Control Engineering

Claudio De Persis

Lecture 1
ver. 2.0.2.2

Organization

- **Course teacher:** Prof. C. De Persis
- Prerequisites: Linear Algebra, Signals and Systems, Calculus – e.g. working knowledge of matrix calculus and linear ordinary differential equations.
- Website for downloads such as lecture slides, tutorial exercises, additional material, etc. in Brightspace.
- Course material: Pdf of the book and reader (modeling). Other useful handouts are available in Brightspace.

Course book

Free internet book. A hard copy of the book can be ordered at Princeton University Press but it is not necessary.

- **Authors:** Karl J. Aström and Richard M. Murray.
- **Title:** Feedback Systems: An Introduction for Scientists and Engineers.
- **Download**:

  http://www.cds.caltech.edu/~murray/books/AM05/pdf/am08-complete_22Feb09.pdf

  (Link checked on 14 April 2024) Version: 28 September 2012.
- Second edition has been released (24 July 2020): Download:

  http://www.cds.caltech.edu/~murray/books/AM08/pdf/fbs-public_24Jul2020.pdf

  (Link checked on 14 April 2024) – we will use the 1st edition in this course, but if I find something in the 2nd edition worth mentioning, I will point it out.
- I sometimes borrow contents from the textbook A.D. Lewis, A Mathematical Approach to Classical Control

  https://mast.queensu.ca/~andrew/teaching/pdf/332-notes.pdf

  (Link checked on 14 April 2024)

Why this book?

- Multi-disciplinary (engineering, physics, biology, etc.)
- Self-contained and accessible
- It is free

Instruction lectures

- Schedule of the lectures, tutorials and computer practicals please refer to the document posted in Nestor

  `https://rooster.rug.nl/#/en/current/schedule/course-WBIE034-05/timeRange=all`

- Tutorials and computer labs are conducted in 6 groups
- The division of the students in groups (independent of their course program) is based on the alphabetical order.

Group PTL	Ardel → Diallo
Group PTL1	Dijkstra → Hermida Iglesias
Group PTL2	Hertoghs → Lungu
Group PTL3	Maaß → Schoonhoven
Group PTL4	Schrauwen → van der Velde
Group PTL5	van Drie → Yazganarikan

  Teaching assistants

Ayoub Chabbari	a.chahbari@student.rug.nl	PTL1 and PTL2
Evie Zwijghuizen	e.zwijghuizen@student.rug.nl	PTL and PTL4
Egor Turcan	e.turcan@student.rug.nl	PTL3
Enrico Portella	e.portella@student.rug.nl	PTL5

- Please **stick** to the schedule and don't change the group you have been assigned to!!!!!

Instruction lectures and practicum

- Presence at the tutorials and the practicals is **obligatory**
- There are 2 **obligatory** Computer Practicals: Matlab/Simulink exercises.
- **Marking scheme** The final mark is determined by the written exam only. To be admitted to the exam and to be able to pass the course, the student should have completed the practical assessments, turned in the reports for them on time and have them judged satisfactory.
- If you have already attended all the tutorials and passed the practicals last year, then you are exempted (but please inform you TA!)

Exam

- Exam scheduled on Thursday 20 June 2024, 15.00-17.00, Exam Hall 1 K14 - R5 Blauwborgje 4.
- Re-Exam scheduled on Thursday 11 July 2024, 11.45-13.45, Exam Hall 1 A1 - C7 Blauwborgje 4.

Organization

- Lectures: treatment of concepts/material, examples.
  I cover much of the book on slides, but I also assign readings from the book or the reader. Towards the end of the course, I cover parts (loop shaping in the frequency domain) whose treatment slightly departs from the textbook, for which I use the book by A.D. Lewis.
- Mondays: tutorials (2 hours), exercises. We post the text of the tutorials before the tutorial takes place. You are supposed to solve the exercises by yourself during the tutorial sessions. The TA will be available to answer your questions.
- Practicals: 2 times, 4 hour sessions. No lab practical

Overview

- **Introduction: objectives**
- Introduction to feedback control
- Introduction to modeling

