocity state by just observing the position state?

The sensors may be noisy and the plant itself may be subject to outside disturbances (process noise) i.e., we are looking for this:

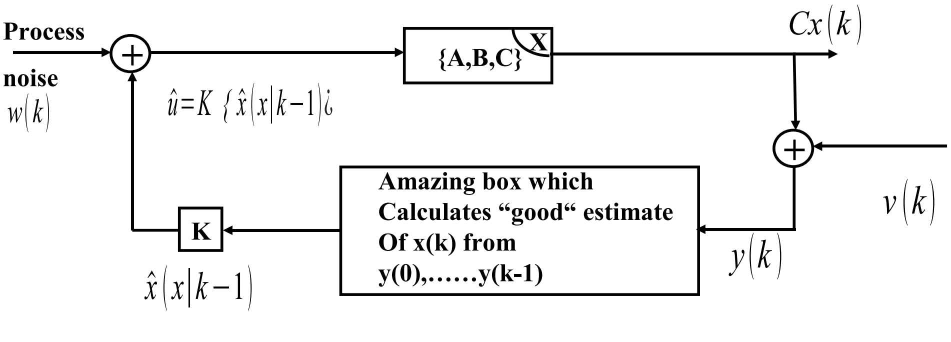


Fig. 50: Formulation of the problem to be solved by the Kalman filter

$x(k+1) = Ax(k) + Bw(k)$

$z(k) = Cx(k)$

$y(k) = Cx(k) + v(k)$. Assumes also $x(0)$ is random and Gaussian and that $x(k), w(k) + v(k)$ are

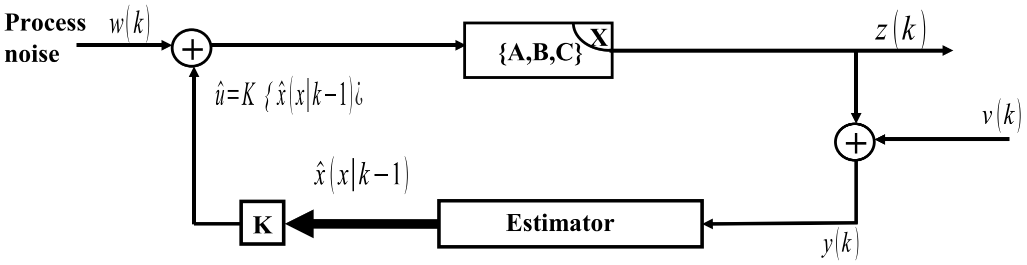


Fig. 51: Concept of the estimator which provides a best state estimate from model and noisy sensor signal

all mutually independent for all k.

Find: $\hat{x}(k|k-1)$ optimal estimate of $x(k)$ given $y_0, ..y_{k-1}$ such that 'mean squared error'.

$E[||x(k) - \hat{x}(k|k-1)||_2^2]$ = minimal.

Fact from statistics: $\hat{x}(k|k-1) = E[x(k)|(y_0, ..., y_{k-1})]$ The Kalman filter is an efficient algorithm that computes the new $\hat{x}_{i+1|i}$ (the linear-least-mean-square estimate) of the system state vector x_{i+1}, given $y_0, ..y_i$, by updating the old estimate $\hat{x}_{i|i-1}$ and old $x_{i|\tilde{i}-1}$ (error). The Kalman filter produces

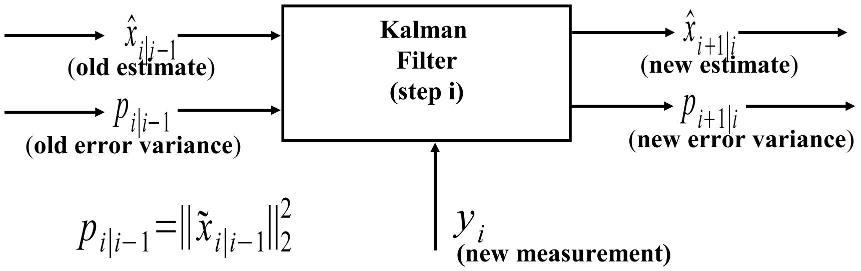


Fig. 52: Update algorithm for the Kalman filter

$\hat{x}_{i+1|i}$ from $\hat{x}_{i|i-1}$ (rather than $\hat{x}_{i|i}$, because it 'tracks' the system 'dynamics'. By the time we compute $\hat{x}_{i|i}$ from $\hat{x}_{i|i-1}$, the system state has changed from x_i.

The Kalman filter algorithm can be divided into a measurement update and a time update:

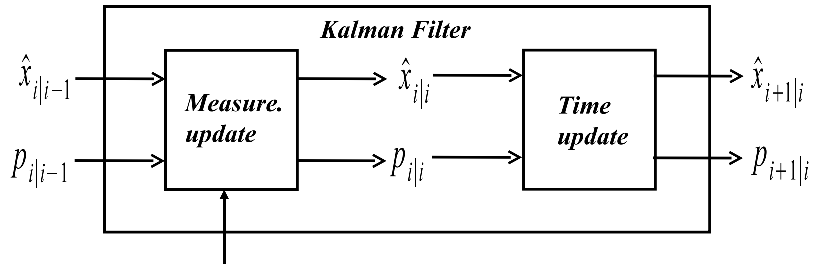


Fig. 53: Separation between measurement and time updates in the Kalman filter

Measurement update(M.U): $\hat{x}_{i|i} = \hat{x}_{i|i-1} + P_{i|i-1}C^T(Cp_{i|i-1}C^T + V)^{-1}(y_i - C\hat{x}_{i|i-1})$

$p_{i|i} = P_{i|i-1} - P_{i|i-1}C^T(CP_{i|i-1}C^T + V)^{-1}Cp_{i|i-1}$.

Time update (T.U.): $\hat{x}_{i+1|i} = A\hat{x}_{i|i}$ and $P_{i+1|i} = Ap_{i|iA^T} + BWB^T$.

With initial conditions: $\hat{x}_{0|-1} = 0$ and $p_{0\hat{|}-1} = X_0$.
By plugging M.U equations in to T.U equations one can do both steps at once:

$$\hat{x}_{i+1|i} = A\hat{x}_{i|i} = A\hat{x}_{i|i-1} + Ap_{i|i-1}C^T(Cp_{i|i-1}C^T + V)^{-1}(y_i - C\hat{x}_{i|i-1}).$$

$\hat{x}_{i+1|i} = A\hat{x}_{i|i-1} + L_i(y_i - C\hat{x}_{i|i-1})$ where $L_i \equiv A\left(p_{i|i-1}C^T(Cp_{i|i-1}C^T + V)^{-1}\right)$

$$p_{i+1|i} = Ap_{i|i}A^T + BWB^T = A[p_{i|i-1} - p_{i|i-1}C^T(Cp_{i|i-1}C^T + V)^{-1}CP_{i|i-1}]A^T + BWB^T$$

$$p_{i+1|i} = Ap_{i|i}A^T + BWB^T - Ap_{i|i-1}C^T(Cp_{i|i-1}C^T + V)^{-1}(Cp_{i|i-1} - 1)A^T$$

