

Systems and Control Theory Part 1: History of control and linear control system design.

Lecture notes of lectures 1-6. L.G. van Willigenburg

Contents

1	History of control, control terminology and concepts.....	3
1.1	Introduction.	3
1.2	The steam engine: an example of feedback control.....	3
1.3	Control system analysis and design in the frequency and time domain.	4
1.3.1	Frequency domain analysis in a nutshell.....	5
1.3.2	Time domain analysis in a nutshell.....	5
1.4	On-Off and PID feedback control.....	6
1.4.1	On-off and PID control of a single variable (single-loop PID control).	6
1.4.2	PID control of multiple variables (multi-loop PID control) and its complications.....	7
1.5	Questioning and improving on PID control: Model based control.....	7
1.6	Model based control: answers in a nutshell.	8
1.7	Building state-space models from first-principles: an example.....	9
1.8	Linearity of dynamic systems.	12
1.9	Linearization about optimal state, input and output trajectories.	14
1.10	Summary.....	16
2	Analysis, simulation and properties of dynamic systems.	17
2.1	Introduction and overview.....	17
2.2	Numerical solutions of the state-equation.....	18
2.3	Asymptotic stability of solutions to the state-equation.....	19
2.4	Responses and asymptotic stability of LTV and LTI systems.	20
2.4.1	State and input versus small state and input perturbations.....	20
2.4.2	The state-transition matrix and responses of LTV and LTI systems.	20
2.4.3	Asymptotic stability of LTI and LTV systems.....	24
2.5	What if the system is linear?.....	25

3	Transfer function analysis and state-space representations of LTI systems.	26
3.1	Introduction.	26
3.2	Transfer functions, black box and white box modelling	27
3.3	Transfer functions/matrices of LTI systems.	27
3.4	General and special internal state-representations of LTI systems	30
3.5	Transfer functions, frequency domain analysis and Fourier and Laplace transforms. ...	33
4	Controllability and observability of dynamic systems.	37
4.1	Partitioning and graphical representation of a dynamical system.....	37
4.2	Controllability and observability as connections to the input and output.....	38
4.3	Canonical state-space representations of dynamical systems.....	39
4.4	Canonical controllability and observability forms	40
4.5	Establishing controllability and observability of dynamical systems represented in state-space form.....	43
4.6	Stabilizability and detectability of LTI systems	46
4.7	Stabilizability and detectability of LTV systems.....	47
5	Pole placement: a state-feedback controller design method for LTI systems.....	47
6	Static and dynamic output-feedback of LTI systems.....	49
6.1	Static versus dynamic output-feedback.	49
6.2	Observer design for LTI systems.....	50
6.3	A prelude to static and dynamic output-feedback for LTV systems.....	52

1 History of control, control terminology and concepts.

1.1 Introduction.

People are dominating planet earth, whether you like it or not. An important reason for this is that we managed to take control over many things, like agriculture and currently even life itself, and also intelligence. All this could be achieved through science and engineering. To control things, you need to 1) be able to manipulate things (actuators), 2) be able to monitor what happens (measurements) and, 3) have some knowledge about how manipulation influences the things you want to control (a model). The modelling dynamic systems course (BCT-20306) took controlling/driving a car as a first example. Here we start with a small historic overview of control. To present it, we also need to introduce some control terminology and concepts. We will continue to use these throughout this course.

After reading this chapter one should:

- 1) Recognise control terminology and concepts.
- 2) Draw a block diagram of a feedback control system.
- 3) Recognise different types of dynamic systems.
- 4) Be able to represent dynamic systems in state-space form.
- 5) Draw a block diagram of an optimal control system.
- 6) Linearize around steady-states and trajectories of dynamic systems.

1.2 The steam engine: an example of feedback control.

A major contribution to the industrial revolution (1760-1840) was the development of engines,

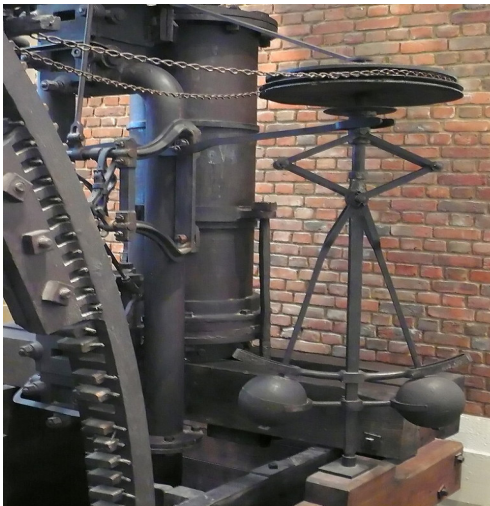


Fig. 1: A governor, measuring the engine speed.

specifically the steam engine. The speed of such an engine depends on steam pressure. The higher the pressure, the higher the speed. To get some control over the steam pressure, steam engines have a valve that can be opened to release some of the pressure. So, by varying the valve opening, steam pressure and thus the speed of the steam engine can be influenced. To prevent the steam engine from going too fast, you want the valve opening to increase with increasing speed. A device was invented, a so-called governor, that was mounted on the part rotated by the engine. As the speed increased, the balls of the governor move outwards and this movement is transferred to the valve that further opens, to decrease the pressure and speed of the engine.

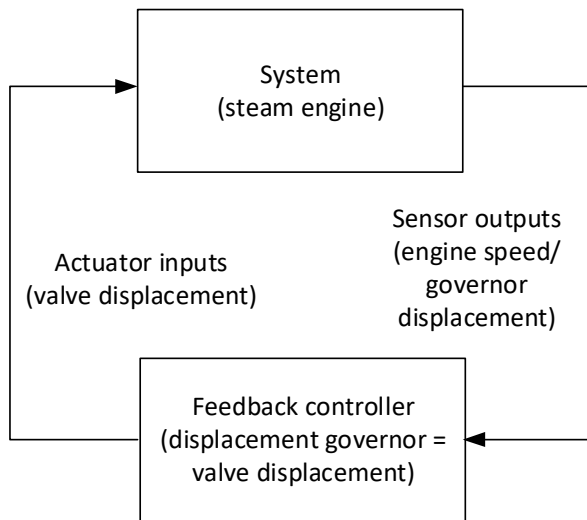


Fig. 2: The governor as a special, simple type of feedback controller.

You may say that the governor acts as a sensor, its displacement being the sensor output measuring engine speed. The valve acts as an actuator, its valve position determining engine speed. The governor displacement (sensor output) is directly “fed back” as a displacement of the valve (actuator input). This simple example beautifully illustrates that feedback control concerns feeding back sensor outputs to actuator inputs to control the system. In the case of the governor, the displacement of the sensor directly determines the displacement of the valve. In general, the link between sensor outputs and actuator inputs may be more sophisticated, to realize

more sophisticated control, called a feedback control system, see Fig. 2. Sensor outputs are generally called outputs of the system, whereas actuator inputs are generally called inputs of the system.

1.3 Control system analysis and design in the frequency and time domain.

Given certain actuator input signals in Fig. 2, the input signals being time-functions, one can record the sensor outputs, which are also time-functions/signals. The system relates the input signals to the output signals.

So one can get an idea of what the system does by analysing these time-functions/signals. If one analyses these time-functions directly, this is called time-domain analysis. Historically, *frequency domain analysis* was very popular initially. This relates to the fact that signals/time-functions can be decomposed into sums of sinusoidal time functions. And the transfer of sinusoidal time-functions from the input to the output of linear time-invariant system (LTI system) is easily described and analysed mathematically. This makes frequency-domain analysis attractive when the system is of the LTI type. The downside of frequency domain analysis is that it only works for the restricted class of LTI systems and requires decomposition and reconstruction into sinusoidal signals of input and output signals. The procedures to do the decomposition and reconstruction are the Laplace or Fourier transform and the inverse Laplace or Fourier transform.

Historically, the design of control systems has been performed either in the frequency or in the time-domain, or sometimes even both. Initially, control system design in the frequency domain was dominant. But the emphasis has shifted towards the time-domain, that will be dominant in this course. Frequency control system analysis and design is called “classical control” whereas time-domain control system analysis and design is called “modern control”.

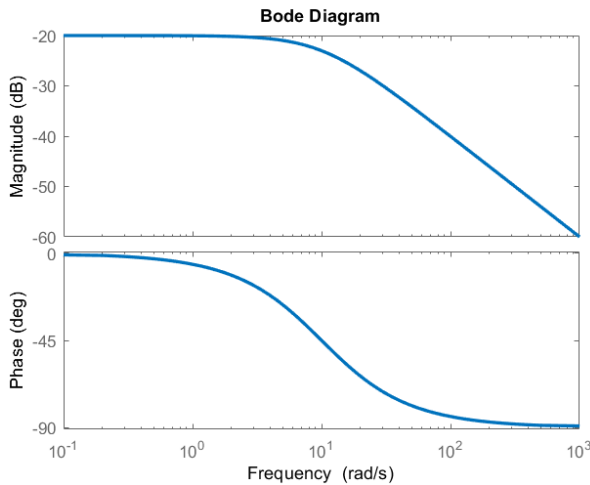


Fig. 3: Bode plot of a first-order system showing the amplitude ratio and phase shift for different frequencies.



knobs covering different frequency bands.

1.3.1 Frequency domain analysis in a nutshell.

LTI systems with a single input and single output (SISO, LTI systems) have the property that a sinusoidal input signal $A_{in} \sin(\omega t + \phi_{in})$, with amplitude A_{in} and phase ϕ_{in} , produces an output signal $A_{out} \sin(\omega t + \phi_{out})$. Note that the sinusoidal character and the angular frequency ω remain unchanged. A Bode plot or diagram plots the amplitude ratio $A_{out}(\omega) / A_{in}(\omega)$ and the phase shift

$\phi_{out}(\omega) - \phi_{in}(\omega)$ against the angular

frequency ω , or the ordinary frequency $f = \omega / (2\pi)$, see Fig. 3. This plot immediately and graphically reveals what the LTI system does to sinusoidal frequency components of an input signal. With an equalizer, used to change the sound of audio signals going in and out, you can change the amplification of audio signals at different frequencies, changing the way the music sounds. An equalizer is an LTI system, of which you can change the Bode plot, by turning different

1.3.2 Time domain analysis in a nutshell.

Time-domain analysis concerns the computation of output signals for arbitrary input signals of systems. It is more straightforward since it does not require decomposition and reconstruction of input and output signals into and from sums/integrals of sinusoidal signals/frequency components. It is also more generally applicable, since it applies to nonlinear, possibly time-varying, dynamic systems. The computation of output signals from input signals must generally be done by some form of numerical integration. Computers can easily and quickly do this. In this course the focus will be very much on time-domain analysis and design of control systems.

1.4 On-Off and PID feedback control.

1.4.1 On-off and PID control of a single variable (single-loop PID control).

The control of systems is still very often performed by so called PID (Proportional, Integral, Derivative) controllers. A control system almost everyone is familiar with is the temperature control in your living room. Generally there is a thermostat on the wall, usually close to the door, where you can read off the measured room temperature as well as set the desired temperature. Then roughly, if the desired temperature exceeds the room temperature the heating system is turned on whereas in all other cases the heater is turned off. The latter is a case of what is called on-off control, which is an even simpler form of control than PID control of the room temperature, depicted in Fig. 4. PID control is more sophisticated since out of the PID controller comes a signal

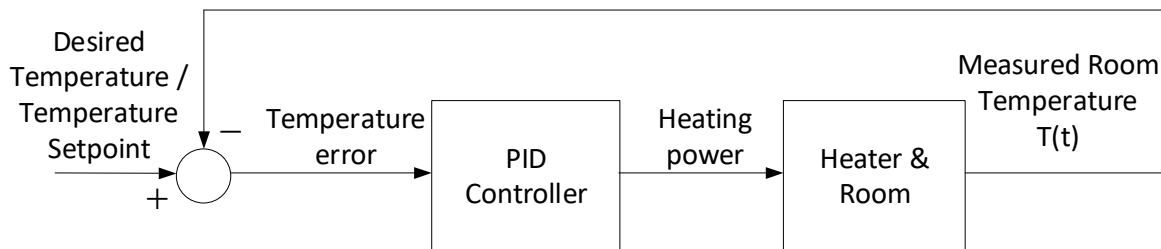


Fig. 4: PID feedback control of room temperature

that is proportional to the temperature error, as well as the integral and derivative of this error over time (therefore the controller is called a Proportional, Integral, Derivative controller). This signal is fed to the heater and determines the heating power given off by the heater. In this case, the heater is not switched on and off but heats the room more smoothly. Also, this assumes the heating system to be one that is able to continuously vary the heating power it gives off. Some important characteristics of this type of PID feedback control are the following:

1. The variable we want to control (room temperature) must be measured.
2. A desired value for this variable (desired temperature/temperature setpoint) is required.
3. An actuator (the heating system) the actuation of which can be controlled continuously is required.
4. The control system is reasonably robust to changes in the heating system and room.
5. A mathematical model of the system (heater and room) describing its behaviour is not required.
6. A temperature error is required to realize control.
7. Changes of the desired temperature/setpoint are not realized immediately but are delayed.
8. Rules of thumb (based on very simple models of the system) are used to determine the magnitude of the Proportional, Integral and Derivative action (setting) of the PID controller.

1.4.2 PID control of multiple variables (multi-loop PID control) and its complications.

In most (industrial) applications one wants to control more than a single variable. As an example we again consider our living room and assume we also want to control humidity. Then we also need an actuator influencing humidity. To that end one may open or close the windows. Let's assume we can change the window openings continuously. This suggests the multi-loop PID control system depicted in Fig. 5. Now one runs into some complications, since temperature is

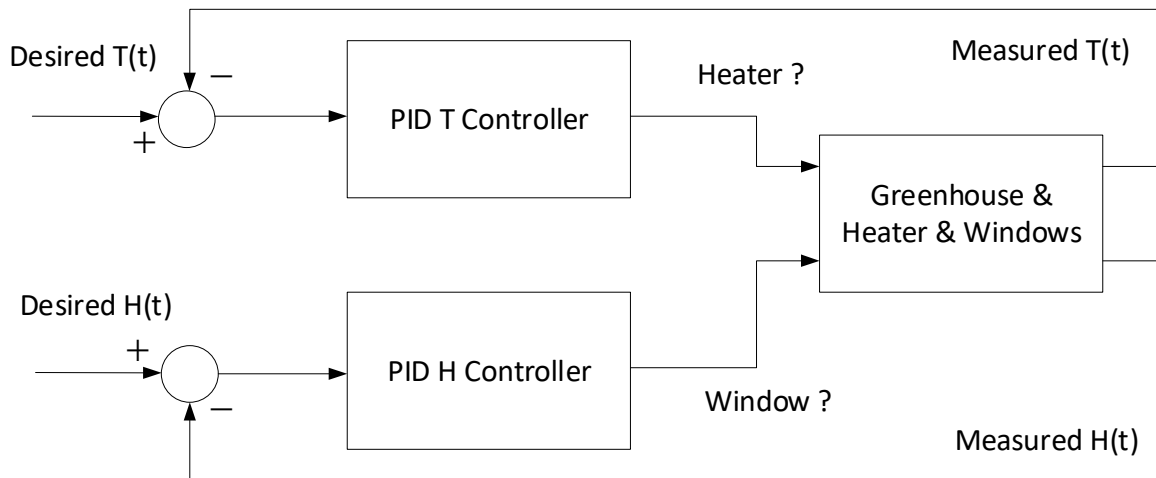


Fig. 5: Multi-loop PID control of temperature and humidity in a room causing complications.

also influenced by the window opening and humidity also by heating. Moreover, humidity and temperature are two variables that are not independent. These phenomena cause interactions between the control loops that deteriorate the proper functioning of the control system. In practice one tries to manipulate the desired temperature and humidity (the setpoints/inputs of the control system) in such a way that a proper behaviour of the control system is obtained. But this is a difficult task, that is often performed experimentally, without guaranteeing a proper result in all cases that may occur. Alternatively, this manipulation is performed at a higher level of control that may involve mathematical modelling and optimization.