Introduction: Objectives course

After this course you should be able

- to **analyze** simple **dynamical systems** obtained from first principle physical modeling
- to distinguish **linear and nonlinear** phenomena
- to **analyze** the dynamic behavior of **continuous-time linear** time-invariant systems
- to **design state and output feedback** controllers
- to **analyze** systems in the **frequency domain** (stability, Nyquist plots and Bode plots)
- to **design** controllers in the frequency domain (PID controllers, pole assignment, internal model, loop shaping)
- to use **Matlab** to simulate dynamical control systems

Last year program of the course is available in Nestor: it provides a detailed description of the contents covered during each lecture.

Introduction: Today

- Chapter 1 of book (Introduction to control systems)

This chapter is very accessible (almost no formula); please read it for an introduction to the history of control and its modern applications

- Part of Chapter 1 of reader and slides: introduction to modeling issues, motion equations for electrical and mechanical systems.

Objectives:

- to understand what control theory is
- to familiarise with simple examples of dynamical control systems

Control theory

If physics is the science of understanding the physical environment, then control theory may be viewed as the science of modifying that environment. Much more than even physics, control is a mathematically oriented science. Control principles are always expressed in mathematical form and are potentially applicable to any concrete situation.

Rudolf Kalman, Control theory, *Encyclopedia Britannica*

- Plant to control
- Controllable inputs (control)
- Uncontrollable inputs (disturbance)
- Output (sensor measurements)
- **Control algorithm (design)**

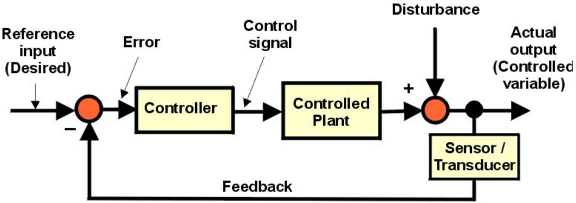


Fig.15: Feedback control system

Cruise control

Car model (see (3.3) in the textbook)

$$m\frac{dv}{dt} = \underbrace{-av^2}_{\text{aerodynamic drag}} + \underbrace{g(v)u}_{\text{driving force}} + \underbrace{d}_{\textit{disturbance}}$$

where

- $v \in \mathbb{R}$ is the velocity of the vehicle (state)
- $u \in \mathbb{R}$ is the throttle position (control input)
- $d \in \mathbb{R}$ is an unknown constant = slope of the road ($F_g = -mg\sin\theta$) + rolling friction (disturbance) – see figure on slide 16

and

- $a > 0$ is a physical parameter
- $g(v)$ is a nonlinear (quadratic) function of the velocity ((3.2) in the textbook): $g(v) = \alpha + \beta v - \gamma v^2$, with $\alpha, \beta, \gamma > 0$ parameters

Cruise control

Car model

$$m\frac{dv}{dt} = \underbrace{-av^2}_{\text{aerodynamic drag}} + \underbrace{g(v)u}_{\text{driving force}} + \underbrace{d}_{disturbance}$$

Equilibrium Let $\overline{v}$ be the cruise velocity and $\overline{u}$ be the constant throttle position such that

$$0 = -a\overline{v}^2 + g(\overline{v})\overline{u}$$

- $\overline{u}$ is the value of the throttle position that keeps the vehicle riding with the constant cruise speed $\overline{v}$ (when $d = 0$)
- Constant cruise speed $\rightarrow$ zero acceleration $\rightarrow$ equilibrium

Cruise control

Car model

$$m\frac{dv}{dt} = \underbrace{-av^2}_{\text{aerodynamic drag}} + \underbrace{g(v)u}_{\text{driving force}} + \underbrace{d}_{\textit{disturbance}}$$

Let $\overline{v}$ be the cruise velocity and $\overline{u}$ such that

$$0 = -a\overline{v}^2 + g(\overline{v})\overline{u}$$

Linearized model around $(\overline{v}, \overline{u})$

$$m\frac{d\tilde{v}}{dt} = -\underbrace{\left(2a\overline{v} + \frac{dg}{dv}\bigg|_{v=\overline{v}}\overline{u}\right)}_{\alpha}\tilde{v} + \underbrace{g(v)|_{v=\overline{v}}}_{\beta}\tilde{u} + d = -\alpha\tilde{v} + \beta\tilde{u} + d$$