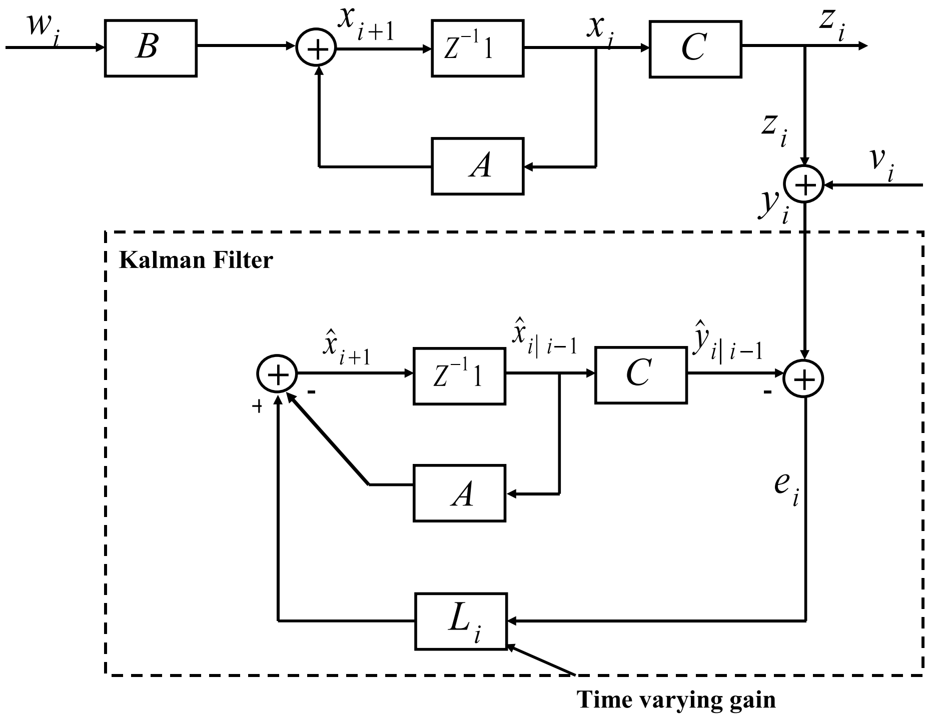


Fig. 54: Model of Kalman filter with time varying gain

Plant equations: $x_{i+1} = Ax_i + Bu_i$ and $y_i = Cx_i + v_i$.

Kalman filter: $\hat{x}_{i+1|i} = A\hat{x}_{i|i-1} + L_i(y_i - \hat{y}_{i|i-1})$ and $y_{i|i-1} = C\hat{x}_{i|i-1}$. If $v = w = 0$ Kalman filter can estimate the state precisely in a finite number of steps.
Remarks:

1. Since $y_i = Cx_i + v_i$ and $\hat{y}_{i|i-1} = C\hat{x}_i$ can write estimator equation as $\hat{x}_{i+1|i} = A\hat{x}_{i|i-1} + L_i(Cx_i + v_i - C\hat{x}_{i|i-1}) = (A - L_iC)x|i\hat{} - 1 + L_iCx_i + v_i$. We can combine this with equation for x_{i+1}

$$\begin{bmatrix} x_{i+1} \\ \hat{x}_{i+1|i} \end{bmatrix} = \begin{bmatrix} A & 0 \\ L_iC & A - L_iC \end{bmatrix} \begin{bmatrix} x_i \\ \hat{x}_{i|i-1} \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_i \\ v_i \end{bmatrix}$$
$$\begin{bmatrix} z_i \\ \hat{y}_{i|i-1} \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} x_i \\ \hat{x}_{i|i-1} \end{bmatrix}$$

2. In practice, the Riccati equation reaches a steady state in a few steps. People often run with steady-state Kalman filters, i.e.

   $L_s s = Ap_{ss}C^T(CP_{ss}C^T + V)^{-1}$,

   where $p_{ss} = Ap_{ss}A^T + BWB^T - Ap_{ss}C^T(CP_{ss}C^T + V)^{-1}CP_{ss}A$.

8.2 LQG controller

Now we are finally ready to solve the full control problem.

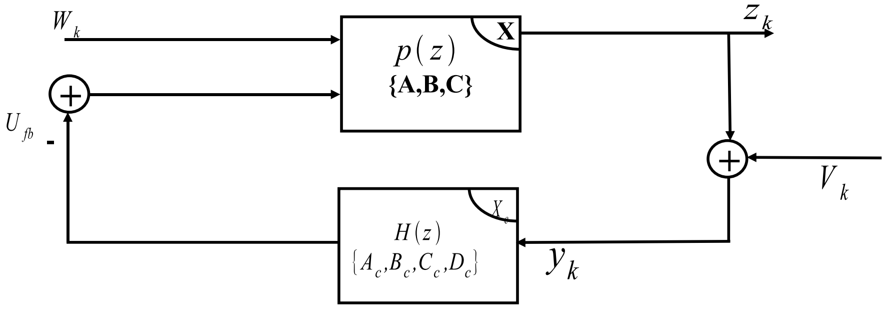


Fig. 55: LQG controller problem definiton

Given: $x_{k+1} = Ax_k + Bu_k + B_w w_k$, $z_k = Cx_k$, $y_k = Cx_k + v_k$, $\langle w_i, w_j\rangle = w\delta_{ij}, \langle v_i, v_j\rangle = V\delta_{ij}$, $\langle w_i, v_j\rangle = 0$, where w_k and v_k are both Gaussian. For Gaussian, K.L. gives the absolute best estimate.
Separation principle: (We will not prove this.) The separation principle states that the LGQ optimal controller is obtained by:

1. using a Kalman filter to obtain least-squares optimal estimate of the plant state,

   i.e., Let $x_c(k) = \hat{x}_{k|k-1}$.
2. Feedback estimated LQR-optimal state feedback

   $u(k) = -K_{LQR}x_c(k) = -K_{LQR}\hat{x}_{k|k-1}$. One can treat problems of optimal feedback and a state estimate separately.

Plant: $||x_{k+1} = Ax_k + (-Bu_k) + B_w w_k, z_k = Cx_k, y_k = Cx_k + v_k||$.

LQG Controller $||\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + Bu_k + L(y_k - C\hat{x}_{k|k-1}), u_k = -K\hat{x}_{k|k-1}$

where $k = -[R + B^TSB]^{-1} + s = A^TSA + Q - A^TSB[R + B^TSB]^{-1}B^TSA$, and

$L = APC^T[V + CPC^T]^{-1} + P = APA^T + BWB^T[V + CPC^T]^{-1}CPC^T$.
We want a controller which takes as input noisy measurements y and produces as output a feedback signal u which will minimize excursions of the regulated plant outputs (if no pole-zero cancellation, then this is equivalent to minimizing state excursions).

We also want to achieve 'regulation' with as little actuator effort u as possible.

Problem statement (mathematically) Find: controller $H(z) = C_c(zI - A_c)^{-1}B_c + D_c$.

