Introduction to Linear Time-Invariant Systems

Control Engineering (Reguleringsteknik)

Christoffer Sloth

`chsl@mmmi.sdu.dk`

SDU Robotics
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

Agenda

Matematiske og grafiske metoder til syntese af lineære tidsinvariante systemer:[1]

- diskret og kontinuert tilstandsbeskrivelse
- analyse i tid og frekvens
- stabilitet, reguleringshastighed, følsomhed og fejl
- digitale PI, PID, LEAD og LAG regulatorer (serieregulatorer)
- tilstandsregulering, pole-placement og tilstands-estimering (observer)
- optimal regulering (least squares) og optimal tilstands-estimation (Kalman-filter)

Færdigheder:
Efter gennemførelse af kurset kan den succesfulde studerende:

- kunne analysere, dimensionere og implementere såvel kontinuert som tidsdiskret regulering af lineære tidsinvariante og stokastiske systemer

Kompetencer:
Efter gennemførelse af kurset kan den succesfulde studerende:

- anvende og implementere klassiske og moderne reguleringsteknikker for at kunne styre og regulere en robot hurtig og præcist

[1] Based on https://odin.sdu.dk/sitecore/index.php?a=fagbesk&id=76852&listid=10980

The twelve lectures of the course are

- **Lecture 1**: Introduction to Linear Time-Invariant Systems
- **Lecture 2**: Stability and Performance Analysis
- **Lecture 3**: Introduction to Control
- **Lecture 4**: Design of PID Controllers
- **Lecture 5**: Root Locus
- **Lecture 6**: The Nyquist Plot
- **Lecture 7**: Dynamic Compensators and Stability Margins
- **Lecture 8**: Implementation
- **Lecture 9**: State Feedback
- **Lecture 10**: Observer Design
- **Lecture 11**: Optimal Control (Linear Quadratic Control)
- **Lecture 12**: The Kalman Filter

The main prerequisites for this course are:

- **Modeling of Dynamical Systems**
  You must be able to setup models of electro-mechanical systems based on fundamental physical laws.
- **Differential Equations**
  You must be able to simulate and interpret solutions to ordinary differential equations.
- **Laplace Transformation**
  You must be able to apply Laplace transformation to obtain transfer functions, and analyze their behavior in frequency domain.
- **$\mathcal{Z}$-Transformation**
  You must be able to apply the z-transformation to obtain discrete transfer functions, and analyze their behavior in frequency domain.
- **Linear Algebra**
  You must be familiar with basic concepts from linear algebra such as eigenvalues, rank, and determinants.

Introduction

Overview of Today's Lecture

$$z^{(n)} - a_{n-1}z^{(n-1)} - \cdots - a_1\dot{z} - a_0 z = b_1 u_1(t) + \cdots + b_m u_m(t)$$

nth-order differential equation

Reformulation

Laplace transform

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

State space model

Laplace transform

Inverse Laplace transform

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

Transfer function

Discretization

Discretization

$$\begin{cases} x_{k+1} = \Phi x_k + \Gamma u_k \\ y_k = Cx_k + Du_k \end{cases}$$

Discrete state space model

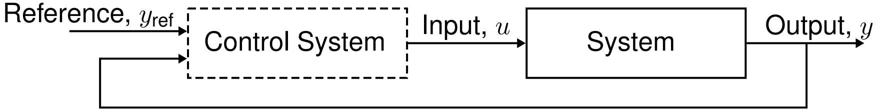
z-transform

Inverse z-transform

$$H(z) = C(zI - \Phi)^{-1}\Gamma + D$$

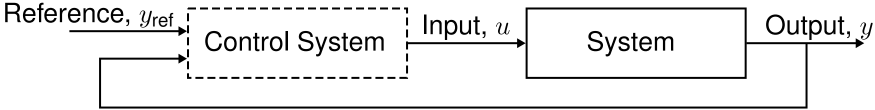
Discrete transfer function

Task: Design a cruise control for a car.



- **Control Input**: Throttle position u
- **Measured Output**: Velocity of the car y
- **Reference Input**: Desired velocity of the car y_{ref}

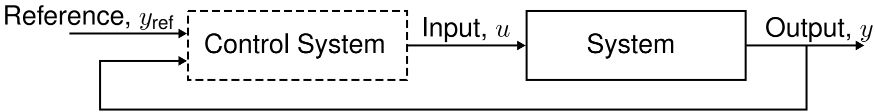
Task: Design a cruise control for a car.



In this course, answers to the following questions are provided:

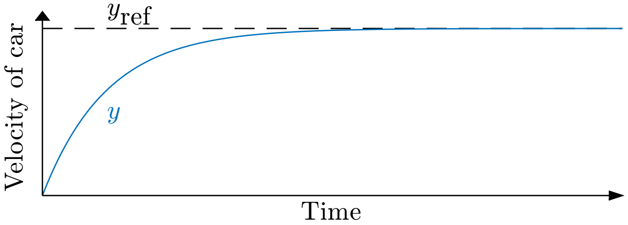
1. How should the control input u be designed such that a desired velocity y_{ref} is reached?

Task: Design a cruise control for a car.

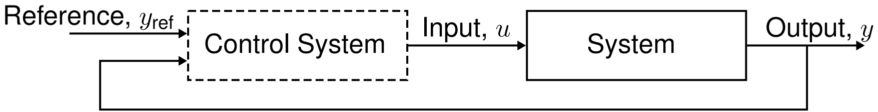


In this course, answers to the following questions are provided:

1. How should the control input u be designed such that a desired velocity y_{ref} is reached?



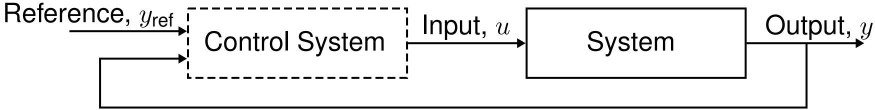
Task: Design a cruise control for a car.



In this course, answers to the following questions are provided:

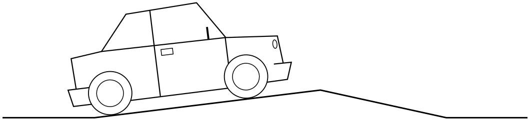
1. How should the control input u be designed such that a desired velocity y_{ref} is reached?
2. Can y_{ref} be reached despite uncertainties in the system?