- The model is obtained from computing the first order derivative of $f(x,u) := -av^2 + g(v)u$ wrt to x and u
- $\tilde{v} = v - \overline{v}$, $\tilde{u} = u - \overline{u}$ are the deviation of the state and the control from the equilibrium
- The linearized model approximates the dynamic behavior of the system around the equilibrium

Cruise control

(Linearized) car model

$$m\frac{d\tilde{v}}{dt} = -\alpha\tilde{v}(t) + \beta\tilde{u}(t) + d$$

Sensing

$$y = \tilde{v}\ (= v - \overline{v})$$

Actuation (throttle position) – PI control (see Chapter 1, page 24)

$$u(t) = \overline{u} + \tilde{u}(t) = \overline{u} + k_p e(t) + k_i \int_0^t e(s)ds, \text{ where } e(t) = -\tilde{y}(t)$$

We will understand in the forthcoming lectures why this $\tilde{u}$, with suitably designed gains k_p, k_i, achieves cruise control.

Note that the controller is a dynamical system as well, with input e and output $\tilde{u}$. In fact, define the controller state $\xi(t) := \int_0^t e(s)ds$; then

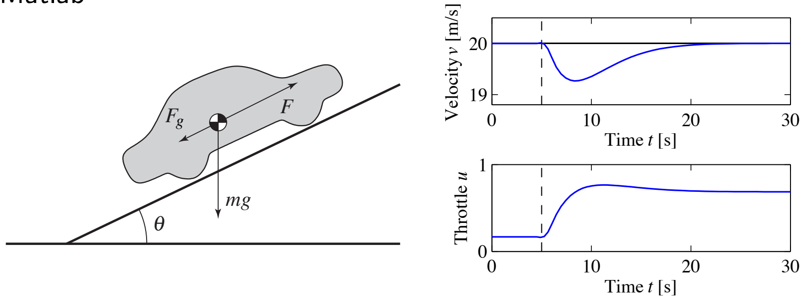
$$\begin{aligned} \dot{\xi} = &\quad e \\ \tilde{u} = &\quad k_i\xi + k_p e \end{aligned}$$

Cruise control

Cruise control response

$$m\frac{dv}{dt} = -av^2 + g(v)\left(\overline{u} + k_p(v(t) - \overline{v}) + k_i \int_0^t (v(s) - \overline{v})ds\right) + d$$

(integral-)differential equation that can be solved (simulated) in Matlab



(a) Effect of gravitational forces

(b) Closed loop response

Figure: Your textbook (Fig. 3.3)

Assignment

Assignment Read Chapter 1 of the textbook for

- Historical perspective on feedback control
- Modern engineering applications

Control is often the hidden technology

Model based control

In modern applications control design relies heavily on understanding the dynamics of the system and uses the (mathematical) **model** of the system

Therefore,

- Understanding of the system is important
- A useful model must be available or has to be built

(State space) system

A system is a mathematical object that relates input, state and output variables. When $\mathbb{T} = \mathbb{R}$ ($\mathbb{T}$ is the time domain) a convenient model of a system is that of a system of *first order* (nonlinear) o.d.e. with inputs and outputs, as follows:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t) \\ y(t) &= h(x(t), u(t), t) \end{aligned} \qquad f(x, u, t) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m, t) \\ f_2(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m, t) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m, t) \end{bmatrix}$$

with

- $u(t) \in \mathbb{U} = \mathbb{R}^m$ the control input
- $x(t) \in \mathbb{X} = \mathbb{R}^n$ the state variable
- $y(t) \in \mathbb{Y} = \mathbb{R}^p$ the output
- $t \in \mathbb{R}$ the time

- $f : \mathbb{X} \times \mathbb{U} \times \mathbb{T} \to \mathbb{X}$
- $h : \mathbb{X} \times \mathbb{U} \times \mathbb{T} \to \mathbb{Y}$