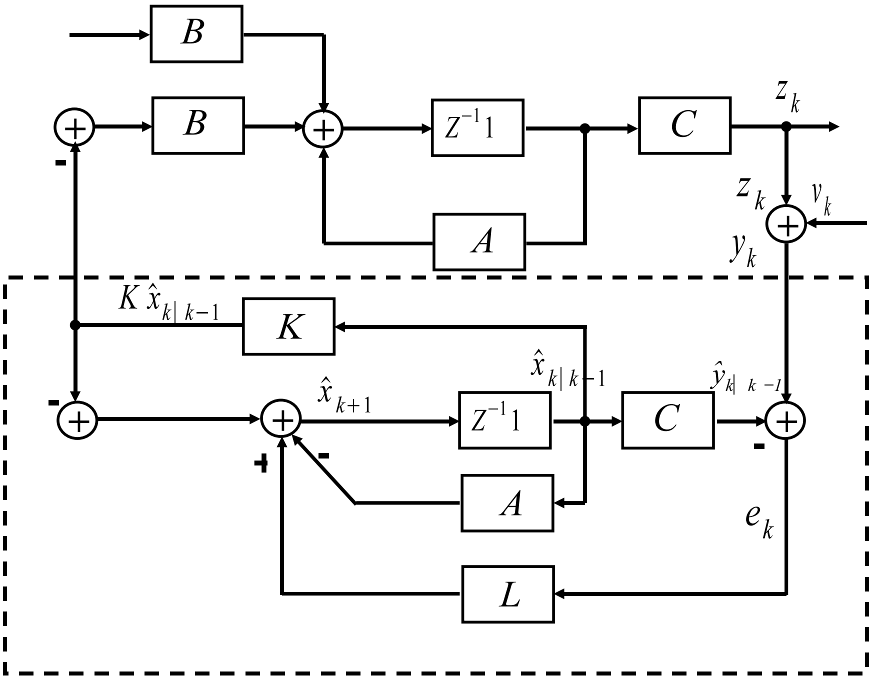


Fig. 56: Visual representation of LQG controller

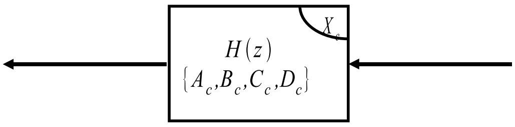


Fig. 57: LQG controller

Controller: $x_c(k+1) = A_c x(k+1) = A_c x(k+1) + B_c y(k), y_c(k) = C_c x_c(k)$ which will minimize the cost, where

$J_{LQG} =_{k\longrightarrow\infty}^{Limit} E[x_k^T Q x_k$ are r.m.s. ‘state’ excursions and $+u_k^T R u_k]$ is the r.m.s. ‘actuator’ effort.

$\|x_{k+1} = Ax_k + (-Bu_k) + B_w w_k, z_k = Cx_k, y_k = Cx_k + v_k\|$

Remarks:

1. Q and R are weighting matrices that allow trading off r.m.s. u and r.m.s. x.
2. If $Q = C^T \rho C; \rho > 0$ then trade off r.m.s. z vs r.m.s. u.
3. In the stochastic LQR case, the only difference is that now we do not have complete state information $y_i = Cx_i + v_i$ we have only noisy observations i.e. cannot use full state feedback.

   Idea: Could we use estimated state feedback? (i.e. $-K\hat{x}_{k|k-1}$)
4. We can let Q/R ratio vary and we will obtain the family of LQG controllers. We can plot r.m.s. Z r.m.s. u for each one.
   So by specifying (1) system model, (2) noise variance, (3) optimally criterion J_{LQG} and plotting trade off curve completely specifies the limit of performance of the system i.e. which combinations of (Z_{rms}, U_{rms}) are achievable by any controller’s good ‘benchmark curve’.

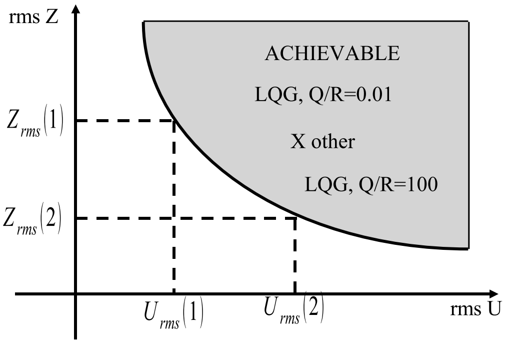


Fig. 58: Tradeoff between control effort and r.m.s. error

9 System identification

System identification is a general term to describe mathematical tools and algorithms that build dynamical models from measured data. A dynamical mathematical model in this context is a mathematical description of the dynamic behaviour of a system or process.
What is system identification? Using experimental data obtained from input/output relations to model dynamic systems.

There are different approaches to system identification depending on the model class.

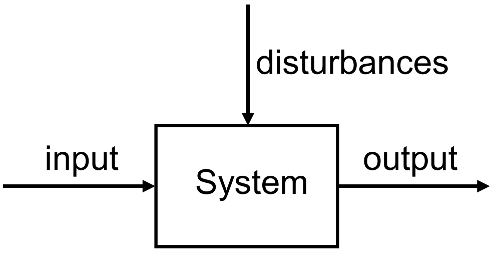


Fig. 59: System with input, output and disturbances. Goal is the identification of the system from measurements of the input and output signals.

1. Linear/non-linear
2. Parametric/non-parametric
   - Non-parametric methods try to estimate a generic model (step responses, impulse responses, frequency responses)
   - Parametric methods estimate parameters in a user-specified model (transfer functions, state-space matrices)

Identification steps: System identification is the experimental approach to process modelling.

System identification includes the following steps:

1. **Experimental design:** Its purpose is to obtain good experimental data, and it includes the choice of the measured variables and of the character of the input signals.
2. **Selection of model structure:** A suitable model structure is chosen using prior knowledge and trial and error.

3. **Choice of the criterion to fit:** A suitable cost function is chosen, which reflects how well the model fits the experimental data.
4. **Parameter estimation:** An optimization problem is solved to obtain the numerical values of the model parameters.
5. **Model validation:** The model is tested in order to reveal any inadequacies.

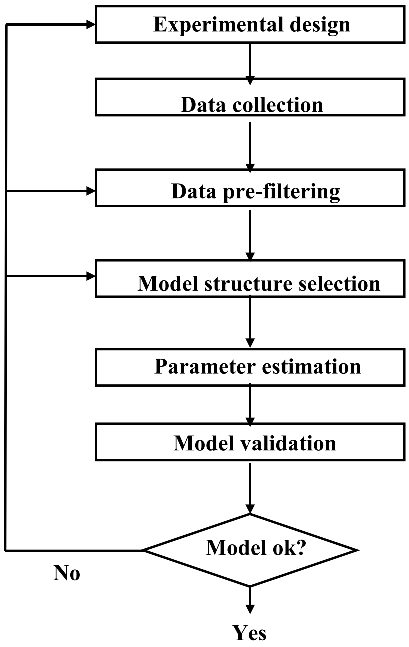


Fig. 60: Procedure for system identification

Different mathematical models: Model description:

1. Transfer functions
2. State-space models
3. Block diagrams (e.g. Simulink)

Notation for continuous time models: Complex Laplace variable s and differential operator p

$$\dot{x}(t) = \frac{\partial x(t)}{\partial t} = px(t) \; .$$

Notation for discrete time models: Complex z transform variable and shift operator q

$$x(k+1) = qx(k) \; .$$

Experiments and data collection: It is often good to use a two-stage approach.

1. Preliminary experiments
   - Step/impulse response tests to get basic understanding of system dynamics
   - Linearity, static gains, time delays constants, sampling interval
2. Data collection for model estimation
   - Carefully designed experiment to enable good model fit

- Operating point, input signal type, number of data points to collect

Preliminary experiments-step response: Useful for obtaining qualitative information about the system.

- Dead-time (delay)
- Static gain
- Time constants
- Resonance frequency

Sample time can be determined from time constants.

Rule-of-thumb: 4–10 sample points over the rise time.

Design experiment for model estimation: Input signal should excite all relevant frequencies

- Estimated model more accurate in frequency ranges where input has high energy.
- Pseudo-Random Binary Sequence (PRBS) is usually a good choice.

Trade-off in selection of signal amplitude

- Large amplitude gives high signal-to-noise ratio (SNR), low parameter variance.
- Most systems are non-linear for large input amplitudes.

Big difference between estimation of system under closed-loop control or not.

Collecting data: Sampling time selection and anti-alias are central.