Task: Design a cruise control for a car.

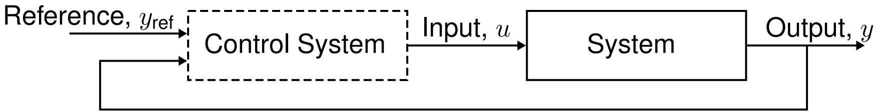


In this course, answers to the following questions are provided:

1. How should the control input u be designed such that a desired velocity y_{ref} is reached?
2. Can y_{ref} be reached despite uncertainties in the system?



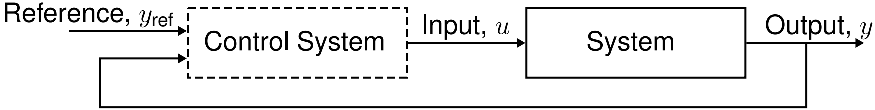
Task: Design a cruise control for a car.



In this course, answers to the following questions are provided:

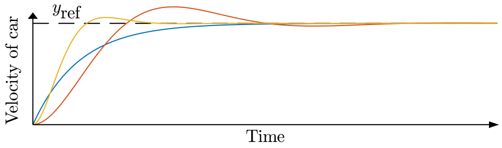
1. How should the control input u be designed such that a desired velocity y_{ref} is reached?
2. Can y_{ref} be reached despite uncertainties in the system?
3. How can a "good" performance be obtained?

Task: Design a cruise control for a car.



In this course, answers to the following questions are provided:

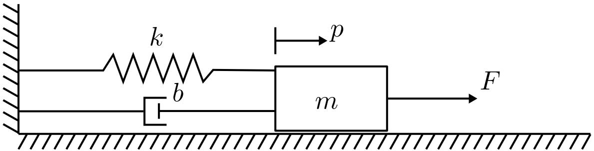
1. How should the control input u be designed such that a desired velocity y_{ref} is reached?
2. Can y_{ref} be reached despite uncertainties in the system?
3. How can a "good" performance be obtained?



To reason about the behavior of a system, we setup ***models*** that describe the input-output behavior of the system.

The models are given by

- Differential equations
- Algebraic relations



- The force F is considered to be the input
- The mass velocity is considered to be the output of the system

The system is of second order, since the dynamics of the system depends on both position and velocity.

The system model is given by Newton's second law[2]

$$m\ddot{p} = -k \cdot p - b \cdot \dot{p} + F$$

where F is the input (corresponds to u in the drawing).

[2] The following shorthand notation is introduced for the time derivative $\dot{p} := \frac{dp}{dt}$

The system model is given by Newton's second law[2]

$$m\ddot{p} = -k \cdot p - b \cdot \dot{p} + F$$

where F is the input (corresponds to u in the drawing).

The output y of the system is given by the relation

$$y = \dot{p}$$

[2]The following shorthand notation is introduced for the time derivative $\dot{p} := \frac{dp}{dt}$

The system model is given by Newton's second law[2]

$$m\ddot{p} = -k \cdot p - b \cdot \dot{p} + F$$

where F is the input (corresponds to u in the drawing).

The output y of the system is given by the relation

$$y = \dot{p}$$

It is seen that the system model comprises

- a 2nd order differential equation (given by Newton's second law) and
- an algebraic relation (output relation)

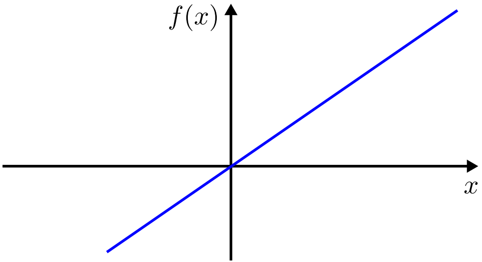
[2] The following shorthand notation is introduced for the time derivative $\dot{p} := \frac{dp}{dt}$

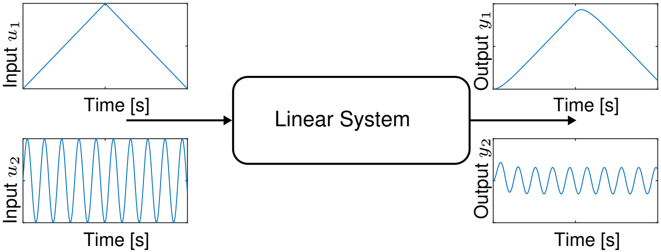
The map $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be ***linear*** if for any $x, y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, the following conditions hold

$$f(x + y) = f(x) + f(y) \qquad \text{Superposition}$$
$$f(\alpha x) = \alpha f(x) \qquad \text{Homogeneity}$$

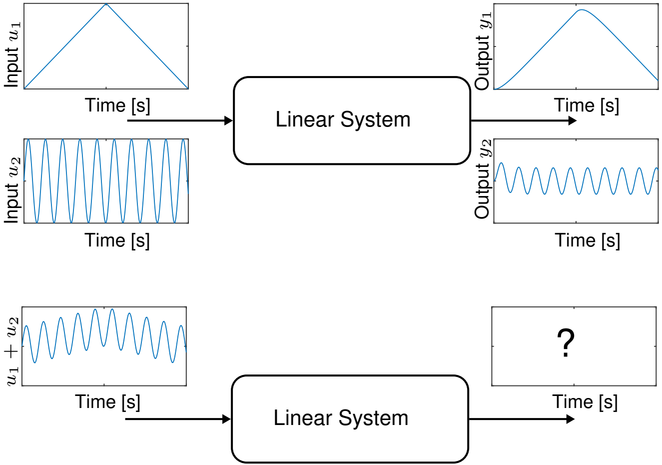
The map $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be ***linear*** if for any $x, y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, the following conditions hold

$$f(x+y) = f(x) + f(y) \qquad \text{Superposition}$$
$$f(\alpha x) = \alpha f(x) \qquad \text{Homogeneity}$$