1.5 Questioning and improving on PID control: Model based control.

The control of industrial robots is still very often performed by PID controllers. Industrial robots have several links or arms, each one connected to the next. Each link/arm position/rotation is controlled by a separate PID controller measuring and controlling the position/rotation of each single link/arm with respect to the previous one. Together these positions/rotations determine what is called a “pose” of the robot. Through their interconnection, the motion of one link/arm influences the motion of the others. Therefore, there is significant interaction between the PID control loops, especially if the robot moves fast, causing problems for this type of multi-loop PID control. Currently one “solves” this problem by simply limiting the speed of the arms of the robot, such that the interactions remain small enough, not disturbing the individual PID control loops too much. But

the robot could move a lot faster if the interactions could be properly accounted for. So one important question coming out of this analysis is:

1. “Can we design control systems that properly account for interactions between different variables in a system?”

Looking back at the characteristics 1, 6, 7 of PID controllers, mentioned in section 1.4.1, one can ask the following three additional questions:

2. “To control some variable within a system, is it necessary to measure this variable?”
3. “Does one need an error first, to control a certain variable?”
4. “Does there have to be a delay in realizing desired changes of variables?”

The answer to the first question is: yes, of course. The answer to the latter three questions is: no, of course not. These four beautiful answers are obtained from the main topic of this course:

Model based control system design.

1.6 Model based control: answers in a nutshell.

Full knowledge of dynamic system behaviour can be captured in a mathematical dynamic model of a system, build from scientific knowledge called “first principles modelling”, as treated in the course Modelling Dynamic Systems, (BCT-20306). The model consists of a series of first-order differential equations represented in state-space form, having n_x state-variables, collected in the state vector $x \in \mathbb{R}^{n_x}$, n_u input variables, collected in the input vector $u \in \mathbb{R}^{n_u}$, and possibly output variables, collected in the output vector $y \in \mathbb{R}^{n_y}$. Having such a model, for arbitrary input signals $u(t)$, $t_0 \leq t \leq t_f$, one can compute by numerical integration the behaviour of the system $x(t)$, and possibly the output $y(t)$, $t_0 \leq t \leq t_f$. Roughly speaking, this allows us to compute in advance the control of the system to realize a certain desired behaviour. So no need in this case for any errors or measurements! Also, since the model describes all interactions between the state-variables, it takes all interactions occurring within the system properly into account! So building dynamic state-space models of systems from first-principles, i.e. scientific knowledge such as conservation laws, is a key tool to perform sophisticated, model based control system design. This topic was extensively treated in the Modelling Dynamic Systems course and serves as the basis for model based control system design, as presented in this course. An example, summarizing all steps required in building a dynamic state-space model of a system, is presented in the next section.

1.7 Building state-space models from first-principles: an example.

The example considered in this section concerns a mass-spring-damper system. The procedure to obtain a state-space model from first principles, starts with making a sketch/drawing of the system, including relevant phenomena and mathematical symbols/variables. For the mass-spring-damper system these are depicted in Fig. 6.

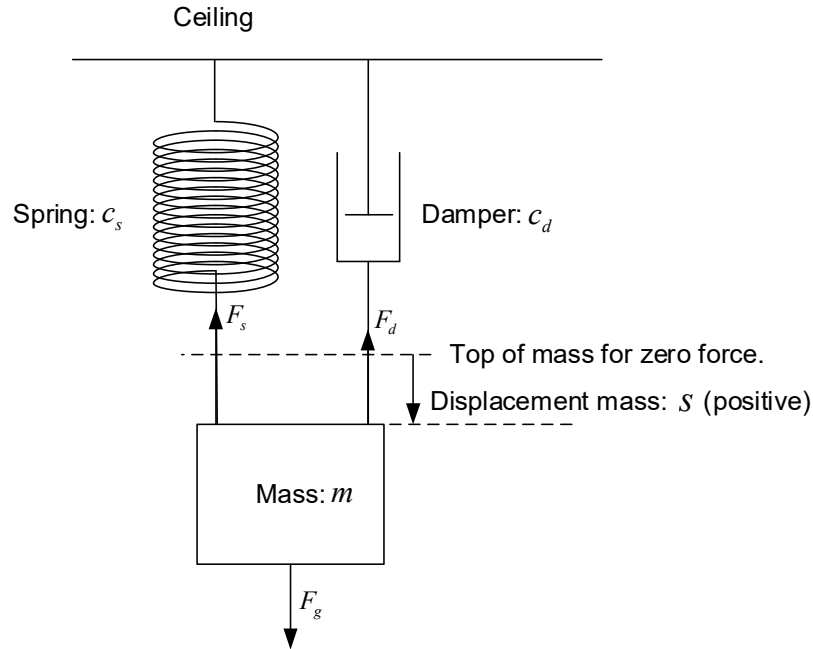


Fig. 6: Drawing with mathematical symbols to build a state-space model of a mass-spring-damper system.

From this figure, first principles must be obtained, linking the mathematical symbols/variables. In our example they concern the three forces acting on the mass being the force of gravity,

$$F_g = mg, \quad g = 9.8 \text{ ms}^{-2}, \quad (1.1)$$

the spring force,

$$F_s = -c_s s, \quad (1.2)$$

where c_s is the spring constant, and the damper force,

$$F_d = -c_d \frac{ds}{dt}. \quad (1.3)$$

where c_d is the damping constant. These three forces, according to Newton's third law, cause acceleration of the mass,

$$\frac{d^2 s}{dt^2} = \frac{F_g + F_s + F_d}{m} = \frac{mg - c_s s - c_d ds/dt}{m} . \quad (1.4)$$

Equation (1.4) is the single, second-order differential equation describing the mass-spring-damper system. To obtain a state-space realization the following procedure must be followed.

1. Find the state-variables denoted by x . These are variables of which time-derivatives appear in the differential equations. Here, there is only 1 differential equation. In it, only time-derivatives of the variable s appear. Therefore s is a state-variable: $x_1 = s$. In addition, all time-derivatives of s are also state-variables, up to, but not including, the highest time-derivative that is found in any of the differential equations. Here, the highest time-derivative is $d^2 s / dt^2$ and therefore $x_2 = ds / dt$ is the only other state-variable.
2. Next find the parameters denoted by p . These are all the constants appearing in the differential equations. Here we have, $p_1 = m$, $p_2 = c_s$, $p_3 = c_d$, $p_4 = g$.
3. Finally determine the inputs denoted by u . These are the remaining variables. In this example there are none, so the mass-spring-damper system has no inputs. To make the example more interesting, assume a variable external force F_e can be applied to the mass. In that case we have $u_1 = F_e$.

The state-variables, parameters, and inputs together determine the state-space mapping in *Table 1*.

Using the state-space mapping and equations (1.1)-(1.4), we obtain the state-equation,

$$\underbrace{\begin{bmatrix} dx_1 / dt \\ dx_2 / dt \end{bmatrix}}_{dx/dt} = \underbrace{\begin{bmatrix} x_2 \\ p_4 + (u_1 - p_1 x_1 - p_2 x_2) / p_3 \end{bmatrix}}_{f(x,u,p)} \quad (1.5)$$

The state-equation contains first derivatives of all state-variables on the left, and algebraic expressions in the state-variables, parameters and inputs on the right, both captured in a vector. Mathematically speaking, the vector function $f(x, u, p)$ fully describes the state-equation, i.e. the dynamics of the state-variables/states.

Notation			
State-space	Field of application	Meaning	Unit
x_1	s	position	m
x_2	ds / dt	velocity	ms^{-1}
p_1	m	mass	kg
p_2	c_s	Spring constant	$N / m = kgs^{-2}$
p_3	c_d	Damper constant	$N / m / s = kgs^{-1}$
p_4	g	Acceleration due to gravity	ms^{-2}
u_1	F_e	External force	$N = kgms^{-2}$

Table 1: State-space mapping of the mass-spring-damper system with an external force.

Suppose the position s of the mass is measured and we are interested in the kinetic energy

$$E_k = \frac{1}{2}mv^2 = \frac{1}{2}m(ds / dt)^2. \quad (1.6)$$

Then we must extend the state-space mapping with output-variables, denoted by y , being the measured variables and variables of interest, as shown in Table 2.

Notation			
State-space	Field of application	Meaning	Unit
y_1	s	position	m
y_2	E_k	Kinetic energy	$J = kgm^2s^{-2}$

Table 2: Extension of the state-space mapping with output-variables.

The output variables are described by the output-equation,

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}}_y = \underbrace{\begin{bmatrix} x_1 \\ \frac{1}{2}p_3x_2^2 \end{bmatrix}}_{g(x,u,p)}. \quad (1.7)$$

The output equation contains all output variables on the left and corresponding algebraic expressions in state-variables, parameters, and inputs on the right, both captured in a vector. Mathematically speaking, the vector function $g(x, u, p)$ fully describes all output variables/outputs. Note that in the output equation (1.7), vector function g does not depend on u .

Together the state-equation (1.5) and the output-equation (1.7) determine the state-space model/representation of the system. Mathematically speaking, this model is fully described by $f(x, u, p)$ and $g(x, p)$. The meaning and units of the state-variables, output variables, and parameters must be obtained from the state-space mapping in *Table 1* and *Table 2*.

1.8 Linearity of dynamic systems.

If a dynamic system possesses certain linearity properties, this may simplify the analysis and control system design. Using the state-space representation, different linearity properties of systems are easily determined and defined.

A system, represented by $f(x, u, p)$ and $g(x, p)$, is linear in the state if and only if $\frac{\partial f}{\partial x}$ and $\frac{\partial g}{\partial x}$ are not functions of x . For the mass-spring-system, excluding outputs, represented by equation (1.5), we obtain

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ -p_1 / p_3 & -p_2 / p_3 \end{bmatrix}, \quad (1.8)$$

which does not depend on x . This implies that the mass-spring-damper system is linear in the state. If we do include the outputs, from the output equation (1.7) we find,

$$\frac{\partial g}{\partial x} = \begin{bmatrix} 1 & 0 \\ 0 & p_3 x_2 \end{bmatrix}. \quad (1.9)$$

Thus, the mass-spring-damper system including the output equation, is not linear in the state since (1.9) does depend on x .

Similarly, a system is linear in the input if and only if $\frac{\partial f}{\partial u}$ and $\frac{\partial g}{\partial u}$ are not functions of u . Establish yourself that the mass-spring-damper with outputs is linear in the input.

Similarly, a system is linear in the parameters if and only if $\frac{\partial f}{\partial p}$ and $\frac{\partial g}{\partial p}$ are not functions of p .

Establish yourself that the mass-spring-damper system with and without outputs, is not linear in the parameters.

A system is linear if and only if,

1. It is linear in the state.
2. It is linear in the input.
3. $f(0, 0, p) = 0, g(0, 0, p) = 0$.

Note that condition 3 is the easiest to check. Therefore, in checking whether a system is linear, first check condition 3. Also note that, to be linear, a system does not have to be linear in the parameters.

Exercise 1.1: Fishing anchovy and salmon.

Let A represent the total weight of anchovy and S the total weight of salmon present in a separate part of the sea where fishermen fish both types with an intensity f_1 and f_2 respectively. The following two differential equations describe this system where c represent constants.

$$\frac{dA}{dt} = c_1 A - c_2 AS - f_1 A, \quad c_1, c_2 > 0, f_1 \geq 0 \quad (1.10)$$

$$\frac{dS}{dt} = -c_3 S + c_4 AS - f_2 S, \quad c_3, c_4 > 0, f_2 \geq 0 \quad (1.11)$$

- a. Determine consistent units for all quantities appearing in equations (1.10), (1.11).
- b. What do the terms $c_1 A$ and $c_3 S$ represent?
- c. What do the terms $f_1 A$ and $f_2 S$ represent?
- d. What do the terms $c_2 AS$ and $c_4 AS$ represent?

Suppose we continuously measure the weight of anchovy and salmon that is fished.

- e. Determine the state-space mapping.
- f. Determine the state-space model of this system.
- g. Is the system linear in the state?
- h. Is the system linear in the input?
- i. Is the system linear?
- j. Is the system linear in the parameters?

1.9 Linearization about optimal state, input and output trajectories.

What is the optimal trajectory of a rocket flying to the moon? What is the cheapest way to start up a chemical or biochemical reactor? What is the best trajectory of a racing car on the race circuit? In all these cases the control objective is not to keep the system in a steady-state. In the second part of this course, using state-space models of systems we will compute such optimal state trajectories $x^*(t)$ and the associated optimal control $u^*(t)$ and output trajectories $y^*(t)$. Furthermore we will design controllers to compensate for small modelling errors and disturbances. The control system obtained in this way is represented in Fig. 7.

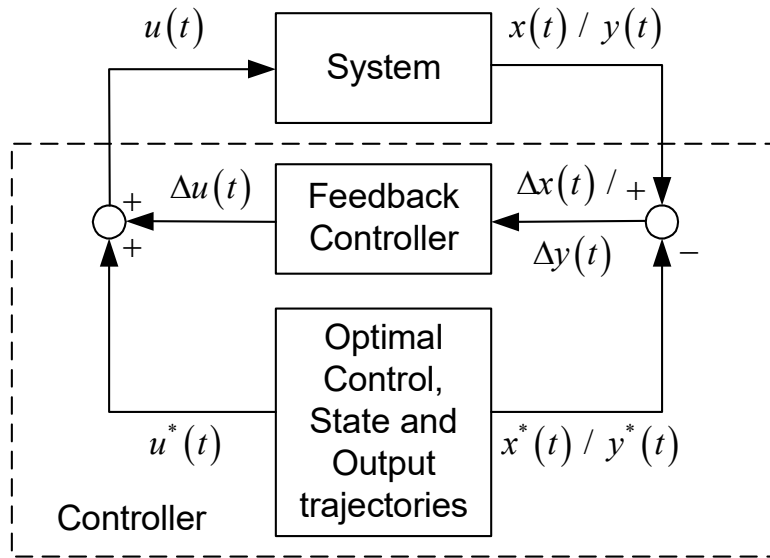


Fig. 7: An optimal control system

Apart from the computation of the optimal state, control and output trajectories, $x^*(t)$, $u^*(t)$, $y^*(t)$ another important issue is the design of the feedback controller to compensate for small disturbances and modelling errors. The feedback controller design is concerned with the problem of computing control corrections $\Delta u(t)$ from the measured state or output deviations $\Delta x(t) / \Delta y(t)$ with the objective to keep them small. Observe that,

$$\Delta x(t) = x(t) - x^*(t), \Delta u(t) = u(t) - u^*(t), \Delta y(t) = y(t) - y^*(t) \quad (1.12)$$

Since the deviations and corrections are small, their dynamic behavior can be very well approximated by a linearized model about the optimal trajectories $x^*(t)$, $u^*(t)$, $y^*(t)$. This linearized model is used for the feedback controller design because the design for linear systems is far easier. This linearized model is mathematically described by

$$\frac{d\Delta x}{dt} \approx \left. \frac{\partial f}{\partial x} \right|_{x=x^*(t), u=u^*(t)} \Delta x + \left. \frac{\partial f}{\partial u} \right|_{x=x^*(t), u=u^*(t)} \Delta u, \quad (1.13)$$

$$\Delta y \approx \left. \frac{\partial g}{\partial x} \right|_{x=x^*(t), u=u^*(t)} \Delta x + \left. \frac{\partial g}{\partial u} \right|_{x=x^*(t), u=u^*(t)} \Delta u. \quad (1.14)$$

In equations (1.13), (1.14), at each time t , the point of linearization is different, namely $x(t) = x^*(t)$, $u(t) = u^*(t)$ and therefore at each time t , matrices $A = \left. \frac{\partial f}{\partial x} \right|_{x(t)=x^*(t), u(t)=u^*(t)}$,

$B = \left. \frac{\partial f}{\partial u} \right|_{x(t)=x^*(t), u(t)=u^*(t)}$, $C = \left. \frac{\partial g}{\partial x} \right|_{x(t)=x^*(t), u(t)=u^*(t)}$, $D = \left. \frac{\partial g}{\partial u} \right|_{x(t)=x^*(t), u(t)=u^*(t)}$ are different. To mathematically

denote this difference, the linearization around the optimal state, control and output trajectory is described by,

$$\frac{d\Delta x}{dt} = A(t)\Delta x(t) + B(t)\Delta u(t) \quad (1.15)$$

$$\Delta y(t) = C(t)\Delta x(t) + D(t)\Delta u(t) \quad (1.16)$$

Equations (1.15), (1.16) represent a linear system which is time-varying. Such linear time-varying systems (LTV systems) are important for the feedback controller design which is part of the optimal control system design, represented by Fig. 7.