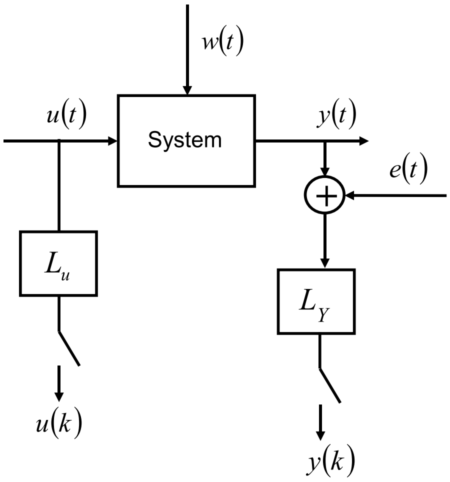


Fig. 61: Concept of the measurement of discretized input and output signals

Data pre-filtering:
Remove:

- Transients needed to reach desired operating point
- Mean values of input and output values, i.e. work with

$$\Delta u[t] = u[t] - \frac{1}{N}\Sigma_{t=1}^{N} u[t]$$

$$\Delta y[t] = y[t] - \frac{1}{N}\Sigma_{t=1}^{N} y[t]$$

- Trends (use ‘detrend’ in MATLAB)
- Outliers (obvious erroneous data points)

General model structure

1. Systems described in terms of differential equations and transfer functions.
2. Provides insight into the system physics and a compact model structure general-linear polynomial model or the general-linear model.

$$y(k) = q^{-k}G(q^{-1},\Theta)u(k) + H(q^{-1},\Theta)e(k)$$

   - $u(k)$ and $y(k)$ are the input and output of the system, respectively.
   - $e(k)$ is zero-mean white noise, or the disturbance of the system.
   - $G(q^{-1},\Theta)$ is the transfer function of the stochastic part of the system.
   - $H(q^{-1},\Theta$ is the transfer function of the stochastic part of the system.
   - General-linear model structure:

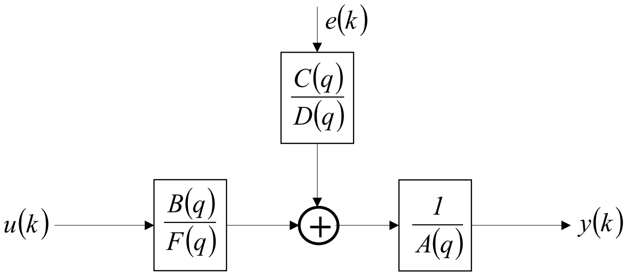


Fig. 62: Transfer functions in universal model structure

3. Provides flexibility for system and stochastic dynamics. Simpler models that are a subset of the linear model structure.

Model structures based on input–output:

Model	$\tilde{p}(q)$	$\tilde{p}_e(q)$
ARX	$\frac{B(q)}{A(q)}$	$\frac{1}{A(q)}$
ARMAX	$\frac{B(q)}{A(q)}$	$\frac{C(q)}{A(q)}$
Box–Jenkins	$\frac{B(q)}{F(q)}$	$\frac{C(q)}{D(q)}$
Output error	$\frac{B(q)}{F(q)}$	1
FIR	$B(q)$	1

$$A(q)y[k] = \frac{B(q)}{F(q)}u[k] + \frac{C(q)}{D(q)}y[k]\text{or}y[k] = \tilde{p}(q)u[k] + \tilde{p}_e(q)e[k]$$

Provides flexibility for system and stochastic dynamics. Simpler models that are a subset of the general linear model structure.

Parametric models:

1. Each of these methods has their own advantages and disadvantages and is commonly used in real-world applications
2. Choice of the model structure to use depends on the dynamics and the noise characteristics of the system.
3. Using a model with more freedom or parameters is not always better as it can result in the modelling of non-existent dynamics and noise characteristics.
4. This is where physical insight into a system is helpful.

AR model

ARX model

ARMAX model

Box–Jenkins model

Output error model

State space model

9.1 AR model:

1. Process model where outputs are only dependent on previous outputs.
2. No system inputs or disturbances are used in the modelling.
3. Class of solvable problems is limited. For signals not for systems.
4. Time series analysis, such as linear prediction coding commonly use the AR model.

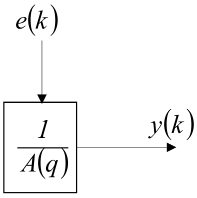


Fig. 63: AR model structure

9.2 ARX model:

Advantages:

- Is the most efficient of the polynomial estimation methods → solving linear regression equations in analytic form.
- Unique solution that satisfies the global minimum of the loss function.
- Preferable, especially when the model order is high.

Disadvantages:

- Disturbances are part of the system dynamics.
- The transfer function of the deterministic part $G(q^{-1}, \Theta)$ of the system and the transfer function of the stochastic part $H(q^{-1}, \Theta)$ of the system have the same set of poles.
- This coupling can be unrealistic.

- The system dynamics and stochastic dynamics of the system do not share the same set of poles all the time.
- However, you can reduce this disadvantages if you have a good signal-to-noise ratio.
- Set the model order higher than the actual model order to minimize the equation error, especially when the signal-to-noise ratio is low.
- However, increasing the model order can change some dynamic characteristics of the model, such as the stability of the model.

$$y[k] = \frac{B(q)}{A(q)}u[k] + \frac{1}{A(q)}e[k]$$

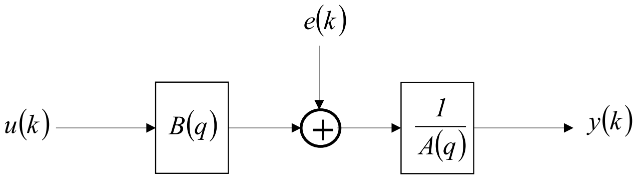


Fig. 64: ARX model structure

9.3 ARMAX model:

1. Includes disturbances dynamics. ARMAX models are useful when you have dominating disturbances that enter early in the process, such as at the input.
   For example, a wind gust affecting an aircraft is a dominating disturbance early in the process.
2. More flexibility in the handling of disturbance modelling.

$$y[k] = \frac{B(q)}{A(q)}u[k] + \frac{C(q)}{A(q)}e[k] \ .$$

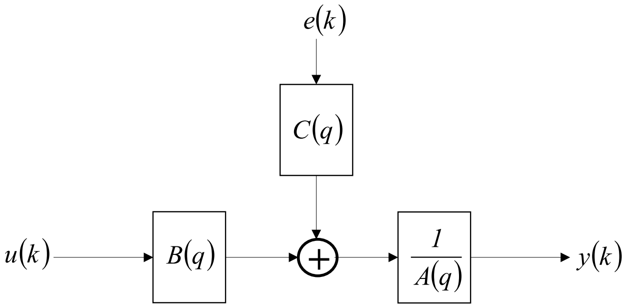


Fig. 65: ARMAX model structure

9.4 Box–Jenkins model:

- Provides a complete model with disturbance properties modelled separately from system dynamics.
- The Box–Jenkins model is useful when you have disturbances that enter late in the process.

  For example, measurement noise on the output is a disturbance late in the process (not output error model).

$$y[k] = \frac{B(q)}{F(q)}u[k] + \frac{C(Q)}{D(q)}e[k]$$

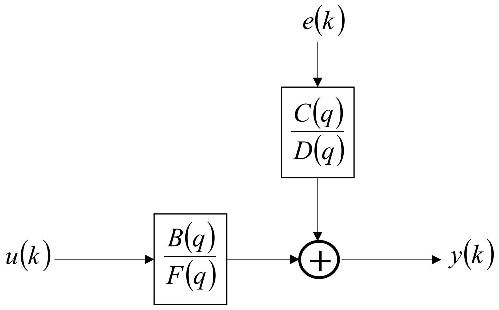


Fig. 66: Box–Jenkins model structure

9.5 Output error model:

- Describes the system dynamics separately.
- No parameters are used for modelling the disturbance characteristics.

$$y[k] = \frac{B(q)}{F(q)}u[k] + e[k]$$

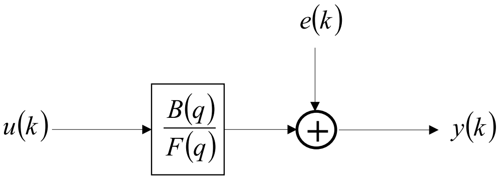


Fig. 67: Output error model structure

9.6 State space model:

1. All previous methods are based on minimizing a performance function.
2. For complex high-order systems the classical methods can suffer from several problems.
3. Find many local minima in the performance function and do not converge to global minima.
4. The user will need to specify complicated parametrization. They also may suffer potential problems with numerical instability and excessive computation time to execute the iterative numerical minimization methods needed.
5. In addition, modern control methods require a state-space model of the system.
6. For cases such as these the State-Space (SS) identification method is the appropriate model structure.