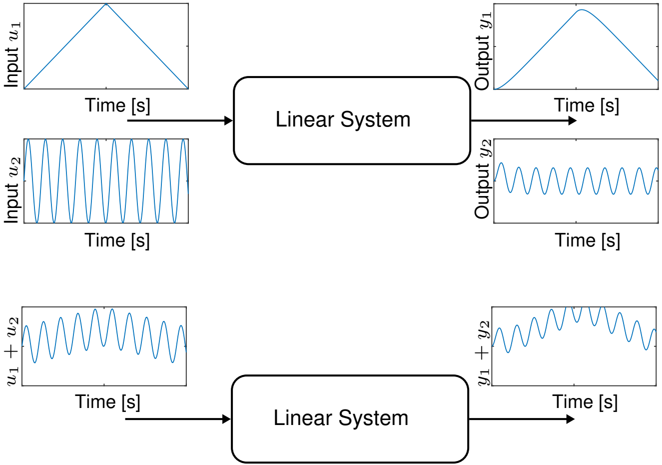




Input u_1
Time [s]
Input u_2
Time [s]
Linear System
Output y_1
Time [s]
Output y_2
Time [s]



Input u_1
Time [s]
Input u_2
Time [s]
Linear System
Output y_1
Time [s]
Output y_2
Time [s]
$u_1 + u_2$
Time [s]
Linear System
?
Time [s]



Input u_1
Time [s]
Input u_2
Time [s]
Linear System
Output y_1
Time [s]
Output y_2
Time [s]
$u_1 + u_2$
Time [s]
Linear System
$y_1 + y_2$
Time [s]

Let $\sigma : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^p$ define the input-output behavior of a system model Σ. The system Σ is ***time-invariant*** if for any input signal $u : \mathbb{R} \to \mathbb{R}^m$ and any delay $\tau \in \mathbb{R}$ the following relation holds

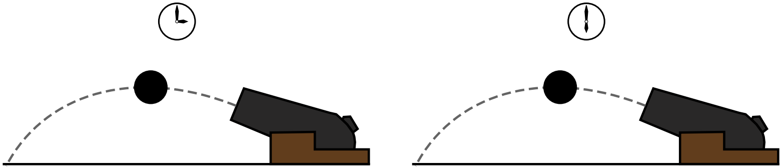
$$y(t - \tau) = \sigma(t, u(t - \tau))$$

for all times $t \in \mathbb{R}$, where y denotes the output signal of the system.

Let $\sigma : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^p$ define the input-output behavior of a system model Σ. The system Σ is ***time-invariant*** if for any input signal $u : \mathbb{R} \to \mathbb{R}^m$ and any delay $\tau \in \mathbb{R}$ the following relation holds

$$y(t - \tau) = \sigma(t, u(t - \tau))$$

for all times $t \in \mathbb{R}$, where y denotes the output signal of the system.



Consider the mass-spring-damper system

$$\ddot{p} = -\frac{k}{m} \cdot p - \frac{b}{m} \cdot \dot{p} + \frac{1}{m} F$$

Consider the mass-spring-damper system

$$\ddot{p} = -\frac{k}{m} \cdot p - \frac{b}{m} \cdot \dot{p} + \frac{1}{m} F$$

Linearity of Solution

- Let $p_1(t)$ be the solution to the differential equation for initial condition $(p_1(t_0), \dot{p}_1(t_0)) = (p_{1,0}, \dot{p}_{1,0})$ and input $F_1(t)$.
- Let $p_2(t)$ be the solution to the differential equation for initial condition $(p_2(t_0), \dot{p}_2(t_0)) = (p_{2,0}, \dot{p}_{2,0})$ and input $F_2(t)$.

Consider the mass-spring-damper system

$$\ddot{p} = -\frac{k}{m} \cdot p - \frac{b}{m} \cdot \dot{p} + \frac{1}{m} F$$

Linearity of Solution

- Let $p_1(t)$ be the solution to the differential equation for initial condition $(p_1(t_0), \dot{p}_1(t_0)) = (p_{1,0}, \dot{p}_{1,0})$ and input $F_1(t)$.
- Let $p_2(t)$ be the solution to the differential equation for initial condition $(p_2(t_0), \dot{p}_2(t_0)) = (p_{2,0}, \dot{p}_{2,0})$ and input $F_2(t)$.

The system is linear if $\alpha p_1(t) + \beta p_2(t)$ is the solution to the differential equation for initial condition $(\alpha p_{1,0} + \beta p_{2,0}, \alpha \dot{p}_{1,0} + \beta \dot{p}_{2,0})$ and input $\alpha F_1(t) + \beta F_2(t)$.

We consider two types of linear time-domain models

- Continuous-time state space models (based on differential equations)

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

- Discrete-time state space models (based on difference equations)

$$\begin{aligned} x_{k+1} &= \Phi x_k + \Gamma u_k \\ y_k &= C x_k + D u_k \end{aligned}$$

Continuous-Time State Space Models

Definition

A continuous-time state space model is a system of first order differential equations and an output equation, given by

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

Continuous-Time State Space Models

Definition

A continuous-time state space model is a system of first order differential equations and an output equation, given by

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

where

- $x \in \mathbb{R}^n$ is the state,
- $u \in \mathbb{R}^m$ is the input,
- $y \in \mathbb{R}^p$ is the output (measurement),
- $A \in \mathbb{R}^{n \times n}$ is the system matrix,
- $B \in \mathbb{R}^{n \times m}$ is the input matrix,
- $C \in \mathbb{R}^{p \times n}$ is the output matrix,
- $D \in \mathbb{R}^{p \times m}$ is the direct feedthrough matrix.

Continuous-Time State Space Models

Transformation to State Space Form (1)

$$z^{(n)} - a_{n-1} z^{(n-1)} - \cdots - a_1 \dot{z} - a_0 z = b_1 u_1(t) + \cdots + b_m u_m(t)$$

nth-order differential equation

Reformulation

Laplace transform

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

State space model

Laplace transform

Inverse Laplace transform

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

Transfer function

Discretization

Discretization

$$\begin{cases} x_{k+1} = \Phi x_k + \Gamma u_k \\ y_k \quad = C x_k + D u_k \end{cases}$$

Discrete state space model

z-transform

Inverse z-transform

$$H(z) = C(zI - \Phi)^{-1} \Gamma + D$$

Discrete transfer function

Given an n^{th} order differential equation

$$z^{(n)} - a_{n-1} z^{(n-1)} - \cdots - a_1 \dot{z} - a_0 z = b_1 u_1(t) + \cdots + b_m u_m(t)$$

where $z^{(k)}$ denotes the k^{th} time derivative of z.
Reformulate the differential equation as a system of first order differential equations on the following form

$$\dot{x} = Ax + Bu$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$.