Example 1.1: Linearization around an optimal trajectory.

Consider the system described by,

$$\frac{dx}{dt} = -x^2 + u, \quad y = xu, \quad x, u, y \in R^1 \quad (1.17)$$

Suppose the optimal state control and output trajectories are,

$$\begin{aligned} x^*(t) &= \sin(t), \quad u^*(t) = \cos(t) + \sin^2(t), \\ y^*(t) &= \sin(t)(\cos(t) + \sin^2(t)) \end{aligned} \quad (1.18)$$

By substituting (1.18) in (1.17) we may verify that the optimal input, state and output response satisfy the system model (1.17), as they should. The time-varying linearized state equation (1.15) about these trajectories is completely determined by the two matrices

$$\begin{aligned} A(t) &= \left. \frac{\partial f}{\partial x} \right|_{x(t)=\sin(t), u(t)=\cos(t)+\sin^2(t)} = [-2x]_{x(t)=\sin(t), u(t)=\cos(t)+\sin^2(t)} = -2\sin(t), \\ B(t) &= \left. \frac{\partial f}{\partial u} \right|_{x(t)=\sin(t), u(t)=\cos(t)+\sin^2(t)} = [1]_{x(t)=\sin(t), u(t)=\cos(t)+\sin^2(t)} = 1. \end{aligned} \quad (1.19)$$

Exercise 1.2

Determine the two matrices $C(t)$, $D(t)$ in (1.16), which determine the linearized output equation.

1.10 Summary.

1. Steam engine control by means of a governor, measuring engine speed, resulted in one of the earliest and simplest feedback control systems.
2. In industry, PID feedback controllers are still very often used. They are reasonably robust and do not require mathematical modelling of the processes they control. Their tuning rules of thumb are based on simple models.
3. In case PID feedback loops strongly interact, control becomes problematic. One tries to fix this by manipulating setpoints. This manipulation is performed at a higher level that may involve modelling and optimization. This does not cure the problems of latency, the input of errors to the controller, and the need to measure all variables that are being controlled.
4. Model based control cures all these problems and also allows for optimal control.
5. To design optimal control systems, linearization along optimal state, control and possibly output trajectories is important, and provides a linearized time-varying system (LTV system) the time-varying system matrices of which are:

$$A(t) = \left. \frac{\partial f}{\partial x} \right|_{x(t)=x^*(t), u(t)=u^*(t)} \quad B(t) = \left. \frac{\partial f}{\partial u} \right|_{x(t)=x^*(t), u(t)=u^*(t)} \quad C(t) = \left. \frac{dg}{dx} \right|_{x(t)=x^*(t), u(t)=u^*(t)} \quad D(t) = \left. \frac{dg}{du} \right|_{x(t)=x^*(t), u(t)=u^*(t)}.$$

The LTV system:

$$\frac{d\Delta x}{dt} = A(t)\Delta x(t) + B(t)\Delta u(t), \quad \Delta y(t) = C(t)\Delta x(t) + D(t)\Delta u(t)$$

describes the behavior of perturbations/deviations,

$$\Delta x(t) = x(t) - x^*(t), \quad \Delta u(t) = u(t) - u^*(t), \quad \Delta y(t) = y(t) - y^*(t)$$

that should preferably be/stay close to zero.

2 Analysis, simulation and properties of dynamic systems.

2.1 Introduction and overview.

Model based control of dynamic systems, requires the ability to analyse and simulate their behavior, and to determine important dynamic system properties for control. An overview of these requirements, that will all be treated in this course, are shown in Fig. 8.

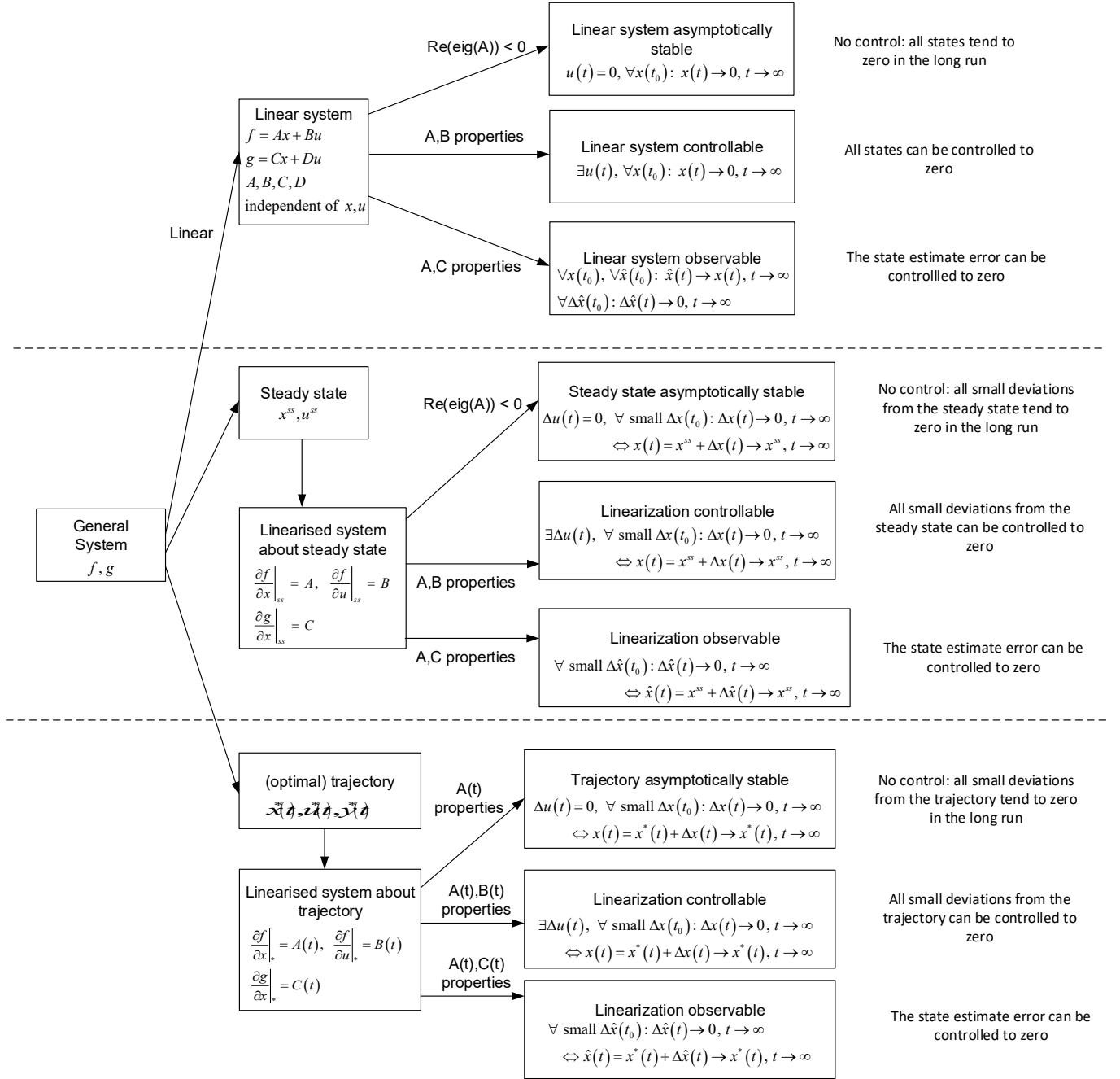


Fig. 8: Overview of the analysis, simulation and properties of different types of dynamic systems.

In the middle on the left in *Fig. 8*, the overview starts out with the most general type of dynamical system, being one that is nonlinear as described in state-space form by $f(x, u, p)$ and $g(x, u, p)$. At the top level on the left, the special case of a linear time-invariant system (LTI system) is represented. To the right the asymptotic stability, controllability and observability properties of an LTI system are represented.

In the middle, to the right of f, g , a steady-state x^{ss}, u^{ss} of this nonlinear system is considered, together with the linearization around this steady-state, being an LTI system. If this LTI system is asymptotically stable, the steady-state is asymptotically stable, implying that all small state deviations $\Delta x = x - x^{ss}$ tend to zero in the long run, without requiring control corrections, i.e. for $\Delta u(t) = u(t) - u^{ss} = 0$ (see *Fig. 7* on page 14). If the LTI system is controllable, then all small state deviations Δx can be (quickly) controlled to zero using small control corrections Δu . If one cannot measure all of the states x , and only measurements y are available (see *Fig. 7* on page 14), we have to use an observer/state-estimator to obtain estimates $\Delta \hat{x}$ of the state-deviations Δx . If the LTI system, obtained from linearization around the steady-state is observable, an observer/state-estimator, that uses Δy and Δu to estimate Δx , can be designed that (quickly) reduces the error in the estimate $\Delta \hat{x}$ of Δx to zero. All these are important properties for optimal control system design (see *Fig. 7* on page 14).

This also holds for the same properties mentioned at the bottom level of *Fig. 8* that considers linearization about a (optimal) control, state and possibly output trajectory $u^*(t)$, $x^*(t)/y^*(t)$ of the nonlinear system f, g . In this case, the linearization is an LTV system (see section 1.8, page 15) for which the same properties exist. Again, these are important properties for the optimal control system design (see *Fig. 7* on page 14). Therefore, these properties and their computation, will be important topics of this course.

After reading this chapter one should:

- 1) Know the overview represented by *Fig. 8*.
- 2) Know how numerical integration provides responses/trajectories/solutions of differential equations represented in state-space form.
- 3) Know the meaning and importance of asymptotic stability.
- 4) Know the state-transition matrix of LTI and LTV systems and how to numerically compute these.
- 5) Know the forced and free response of LTI and LTV systems.

2.2 Numerical solutions of the state-equation.

Recall the state-equation of a dynamic system given by,

$$\frac{dx}{dt} = f(x, u, p). \quad (2.1)$$

Given the initial state $x(t_0)$ and the input $u(t)$, $t_0 \leq t \leq t_f$, we may compute the system response/behavior $x(t)$, $t_0 \leq t \leq t_f$ using Euler forward numerical integration that uses a first-order Taylor series approximation:

$$x(t + \Delta t) \cong x(t) + \frac{dx}{dt} \Delta t \Leftrightarrow x(t + \Delta t) = x(t) + f(x(t), u(t), p) \Delta t \quad (2.2)$$

where Δt is a small enough time-step such that the error due to the first-order Taylor series approximation (2.2) remains small. Starting from the state and input at time t_0 , i.e. $x(t_0)$, $u(t_0)$, one can compute from the state-space model (2.1) and the first-order Taylor series approximation (2.2) the value of the state at time $t_0 + \Delta t$, i.e. $x(t_0 + \Delta t)$. Doing this repeatedly we obtain $x(t_0 + k\Delta t)$, $k = 0, 1, 2, \dots, K$, $K = (t_f - t_0) / \Delta t$. To gain accuracy, more advanced numerical integrators approximate also higher-order terms in a Taylor series expansion. To obtain values of the state at times different from $t_0 + k\Delta t$, we can interpolate. In this way $x(t)$, $t_0 \leq t \leq t_f$ is obtained, a solution of the differential equations (2.1) with initial state $x(t_0)$ and input $u(t)$, $t_0 \leq t \leq t_f$.

A special type of solution of a differential equation is a steady-state solution $x(t) = x^{ss}$, $t_0 \leq t \leq t_f$. Then, from equation (2.1), $u(t) = u^{ss}$, $f(x^{ss}, u^{ss}, p) = 0$, $t_0 \leq t \leq t_f$ must hold.

2.3 Asymptotic stability of solutions to the state-equation.

Roughly speaking, a (desired) solution $x^*(t)$, $t_0 \leq t < \infty$ of a state-equation is asymptotically stable if small deviations $\Delta x(t) = x(t) - x^*(t)$ from the solution tend to zero as time tends to infinity. Formally, (asymptotic) stability considers solutions over infinitely long time-periods. In practice, (asymptotic) stability tells you what happens “in the long run”. Small state deviations $\Delta x(t)$ tending to zero in the long run, is a desirable property as it comes to control, because it means that small disturbances will not affect a (desired) solution too much.

But what if a desired solution is not asymptotically stable? Then disturbances will not tend to zero in the long run and will seriously affect the solution. Is there anything we can do about this? To that end, consider again the optimal control system represented by Fig. 7 on page 14, where $x^*(t)$ is the (optimal/desired) solution. Observe that we may introduce perturbation state-

feedback, i.e. compute control corrections $\Delta u(t)$ from the measured state-deviations $\Delta x(t)$ with the objective to control $\Delta x(t)$ to zero.

Asymptotic stability considers the behavior of small state-perturbations $\Delta x(t)$ from a (optimal/desired) solution $x^*(t)$ of the state-equation. Since they are small, the behavior of state-deviations $\Delta x(t)$ may be very well approximated by linearizing the state-equation (2.1) around the (desired) solution $u^*(t)$, $x^*(t)$, $t_0 \leq t \leq t_f$. This linearization is the LTV system described in section 1.9, equations (1.13), (1.15). This reveals that asymptotic stability of a (optimal/desired) solution of the state-equation, is fully determined by stability properties of the linearization about this trajectory. If the solution of the state-equation is a steady-state, i.e. $x(t) = x^{ss}$, $u(t) = u^{ss}$, $t_0 \leq t \leq t_f$, then the linearization is an LTI system. So stability of both LTV and LTI systems will decide on the stability of solutions to the state-equation.

2.4 Responses and asymptotic stability of LTV and LTI systems.

2.4.1 State and input versus small state and input perturbations.

LTV and LTI systems are obtained in section 2.3 as descriptions of small perturbations $\Delta x(t) = x(t) - x^*(t)$, $\Delta u(t) = u(t) - u^*(t)$ from desired responses/trajectories $x^*(t)$, $u^*(t)$ of a general nonlinear state-equation (2.1). These small perturbations are the state and input of the LTV and LTI systems. In considering and analysing systems, their state and input are generally referred to as x, u respectively, as we will also do next. So keep in mind that x, u in this section may be representing perturbations Δx , Δu . Of course, the original state equation (2.1) may be of the LTI or LTV type itself. Then x, u in this section refer directly to state equation (2.1) and do not represent small perturbations.

2.4.2 The state-transition matrix and responses of LTV and LTI systems.

To consider asymptotic stability of LTV and LTI systems, we must first introduce the state-transition matrix of a general LTV system

$$\frac{dx}{dt} = A(t)x + B(t)u. \quad (2.3)$$

When $u(t) = 0$, the solution or state-response $x(t)$ is given by,

$$x(t) = \Phi(t, t_0)x(t_0), \quad (2.4)$$

where $\Phi(t, t_0)$ is a square $n_x \times n_x$ matrix that is invertible. This matrix is called the state-transition matrix of the LTV system (2.3). The state-transition matrix has the following properties,

$$\Phi(t_0, t_0) = I \quad (2.5)$$

$$\frac{d}{dt} \Phi(t, t_0) = A(t) \Phi(t, t_0) \quad (2.6)$$

$$\Phi(t_2, t_1) \Phi(t_1, t_0) = \Phi(t_2, t_0) \quad (2.7)$$

For a nonzero $u(t)$ the solution or state-response $x(t)$ is given by,

$$x(t) = \underbrace{\Phi(t, t_0) x(t_0)}_{\text{Free response}} + \underbrace{\int_{t_0}^t \Phi(\tau, t_0) B(\tau) u(\tau) d\tau}_{\text{Forced response}} \quad (2.8)$$

This follows from direct substitution of (2.8) into (2.3) and the properties (2.5), (2.6). The first term on the right in equation (2.8) is called the free response, which is caused by the initial state $x(t_0)$, whereas the second term, called the forced response, is caused by the input $u(t)$. For LTI systems, the state-transition matrix is also called the matrix exponential because,

$$\Phi(t, t_0) = e^{A(t-t_0)} \quad (2.9)$$

where,

$$e^{A(t-t_0)} = I + A(t-t_0) + \frac{A^2(t-t_0)^2}{2!} + \frac{A^3(t-t_0)^3}{3!} + \frac{A^4(t-t_0)^4}{4!} + \dots \quad (2.10)$$

If we take $\Delta t = t - t_0$ small, then equation (2.10) can be very well approximated by,

$$e^{A\Delta t} \cong I + A\Delta t + \frac{A^2\Delta t^2}{2!} \quad (2.11)$$

From the property (2.7), taking $t_1 = t_0 + \Delta t$ and $t_2 = t_1 + \Delta t$ we find,

$$e^{A2\Delta t} = e^{A\Delta t} e^{A\Delta t} \quad (2.12)$$

and, in general,

$$e^{Ak\Delta t} = \underbrace{e^{A\Delta t} e^{A\Delta t} \dots e^{A\Delta t}}_{k \text{ times}}. \quad (2.13)$$

In equation (2.13), we can use the approximation (2.11) to compute the matrix exponential. This is called “scaling and squaring”. The scaling refers to choosing Δt sufficiently small. The squaring refers to equation (2.12) that is repeated in equation (2.13). Scaling and squaring is a popular, simple, and accurate method to compute the matrix exponential in any case. There are many alternatives to compute the matrix exponential, one of which has been presented in the modelling dynamic systems course. That method relied on eigenvectors and eigenvalues of the matrix A . But it does not work if there are eigenvalues having multiple eigenvectors.