Hint: When the model order is high, use an ARX model because the algorithm involved in ARX model estimation is fast and efficient when the number of data points is very large. The state-space model estimation with a large number of data points is slow and requires a large amount of memory. If you must use a state-space model, for example in modern control methods, reduce the sampling rate of the signal in case the sampling rate is unnecessarily high.

$$x(k+1) = Ax(k) + Bu(k) + Ke(k)$$

$$y(k) = Cx(k) + Du(k) + e(k)$$

$x(k)$: state vector (Dim. setting you need to provide for the state-space model)
$y(k)$: system output
$u(k)$: System input
$e(k)$: Stochastic error. A, B, C, D, K are system matrices.

7. In general, they provides a more complete representation of the system, especially for complex MIMO systems, because similar to first-principle models.
8. The identification procedure does not involve non-linear optimization so the estimation reaches a solution regardless of the initial guess.
9. Parameter settings for the state-space model are simpler.
10. You need to select only the order/states, of the model.
11. Prior knowledge of the system or by analysing the singular values of A.

Determine parameters: Determining the delay and model order for the prediction error methods, ARMAX, BJ, and OE, is typically a trial-and-error process.

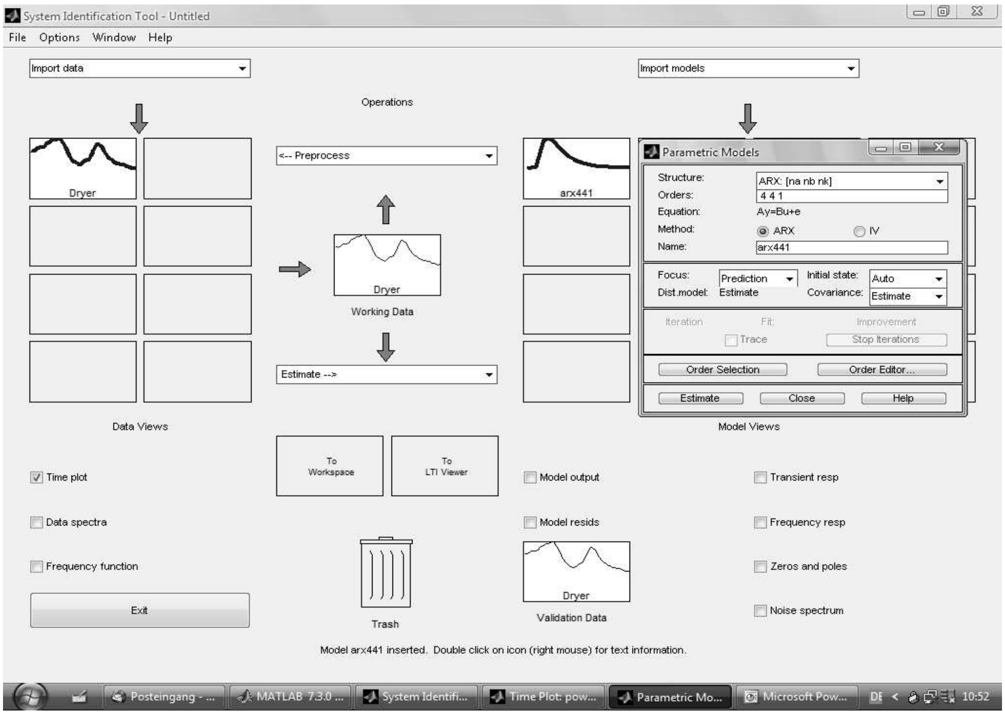
The following is a useful set of steps that can lead to a suitable model (this is not the only methodology you can use, nor is this a comprehensive procedure).

1. Obtain useful information about the model order by observing the number of resonance peaks in the non-parametric frequency response function. Normally, the number of peaks in the magnitude response equals half the order of $A(q), F(q)$.
2. Obtain a reasonable estimate of delay using correlation analysis and/or by testing reasonable values in a medium-sized ARX model. Choose the delay that provides the best model fit based on prediction errors or other fit criteria.
3. Test various ARX model orders with this delay choosing those that provide the best fit.
4. Since the ARX model describes both the system dynamics and noise properties using the same set of poles, the resulting model may be unnecessarily high in order.
5. By plotting the zeros and poles (with the uncertainty intervals) and looking for cancellations you can reduce the model order. The resulting order of the poles and zeros are a good starting point for ARMAX, OE and/or BJ models with these orders used as the B and F model parameters and first- or second-order models for the noise characteristics.
6. Are there additional signals? Measurements can be incorporated as extra input signals!
7. If you cannot obtain a suitable model following these steps, additional physical insight into the problem might be necessary (compensating for non-linear sensors or actuators and handling of important physical non-linearities are often necessary in addition to using a ready-made model).

From the prediction error standpoint, the higher the order of the model, the better the model fits the data because the model has more degrees of freedom. However, you need more computation time and memory for higher orders. The parsimony principle advocates choosing the model with the smallest degree of freedom, or number of parameters, if all the models fit the data well and pass the verification test.

Conclusion:

- Variety of model structures available
- Choice is based upon an understanding of the system identification method and algorithm
- System and disturbance are important
- Handling a wide range of system dynamics without knowing system physics
- Reduction of engineering effort
- Identification tools (Matlab, Matrix or LabVIEW) are available for developing, prototyping, and deploying control algorithms.



Structure: ARX: [na nb nk]
Orders: 4 4 1
Equation: Ay=Bu+e
Method: ● ARX ○ IV
Name: arx441
Focus: Prediction
Dist.model: Estimate
Initial state: Auto
Covariance: Estimate

Model arx441 inserted. Double click on icon (right mouse) for text information.

Fig. 68: Matlab system identification tool

10 Non-linear systems

In mathematics, a non-linear system is a system which is not linear i.e. a system which does not satisfy the superposition principle. Less technically, a non-linear system is any problem where the variable(s) to be solved for cannot be written as a linear sum of independent components. A non-homogeneous system, which is linear apart from the presence of a function of the independent variables, is non-linear according to a strict definition, but such systems are usually studied alongside linear systems because they can be transformed to a linear system as long as a particular solution is known. Generally, non-linear problems are difficult (if possible) to solve and are much less understandable than linear problems. Even if not exactly solvable, the outcome of a linear problem is rather predictable, while the outcome of a non-linear is inherently not.

Properties of non-linear systems: Some properties of non-linear dynamic systems are:

- They do not follow the principle of **superposition** (linearity and homogeneity).
- They may have multiple isolated equilibrium points (**linear systems** can have only one).
- They may exhibit properties such as **limit-cycle**, **bifurcation**, **chaos**

- Finite escape time: The state of an **unstable** non-linear system can go to infinity in finite time.
- For a **sinusoidal** input, the output signal may contain many harmonics and sub-harmonics with various amplitudes and phase differences (a linear system's output will only contain the sinusoid at the output).