Continuous-Time State Space Models

Transformation to State Space Form (2)

To transform a differential equation into state space form, the following procedure can be followed, when z is the unknown function.

1. Define variables x_i according to $x_1 = z$, $x_2 = \dot{z}, \ldots, x_n = z^{(n-1)}$.
2. Define the system matrix $A \in \mathbb{R}^{n \times n}$ and input matrix $B \in \mathbb{R}^{n \times m}$ as

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix}}_{=\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \end{bmatrix}}_{=A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}}_{=x} + \underbrace{\begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \\ b_1 & \cdots & b_m \end{bmatrix}}_{=B} \underbrace{\begin{bmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{bmatrix}}_{=u(t)}$$

To transform a differential equation into state space form, the following procedure can be followed, when z is the unknown function.

1. Define variables x_i according to $x_1 = z$, $x_2 = \dot{z}, \dots, x_n = z^{(n-1)}$.
2. Define the system matrix $A \in \mathbb{R}^{n \times n}$ and input matrix $B \in \mathbb{R}^{n \times m}$ as

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix}}_{=\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \end{bmatrix}}_{=A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}}_{=x} + \underbrace{\begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \\ b_1 & \cdots & b_m \end{bmatrix}}_{=B} \underbrace{\begin{bmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{bmatrix}}_{=u(t)}$$

We associate the following output equation to the system description

$$y = Cx + Du$$

where $y \in \mathbb{R}^p$ is the output (vector of measured variables).

Recall the dynamics of the mass-spring-damper system

$$m\ddot{p} = -k \cdot p - b \cdot \dot{p} + F$$

where p is the position of the mass and F is the input force.

Recall the dynamics of the mass-spring-damper system

$$m\ddot{p} = -k \cdot p - b \cdot \dot{p} + F$$

where p is the position of the mass and F is the input force.

1. Define the state vector as

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix}$$

Recall the dynamics of the mass-spring-damper system

$$m\ddot{p} = -k \cdot p - b \cdot \dot{p} + F$$

where p is the position of the mass and F is the input force.

1. Define the state vector as

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix}$$

2. Define the system equation as

$$\underbrace{\begin{bmatrix} \dot{p} \\ \ddot{p} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}}_{=A} \underbrace{\begin{bmatrix} p \\ \dot{p} \end{bmatrix}}_{=x} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{=B} \underbrace{F}_{=u}$$

Recall the dynamics of the mass-spring-damper system

$$m\ddot{p} = -k \cdot p - b \cdot \dot{p} + F$$

where p is the position of the mass and F is the input force.

1. Define the state vector as

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix}$$

2. Define the system equation as

$$\underbrace{\begin{bmatrix} \dot{p} \\ \ddot{p} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}}_{=A} \underbrace{\begin{bmatrix} p \\ \dot{p} \end{bmatrix}}_{=x} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{=B} \underbrace{F}_{=u}$$

3. Define the output equation as (we measure the velocity, i.e., $\dot{p}$)

$$y = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{=C} \underbrace{\begin{bmatrix} p \\ \dot{p} \end{bmatrix}}_{=x} + \underbrace{0}_{=D} \cdot \underbrace{F}_{=u}$$

Recall that one can write an analytical expression for the solution to first order linear ODEs. An example is

$$\frac{dx}{dt} + ax = u(t)$$

where $a \in \mathbb{R}$ is some constant, and $u(t)$ is an input.

Recall that one can write an analytical expression for the solution to first order linear ODEs. An example is

$$\frac{dx}{dt} + ax = u(t)$$

where $a \in \mathbb{R}$ is some constant, and $u(t)$ is an input.

Assume that $u(t) = 0$, then we obtain the first order linear ***homogeneous*** ODE

$$\frac{dx}{dt} + ax = 0.$$

Recall that one can write an analytical expression for the solution to first order linear ODEs. An example is

$$\frac{dx}{dt} + ax = u(t)$$

where $a \in \mathbb{R}$ is some constant, and $u(t)$ is an input.

Assume that $u(t) = 0$, then we obtain the first order linear ***homogeneous*** ODE

$$\frac{dx}{dt} + ax = 0.$$

Separate the variables as

$$\frac{1}{x}dx = -adt$$

Recall that one can write an analytical expression for the solution to first order linear ODEs. An example is

$$\frac{dx}{dt} + ax = u(t)$$

where $a \in \mathbb{R}$ is some constant, and $u(t)$ is an input.

Assume that $u(t) = 0$, then we obtain the first order linear ***homogeneous*** ODE

$$\frac{dx}{dt} + ax = 0.$$

Separate the variables as

$$\frac{1}{x}dx = -adt$$

Integrate the expression

$$\int \frac{1}{x}dx = \int -adt$$

Assume that $u(t) = 0$, then we obtain the first order linear ***homogeneous*** ODE

$$\frac{dx}{dt} + ax = 0.$$

Separate the variables as

$$\frac{1}{x}dx = -adt$$

Integrate the expression

$$\int \frac{1}{x}dx = \int -adt$$
$$\log|x| + c_1 = -at + c_2$$

Assume that $u(t) = 0$, then we obtain the first order linear ***homogeneous*** ODE

$$\frac{dx}{dt} + ax = 0.$$

Separate the variables as

$$\frac{1}{x}dx = -adt$$

Integrate the expression

$$\begin{aligned} \int \frac{1}{x}dx &= \int -adt \\ \log|x| + c_1 &= -at + c_2 \\ \log|x| &= -at + c_3 \end{aligned}$$