The “scaling and squaring” method to compute the matrix exponential can also be used to compute the state-transition matrix of LTV systems. Since Δt must be small, we can approximate the LTV system in between $t_0 + (i-1)\Delta t$ and $t_0 + i\Delta t$, $i = 1, 2, \dots$ by an LTI system,

$$\begin{aligned} \frac{dx}{dt} &= A_i x + B_i u, \\ A_i &= \frac{A(t_0 + (i-1)\Delta t) + A(t_0 + i\Delta t)}{2}, \quad B_i = \frac{B(t_0 + (i-1)\Delta t) + B(t_0 + i\Delta t)}{2}, \\ t_0 + (i-1)\Delta t &\leq t \leq t_0 + i\Delta t \end{aligned} \quad (2.14)$$

Using the property (2.7), from equation (2.11) with A replaced by A_i we obtain,

$$\Phi(t_0, t_0 + k\Delta t) = e^{A_1\Delta t} e^{A_2\Delta t} \dots e^{A_k\Delta t}, \quad e^{A_i\Delta t} \cong I + A_i\Delta t + \frac{A_i^2\Delta t^2}{2}, \quad i = 1, 2, \dots, k. \quad (2.15)$$

Equation (2.15) together with the expressions for A_i in equation (2.14), represents a way to compute the state-transition matrix for LTV systems. If an LTV system has measured variables, or variables of interest, these are described by the output equation

$$y(t) = C(t)x(t) \quad (2.16)$$

From equations (2.16), (2.8) the output-response $y(t)$ satisfies

$$y(t) = \underbrace{C(t)\Phi(t, t_0)x(t_0)}_{\text{Free response}} + \underbrace{C(t)\int_{t_0}^t \Phi(\tau, t_0)B(\tau)u(\tau)d\tau}_{\text{Forced response}}. \quad (2.17)$$

If the system is LTI, then the state-response (2.8) and the output response (2.17) become

$$x(t) = \underbrace{e^{A(t-t_0)}x(t_0)}_{\text{Free response}} + \underbrace{\int_{t_0}^t e^{A(\tau-t_0)}Bu(\tau)d\tau}_{\text{Forced response}}. \quad (2.18)$$

$$y(t) = \underbrace{Ce^{A(t-t_0)}x(t_0)}_{\text{Free response}} + \underbrace{C\int_{t_0}^t e^{A(\tau-t_0)}Bu(\tau)d\tau}_{\text{Forced response}}. \quad (2.19)$$

In analysing LTI systems, as we will do in the next chapter, the transfer from each input to each output is considered separately. So the analysis concerns SISO LTI systems. The response of SISO LTI systems to two specific inputs $u(t)$ are generally considered. The first input is an impulse $u(t) = \delta(t)$ which is a time-function that is only non-zero at time $t = 0$ while having a surface underneath equal to one. This input has the special property that the state-response (2.18) of an SISO LTI system equals

$$x(t) = \underbrace{e^{A(t-t_0)}x(t_0)}_{\text{Free response}} + \underbrace{e^{A(t-t_0)}B}_{\text{Forced response}}, \quad (2.20)$$

whereas the output-response (2.19) equals

$$y(t) = \underbrace{Ce^{A(t-t_0)}x(t_0)}_{\text{Free response}} + \underbrace{Ce^{A(t-t_0)}B}_{\text{Forced response}}. \quad (2.21)$$

What is called the impulse response of a SISO LTI system neglects the free response and assumes the LTI system has an output. From equation (2.21) the impulse response thus equals

$$y(t) = Ce^{A(t-t_0)}B. \quad (2.22)$$

The other input signal that is generally considered is the step input $u(t) = 1, t \geq t_0$. In this case the state-response (2.18) is

$$x(t) = \underbrace{e^{A(t-t_0)}x(t_0)}_{\text{Free response}} + \underbrace{\int_{t_0}^t e^{A(\tau-t_0)}Bd\tau}_{\text{Forced response}}, \quad (2.23)$$

whereas the output-response is

$$y(t) = \underbrace{Ce^{A(t-t_0)}x(t_0)}_{\text{Free response}} + \underbrace{C \int_{t_0}^t e^{A(\tau-t_0)}Bd\tau}_{\text{Forced response}}. \quad (2.24)$$

What is called the step response of a SISO LTI system neglects the free response and assumes the LTI system has an output. From equation (2.24) the step response thus equals

$$y(t) = C \int_{t_0}^t e^{A(\tau-t_0)}Bd\tau. \quad (2.25)$$

2.4.3 Asymptotic stability of LTI and LTV systems.

An alternative method to compute the matrix exponential, obtained from the modelling dynamic systems course, is

$$e^{A(t-t_0)} = Ve^{D(t-t_0)}V^{-1}. \quad (2.26)$$

In equation (2.26), D is an $n_x \times n_x$ diagonal matrix with the eigenvalues of matrix A on the diagonal. V is also an $n_x \times n_x$ matrix. The i^{th} column of matrix V is the eigenvector corresponding to the i^{th} eigenvalue on the diagonal of matrix D . Both matrices D and V can be obtained from an eigenvalue decomposition of matrix A (function eig in Matlab). Equation (2.26) does not work when matrix V is not invertible, which happens if some eigenvalue has multiple eigenvectors.

On the other hand, equation (2.26) immediately reveals if an LTI system is asymptotically stable. To see this, observe from equation (2.10) on page 20, that if D is a diagonal matrix,

$$e^{D(t-t_0)} = \begin{bmatrix} e^{\lambda_1(t-t_0)} & 0 & 0 & 0 \\ 0 & e^{\lambda_2(t-t_0)} & 0 & 0 \\ 0 & 0 & \vdots & 0 \\ 0 & 0 & 0 & e^{\lambda_{n_x}(t-t_0)} \end{bmatrix}. \quad (2.27)$$

From equation (2.27) we find that $e^{D(t-t_0)} \rightarrow 0$ as $t \rightarrow \infty$ if and only if all eigenvalues λ_i , $i=1,2,\dots,n_x$ have a negative real part. To see this, recall that the imaginary parts of the eigenvalues only determine the frequencies of the oscillatory state-response whereas the real parts determine the exponential growth or decay of the state-response. Then, from equation (2.26) we find that: an LTI system is asymptotically stable if and only if all eigenvalues λ_i , $i=1,2,\dots,n_x$ of the $n_x \times n_x$ matrix A have a negative real part.

In general, for LTV systems one cannot find properties of the matrices $A(t)$ that determine asymptotic stability of the LTV system. The only thing one can do is compute $\Phi(t, t_0)$ and $\Phi(t, t_0)x(t_0)$ to see if $x(t) \rightarrow 0$. This implies that we can easily check the asymptotic stability of steady-states $u^*(t) = u^{ss}$, $x^*(t) = x^{ss}$, $f(x^{ss}, u^{ss}, p) = 0$ of dynamic systems, by considering the asymptotic stability of their linearization, which is an LTI system. But to determine the asymptotic stability of a solution $u^*(t)$, $x^*(t)$, that is not a steady-state of a dynamic system, is difficult, since the linearization about the solution is an LTV system. If solutions $u^*(t)$, $x^*(t)$, are not asymptotically stable, we may use control corrections $\Delta u(t) = u(t) - u^*(t)$ to try to control $\Delta x(t) = x(t) - x^*(t)$ to zero (see Fig. 7 on page 14). Whether this can be done successfully depends critically on the controllability and stabilizability properties of the linearizations around the solution $u^*(t)$, $x^*(t)$. These properties will be considered in chapter 4.

2.5 What if the system is linear?

Wageningen University is a “Life Science University”. As a result, most dynamic systems in Wageningen are “alive” and nonlinear. At Technical Universities, one much more often deals with systems that are linear and time-invariant (LTI), such as electrical systems and circuits. These LTI systems are much more easy to analyse and design control systems for. Probably therefore, historically, control engineering for a long time just focussed on LTI systems.

In this chapter, we have considered solutions $u^*(t)$, $x^*(t)$, and their asymptotic stability, of dynamic systems that may be nonlinear. So what changes as to the asymptotic stability of

solutions, if the system is LTI? Verify yourself that in this case, irrespective of the solution $u^*(t)$, $x^*(t)$, the linearization is the very same LTI system as the original. Also verify yourself, that this linearization is no longer an approximate, but an exact description of the behavior of deviations, which no longer must be small, to be described properly.

Summarizing: when the system is nonlinear, asymptotic stability of a solution guarantees that small perturbations $\Delta x(t)$ tend to zero in the long run. When the system is LTI, asymptotic stability guarantees that arbitrary large perturbations tend to zero in the long run.

3 Transfer function analysis and state-space representations of LTI systems.

3.1 Introduction.

As we remarked in section 2.5, historically, the emphasis of control engineering was on LTI systems and lasted for a long time, and still continues. Since most dynamic systems in Wageningen are “alive”, and therefore nonlinear, we started this course by considering nonlinear dynamic systems. But we have seen in the previous chapter, that linearization is required to find out about the stability of solutions. Also, we have seen that linearizations provide a reasonable description, as long as higher-order terms in a Taylor series expansion remain negligible. This section will focus on realizations and analysis of LTI systems relying on transfer functions. Transfer functions are descriptions providing a frequency-domain analysis, applicable only to LTI systems. Although this course relies mostly on time-domain analysis, that is much more generally applicable, given the popularity and additional insight frequency domain analysis provides, it is presented as part of this course.

After reading this chapter one should:

- 1) Know the difference between white box and black box modelling.
- 2) Convert state-space representations of LTI systems to transfer functions and vice-versa.
- 3) Know the meaning of poles of transfer functions and their relation to eigenvalues of state-space representations.
- 4) Know the characteristic polynomial associated with transfer functions.
- 5) Know how Bode plots represent properties of LTI systems in the frequency domain.

3.2 Transfer functions, black box and white box modelling

A dynamic system may be considered a box with inputs and outputs, see Fig. 9. One way to find

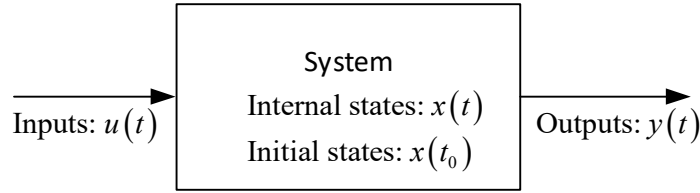


Fig. 9: Dynamic system as a box with internal states, inputs and outputs

out about the workings of the box is to find relations between the inputs and the outputs. Another is to open the box, and see what is inside. The first does not use any knowledge of what is inside the box and is therefore called black box modelling. The latter does use knowledge of what is inside the box and is therefore called white box modelling. Building mathematical models from scientific knowledge, called “first-principles modelling”, requires looking inside the box, and thus falls into the category of white box modelling. Transfer functions of LTI systems relate inputs to outputs of the system. Parts of the system that are not connected to the input, or not connected to the output, are ignored by transfer functions, because they play no role in relating the inputs to the outputs. Also, different systems/boxes may relate inputs and outputs in exactly the same manner. Transfer functions do not distinguish between these systems/boxes. In terms of state-space models, transfer functions do not care about the state-representation inside the box. Any internal state-representation that provides the input-output relations of the LTI system will do. Also, transfer functions generally ignore the free response of the system caused by a nonzero initial condition $x(t_0)$ and only consider the forced response in equation (2.19).

3.3 Transfer functions/matrices of LTI systems.

Starting from the state-space model of an LTI system

$$\frac{dx}{dt} = Ax + Bu, \quad (3.1)$$

$$y = Cx, \quad (3.2)$$

its transfer function can be obtained by replacing d/dt by s . Behind this replacement are the Laplace and Fourier transforms, as explained in section 3.5 on page 33. The Laplace transform changes time functions into functions of s . To indicate this change, after transformation, the time functions $u(t), x(t), y(t)$ will be denoted by $U(s), X(s), Y(s)$. These replacements turn the state-space model (3.1), (3.2) into

$$sX(s) = AX(s) + BU(s), \quad (3.3)$$

$$Y(s) = CX(s). \quad (3.4)$$

The Laplace transformation changes the differential equations (3.1) into the algebraic equations (3.3). Solving differential equations then changes into solving algebraic equations, which is easier. The transfer function that relates the output to the input is

$$H(s) = \frac{Y(s)}{U(s)}. \quad (3.5)$$

From equation (3.3) we obtain $(sI - A)X(s) = BU(s)$ and therefore $X(s) = (sI - A)^{-1}BU(s)$. Next, from equation (3.4) we obtain $Y(s) = C(sI - A)^{-1}BU(s)$. Then from equation (3.5) we find

$$H(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B. \quad (3.6)$$

Example 3.1: Transfer function of a mass-spring-damper system.

Reconsider the mass-spring-damper system on pages 9, 10. Take as spring and damper constants $c_s = 1$, $c_d = 1$, and assume the mass $m = 1$. Furthermore, neglect the influence of gravity by taking $g = 0$. Also assume only the position of the mass is measured. Verify from equations (1.4), (1.5) and the state-space mapping in Fig. 6 that this mass-spring-damper system is a single input-single output (SISO) LTI system described by

$$\frac{dx}{dt} = Ax + Bu, \quad A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (3.7)$$

$$y = Cx, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \quad (3.8)$$

From equations (3.6)-(3.8) its transfer function is

$$H(s) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_C \underbrace{\begin{bmatrix} s & -1 \\ 1 & s+1 \end{bmatrix}^{-1}}_{(sI-A)^{-1}} \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_B = \quad (3.9)$$

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \frac{1}{s^2 + s + 1} \begin{bmatrix} s+1 & 1 \\ -1 & s \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{s^2 + s + 1}.$$

If an LTI system has n_u inputs and n_y outputs, the transfer function $H(s)$ in equation (3.6) is a transfer matrix with dimensions $n_y \times n_u$. As a simple example take $A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$, $B = C = I$.

Then

$$H(s) = \underbrace{I \begin{bmatrix} s & -1 \\ 1 & s+1 \end{bmatrix}^{-1} I}_{(sI-A)^{-1}} = g = p_4 = 0 \quad (3.10)$$

$$\frac{1}{s^2 + s + 1} \begin{bmatrix} s+1 & 1 \\ -1 & s \end{bmatrix} = \begin{bmatrix} \frac{s+1}{s^2 + s + 1} & \frac{1}{s^2 + s + 1} \\ \frac{-1}{s^2 + s + 1} & \frac{s}{s^2 + s + 1} \end{bmatrix}.$$

In general, element i, j of the transfer matrix $H(s)$ is the transfer function $Y_i(s)/U_j(s)$ from the j^{th} input to the i^{th} output.

In section 3.2 we remarked that transfer functions/matrices do not represent or care about the internal state-representation of the system. Any internal state-representation will do. This suggests that there must be alternative internal state-representations of the system having the same transfer function. Observe that Example 3.1 started out from a certain internal state-representation of the system, namely the one obtained from first-principles on pages 9, 10. From this internal state-representation the transfer function was obtained. But how about alternative internal state-representations? These can be obtained by changing coordinates that represent the internal state. Changes of coordinates can simplify the underlying mathematics and computations but obscure the physical interpretation of state-variables.