Analysis and control of non-linear systems: There are several well-developed techniques for analysing non-linear feedback systems:

- Describing function method
- Phase plane method
- Lyapunov stability analysis
- Singular perturbation method
- Popov criterion (described in The Luré Problem below)
- Centre manifold theorem
- Small-gain theorem
- Passivity analysis

Control design of non-linear systems: Control design techniques for non-linear systems also exist. These can be subdivided into techniques which attempt to treat the system as a linear system in a limited range of operation and use (well-known) linear design techniques for each region:

- Gain scheduling
- Adaptive control

Those that attempt to introduce auxiliary non-linear feedback in such a way that the system can be treated as linear for purposes of control design:

- Feedback linearization
- Lyapunov based methods
- Lyapunov redesign
- Back-stepping
- Sliding mode control

10.1 Describing function method:

- Procedure for analysing non-linear control problems based on quasi-linearization.
- Replacement of the non-linear system by a system that is linear except for a dependence on the amplitude of the input waveform. Must be carried out for a specific family of input waveforms.
- An example might be the family of sine-wave inputs; in this case the system would be characterized by an SIDF or sine input describing function $H(A, \omega)$ giving the system response to an input consisting of a sine wave of amplitude A and frequency ω (this SIDF is a generalization of the transfer function $H(\omega)$ used to characterize linear systems).
- Other types of describing functions that have been used are DFs for level inputs and for Gaussian noise inputs. The DFs often suffice to answer specific questions about control and stability.

10.2 Phase plane method

- Refers to graphically determining the existence of limit cycles.

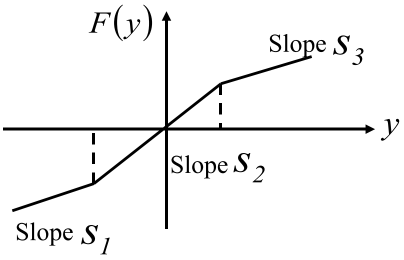


Fig. 69: Example of a piecewise linear function

- The phase plane, applicable for second-order systems only, is a plot with axes being the values of the two state variables, $x2$ vs $x1$.
- Vectors representing the derivatives at representative points are drawn. With enough of these arrows in place the system behaviour over the entire plane can be visualized and limit cycles can be easily identified.

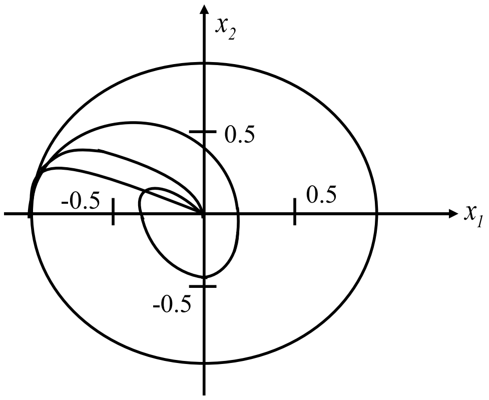


Fig. 70: Identification of limit cycles in the phase plane

10.3 Lyapunov stability

- Lyapunov stability occurs in the study of dynamical systems. In simple terms, if all solutions of the dynamical system that start out near an equilibrium point x_e stay near x_e forever, then x_e is Lyapunov stable.
- More strongly, if all solutions that start out near x_e converge to x_e, then x_e is asymptotically stable.
- The notion of exponential stability guarantees a minimal rate of decay, i.e., an estimate of how quickly the solutions converge.
- The idea of Lyapunov stability can be extended to infinite-dimensional manifolds, where it is known as **structural stability**, which concerns the behaviour of different but ‘nearby’ solutions to differential equations.

10.4 Luré’s problem

1. Control systems have a forward path that is linear and time-invariant, and a feedback path that contains a memoryless, possibly time-varying, static non-linearity.
2. The linear part can be characterized by four matrices (A, B, C, D), while the non-linear part is $\Phi(y)$ with a sector non-linearity.

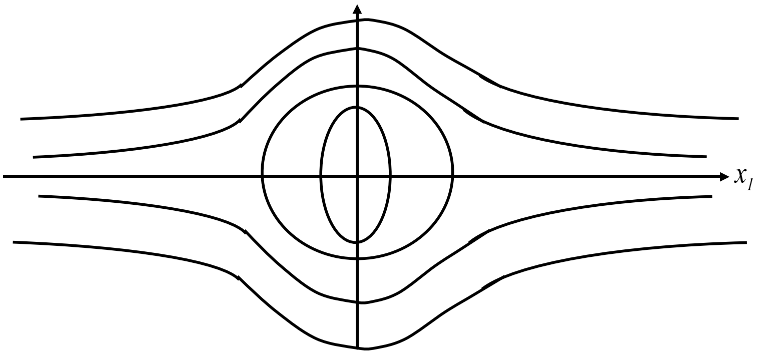


Fig. 71: Lyapunov stability analysis in the phase plane

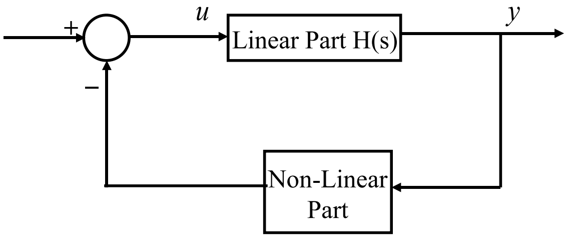


Fig. 72: The Luré problem decomposes the control problem in a linear forward path and a non-linear feedback path

3. by A.I. Luré.

10.5 Popov criterion

1. The sub-class of Luré's systems studied by **Popov** is described by:

$$\dot{x} = Ax + bu, \dot{\xi} = u$$

$$y = cx + d\xi, u = -\Phi(y) \text{ , where}$$

2. $x\epsilon R\ n, \xi, u, y$ are scalars and A, b, c, d have commensurate dimensions. The non-linear element $\Phi : R \to R$ is a time-invariant non-linearity belonging to open sector $(0, \infty)$. This means that

$$\Phi(0) = 0, y\Phi(y) > 0 \forall y \neq 0 \text{ ;}$$

Popov stability criteria: →Absolute stability. The marked region in the figure is called a sector

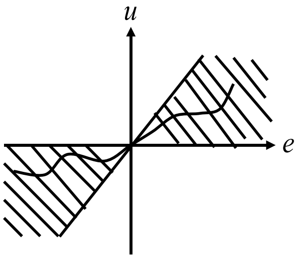


Fig. 73: Stability in a sector described by the Popov inequality

$$0 \leq F(e) \leq ke \text{ for } e > 0$$

$$ke \leq F(e) \leq 0 \text{ for } e < 0$$

or shorter

$$0 \leq \frac{F(e)}{e} \leq k \, .$$

The standard control loop is absolute stable in sector $[0, k]$, if the closed loop only has global asymptotic stable equilibrium for any characteristic $F(e)$ lying in this sector.

Prerequisities:

$$G(s) = \frac{1}{s^p} \frac{1 + b_1 s + ... + b_m s^m}{1 + a_1 s + ...a_n s^n}, \; m < n + p \, .$$

No common zeros of numerators and denominators.

$F(e)$ is unique and at least steady in intervals $F(0) = 0$.

One distinguishes the main case $p = 0$, i.e. without integrator eigen value and the singular case $p \neq 0$.