Assume that $u(t) = 0$, then we obtain the first order linear ***homogeneous*** ODE

$$\frac{dx}{dt} + ax = 0.$$

Separate the variables as

$$\frac{1}{x}dx = -adt$$

Integrate the expression

$$\begin{aligned} \int \frac{1}{x}dx &= \int -adt \\ \log|x| + c_1 &= -at + c_2 \\ \log|x| &= -at + c_3 \\ |x| &= e^{-at}e^{c_3} \end{aligned}$$

Separate the variables as

$$\frac{1}{x}dx = -adt$$

Integrate the expression

$$\begin{aligned}\int \frac{1}{x}dx &= \int -adt\\ \log|x| + c_1 &= -at + c_2\\ \log|x| &= -at + c_3\\ |x| &= e^{-at}e^{c_3}\end{aligned}$$

The solution to the first order linear homogeneous ODE is then

$$x(t) = e^{-at}x(0).$$

The solution to a system of first order differential can be found numerically in different software tools - In this course I use MATLAB.

```
%% Define system dynamics
A = [0 1;-1 -1];
B = [0; 1];
C = [1 0];
D = 0;
% Create object for dynamical system
sys = ss(A,B,C,D);
%% Simulate the system
figure
step(sys)
figure
impulse(sys)
```

Continuous-Time State Space Models

Simulation in MATLAB

There are numerous methods for simulation in MATLAB; however, the following is quite general.

```
tEnd = 10; % End time of simulation
x0 = [1;2]; % Initial condition
% Specification of input
tu = linspace(0,tEnd)';
u = cos(tu);
% Simulate the system
options = []; % Solver options (can be changed with odeset)
[tSim,xSim] = ode45(@sysDesc,[0 tEnd],x0,options,[tu u],sys);
% Plot result
figure;plot(tSim,xSim)

function dxdt = sysDesc(t,x,u,sys)
    u = interp1(u(:,1),u(:,2),t);
    dxdt = sys.A*x+sys.B*u;
end
```

A discrete-time state space model is a system of first order difference equations and an output equation, given by

$$
\begin{aligned}
x_{k+1} &= \Phi x_k + \Gamma u_k \\
y_k &= C x_k + D u_k
\end{aligned}
$$

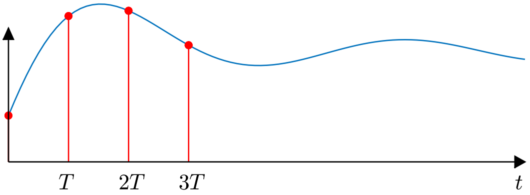
A discrete-time state space model is a system of first order difference equations and an output equation, given by

$$\begin{aligned} x_{k+1} &= \Phi x_k + \Gamma u_k \\ y_k &= C x_k + D u_k \end{aligned}$$

where

- $x_k \in \mathbb{R}^n$ is the state,
- $u_k \in \mathbb{R}^m$ is the input,
- $y_k \in \mathbb{R}^p$ is the output (measurement),
- $\Phi \in \mathbb{R}^{n \times n}$ is the system matrix,
- $\Gamma \in \mathbb{R}^{n \times m}$ is the input matrix,
- $C \in \mathbb{R}^{p \times n}$ is the output matrix,
- $D \in \mathbb{R}^{p \times m}$ is the direct feedthrough matrix.

The objective of the discrete-time model is to describe the solution at sampling instances.



Discrete-Time State Space Models

Transformation from Continuous-Time to Discrete-Time

$$z^{(n)} - a_{n-1} z^{(n-1)} - \cdots - a_1 \dot{z} - a_0 z = b_1 u_1(t) + \cdots + b_m u_m(t)$$

nth-order differential equation

Reformulation

Laplace transform

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

State space model

Laplace transform

Inverse Laplace transform

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

Transfer function

Discretization

Discretization

$$\begin{cases} x_{k+1} = \Phi x_k + \Gamma u_k \\ y_k \quad\;\; = C x_k + D u_k \end{cases}$$

Discrete state space model

z-transform

Inverse z-transform

$$H(z) = C(zI - \Phi)^{-1} \Gamma + D$$

Discrete transfer function

A system on state space form

$$\dot{x} = Ax + Bu$$

has solution

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau$$

A system on state space form

$$\dot{x} = Ax + Bu$$

has solution

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau$$

The solution at time $t_{k+1} = t_k + T$ can be written as

$$x(t_{k+1}) = e^{AT}x(t_k) + \int_{t_k}^{t_k+T} e^{A(t_k+T-\tau)}Bu(\tau)d\tau$$

A system on state space form

$$\dot{x} = Ax + Bu$$

has solution

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau$$

The solution at time $t_{k+1} = t_k + T$ can be written as

$$x(t_{k+1}) = e^{AT}x(t_k) + \int_{t_k}^{t_k+T} e^{A(t_k+T-\tau)}Bu(\tau)d\tau$$

Assumption: The input is constant between samples, i.e., $u(\tau) = u(t_k)$ for all $\tau \in [t_k, t_k + T)$

A system on state space form

$$\dot{x} = Ax + Bu$$

has solution

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau$$

The solution at time $t_{k+1} = t_k + T$ can be written as

$$x(t_{k+1}) = e^{AT}x(t_k) + \int_{t_k}^{t_k+T} e^{A(t_k+T-\tau)}Bu(\tau)d\tau$$

Assumption: The input is constant between samples, i.e., $u(\tau) = u(t_k)$ for all $\tau \in [t_k,\, t_k + T)$; thus,

$$x(t_{k+1}) = e^{AT}x(t_k) + \left(\int_{t_k}^{t_k+T} e^{A(t_k+T-\tau)}Bd\tau\right)u(t_k)$$

The solution at time $t_{k+1} = t_k + T$ can be written as

$$x(t_{k+1}) = e^{AT}x(t_k) + \int_{t_k}^{t_k+T} e^{A(t_k+T-\tau)}Bu(\tau)d\tau$$

Assumption: The input is constant between samples, i.e., $u(\tau) = u(t_k)$ for all $\tau \in [t_k, t_k + T)$; thus,

$$x(t_{k+1}) = e^{AT}x(t_k) + \left(\int_{t_k}^{t_k+T} e^{A(t_k+T-\tau)}Bd\tau\right)u(t_k)$$

$$x(t_{k+1}) = e^{AT}x(t_k) + \left(\int_{0}^{T} e^{A(T-\tau)}Bd\tau\right)u(t_k)$$