One way to simplify computations and the mathematical analysis of LTI systems is to change matrix A into a diagonal matrix D by means of a change of coordinates. The change of coordinates is found from an eigenvalue decomposition of matrix

$$A = VDV^{-1}. \quad (3.11)$$

Diagonal matrix D has the eigenvalues of matrix A on its diagonal. Matrix V contains as columns the corresponding eigenvectors. Changing matrix A into D is achieved by a change of coordinates determined by matrix V^{-1} as follows. Each column of matrix V^{-1} is a basis vector of the new basis in which the internal state x is expressed. The fact that V^{-1} is invertible guarantees

that its columns are independent and thus form a basis. In this new basis, the original internal state x is represented as

$$x' = V^{-1}x \quad (3.12)$$

The corresponding change of the LTI dynamics is

$$\frac{dx'}{dt} = \underbrace{V^{-1}AV}_{A'=D} x' + \underbrace{V^{-1}B}_{B'} u, y = \underbrace{CV}_{C'} x' \quad (3.13)$$

Example 3.2: Diagonalizing the mass-spring-damper system by a change of coordinates.

The LTI system matrices of the mass-spring-damper system (3.7), (3.8) are $A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$,

$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $C = [1 \ 0]$. From an eigenvalue decomposition of matrix A we obtain

$$V = \begin{bmatrix} 0.707 & 0.707 \\ -0.354 + 0.612i & -0.354 - 0.612i \end{bmatrix}, V^{-1} = \begin{bmatrix} 0.707 - 0.408i & -0.817i \\ 0.707 + 0.408i & +0.817i \end{bmatrix}$$

$$D = A' = \begin{bmatrix} -0.5 + 0.866i & 0 \\ 0 & -0.5 - 0.866i \end{bmatrix}, B' = V^{-1}B = \begin{bmatrix} 0.817i \\ 0.817i \end{bmatrix}, C' = [0.707 \ 0.707].$$

Observe that the columns of matrix V^{-1} , that represent the new basis vectors, contain complex numbers! This happens because the eigenvalues and eigenvectors contain complex numbers. As a result the system matrices $A' = D$, B' also contain complex numbers like the new state-vector x' . With this, the physical interpretation is lost completely. However, all this leaves unchanged the relation between the input and output of the LTI system.

3.4 General and special internal state-representations of LTI systems

Recall the general transfer matrix of an LTI system given by

$$H(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1} B. \quad (3.14)$$

The square matrix $(sI - A)^{-1}$, that largely determines properties of the transfer matrix $H(s)$, has the following form

$$(sI - A)^{-1} = \frac{1}{p(s)} \begin{bmatrix} q_{11}(s) & \cdots & q_{1n}(s) \\ \vdots & \ddots & \vdots \\ q_{n1}(s) & \cdots & q_{nn}(s) \end{bmatrix}, \quad (3.15)$$

where p , q are polynomials in s , specifically

$$p(s) = \det(sI - A) = s^{n_x} + p_{n_x-1}s^{n_x-1} + \dots + p_1s + p_0 \quad (3.16)$$

$$q_{ij}(s) = \text{determinant of } (sI - A) \text{ with row } i \text{ and column } j \text{ dropped} \quad (3.17)$$

Note that the degree of $q_{ij}(s)$ is at least one less than n_x , the degree of $p(s)$ that is called the characteristic polynomial of the transfer matrix $H(s)$. The zeros of $p(s)$ are called the poles of the transfer matrix $H(s)$ for which it becomes infinite. The interpretation of this situation is that there is a non-zero output for zero input. In this case, the non-zero output reflects the response to non-zero initial conditions. If s_0 is a pole then it is also an eigenvalue of matrix A of the LTI system. The reverse is not necessarily true as illustrated by the following example.

Example 3.3: LTI system with an eigenvalue not being a pole

Consider the LTI system with $A = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$. Its transfer function is

$$\begin{aligned} H(s) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s+1 & 0 \\ 0 & s \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{s}{s(s+1)} & 0 \\ 0 & \frac{s+1}{s(s+1)} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= \frac{s}{s(s+1)} = \frac{1}{s+1} \end{aligned} \quad (3.18)$$

So $p(s) = s+1$ and $s_1 = -1$ is the single pole. The eigenvalues of matrix A are $\lambda_1 = 1$, $\lambda_2 = 0$ showing that $\lambda_2 = 0$ does not correspond to a pole. Here, this is due to the fact that state x_2 does not influence the output, and therefore the transfer function. But obviously, state x_2 does influence the state-response of the system.

Starting from a given transfer matrix $H(s)$, infinitely many internal state-representations, also called realizations, can be obtained since one can choose infinitely many bases to represent the internal state. For SISO LTI systems the transfer matrix $H(s)$ is a single transfer function

$$H(s) = \frac{Y(s)}{U(s)} = \frac{q_{n_x-1}s^{n_x-1} + \dots + q_1s + q_0}{s^{n_x} + p_{n_x-1}s^{n_x-1} + \dots + p_1s + p_0}. \quad (3.19)$$

Recalling that s replaces d/dt , a straightforward internal state-representation is

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ \vdots \\ x_{n_x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & 0 \\ 0 & \dots & & 0 & 1 \\ -p_0 & -p_1 & \dots & \dots & -p_{n_x-1} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n_x} \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u, \quad y = [q_0 \quad \dots \quad q_{n_x-1}] \begin{bmatrix} x_1 \\ \vdots \\ x_{n_x} \end{bmatrix} \quad (3.20)$$

To see this, observe that the first $n_x - 1$ state differential equations imply $X_{i+1}(s) = sX_i(s)$ so that $X_{i+1}(s) = s^i X_1(s)$, $i = 0, 1, \dots, n_x - 1$. Then from the last differential equation we find

$U(s) = (p_0 + p_1s + p_2s^2 + \dots + p_{n_x-1}s^{n_x-1} + s^{n_x})X_1(s)$. The output equation implies

$Y(s) = (q_{n_x-1}s^{n_x-1} + \dots + q_1s + q_0)X_1(s)$. Then $Y(s)/X(s)$ is indeed the transfer function (3.19).

The realization (3.20) is called the controller canonical realization. Another realization is obtained as follows. Since the transfer function $H(s)$ of a SISO LTI system is a scalar, we have

$$\underbrace{H(s)}_{1 \times 1} = H^T(s) = \underbrace{B^T}_{1 \times n_x} (sI - A^T)^{-1} \underbrace{C^T}_{n_x \times 1} \quad (3.21)$$

From equation (3.21) and equation (3.14), observe that $A' = A^T$, $B' = C^T$, and $C' = B^T$ is also a realization of the SISO LTI system. This realization is called observer canonical realization given by

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 & -p_0 \\ 1 & \ddots & \vdots & -p_1 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & -p_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} q_0 \\ \vdots \\ q_{n-1} \end{bmatrix} u, \quad y = [0 \quad \dots \quad 0 \quad 1] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}. \quad (3.22)$$

A third realization of an LTI SISO system obtained from the transfer function (3.19) is obtained using

$$p(s) = (s - a_1)(s - a_2) \dots (s - a_{n_x}), \quad (3.23)$$

where $a_1, a_2, \dots, a_{n_x}$ are the zeros (roots) of $p(s)$. Then the fractional expansion

$$\frac{q(s)}{p(s)} = \frac{A_1}{s - a_1} + \dots + \frac{A_n}{s - a_n}. \quad (3.24)$$

provides the following diagonal realization

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & & 0 \\ 0 & \dots & 0 & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} A_1 & \dots & A_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}. \quad (3.25)$$

Observe that Example 3.2 on page 30 resulted “almost” in this type of realization. Try to find yourself a state-transformation that scales up the two state-variables by the same imaginary number to obtain the diagonal realization (3.25).

3.5 Transfer functions, frequency domain analysis and Fourier and Laplace transforms.

LTI SISO systems have the special property that a sine wave at the input $u(t) = A_{in} \cos(\omega t + \phi_1)$ produces a sine wave at the output $y(t) = A_{out} \cos(\omega t + \phi_{out})$, if we neglect the free response. The angular frequency ω of both sine waves is identical, only the amplitudes A_{in}, A_{out} and the phases ϕ_{in}, ϕ_{out} are different. The phase shift $\phi_s = \phi_{out} - \phi_{in}$ and the amplitude ratio $A_r = A_{out} / A_{in}$ depend on the angular frequency ω as well as properties of the LTI SISO system. Both can be calculated from the transfer function $H(s)$ when we substitute $s = j\omega$,

$$A_r(\omega) = A_{out}(\omega) / A_{in}(\omega) = |H(j\omega)|, \quad \phi_s(\omega) = \phi_{out}(\omega) - \phi_{in}(\omega) = \arg(H(j\omega)). \quad (3.26)$$

The amplitude ratio A_r and phase shift ϕ_s , as a function of $\omega \geq 0$, make up the *Bode plot* of an LTI SISO system, see Fig. 3 on page 5.

Every input $u(t)$ can be considered as an infinite sum (integral) of sine waves with different phases, frequencies and infinitely small amplitudes (amplitude densities). Suppose for a certain arbitrary input signal $u(t)$ we know all these phases $\phi(\omega)$ and amplitude densities $A(\omega)$ for each angular frequency $0 \leq \omega < \infty$. Knowing also all amplitude ratios $A_r(\omega) = A_{out}(\omega) / A_{in}(\omega)$

and phase shifts $\phi_s(\omega) = \phi_{out}(\omega) - \phi_{in}(\omega)$ for each angular frequency ω , by applying superposition, the associated output signal is obtained as an (infinite) sum of sine waves, see Fig. 10 on page 35. Given $u(t)$, finding its phases $\phi_{in}(\omega)$ and amplitude densities $A_{in}(\omega)$ for each angular frequency $0 \leq \omega < \infty$ is performed by the Fourier Transform given by

$$U(\omega) = \int_{-\infty}^{\infty} e^{-j\omega t} u(t) dt, \quad -\infty < \omega < +\infty. \quad (3.27)$$

Note that for every $\omega \geq 0$, $U(\omega)$ is a complex number and $U(-\omega) = U^*(\omega)$. From $U(\omega)$ we obtain the so called amplitude spectrum $A(\omega)$ and phase spectrum $\phi(\omega)$ of the input $u(t)$

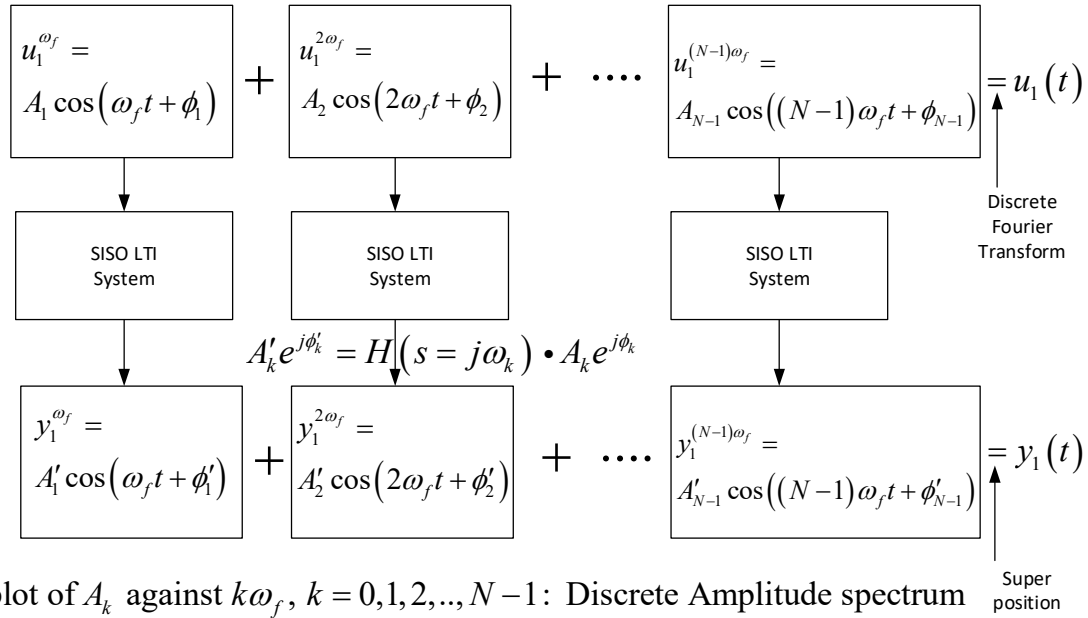
$$A_{in}(\omega) = |U(\omega)|, \quad \phi_{in}(\omega) = \arg(U(\omega)). \quad (3.28)$$

Note that $A_{in}(\omega)$ and $\phi_{in}(\omega)$ are real. For each $\omega \geq 0$, multiplying the amplitudes $A_{in}(\omega)$ with the ratios $A_r(\omega) = |H(j\omega)|$ to become $A_{out}(\omega)$ and shifting the phases $\phi_{in}(\omega)$ by $\phi_s(\omega) = \arg(H(j\omega))$ to become $\phi_{out}(\omega)$ comes down to multiplication by the transfer function $H(j\omega)$. By applying superposition, the output signal $Y(\omega)$ is obtained as an infinite sum (integral) of sine waves $A_{out} \cos(\omega t + \phi_{out})$. For each $\omega \geq 0$, the transfer from $U(\omega)$ to $Y(\omega)$ is given by $Y(\omega) = H(j\omega)U(\omega)$ which complies with $H(s)$ in equation (3.5) for $s = j\omega$. To find the corresponding output $y(t)$ from $Y(\omega)$, we have to apply the inverse Fourier transform

$$y(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} Y(\omega) e^{j\omega t} d\omega, \quad -\infty < t < +\infty. \quad (3.29)$$

The whole procedure of 1) Fourier transformation of the input $u(t)$ to $U(\omega)$, 2) Multiplication with the transfer function: $Y(\omega) = H(j\omega)U(\omega)$ for every $\omega \geq 0$ and 3) Inverse Fourier transformation to obtain the output $y(t)$ from $Y(\omega)$ is represented by Fig. 10 for a finite number of discrete frequencies $\omega = k\omega_f$, $k = 1, 2, \dots, N-1$ with $A_{in}(k\omega_f) = A_k$, $\phi_{in}(k\omega_f) = \phi_k$, $A_{out}(k\omega_f) = A'_k$, $\phi_{out}(k\omega_f) = \phi'_k$. Fig. 11 reveals how $A_k \cos(k\omega_f t + \phi_k)$ is represented by the complex number $A_k e^{j\phi_k}$. As opposed to this frequency-domain, transfer function approach, the time-domain approach employs numerical integration to obtain both $x(t)$ and $y(t)$ directly from $u(t)$.