Cases	p	Sector	Inequality
Ia	0	$\frac{F(e)}{e} \geq 0$	$\Re((1 + qj\omega)G(j\omega)) > 0$
Ib	0	$k \geq \frac{F(e)}{e} \geq 0$	$\Re((1 + qj\omega)G(j\omega)) > -\frac{1}{k}$
IIa	$1, ..., n - m$	$k \geq \frac{F(e)}{e} \geq 0$	$\Re((1 + qj\omega)G(j\omega)) > -\frac{1}{k}$
IIb	$1, ..., n - m$	$\frac{F(e)}{e} > 0$	$\Re((1 + qj\omega)G(j\omega)) \geq 0$

- The control loop is absolute stable in the given sector, when a real number q is found for all $\omega \leq 0$ apply on the right side of inequality.
- The Popov criteria is a sufficient condition for absolute stability of the control loop.

Geometric interpretion of the Popov inequality:

- For the case Ib and IIa, the inequality can be rewritten as follows

$$\Re((1 + qj\omega)G(j\omega)) > -\frac{1}{k}$$

$$\Re(G(j\omega)) + q\Re(j\omega(G(j\omega))) > -\frac{1}{k}$$

$$\Re(G(j\omega)) + q\Re(j\omega\Re(G(j\omega)) > -\omega\Im(G(j\omega))) > -\frac{1}{k}$$

$$\Re(G(j\omega)) - q\omega\Im(G(j\omega)) > -\frac{1}{k} \longrightarrow 1$$

- One defines the modified frequency plot

$$G_p(j\omega) = \Re(G(j\omega)) + j\omega\Im(G(j\omega))$$

- This frequency plot is called Popov curve, so equation 1 can be represented as

$$\Im(G_p(j\omega)) < \frac{1}{q}(\Re(G_p(j\omega)) + \frac{1}{k}) \, .$$

Example of Popov criteria: The non-linearity is shown as a three-point characteristic which already hosts the amplification K of the linear subsystem. Popov frequency response locus $Z = Lp(j\omega)$. The line g_{krit} is tangential to the Popov frequency response locus curve and intersects the real axis at 0.1.

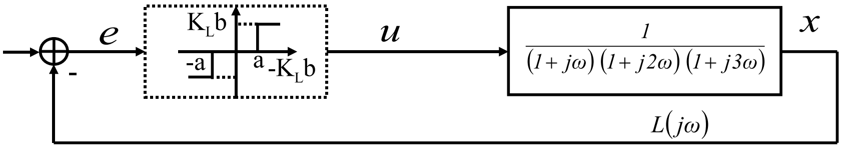


Fig. 74: Example for controller non-linearity described by a three-point characteristic

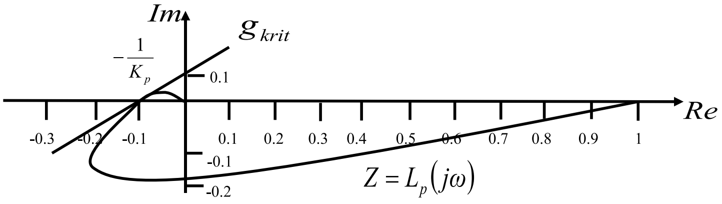


Fig. 75: Graphical determination of g_{krit}

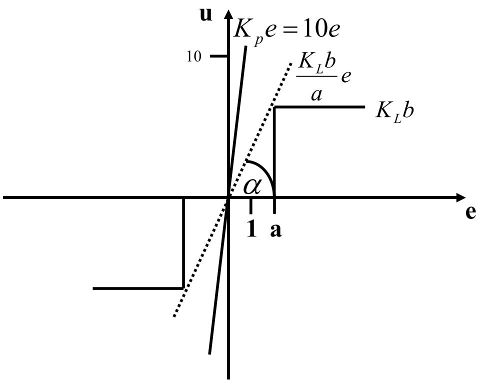


Fig. 76: Stability sector for the example

Popov sector is defined by the line $K_p e$. To verify Popov criteria the gradient of three-point characteristic must be smaller the calculated gradient

$$\tan\alpha = \frac{K_L b}{a}$$

$$\frac{K_L b}{a} < 10\ .$$

If this is true for the given inequality, then it is idle state global asymptotic stable.

11 Non-linear control

Non-linear control is a sub-division of control engineering which deals with the control of non-linear systems. The behaviour of a non-linear system cannot be described as a linear function of the state of that system or the input variables to that system. For linear systems, there are many well-established control techniques, for example root-locus, Bode plot, Nyquist criterion, state-feedback, pole-placement etc.

11.1 Gain scheduling

Gain scheduling is an approach to control of **non-linear systems** that uses a family of **linear controllers**, each of which provides satisfactory control for a different operating point of the system.

One or more observable variables, called the scheduling variables, are used to determine what operating region the system is currently in and to enable the appropriate linear controller.

Example: In an aircraft flight control system, the altitude and Mach number might be the scheduling variables, with different linear controller parameters available (and automatically plugged into the controller) for various combinations of these two variables.

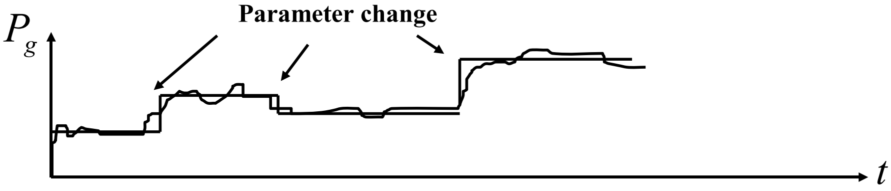


Fig. 77: Concept of gain scheduling of controller parameters based on the scheduling variable

11.2 Adaptive control

1. Modifying the controller parameters to compensate slow time Ű varying behaviour or uncertainties.
2. For example, as an aircraft flies, its mass will slowly decrease as a result of fuel consumption; we need a control law that adapts itself to such changing conditions.
3. Adaptive control does not need a priori information about the bounds on these uncertain or time-varying parameters.
4. Robust control guarantees that if the changes are within given bounds the control law need not be changed.
5. Adaptive control is precisely concerned with control law changes.

There are several broad categories of feedback adaptive control (classification can vary):

Dual adaptive controllers optimal controllers. – Suboptimal dual controllers

Non-dual adaptive controllers – Model Reference Adaptive Controllers (MRACs) [incorporate a reference model defining desired closed loop performance]

MRAC

MIA – Gradient optimization MRACs [use local rule for adjusting parameters when performance differs from reference]

- Stability optimized MRACs
- Model Identification Adaptive Controllers (MIACs) [perform **system identification** while the system is running]
- Cautious adaptive controllers [use current SI to modify control law, allowing for SI uncertainty]. Non-parametric adaptive controllers
- Parametric adaptive controllers
- Explicit parameter adaptive controllers
- Implicit parameter adaptive controllers

11.3 Feedback linearization

1. Common approach used in controlling nonlinear systems.
2. Transformation of the non-linear system into an equivalent linear system through a change of variables and a suitable control input.
3. Feedback linearization may be applied to non-linear systems of the following form:

$$\dot{x} = f(x) + g(x)u$$

$$y = h(x)$$

   where x is the state vector, u is the vector of inputs, and y is the vector of outputs.
4. The goal, then, is to develop a control input u that renders either the input–output map linear, or results in a linearization of the full state of the system.