The solution at time $t_{k+1} = t_k + T$ can be written as

$$x(t_{k+1}) = e^{AT} x(t_k) + \int_{t_k}^{t_k+T} e^{A(t_k+T-\tau)} B u(\tau) d\tau$$

Assumption: The input is constant between samples, i.e., $u(\tau) = u(t_k)$ for all $\tau \in [t_k, t_k + T)$; thus,

$$x(t_{k+1}) = e^{AT} x(t_k) + \left(\int_{t_k}^{t_k+T} e^{A(t_k+T-\tau)} B d\tau \right) u(t_k)$$

$$x(t_{k+1}) = e^{AT} x(t_k) + \left(\int_{0}^{T} e^{A(T-\tau)} B d\tau \right) u(t_k)$$

$$x(t_{k+1}) = \underbrace{e^{AT}}_{=\Phi} x(t_k) + \underbrace{\left(\int_{0}^{T} e^{A(T-\tau)} B d\tau \right)}_{=\Gamma} u(t_k)$$

Recall the dynamics of the mass-spring-damper system

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

with

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} p \\ \dot{p} \end{bmatrix}$$

Recall the dynamics of the mass-spring-damper system

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

with

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} p \\ \dot{p} \end{bmatrix}$$

The discrete time system is given by

$$x_{k+1} = \Phi x_k + \Gamma u_k$$
$$y_k = C x_k$$

Recall the dynamics of the mass-spring-damper system

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

with

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} p \\ \dot{p} \end{bmatrix}$$

The discrete time system is given by

$$x_{k+1} = \Phi x_k + \Gamma u_k$$
$$y_k = C x_k$$

where (for $m = 1$, $b = 3$, $k = 2$, $T = 0.1$)[3]

$$\Phi = e^{AT} \approx \begin{bmatrix} 0.9909 & 0.0861 \\ -0.1722 & 0.7326 \end{bmatrix}, \quad \Gamma = \int_0^T e^{A(T-\tau)} B d\tau \approx \begin{bmatrix} 0.0045 \\ 0.0861 \end{bmatrix}, \quad x_k = \begin{bmatrix} p(t_k) \\ \dot{p}(t_k) \end{bmatrix}$$

[3] The matrix exponential can be computed with `expm` in MATLAB.

Frequency-Domain Models

A transfer function of a single-input single-output system is given by

$$G(s) = \frac{Q(s)}{P(s)}$$

where numerator $Q(s)$ and denominator $P(s)$ are polynomials in the variable s.

A transfer function of a single-input single-output system is given by

$$G(s) = \frac{Q(s)}{P(s)}$$

where numerator $Q(s)$ and denominator $P(s)$ are polynomials in the variable s.

- The roots of $Q(s)$ are called **the zeros of** $G(s)$.
- The roots of $P(s)$ are called **the poles of** $G(s)$.

A transfer function of a single-input single-output system is given by

$$G(s) = \frac{Q(s)}{P(s)}$$

where numerator $Q(s)$ and denominator $P(s)$ are polynomials in the variable s.

- The roots of $Q(s)$ are called **the zeros of** $G(s)$.
- The roots of $P(s)$ are called **the poles of** $G(s)$.

Only proper transfer functions are considered, i.e., it is assumed that the degree of $P(s)$ is higher than $Q(s)$.

The Laplace transform is defined as

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty f(t)e^{-st}dt.$$

The Laplace transform is defined as

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty f(t)e^{-st}dt.$$

The main property that is exploited is

$$\mathcal{L}\{\dot{f}(t)\} = sF(s) - f(0).$$

State Space Model to Transfer Function

$$z^{(n)} - a_{n-1} z^{(n-1)} - \cdots - a_1 \dot{z} - a_0 z = b_1 u_1(t) + \cdots + b_m u_m(t)$$

nth-order differential equation

Reformulation

Laplace transform

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

State space model

Laplace transform

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

Transfer function

Inverse Laplace transform

Discretization

Discretization

$$\begin{cases} x_{k+1} = \Phi x_k + \Gamma u_k \\ y_k \quad = C x_k + D u_k \end{cases}$$

Discrete state space model

z-transform

$$H(z) = C(zI - \Phi)^{-1}\Gamma + D$$

Discrete transfer function

Inverse z-transform

State Space Model to Transfer Function

Taking Laplace transforms of the system and assuming that $x_0 = 0$

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

yields

State Space Model to Transfer Function

Taking Laplace transforms of the system and assuming that $x_0 = 0$

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

yields

$$
\begin{aligned}
sx(s) &= Ax(s) + Bu(s) \\
y(s) &= Cx(s) + Du(s)
\end{aligned}
$$

rearranging, we obtain:

State Space Model to Transfer Function

Taking Laplace transforms of the system and assuming that $x_0 = 0$

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

yields

$$\begin{aligned} sx(s) &= Ax(s) + Bu(s) \\ y(s) &= Cx(s) + Du(s) \end{aligned}$$

rearranging, we obtain:

$$\begin{aligned} (sI - A)\, x(s) &= Bu(s) \\ y(s) &= Cx(s) + Du(s) \end{aligned}$$

State Space Model to Transfer Function

$$
\begin{aligned}
sx(s) &= Ax(s) + Bu(s) \\
y(s) &= Cx(s) + Du(s)
\end{aligned}
$$

rearranging, we obtain:

$$
\begin{aligned}
(sI - A)\,x(s) &= Bu(s) \\
y(s) &= Cx(s) + Du(s)
\end{aligned}
$$

Premultiplying with $(sI - A)^{-1}$ on either side of the system equation, results in

$$
\begin{aligned}
x(s) &= (sI - A)^{-1}\,Bu(s) \\
y(s) &= Cx(s) + Du(s)
\end{aligned}
$$

State Space Model to Transfer Function

$$
\begin{aligned}
(sI - A)\, x(s) &= Bu(s) \\
y(s) &= Cx(s) + Du(s)
\end{aligned}
$$

Premultiplying with $(sI - A)^{-1}$ on either side of the system equation, results in

$$
\begin{aligned}
x(s) &= (sI - A)^{-1}\, Bu(s) \\
y(s) &= Cx(s) + Du(s)
\end{aligned}
$$