The Fourier and inverse Fourier transform (3.27), (3.29) are special cases of the Laplace transform and inverse Laplace transform taking $j\omega$ for the complex number s .



plot of A_k against $k\omega_f$, $k = 0, 1, 2, \dots, N-1$: Discrete Amplitude spectrum

plot of ϕ_k against $k\omega_f$, $k = 0, 1, 2, \dots, N-1$: Discrete Phase spectrum

Complex number representing A_k and ϕ_k : $A_k e^{j\phi_k} = A_k (\cos(\phi_k) + j \sin(\phi_k))$

Similar arguments apply to A'_k , ϕ'_k

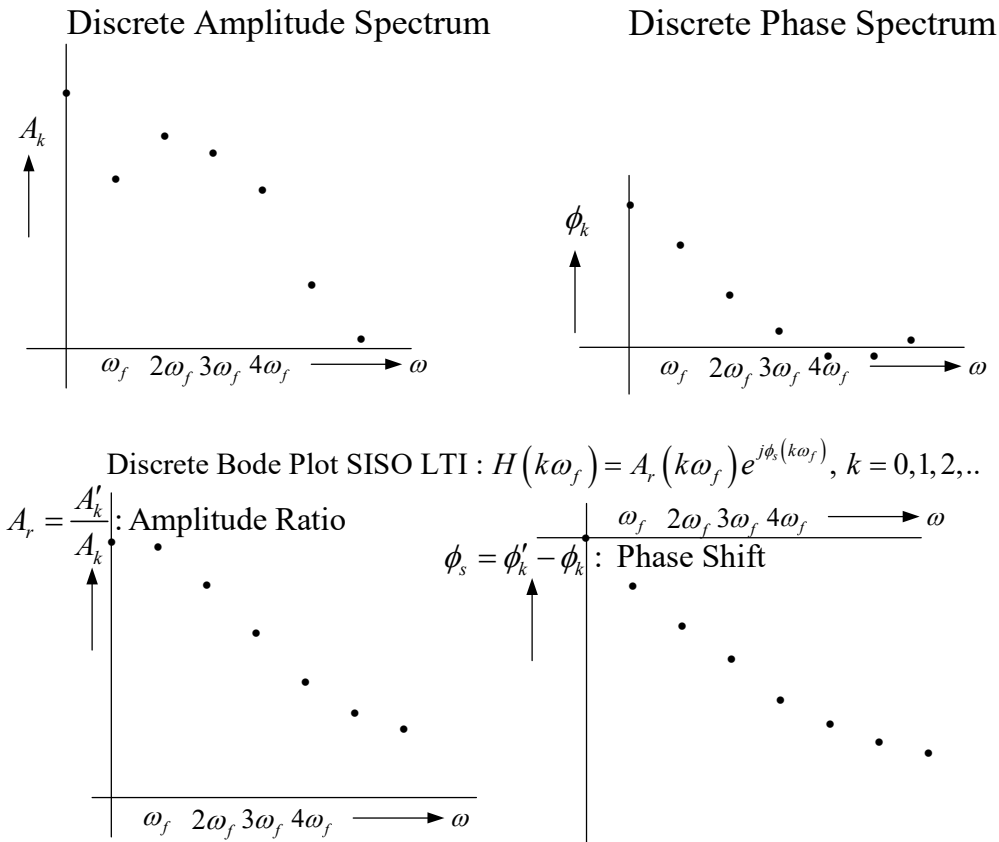


Fig. 10: Discrete Bode plot of a LTI SISO system. By interpolation one obtains the continuous Bode plot.

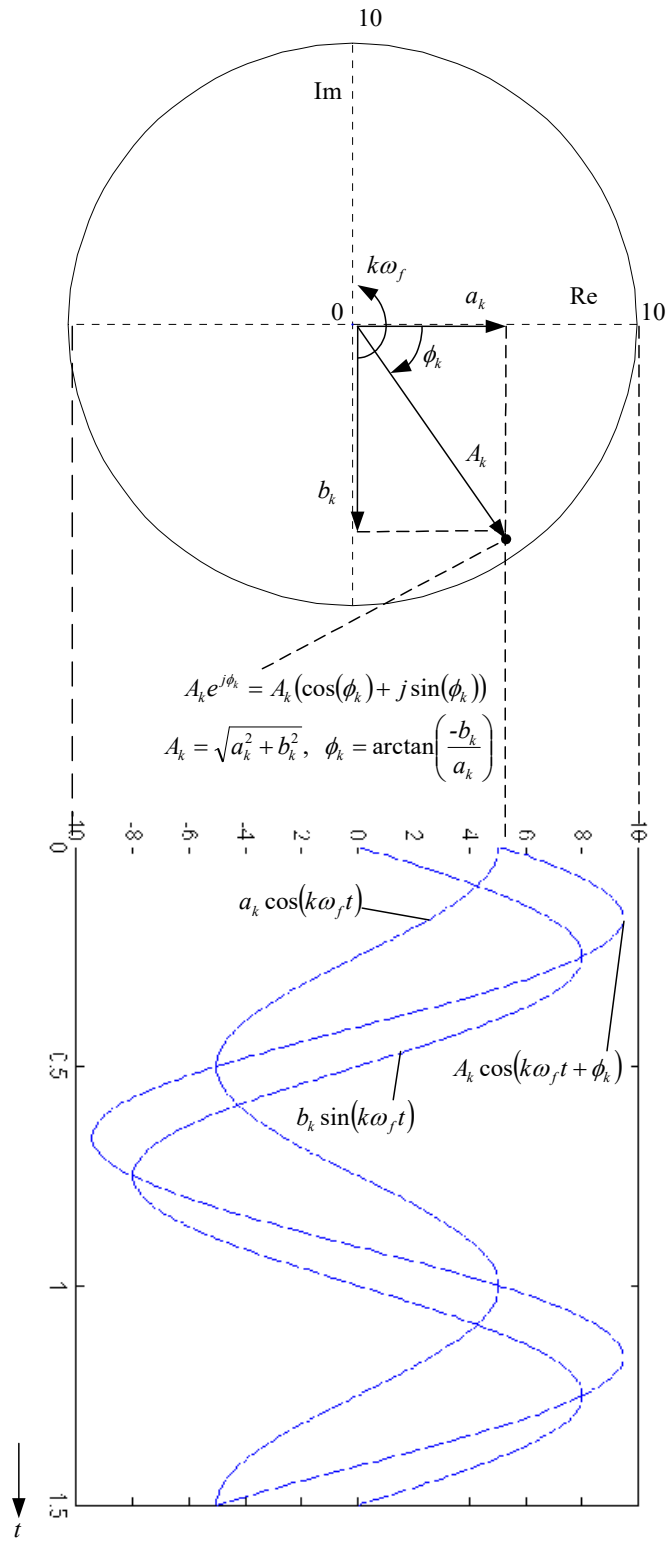


Fig. 11: Complex number $A_k e^{j\phi_k}$ representing frequency component $A_k \cos(k\omega_f + \phi_k)$

4 Controllability and observability of dynamic systems.

After reading this chapter one should:

- 1) Know the meaning of controllability and observability of dynamic systems.
- 2) Know the importance of these two properties for control system design.
- 3) Know how to detect these properties from state-space representations.
- 4) Know the controllability canonical and observability canonical representation of systems.
- 5) Know the properties stabilizability and detectability for LTI and LTV systems.
- 6) Know how to detect these two properties for LTI systems.

4.1 Partitioning and graphical representation of a dynamical system.

The three types of variables characterizing the state-space description

$$\frac{dx}{dt} = f(x, u, p), \quad y = g(x), \quad x \in \mathbb{R}^{n_x}, \quad u \in \mathbb{R}^{n_u}, \quad y \in \mathbb{R}^{n_y} \quad (4.1)$$

of dynamical systems are graphically represented in Fig. 12. The input variables $u_i \in \mathbb{R}$, $i = 1, 2, \dots, n_u$, contained in the input $u \in \mathbb{R}^{n_u}$ are *external* quantities that can often be manipulated and influence the state-variables x_i , $i = 1, 2, \dots, n_x$, contained in the state $x \in \mathbb{R}^{n_x}$ that *internally* represents the system. The output variables y_i , $i = 1, 2, \dots, n_y$, contained in the output $y \in \mathbb{R}^{n_y}$, are observable *external* quantities or quantities of interest that depend on the state-variables.

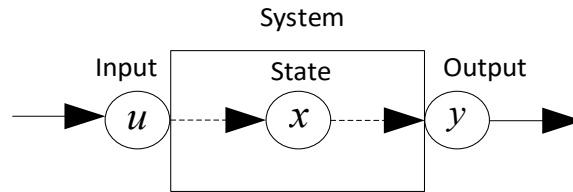


Fig. 12: A system having an input $u \in \mathbb{R}^{n_u}$, output $y \in \mathbb{R}^{n_y}$ and internal state $x \in \mathbb{R}^{n_x}$.

The so-called state-space is made up of the state $x \in \mathbb{R}^{n_x}$ that represents the system *internally*. The state-space naturally partitions into four parts, shown in Fig. 13. One part, denoted by A , is connected to the input but does not connect to the output. A second part, denoted by B , is connected to the input and does connect to the output. A third part, denoted by C , is not connected to the input and does connect to the output. The fourth part, denoted by D , is not connected to the input but does connect to the output. The arrows in Fig. 13 represent connections each part can have with the other parts. Some arrows may be absent as long as the

connection of each system part with the input and output, as described above, is maintained. One may for instance drop the arrow from B to A and/or the one from D to B . Of course, not all four parts A, B, C , and D have to be present in a system.

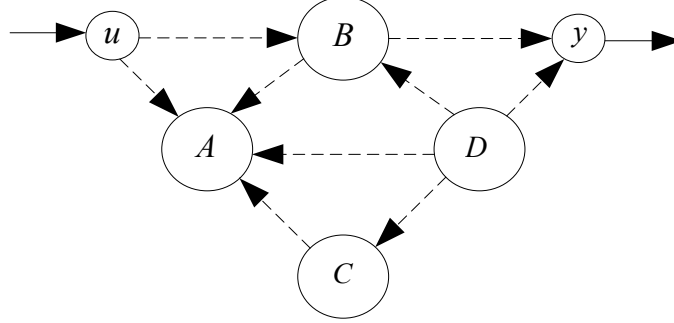


Fig. 13: Graphical representation and partitioning of a system with input $u \in \mathbb{R}^{n_u}$ and output $y \in \mathbb{R}^{n_y}$. The internal state-space, represented by the state $x \in \mathbb{R}^{n_x}$, is partitioned into four parts A, B, C , and D . The partitioning is based on whether or not system parts connect to the input and output. The partitioning seems to have been first recognized and presented by Kalman. It is therefore called the Kalman decomposition.

Exercise 4.1: Partitioning of an LTI system.

Consider the LTI system described by,

$$\frac{dx}{dt} = \begin{bmatrix} dx_1/dt \\ dx_2/dt \\ dx_3/dt \\ dx_4/dt \end{bmatrix} = Ax + Bu, \quad A = \begin{bmatrix} -1 & 1 & 2 & 3 \\ 0 & -2 & 0 & 4 \\ 0 & 0 & -3 & 5 \\ 0 & 0 & 0 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix}, \quad (4.2)$$

$$y = Cx, \quad C = \begin{bmatrix} 0 & 2 & 0 & 1 \end{bmatrix}$$

Verify that x_1 represents part A , x_2 part B , x_3 part C , and x_4 part D of the LTI system. Also verify that changing any of the zeros in the system matrices A , B , and C , into a non-zero number, destroys this result.

4.2 Controllability and observability as connections to the input and output.

From the graphical representation and partitioning in Fig. 13, it is immediately clear that the control of systems by means of the input variables is restricted to parts A and B , since only these parts are connected to the input. Together these are called the controllable part of the system. This naturally leads to a partitioning of the system into a controllable part, $A + B$, and an uncontrollable part, $C + D$, as shown in Fig. 14.

Similarly, the observation of systems through the output is restricted to parts B and D since only these parts connect to the output. Collectively, these are called the observable part of the system. This naturally leads to a partitioning of the system into an unobservable part, $A + C$, and an observable part, $B + D$, as shown in Fig. 15.

From Fig. 13, the only system part that can be controlled as well as observed is B . Therefore, only this part of the system lends itself to feedback control. The behavior of a system is fully described by the state x , and all four parts A, B, C , and D contribute to it. Given the former remark concerning feedback control, this immediately implies that, apart from part B , feedback controllers cannot control parts A, C , and D of a system!

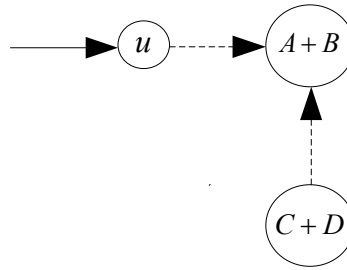


Fig. 14: Graphical representation and partitioning of a system with an input u . The state-space naturally partitions into the controllable part, $A + B$, and the uncontrollable part, $C + D$.

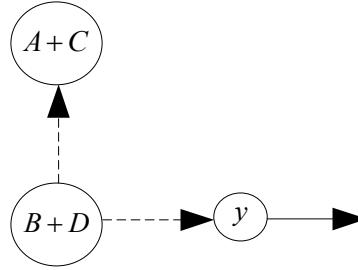


Fig. 15: Graphical representation and partitioning of a system with an output y . The state-space naturally partitions into an observable part, $B + D$, and an unobservable part, $A + C$.

4.3 Canonical state-space representations of dynamical systems.

Canonical state-space representations of systems have the special property that separate parts of the system like A , B , C , and D , as well as $A + B$, $C + D$, $A + C$ and $B + D$ in Fig. 13-Fig. 15 are represented by subsets of the state-variables x_i , $i = 1, 2, \dots, n_x$. One example is the system in Exercise 4.1 on page 38 that has each part A , B , C , and D represented by a single state-variable. In general, these parts are represented by scalar functions of all state-variables, referred to as modes. A state-transformation relates these modes to corresponding state-variables in a

canonical representation, see Fig. 16. A huge advantage of canonical state-space representations is that they reveal connections to the input and output in a straightforward manner.

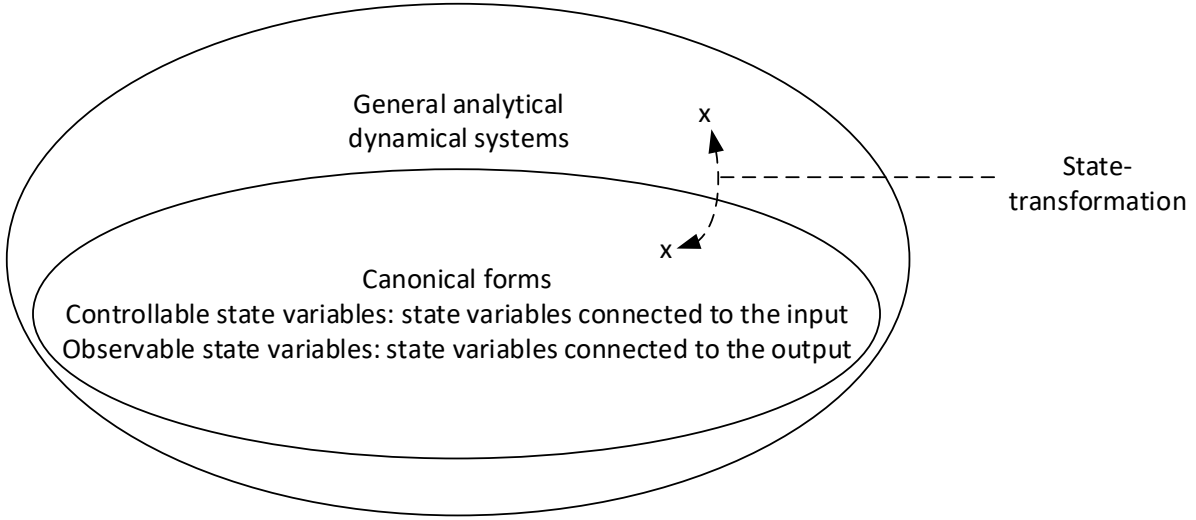


Fig. 16: Canonical forms facilitating simple definitions/explanations of controllability and observability as connectivity's to the input and output representing structural properties of dynamical systems. State-transformations connect general analytical dynamical systems to their canonical forms.

After a state-transformation, these connections become “hidden”, i.e. distributed over all state-variables. Unfortunately, state-space representations are not generally in canonical form. This raises the problem of either transforming them into canonical forms, through a state-transformation, or to detect controllable and uncontrollable as well as observable and unobservable modes directly. These will be the topic of section 4.5. First, we will further explore canonical representations.