Existence of a state feedback control?

$$u = \alpha(x) + \beta(x)y$$

And variable change?

$$z = T(x)$$

Pendulum example: Task: Stabilization of origin of pendulum eq.

$$\dot{x_1} = x_2$$

$$\dot{x_2} = a[\sin(x_1 + \delta) - \sin(\delta)] - bx_2 + cu$$

1. Choose u as follows to cancel non-linear term:

$$u = \frac{a}{c}[\sin(x_1 + \delta) - \sin(\delta)] + \frac{v}{c}$$

   From the above two equations: linear system

$$\dot{x_1} = x_2$$

$$\dot{x_2} = -b.x_2 + v$$

2. Stabilizing feedback controller

$$v = -k_1x_1 - k_2x_2$$

   Closed loop

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = k_1x_1 - (k_2 + b)x_2$$

3. Re-substituting to or feedback control law

$$u = \frac{a}{c}[\sin(x_1 + \delta) - \sin(\delta)] - \frac{1}{c}(k_1x_1 + k_2x_2)$$

11.4 Sliding mode control

In control theory sliding mode control is a type of variable structure control where the dynamics of a non-linear system is altered via application of a high-frequency switching control. This is a state feedback control scheme where the feedback is not a continuous function of time.

This control scheme involves the following two steps:

- Selection of a hyper surface or a manifold such that the system trajectory exhibits desirable behaviour when confined to this manifold.
- Finding feedback gains so that the system trajectory intersects and stays on the manifold.

Advantages:

- Controllers with switching elements are easy to realize.
- The controller outputs can be restricted to a certain region of operation.
- Can also be used inside the region of operation to enhance the closed-loop dynamics.

Two types of switching controllers

- Switching between constant values
- Switching between state or output dependent values

Design of a switching controller therefore needs:

- The definition of a switching condition
- The choice of the control law inside the areas separated by the switching law

Phase plot of a closed-loop system with sliding mode controller

$$(y = 0, \dot{y} = 1 \ \text{and} \ m = 1) \ .$$

Consider a linear system

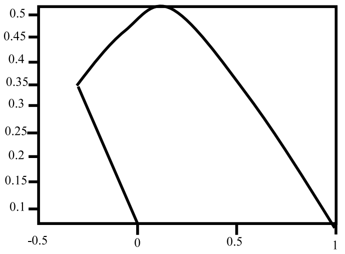


Fig. 78: Phase plot of a closed-loop system with sliding mode controller

$$\ddot{y} = u$$

The control law

$$u = -ky \ \text{with} \ k > 0 \ .$$

It is clear that the controller is not able to stabilize the semi stable system that has two open-loop eigen values of zero. The root locus plot with two branches along the imaginary axis.

Switching function:

$$s(y, \dot{y}) = my + \dot{y} \ .$$

With a scalar $m > 0$ then a variable structure control law

$$u = -1 \ \ s(y, \dot{y}) > 0, 1 \ \ s(y, \dot{y}) < 0 \ .$$

Phase portrait with example: Consider a second-order system

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = h(x) + g(x)u$$

$g(x)$ and $h(x)$ are unknown non-linear functions.

On the manifolds $s = 0$ motion

$$s = a_1 x_1 + x_2 = 0 \to \dot{x} = -ax \ .$$

Independent of $g(x)$ and $h(x)$

Goal: state feedback → stabilize the origin.

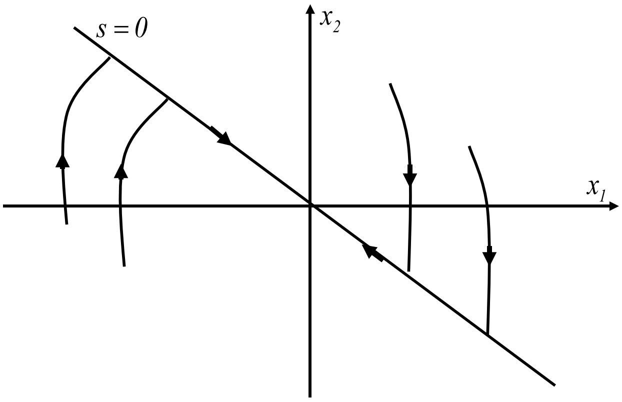


Fig. 79: Manifolds for the example

Chattering on sliding manifold

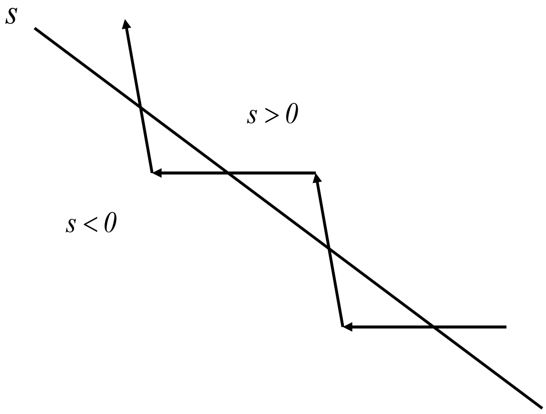


Fig. 80: Chattering on sliding manifold

- Caused by delays (no ideal would run on manifold).
- Region $s < 0 \to$ trajectory reverses
- Delay leads to crossing again

Disadvantages

– Low control accuracy
– Power losses
– Excitation of un-modelled of dynamics
– Can result in instabilities.

Bibliography

[1] C.T. Chen, *Linear System Theory and Design,* 3rd. ed. (Oxford University Press, Oxford, 1999).

[2] H.K. Khalil, *Nonlinear Systems,* 3rd. ed. (Prentice-Hall, Upper Saddle River, NJ, 2002), (ISBN $0-13-067389-7$).

[3] N.S. Nise, *Control Systems Engineering,* 4th ed. (John Wiley and Sons, Inc., Chichester, 2004).

[4] D. Hinrichsen and A.J. Pritchard, *Mathematical Systems Theory I, Modelling, State Space Analysis, Stability and Robustness* (Springer, Berlin, 2003).

[5] E.D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems,* 2nd ed. (Springer, New York, 1998).

[6] C. Kilian, *Modern Control Technology: Component and Systems*, 3rd ed. (Delmar Thompson Learning, 2005).

[7] V. Bush, *Operational Circuit Analysis* (John Wiley and Sons, Inc., New York, 1929).

[8] R.F. Stengel, *Optimal Control and Estimation* (Dover Publications, New York, 1994) (ISBN $0-486-68200-5$, ISBN$-13: 978-0-486-68200-6$.

[9] G.F. Franklin, A. Emami-Naeini and J.D. Powell, *Feedback Control of Dynamic Systems*, 4th ed. (Prentice-Hall, Upper Saddle River, NJ, 2002) (ISBN $0-13-032393-4$).

[10] J.L. Hellerstein, D.M. Tilbury and S. Parekh , *Feedback Control of Computing Systems* (John Wiley and Sons, Newark, NJ, 2004).

[11] G.E. Carlson, *Signal and Linear System Analysis with Matlab*, 2nd ed. (Wiley, Chichester, 1998).

[12] J.G. Proakis and D.G. Mandalokis, *Digital Signal Processing Principals, Algorithms and Applications*, 3rd ed. (Prentice-Hall, Upper Saddle River, NJ, 1996).

[13] R.E. Ziemer, W.H. Tranter and D.R. Fannin, *Signals and Systems: Continuous and Discrete*, 4th ed. (Prentice-Hall, Upper Saddle River, NJ, 1998).

[14] B. Porat, *A Course in Digital Signal Processing* (Wiley, Chichester, 1997).

[15] L. Ljung, *Systems Identification: Theory for the User*, 2nd ed. (Prentice-Hall, Englewood Cliffs, NJ, 1999).

[16] Jer-Nan Juang, *Applied System Identification* (Prentice-Hall, Englewood Cliffs, NJ, 1994).

[17] A.I. Luré and V.N. Postnikov, *On the theory of stability of control systems, Applied Mathematics and Mechanics*, **8** (3), 1944, in Russian.

[18] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed. (Prentice-Hall, Englewood Cliffs, NJ, 1993).

[19] A. Isidori, *Nonlinear Control Systems*, 3rd ed. (Springer, London, 1995).