Finally, we obtain:

$$
\begin{aligned}
x(s) &= (sI - A)^{-1}\, Bu(s) \\
y(s) &= C\, (sI - A)^{-1}\, Bu(s) + Du(s)
\end{aligned}
$$

State Space Model to Transfer Function

Premultiplying with $(sI - A)^{-1}$ on either side of the system equation, results in

$$\begin{aligned} x(s) &= (sI - A)^{-1} Bu(s) \\ y(s) &= Cx(s) + Du(s) \end{aligned}$$

Finally, we obtain:

$$\begin{aligned} x(s) &= (sI - A)^{-1} Bu(s) \\ y(s) &= C\,(sI - A)^{-1} Bu(s) + Du(s) \end{aligned}$$

Consequently,

$$y(s) = G(s)u(s)\,, \quad \text{where:}$$

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

Example: Mass-Spring-Damper

For the spring-mass-damper system with $m = 1$, $b = 3$, $k = 2$, the state space representation is:

$$\begin{pmatrix} \dot{p} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix} \begin{pmatrix} p \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} F$$

$$v = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} p \\ v \end{pmatrix} + \begin{pmatrix} 0 \end{pmatrix} F$$

Thus, the transfer function becomes:

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

Example: Mass-Spring-Damper

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

$$= \begin{pmatrix} 0 & 1 \end{pmatrix} \left(\begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix} \right)^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \end{pmatrix}$$

Example: Mass-Spring-Damper

$$
\begin{aligned}
G(s) &= C\left(sI - A\right)^{-1} B + D \\
&= \begin{pmatrix} 0 & 1 \end{pmatrix} \left(\begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix} \right)^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} s & -1 \\ 2 & s+3 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix}
\end{aligned}
$$

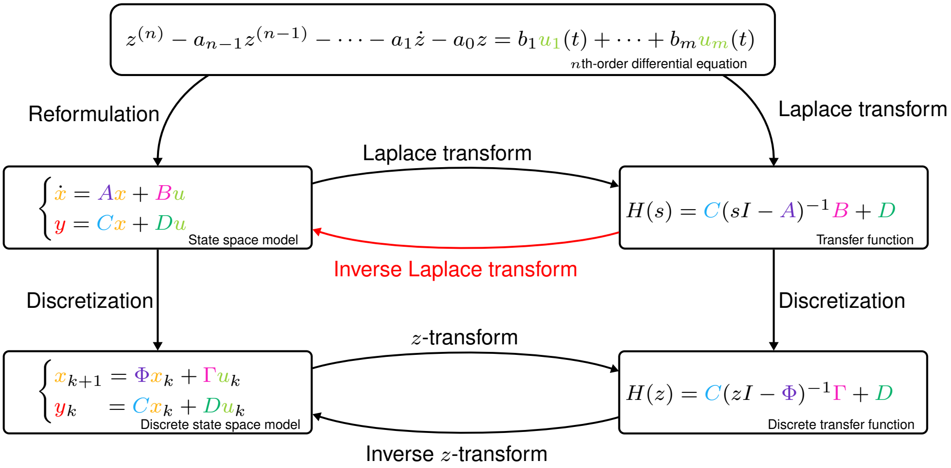
Example: Mass-Spring-Damper

$$
\begin{aligned}
G(s) &= C\,(sI - A)^{-1}\,B + D \\
&= \begin{pmatrix} 0 & 1 \end{pmatrix} \left(\begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix} \right)^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} s & -1 \\ 2 & s+3 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\
&= \frac{1}{s^2 + 3s + 2} \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} s+3 & 1 \\ -2 & s \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}
\end{aligned}
$$

Example: Mass-Spring-Damper

$$
\begin{aligned}
G(s) &= C\,(sI - A)^{-1}\,B + D \\
&= \begin{pmatrix} 0 & 1 \end{pmatrix} \left(\begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix} \right)^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} s & -1 \\ 2 & s+3 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\
&= \frac{1}{s^2 + 3s + 2} \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} s+3 & 1 \\ -2 & s \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\
&= \frac{s}{s^2 + 3s + 2}
\end{aligned}
$$

Transfer Function to State Space Model



Transfer Function to State Space Model

Consider the transfer function $g(s) = \frac{1}{s^2+a_1 s+a_2}$. From the relationship

$$y(s) = \frac{1}{s^2 + a_1 s + a_2} u(s)$$

we infer

$$s^2 y(s) + a_1 s y(s) + a_2 y(s) = u(s)$$

Taking inverse Laplace transform, this becomes:

$$\ddot{y}(t) + a_1 \dot{y}(t) + a_2 y(t) = u(t)$$

Transfer Function to State Space Model

$$\ddot{y}(t) + a_1\dot{y}(t) + a_2 y(t) = u(t)$$

A possible choice of states is: $x_1 = y$, $x_2 = \dot{y}$. With this choice, the system equations become:

$$\begin{aligned}
\dot{x}_1 &= \dot{y} = x_2 \\
\dot{x}_2 &= \ddot{y} = -a_1\dot{y} - a_2 y + u = -a_2 x_1 - a_1 x_2 + u
\end{aligned}$$

In matrix form, we obtain:

Transfer Function to State Space Model

$$\ddot{y}(t) + a_1\dot{y}(t) + a_2 y(t) = u(t)$$

A possible choice of states is: $x_1 = y$, $x_2 = \dot{y}$. With this choice, the system equations become:

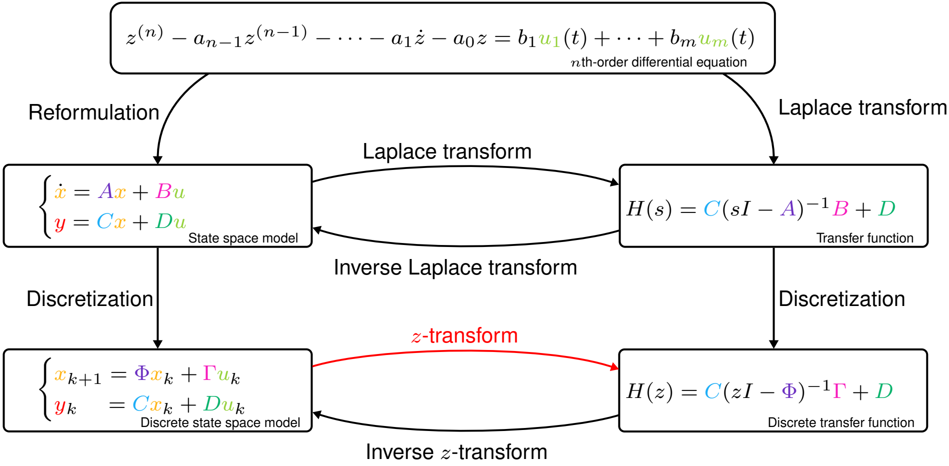
$$\begin{aligned} \dot{x}_1 &= \dot{y} = x_2 \\ \dot{x}_2 &= \ddot{y} = -a_1\dot{y} - a_2 y + u = -a_2 x_1 - a_1 x_2 + u \end{aligned}$$

In matrix form, we obtain:

$$\begin{aligned} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ -a_2 & -a_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \\ y &= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \end{pmatrix} u \end{aligned}$$

Discrete Transfer Functions

Z-Transform



The z-transform is defined as

$$\mathcal{Z}\{f(k)\} = F(z) = \sum_{k=0}^{\infty} f(k) z^{-k}.$$

The z-transform is defined as

$$\mathcal{Z}\{f(k)\} = F(z) = \sum_{k=0}^{\infty} f(k) z^{-k}.$$

The main property that is exploited is

$$\mathcal{Z}\{f(k-1)\} = z^{-1}(F(z) - f(0)).$$

Consider the difference equation

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) + b_0 u(k) + b_1 u(k-1) + b_2 u(k-2)$$

Consider the difference equation

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) + b_0 u(k) + b_1 u(k-1) + b_2 u(k-2)$$

The z-transformation of $y(k)$ is given by

$$Y(z) = (-a_1 z^{-1} - a_2 z^{-2}) Y(z) + (b_0 + b_1 z^{-1} + b_2 z^{-2}) U(z)$$

Consider the difference equation

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) + b_0 u(k) + b_1 u(k-1) + b_2 u(k-2)$$

The z-transformation of $y(k)$ is given by

$$Y(z) = (-a_1 z^{-1} - a_2 z^{-2}) Y(z) + (b_0 + b_1 z^{-1} + b_2 z^{-2}) U(z)$$

This leads to the discrete transfer function

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Taking the z-transform of the discrete-time state space model

$$
\begin{aligned}
x_{k+1} &= \Phi x_k + \Gamma u_k \\
y_k &= C x_k + D u_k
\end{aligned}
$$

yields

Taking the z-transform of the discrete-time state space model

$$\begin{aligned} x_{k+1} &= \Phi x_k + \Gamma u_k \\ y_k &= C x_k + D u_k \end{aligned}$$

yields

$$\begin{aligned} z x(z) &= \Phi x(z) + \Gamma u(z) \\ y(z) &= C x(z) + D u(z) \end{aligned}$$

rearranging, we obtain:

Taking the z-transform of the discrete-time state space model

$$
\begin{aligned}
x_{k+1} &= \Phi x_k + \Gamma u_k \\
y_k &= C x_k + D u_k
\end{aligned}
$$

yields

$$
\begin{aligned}
z x(z) &= \Phi x(z) + \Gamma u(z) \\
y(z) &= C x(z) + D u(z)
\end{aligned}
$$

rearranging, we obtain:

$$
G(z) = C(zI - \Phi)^{-1}\Gamma + D
$$

Discrete Transfer Functions

Discretization

$$z^{(n)} - a_{n-1} z^{(n-1)} - \cdots - a_1 \dot{z} - a_0 z = b_1 u_1(t) + \cdots + b_m u_m(t)$$

nth-order differential equation

Reformulation

Laplace transform

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

State space model

Laplace transform

Inverse Laplace transform

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

Transfer function

Discretization

Discretization

$$\begin{cases} x_{k+1} = \Phi x_k + \Gamma u_k \\ y_k \quad = C x_k + D u_k \end{cases}$$

Discrete state space model

z-transform

Inverse z-transform

$$H(z) = C(zI - \Phi)^{-1} \Gamma + D$$

Discrete transfer function

There are many methods for transforming a transfer function from s-domain to z-domain including

- Matched z-transformation
- Impulse invariant z-transformation
- Bilinear z-transformation

There are many methods for transforming a transfer function from s-domain to z-domain including

- Matched z-transformation
- Impulse invariant z-transformation
- Bilinear z-transformation

These methods have different properties, as you have seen previously in connection with design of IIR filters.

The equivalent characteristics in the z-plane are related to those in the s-plane by the expression

$$z = e^{sT}$$

where T is the sample period.

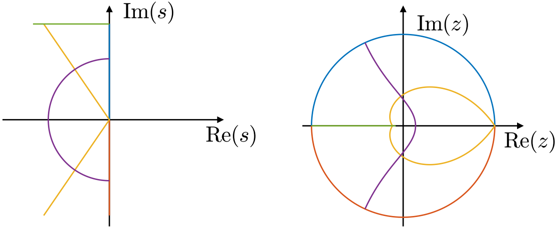
Discrete Transfer Functions

Relation between s and z

The equivalent characteristics in the z-plane are related to those in the s-plane by the expression

$$z = e^{sT}$$

where T is the sample period.



Summary

Relation between Models

$$z^{(n)} - a_{n-1} z^{(n-1)} - \cdots - a_1 \dot{z} - a_0 z = b_1 u_1(t) + \cdots + b_m u_m(t)$$

nth-order differential equation

Reformulation

Laplace transform

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

State space model

Laplace transform

Inverse Laplace transform

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

Transfer function

Discretization

Discretization

$$\begin{cases} x_{k+1} = \Phi x_k + \Gamma u_k \\ y_k = C x_k + D u_k \end{cases}$$

Discrete state space model

z-transform

Inverse z-transform

$$H(z) = C(zI - \Phi)^{-1} \Gamma + D$$

Discrete transfer function