4.4 Canonical controllability and observability forms

The canonical controllability form of systems has its state-variables partitioned into controllable state-variables, i.e. state-variables connected to the input, contained in the vector x^u , and uncontrollable state-variables, i.e. state-variables not connected to the input, contained in the vector $x^{\bar{u}}$. The state-variables in x^u represent parts $A + B$ of the system, depicted in Fig. 13 and Fig. 14, while the state-variables in $x^{\bar{u}}$ represent parts $C + D$. Given that x^u connects to the input

whereas $x^{\bar{u}}$ does not, the state equation $\frac{dx}{dt} = f(x, u, p)$ takes the following form

$$\frac{dx}{dt} = \begin{bmatrix} dx^u / dt \\ dx^{\bar{u}} / dt \end{bmatrix} = f(x, u) = \begin{bmatrix} f^u(x^u, x^{\bar{u}}, u) \\ f^{\bar{u}}(x^{\bar{u}}) \end{bmatrix}. \quad (4.3)$$

Equation (4.3) represents the state equation of a dynamical system in canonical controllability form. In a similar manner the canonical observability form has its state-variables partitioned into observable state-variables, i.e. state-variables connected to the output, contained in the vector x^y , and unobservable state-variables, i.e. state-variables not connected to the output, contained in the vector $x^{\bar{y}}$. The state-variables in x^y thus represent parts $B + D$ of the system, depicted in Fig. 13 and Fig. 15, while the state-variables in $x^{\bar{y}}$ represent parts $A + C$. Since x^y connects to the output, whereas $x^{\bar{y}}$ does not, the state-space representation (4.1) takes the form

$$\frac{dx}{dt} = \begin{bmatrix} dx^y / dt \\ dx^{\bar{y}} / dt \end{bmatrix} = f(x, u) = \begin{bmatrix} f^y(x^y, u) \\ f^{\bar{y}}(x^y, x^{\bar{y}}, u) \end{bmatrix}, \quad y = g(x) = g(x^y). \quad (4.4)$$

Equation (4.4) represents a state-space description of a dynamical system in canonical observability form. This form concerns both the state equation and output equation (4.1). If the system is linear $f, f^u, f^{\bar{u}}$ are linear homogeneous functions of x and u , while g is a linear homogeneous function of x . Then the canonical controllability form in (4.3) specializes to

$$\frac{dx}{dt} = \begin{bmatrix} dx^u / dt \\ dx^{\bar{u}} / dt \end{bmatrix} = Ax + Bu = \begin{bmatrix} A^{uu} & A^{u\bar{u}} \\ \theta^{\bar{u}u} & A^{\bar{u}\bar{u}} \end{bmatrix} \begin{bmatrix} x^u \\ x^{\bar{u}} \end{bmatrix} + \begin{bmatrix} B^u \\ \theta^{\bar{u}} \end{bmatrix} u, \quad (4.5)$$

where $A^{u\bar{u}}$ represents the influence of $x^{\bar{u}}$ on x^u , and similarly for the other matrices, while θ represents a zero matrix. All matrices have appropriate dimensions and may be explicit functions of time t . If the system is linear, $g(y)$ is also a linear and homogeneous function and the canonical observability form in (4.4) specializes to

$$\frac{dx}{dt} = \begin{bmatrix} dx^y / dt \\ dx^{\bar{y}} / dt \end{bmatrix} = Ax + Bu = \begin{bmatrix} A^{yy} & \theta^{y\bar{y}} \\ A^{\bar{y}y} & A^{\bar{y}\bar{y}} \end{bmatrix} \begin{bmatrix} x^y \\ x^{\bar{y}} \end{bmatrix} + \begin{bmatrix} B^y \\ B^{\bar{y}} \end{bmatrix} u, \quad y = Cx = \begin{bmatrix} C^y & \theta^{\bar{y}} \end{bmatrix} \begin{bmatrix} x^y \\ x^{\bar{y}} \end{bmatrix}. \quad (4.6)$$

Again, the matrices in (4.6) have appropriate dimensions and may explicitly depend on time t .

Exercise 4.2: Example of an LTI system represented in the canonical controllability form.

Consider the system in Exercise 4.1 on page 38. Establish that this system can be represented in

the canonical form (4.5) with $x^u = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $x^{\bar{u}} = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$, $A^{uu} = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix}$, $A^{u\bar{u}} = \begin{bmatrix} 2 & 3 \\ 0 & 4 \end{bmatrix}$,

$$A^{\bar{u}\bar{u}} = \begin{bmatrix} -3 & 5 \\ 0 & -4 \end{bmatrix}, \quad B^u = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Exercise 4.3: Example of an LTI system represented in the canonical observability form.

Consider once more the system in Exercise 4.1 on page 38. Establish that this system can be

represented in the canonical observability form (4.6) with $x^y = \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}$, $x^{\bar{y}} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$, $A^{yy} = \begin{bmatrix} -2 & 4 \\ 0 & -4 \end{bmatrix}$,
 $A^{\bar{y}y} = \begin{bmatrix} 1 & 3 \\ 0 & 5 \end{bmatrix}$, $A^{\bar{y}\bar{y}} = \begin{bmatrix} -1 & 2 \\ 0 & -3 \end{bmatrix}$, $C^y = [2 \ 1]$.

The partitioning in Fig. 13 on page 38 into four system parts immediately suggests a canonical controllability observability form or Kalman canonical form, where each system part is also represented by corresponding subsets of state-variables

$$\frac{dx}{dt} = \begin{bmatrix} dx^{u\bar{y}} / dt \\ dx^{u^y} / dt \\ dx^{\bar{u}\bar{y}} / dt \\ dx^{\bar{u}^y} / dt \end{bmatrix} = f(x, u) = \begin{bmatrix} f^{u\bar{y}}(x^{u\bar{y}}, x^{u^y}, x^{\bar{u}\bar{y}}, x^{\bar{u}^y}, u) \\ f^{u^y}(x^{u^y}, x^{\bar{u}^y}, u) \\ f^{\bar{u}\bar{y}}(x^{\bar{u}\bar{y}}, x^{\bar{u}^y}) \\ f^{\bar{u}^y}(x^{\bar{u}\bar{y}}) \end{bmatrix}, \quad y = g(x) = g(x^{u^y}, x^{\bar{u}^y}). \quad (4.7)$$

In equation (4.7), $x^{u\bar{y}}$ contains state-variables that are controllable but unobservable, representing part A , since $x^{u\bar{y}}$ and part A connect to the input but not to the output, as seen from equation (4.7) and Fig. 13. Similarly, x^{u^y} contains state-variables that are controllable and observable, representing part B . Furthermore, $x^{\bar{u}\bar{y}}$ describes the uncontrollable and unobservable part, C and $x^{\bar{u}^y}$ the uncontrollable observable part, D . If the system is linear, equation (4.7) becomes,

$$\frac{dx}{dt} = \begin{bmatrix} dx^{u\bar{y}} / dt \\ dx^{u^y} / dt \\ dx^{\bar{u}\bar{y}} / dt \\ dx^{\bar{u}^y} / dt \end{bmatrix} = f(x, u) = Ax + Bu = \begin{bmatrix} A^{AA} & A^{AB} & A^{AC} & A^{AD} \\ \theta^{BA} & A^{BB} & \theta^{BC} & A^{BD} \\ \theta^{CA} & \theta^{CB} & A^{CC} & A^{CD} \\ \theta^{DA} & \theta^{DB} & \theta^{DC} & A^{DD} \end{bmatrix} \begin{bmatrix} x^{u\bar{y}} \\ x^{u^y} \\ x^{\bar{u}\bar{y}} \\ x^{\bar{u}^y} \end{bmatrix} + \begin{bmatrix} B^A \\ B^B \\ \theta^C \\ \theta^D \end{bmatrix} u, \quad (4.8)$$

$$y = g(x) = Cx = \begin{bmatrix} \theta^A & C^B & \theta^C & C^D \end{bmatrix} \begin{bmatrix} x^{u\bar{y}} \\ x^{u^y} \\ x^{\bar{u}\bar{y}} \\ x^{\bar{u}^y} \end{bmatrix}$$

The number of state-variables within $x^{u\bar{y}}$, x^{u^y} , $x^{\bar{u}\bar{y}}$ and $x^{\bar{u}^y}$ are the *dimensions* of the corresponding system parts A, B, C , and D . They are denoted by n_A, n_B, n_C , and n_D ,

respectively. Together n_A, n_B, n_C , and n_D determine the *system structure*. In most cases, one is content with knowing just $n_A + n_B$, the number of controllable state-variables in the canonical forms (4.3) and (4.5), and $n_B + n_D$, the number of observable state-variables in the canonical forms (4.4) and (4.6).

Exercise 4.4: An LTI system represented in the Kalman canonical form.

Consider the system in Exercise 4.1 on page 38. Establish that it is represented in the Kalman canonical form and determine $n_A, n_B, n_C, n_D, n_A + n_B, n_C + n_D$.

Remark 1

The mathematical canonical state-space representations (4.3)-(4.8) are *special* and *simple* because each system part is precisely represented by a subset of state-variables that do or do not connect to the input and output, instead of being described by scalar functions of all state-variables, called modes, obtained after a state-basis transformation. As a result, any failure of a state-variable to connect to the input or output, can be directly observed from missing dependencies in equations (4.3), (4.4) and (4.7), which are represented by zero parts in equations (4.5), (4.6) and (4.8).

Remark 2

To *construct* systems having certain controllability/observability properties one can select a corresponding canonical form and *choose* the (nonzero) system parts arbitrarily. This *generically* realizes the corresponding controllability/observability properties, since there is a small chance that an arbitrary choice of system parts causes additional uncontrollable/unobservable modes. Having realized the appropriate controllability/observability properties this way, we may subsequently “hide” them by performing a state-transformation.

Remark 3

Following Remark 2, all canonical forms have the property that the (nonzero) system parts do not cause additional uncontrollable/unobservable modes.

4.5 Establishing controllability and observability of dynamical systems represented in state-space form.

From the results presented in the previous sections, we learned that the sensitivity of uncontrollable state variables or modes to the input is zero. Similarly, the sensitivity of the output to unobservable state variables or modes is zero. An algorithm that computes such sensitivities, and is especially suitable for large-scale systems, for which other algorithms tend to fail, has been developed at Biometris. It can be used to check controllability and observability of any dynamical system represented in state-space form. Moreover, it also finds the state-variables that make up each uncontrollable or unobservable mode, if any. Given an arbitrary initial state and input, by

means of numerical integration, the algorithm computes the state-response/trajectory of the system during a very short time and also a sensitivity matrix that captures sensitivities along this state-response/trajectory. A singular value decomposition (SVD) applied to this sensitivity matrix determines controllability/observability as well as the state-variables making up possible uncontrollable/unobservable modes. An SVD of an $n \times m$ sensitivity matrix M is

$$\underbrace{M}_{n \times m} = \underbrace{U}_{n \times n} \underbrace{S}_{n \times m} \underbrace{V^T}_{m \times m} \quad (4.9)$$

Matrix S is diagonal with the $\min(n, m)$ singular values of matrix M on the diagonal. Matrix U contains as columns the left singular vectors of matrix M , each one corresponding to a singular value. Matrix V contains as columns the right singular vectors of matrix M , each one corresponding to a singular value. If M is a sensitivity matrix then zero singular values indicate uncontrollability/unobservability. The corresponding singular vectors corresponding to the smallest dimension of M , are the uncontrollable/unobservable modes whereas the remaining ones are the controllable/observable modes.

The rank of a matrix is the number of non-zero singular values. For an $n \times m$ matrix, if any of the $\min(n, m)$ singular values is zero the matrix is not full-rank (rank deficient) so having a rank less than $\min(n, m)$. Only if the $n \times m$ sensitivity matrix M is full-rank, controllability/observability holds.

If the system is linear and time-varying (LTV), alternative algorithms rely on the following condition obtained from the state-response (2.8) of LTV systems,

$$\text{rank}(W(t, t_0)) = n_x, \quad W(t, t_0) = \int_{t_0}^t \Phi(\tau, t_0) B(\tau) B^T(\tau) \Phi^T(\tau, t_0) d\tau. \quad (4.10)$$

Condition (4.10) guarantees that in the state-response (2.8), the state $x(t)$ can reach any value by selecting $u(\tau)$, $t_0 \leq \tau \leq t$ appropriately. The latter is another way to define controllability, instead of having no state-variables or modes that are disconnected from the input. At any time $t \geq t_0$, the so called controllability gramian $W(t, t_0)$ in (4.10) can be computed using numerical integration. In the same way as described below equation (4.9), an SVD of $W(t, t_0)$ determines controllability (when full-rank) as well as uncontrollable modes, in case controllability does not hold.

Similarly, the following condition, obtained from the output response (2.17) of LTV systems, applies to observability

$$\text{rank}(M(t, t_0)) = n_x, \quad M(t, t_0) = \int_{t_0}^t \Phi^T(\tau, t_0) C^T(\tau) C(\tau) \Phi(\tau, t_0) d\tau. \quad (4.11)$$

This condition guarantees that the output is affected by any change of state, which is another way to define observability. Matrix $M(t, t_0)$ is called the observability gramian. It can also be computed using numerical integration.

If the system is of the LTI type (3.1), (3.2), instead of using the controllability gramian, the following condition also guarantees controllability

$$\text{rank}(\mathcal{R}) = n_x, \quad \mathcal{R} = [B, AB, A^2B, \dots, A^{n_x-1}B] \in \mathbb{R}^{n_x \times n_x n_u} \quad (4.12)$$

This condition, which is mostly used, is also based on the state-response (2.8), that specializes to (2.18) and relies on a Taylor series expansion found on page 55 of Kwakernaak & Sivan. The matrix $\mathcal{R}$ in (4.12) is called the controllability matrix (computed by the Matlab function ctrb). An SVD of this matrix provides exactly the same information concerning controllability and controllable/uncontrollable modes as an SVD of the sensitivity matrix described above.

Similarly, the following condition applies to observability of LTI systems. It is based on a Taylor series expansion found on page 65 of Kwakernaak & Sivan, of the output-response (2.19) on page 23 of these lecture notes.

$$\text{rank}(\mathcal{O}) = n_x, \quad \mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \in \mathbb{R}^{n_y n_x \times n_x} \quad (4.13)$$

The matrix $\mathcal{O}$ is called the observability matrix (computed by the Matlab function obsv). An SVD of this matrix provides exactly the same information concerning observability and observable/unobservable modes as the SVD's mentioned earlier.

Controllability of LTV systems is fully determined by the system matrices $A(t), B(t)$ whereas observability is fully determined by the system matrices $A(t), C(t)$. They are dual properties meaning that observability of $A(t), C(t)$ is equivalent to controllability of $A^T(t), C^T(t)$.

One can observe this duality from the controllability and observability gramian (4.10), (4.11). The same duality also applies to LTI systems, as can be seen from the controllability and observability matrices (4.12), (4.13). Therefore, the same algorithm to compute controllability of an LTV/LTI system can thus be used to compute observability of an LTV/LTI system.

4.6 Stabilizability and detectability of LTI systems

Consider the canonical controllability form

$$\frac{dx}{dt} = \begin{bmatrix} dx^u / dt \\ dx^{\bar{u}} / dt \end{bmatrix} = Ax + Bu = \begin{bmatrix} A^{uu} & A^{u\bar{u}} \\ \theta^{\bar{u}u} & A^{\bar{u}\bar{u}} \end{bmatrix} \begin{bmatrix} x^u \\ x^{\bar{u}} \end{bmatrix} + \begin{bmatrix} B^u \\ \theta^{\bar{u}} \end{bmatrix} u, \quad (4.14)$$

Observe that $x^{\bar{u}}$ is the state of an autonomous LTI system, not influenced by the input and fully represented by the square matrix $A^{\bar{u}\bar{u}}$. It represents the uncontrollable part of the system. If this system part is uniformly stable, which is equivalent to all eigenvalues of $A^{\bar{u}\bar{u}}$ having negative real parts, for any $x^{\bar{u}}(0)$, $x^{\bar{u}}(t) \rightarrow 0$ as $t \rightarrow \infty$. The system is called stabilizable in this case because,

given controllability of the state x^u , this implies that for any $x(0) = \begin{bmatrix} x^u(0) \\ x^{\bar{u}}(0) \end{bmatrix}$ a control exists such

that $x(t) \rightarrow 0$ as $t \rightarrow \infty$. If some eigenvalue of $A^{\bar{u}\bar{u}}$ does not have a negative real part the LTI system is not stabilizable. If an LTI system is not represented in the canonical controllability form, one can always compute a state-transformation into this form, and then determine stabilizability in the way just described. This is the way algorithms that determine stabilizability of LTI systems work (function `stblzb` in Matlab).

Exercise 4.5

Given the canonical observability form

$$\frac{dx}{dt} = \begin{bmatrix} dx^y / dt \\ dx^{\bar{y}} / dt \end{bmatrix} = Ax + Bu = \begin{bmatrix} A^{yy} & \theta^{y\bar{y}} \\ A^{\bar{y}y} & A^{\bar{y}\bar{y}} \end{bmatrix} \begin{bmatrix} x^y \\ x^{\bar{y}} \end{bmatrix} + \begin{bmatrix} B^y \\ B^{\bar{y}} \end{bmatrix} u, \quad y = Cx = \begin{bmatrix} C^y & \theta^{\bar{y}} \end{bmatrix} \begin{bmatrix} x^y \\ x^{\bar{y}} \end{bmatrix}. \quad (4.15)$$

for $u(t)=0$, $x^y(0)=0$, determine the conditions under which $x^{\bar{y}}(t) \rightarrow 0$ as $t \rightarrow \infty$ for every $x^{\bar{y}}(0)$.

Under these conditions the LTI system (4.15) is called detectable because the part $x^{\bar{y}}$ of the state that cannot be observed in this case, tends to zero ultimately. If the LTI system is not represented in the observability canonical form, one can always compute a state-transformation into this form and then check detectability based on the condition found from Exercise 4.5. Detectability is fully determined by the LTI system matrices A, C in (4.15) while stabilizability is fully determined by the LTI system matrices A, B in (4.14). Like controllability and observability, they are dual properties, implying that detectability of A, C is equivalent to stabilizability of A^T, C^T . Therefore one can also compute detectability using the Matlab function `stblzb`.

4.7 Stabilizability and detectability of LTV systems.

Like stability, stabilizability is a much more complicated property for LTV systems. One reason is that stability and stabilizability are concerned with behavior as time tends to infinity. For time-varying systems this requires a specification of an LTV system over an infinitely long time-interval. Since the differential equations are time-varying such a description is often problematic. For the optimal controller design part of this course (Part 2, chapters 7-12) we need the design of controllers for LTV systems. Fortunately, the design has to be performed only over a finite-time interval. Therefore, the design does not require the LTV system to be controllable, stabilizable, observable or detectable. On the other hand, controllability and observability are desirable properties for the LTV controller design, as further explained in Part 2 of this course.

5 Pole placement: a state-feedback controller design method for LTI systems.

After reading this chapter the student should:

- 1) Know pole placement as a controller design method fixing the eigenvalues/poles of the closed loop system by means of state-feedback.
- 2) Know how to compute and apply the state-feedback gain to arrive at the closed loop system with the specified eigenvalues/poles.
- 3) Know the conditions under which pole placement realizes arbitrary eigenvalues/poles.
- 4) Know the effect of prescribed eigenvalues on closed-loop system performance.

Recall the state-equation of an LTI system

$$\frac{dx}{dt} = Ax + Bu . \quad (5.1)$$

Suppose all the state-variables are measured. Then the output equation is

$$y = x \Leftrightarrow y = Cx, \quad C = I . \quad (5.2)$$

This is the most favourable situation for control system design called “having full state-information”. Then we may use the measurements $y = x$ to generate a control

$$u = -Ly = -Lx , \quad (5.3)$$

Equation (5.3) feeds back the output=state to the input. Therefore $L \in \mathbb{R}^{n_u \times n_x}$ is called a state-feedback matrix. Substituting equation (5.3) into equation (5.1) we obtain

$$\frac{dx}{dt} = \left(\underbrace{A}_{n_x \times n_x} - \underbrace{B}_{n_x \times n_u} \underbrace{L}_{n_u \times n_x} \right) x \quad (5.4)$$

which is an autonomous system. The stability of this autonomous system is determined by the eigenvalues of matrix $A - BL$. If the LTI system (5.1), or equivalently the matrix pair (A, B) , is controllable, by choosing the feedback matrix L appropriately, we can place the eigenvalues/poles of matrix $A - BL$ at arbitrary locations. This is called pole placement. Since the eigenvalues determine the stability properties of the system (5.4), these stability properties can be manipulated at will. Algorithms have been developed that compute the state-feedback matrix L from the controllable matrix pair (A, B) and specified locations of the eigenvalues/poles (place in Matlab). Of course, if complex eigenvalues are specified causing oscillatory behavior, they must be specified as complex conjugate pairs. Generally, one prefers exponential non-oscillatory decay of state variables. This requires specifying real eigenvalues that are negative.

The ability to place the eigenvalues at specified locations is due to the following. If the matrix pair (A, B) , or equivalently the system (5.1), is controllable, then using L one can realize an arbitrary characteristic polynomial

$$\det(A - BL - \lambda I) = \lambda^{n_x} + r_{n_x-1} \lambda^{n_x-1} + r_{n_x-2} \lambda^{n_x-2} + \dots + r_1 \lambda + r_0 \quad (5.5)$$

with given coefficients $r_{n_x-1}, r_{n_x-2}, \dots, r_1, r_0$. These coefficients uniquely determine the eigenvalues of matrix $A - BL$ being the zeros of the characteristic polynomial (5.5).

Exercise 5.1: Finding a feedback matrix L realizing specified eigenvalues of $A - BL$.

Reconsider the state-equation of the mass-spring-damper system with the mass, damping and spring constant all equal to 1:

$$\frac{dx}{dt} = Ax + Bu, \quad A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (5.6)$$

- Verify that the eigenvalues of matrix A are $-\frac{1}{2} \pm \frac{1}{2}i\sqrt{3}$.
- Is the system asymptotically stable? Explain.
- Is the state-response oscillatory if there is no input? Explain.

Assume all state variables, i.e. both the displacement and velocity of the mass are measured.

- d) Write a mathematical expression for the characteristic polynomial $\det(A - BL)$ in the components of the state-feedback matrix L . Use it to find the state-feedback matrix L that realizes poles/eigenvalues being $-1, -2$.
- e) Describe how this state-feedback improves the system behavior.

A system with feedback from the output to the input is called a closed loop system. Without this type of feedback the system is called open-loop.

Exercise 5.2: Pole placement for systems having multiple inputs.

In Exercise 5.1, the matrix L was uniquely determined by the prescribed poles. Explain why this is no longer the case if the system has multiple inputs. The Matlab function `place` then uses additional rules to arrive at a proper feedback matrix.

6 Static and dynamic output-feedback of LTI systems.

After reading this section one should:

- 1) Know the difference between static and dynamic output feedback.
- 2) Know the advantages and disadvantages of static versus dynamic output feedback.
- 3) Know what an observer/state estimator does.
- 4) Know how to design an observer/state-estimator using pole placement.
- 5) Know how dynamic output feedback determines eigenvalues of the to be controlled system as well those of the state-estimator errors.

6.1 Static versus dynamic output-feedback.

For LTI systems, state-feedback allowed us to place the eigenvalues of the closed loop system at arbitrary locations if the open loop system, that is the system without feedback, is controllable. State-feedback is only possible if all the state variables can be measured (called “having full state-information”) which is the most favourable situation for control system design. In most practical applications, one cannot measure all state-variables, or, measuring all of them, is too complicated and/or expensive. This chapter considers feedback mechanisms that use the available measurements, described by the output equation.

Reconsider once more the mass-spring-damper system with the mass, damping and spring constant all equal to 1 of which only the first state-variable, i.e. the displacement of the mass, is measured:

$$\frac{dx}{dt} = Ax + Bu, \quad y = Cx, \quad A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \quad (6.1)$$

Similar to the state-feedback $u = -Lx$, $L \in \mathbb{R}^{n_u \times n_x}$ we may apply the output-feedback $u = -Ly$, $L \in \mathbb{R}^{n_u \times n_y}$. Since now there is only one output, matrix $L \in \mathbb{R}^{n_u \times n_y}$ reduces to a scalar so with only one component to choose. This no longer allows us to place the two poles of the closed-loop system at arbitrary locations. In the special case that we have n_x outputs, output matrix C of the LTI system becomes square. If this matrix is also invertible we can recover the state from the output as follows

$$x = C^{-1}y \quad (6.2)$$

In this special case we can use the output-feedback $u = -L'y$ to realize state-feedback as follows

$$u = -Lx = -LC^{-1}y = -L'y, \quad L' \in \mathbb{R}^{n_u \times n_y}, \quad n_y = n_x \quad (6.3)$$

Obviously, in most cases the output matrix C of the LTI system is not square and invertible. Given the advantages of state-feedback and the fact that “having full-state information” is most desirable for control system design, the following question presents itself: “Is there a way to recover the state x if matrix C is not square and invertible”? The answer is “yes” and the methodology to recover the state is by means of an observer/state-estimator. This observer will turn out to be a dynamic system itself. Therefore output-feedback realized by an observer is called dynamic output feedback as opposed to static output-feedback as described by equation (6.3).

6.2 Observer design for LTI systems.

To reconstruct the state x from measurements y of an LTI system, an observer contains a copy of the LTI system. The state of the observer is denoted by $\hat{x}$ to distinguish it from the state x of the LTI system and to also indicate that $\hat{x}$ is an estimate of the state x , that in general contains an error $e = \hat{x} - x$. The observer, apart from having a copy of the LTI system, must use the measurements y to obtain estimates $\hat{x}$ of x that keep the error $e = \hat{x} - x$ small. Below we state the state-space representation of an observer for an LTI system

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + K \left(y - \underbrace{C\hat{x}}_{\hat{y}} \right). \quad (6.4)$$

The first two terms $A\hat{x} + Bu$ on the right represent a copy of the state-equation of the LTI system. The third term $K(y - \hat{y})$ uses the difference between the measurement y and the estimated measurement $\hat{y} = C\hat{x}$ to correct the state-estimate $\hat{x}$ to keep the error $e = \hat{x} - x$ small. Matrix $K \in \mathbb{R}^{n_x \times n_y}$ is the observer gain matrix. The difference $y - \hat{y}$ is called the innovation that is

multiplied by the observer gain matrix K to generate corrections, intended to keep the error $e = \hat{x} - x$ small. The observer gain matrix K is the major part of the observer design, since the matrices A, B and C of the LTI system are fixed in principle. The observer gain matrix K determines how the errors $e = \hat{x} - x$ propagate. Note that the observer (6.4) has two inputs: the input u applied to the LTI system as well as the output y coming out of the LTI system of which the observer estimates the state. Having the state estimate $\hat{x}$ obtained from the observer, enables us to apply the state-feedback

$$u = -L\hat{x}, \quad L \in \mathbb{R}^{n_u \times n_x}. \quad (6.5)$$

The interconnection of the LTI system and the observer combined with the state-feedback (6.5), is represented in Fig. 17.

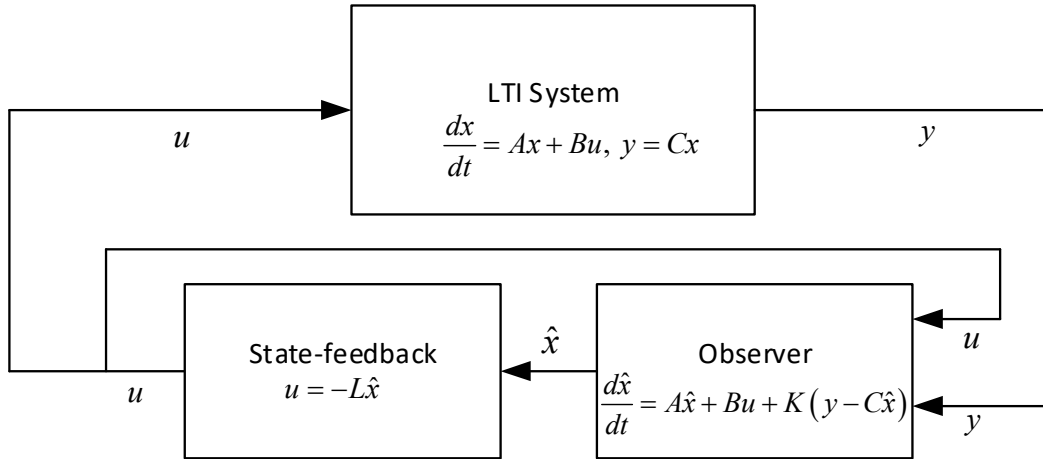


Fig. 17: Closed loop dynamic output-feedback control of an LTI system by means of an observer and state-feedback.

This closed-loop system is one with dynamic output-feedback because the feedback contains a dynamic system, namely the observer, that uses the output y of the LTI system as an input to realize feedback. The closed loop system performance and stability is influenced by the choice of the observer gain matrix K with dimensions $n_x \times n_y$ and the state-feedback matrix L with dimensions $n_u \times n_x$.

As explained in the previous chapter, the state-feedback gain matrix L can be selected based on pole placement, if the LTI system is controllable. Pole placement can also be used to choose the observer gain matrix K . To see this, we have to consider the propagation of the state-estimate error $e = \hat{x} - x$. To obtain it, we start from the state-space representation of the closed loop system that has as state-variables the ones of the LTI system, x , as well as the ones of the observer, $\hat{x}$, see Fig. 17. This state-representation is

$$\begin{bmatrix} dx/dt \\ d\hat{x}/dt \end{bmatrix} = \begin{bmatrix} A & -BL \\ KC & A-BL-KC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}. \quad (6.6)$$

To obtain the propagation of the error $e = \hat{x} - x$, use $de/dt = d(\hat{x} - x)/dt = d\hat{x}/dt - dx/dt$ to obtain the following state-space model describing the propagation of x and e

$$\begin{aligned} \begin{bmatrix} dx/dt \\ de/dt \end{bmatrix} &= \begin{bmatrix} Ax + Bu \\ (A-KC)e \end{bmatrix} = \begin{bmatrix} Ax - BL\hat{x} \\ (A-KC)e \end{bmatrix} \\ &= \begin{bmatrix} Ax - BL\hat{x} + BLx - BLx \\ (A-KC)e \end{bmatrix} = \begin{bmatrix} (A-BL)x + BL e \\ (A-KC)e \end{bmatrix} \\ &= \begin{bmatrix} A-BL & BL \\ 0 & A-KC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}. \end{aligned} \quad (6.7)$$

Since the square block matrix on the last line of equation (6.7) is upper triangular, the eigenvalues of this matrix are the eigenvalues of the two block matrices $A-BL$ and $A-KC$ on the diagonal. Recall that if (A, B) is controllable, the eigenvalues of $A-BL$ can be placed arbitrary by choosing L . Dually, if (A, C) is observable, which is equivalent to (A^T, C^T) being controllable, the eigenvalues of $A^T - C^T K^T = (A-KC)^T$ can be placed arbitrarily by choosing K^T . Since $(A-KC)^T$ and $A-KC$ have the same eigenvalues, pole placement allows us to fix all the poles of the closed-loop system (6.7) that describes the propagation of the state x of the LTI system and the state-estimation error $e = \hat{x} - x$. Moreover, the state-feedback gain matrix L and the observer gain matrix K can be selected independent of one another. This is called the separation property of this type of control system design. From equation (6.7) observe that L fixes the poles associated with the state x of the LTI system, whereas K fixes the poles of the error-propagation $e = \hat{x} - x$. Also, the error e acts as an input to the state x of the LTI system.

6.3 A prelude to static and dynamic output-feedback for LTV systems.

In Part 2 of the systems and control theory course we will be very much concerned with the optimal control and feedback design of general nonlinear systems, as represented by Fig. 7 on page 14. For general nonlinear dynamic systems represented in state-space form

$$\frac{dx}{dt} = f(x, u, p), \quad (6.8)$$

$$y = g(x, p), \quad (6.9)$$

you will learn how to obtain so called optimal responses/trajectories

$$u^*(t), x^*(t), y^*(t), t_0 \leq t \leq t_f, \quad (6.10)$$

shown in the bottom part of Fig. 7. The feedback part in the middle of Fig. 7, represents either state-perturbation feedback, when “full-state information” is available, or output-perturbation feedback if “full-state information” is not available. Both types of feedback are designed using a linearized model around the optimal trajectory (6.10).

If this trajectory is a steady-state, the linearization is an LTI system. In that case we may apply the state or output feedback design presented in the previous section, based on pole placement. Keep in mind that the state, input and output of this design represent the perturbations/corrections Δx , Δy and Δu in Fig. 7. Alternatively, instead of pole placement, we will use optimal control theory to obtain both the perturbation state-feedback gain L as well as the observer gain K in Part 2 of this course.

If the optimal trajectory (6.10) is not a steady-state, then the linearized model is an LTV system. Roughly speaking, for LTV systems, the state and output feedback design is similar to the one of LTI systems. However, system matrices A, B, C as well as the state-feedback and observer gains L, K become functions of time. Pole placement techniques no longer apply in this case. But optimal control theory does provide $L(t)$, $K(t)$ as will also become clear in Part 2 of this course.