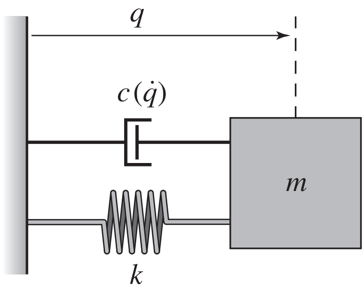
Most of time, f, h do not depend on t (**time invariant** systems).
In addition, if f, h are linear in x, u the system is linear time-invariant (LTI)

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

Different types of mathematical models

Deterministic exact relationship between measurable and derived variables, no stoch. uncertainty	**Stochastic** contains quantities described using stochastic variables or processes	
Dynamic variables changing with time, differential or difference equations	**Static** instantaneous links between variables, e.g., Ohm's law	
Continuous time $t \in \mathbb{R}$ relations with help of differential equations	**Discrete time** $k \in \mathbb{Z}$ relations with help of difference equations	**Hybrid** $(t, k) \in \mathbb{R} \times \mathbb{Z}$ mix of differential difference equations
Lumped Ordinary differential eqs. finite # of variables, discrete set of variables variables in space	**Distributed** Partial differential eq.s infinite # of variables, continuous set of variables variables in space	

Example: mass-spring-damper system



$m\ddot{q} + c(\dot{q}) + kq = F$

m mass

k spring coefficient

q cart displacement from rest pos.

$c(\dot{q}) = c\dot{q}$, $c > 0$, for simplicity

c damping coefficient

$\dot{q} = \frac{dq(t)}{dt}$, $\ddot{q} = \frac{d^2q(t)}{dt^2}$

F external force or control acting on the mass along the direction of motion

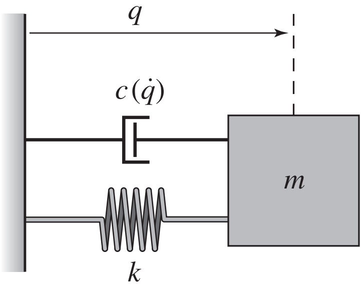
Equations obtainable via

$$\text{EL equations} \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial \mathcal{D}}{\partial \dot{q}} = F$$

where

Kinetic (co-)energy (mass)	$T^*(\dot{q}) = \frac{1}{2}m\dot{q}^2$
Potential energy (spring)	$E(q) = \frac{1}{2}kq^2$
Lagrangian function	$L(q,\dot{q}) = T^*(\dot{q}) - E(q)$
Rayleigh dissipation function	$\mathcal{D}(\dot{q}) = \frac{1}{2}c\dot{q}^2$

Example: mass-spring-damper system



$m\ddot{q} + c\dot{q} + kq = F$
m mass
k spring coefficient
q cart displacement from rest pos.
F external force

Define **output** $y(t) = q(t)$, **input** $u(t) = F(t)$, **states** $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix}$. Then

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} x_2(t) \\ -\frac{k}{m}x_1(t) - \frac{c}{m}x_2(t) + \frac{1}{m}u(t) \end{bmatrix} &&=: f(x(t), u(t)) \\ y(t) &= x_1(t) &&=: h(x(t)) \end{aligned}$$

is the state space representation of the system

Example: mass-spring-damper system

In fact the state space model

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} x_2(t) \\ -\frac{k}{m}x_1(t) - \frac{c}{m}x_2(t) + \frac{1}{m}u(t) \end{bmatrix} &&=: f(x(t), u(t)) \\ y(t) &= x_1(t) &&=: h(x(t)) \end{aligned}$$

is a <u>linear</u> state space model given by

$$\begin{aligned} \dot{x}(t) &= \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{B} u(t) \\ y(t) &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} x(t) \end{aligned}$$

Example: Keynes' model for an economy

An example of discrete-time system (Textbook, Exercise 2.4, p. 62)
$y(t)$: gross national product (GNP), year t [output]
$c(t)$: total consumption, year t [state]
$i(t)$: total investments, year t [state]
$g(t)$: expenses of government, year t [input]

$$y(t) = c(t) + i(t) + g(t)$$

Assumptions:

1. $c(t+1) = ay(t)$, $a > 0$ (consumption increases with GNP with 1-year delay)
2. $i(t+1) = b(c(t+1) - c(t))$, $b > 0$ (investment proportional to rate of change of consumption)

Then

$$\begin{array}{rcl} c(t+1) & = & ac(t) + ai(t) + ag(t) \\ i(t+1) & = & b(ac(t) + ai(t) + ag(t) - c(t)) \\ & & (ba - b)c(t) + ba\, i(t) + ba\, g(t) \\ y(t) & = & c(t) + i(t) + g(t), \quad t = 0, 1, 2, \ldots \end{array}$$

Example: Keynes' model for an economy

If we define the state vector and the control input as

$$x := \begin{bmatrix} c \\ i \end{bmatrix}, \quad u := g$$

we obtain the linear state space model of the system

$$\begin{aligned} x(t+1) &= \underbrace{\begin{bmatrix} a & a \\ ba - b & ba \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} a \\ ba \end{bmatrix}}_{B} u(t) \\ y(t) &= \underbrace{\begin{bmatrix} 1 & 1 \end{bmatrix}}_{C} x(t) + \underbrace{1}_{D} u(t) \end{aligned}$$

Linear discrete-time systems

The model

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t), \quad t = 0, 1, 2, \ldots \end{aligned}$$

is a forced or <u>controlled difference equation</u> representing a discrete-time system.

Given an input signal $u(t)$, $t = 0, 1, 2, \ldots$ and the state $x(0)$ at time 0, the state at each discrete time $t \geq 1$ can be recursively computed:

$$\begin{aligned} x(1) &= Ax(0) + Bu(0) \\ x(2) &= Ax(1) + Bu(1) = A^2x(0) + ABu(0) + Bu(1) \\ &\vdots \\ x(t) &= A^t x(0) + \textstyle\sum_{i=0}^{t-1} A^{t-1-i}Bu(i) \end{aligned}$$

and so its output

$$y(t) = CA^t x(0) + \sum_{i=0}^{t-1} CA^{t-1-i}Bu(i) + Du(t)$$

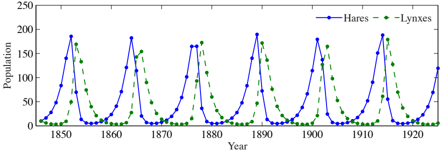
Discrete-time systems are extensively covered in the course **Digital and Hybrid Control Systems**

Example: Predator-prey model

$$\begin{aligned} x_1(k+1) &= x_1(k) + u(k)x_1(k) - ax_1(k)x_2(k) \\ x_2(k+1) &= x_2(k) - dx_2(k) + cx_1(k)x_2(k) \end{aligned}$$

- $x(k) \in \mathbb{R}^2$ population of prey and predator
- $u(k) \in \mathbb{R}$ birth rate per unit period controllable via food supply
- $ax_1(k)x_2(k)$ rate of predation, $cx_1(k)x_2(k)$ growth rate due to predation
- $dx_2(k)$ death rate

Population evolution



Evolution of the population of predators and preys as predicted by the model with $a = c = 0.014$, $u(k) = 0.6$ for all time instants k, $d = 0.7$.

Nonlinear discrete-time systems

The predator-prey model is a nonlinear discrete-time system

$$\begin{aligned} x(t+1) &= \quad f(x(t), u(t)) \\ y(t) &= \quad h(x(t), u(t)), \quad t = 0, 1, 2, \ldots \end{aligned}$$

In the example

$$f(x, u) = \begin{bmatrix} x_1 - a x_1 x_2 + u x_1 \\ x_2 + b x_1 x_2 - d x_2 \end{bmatrix}$$

Similarly to the linear case, solutions for nonlinear discrete-time systems can be recursively computed from the state at time $t = 0$ and an input sequence $u(0), u(1), u(2), \ldots$

$$\begin{aligned} x(1) &= \quad f(x(0), u(0)) \\ x(2) &= \quad f(x(1), u(1)) = f(f(x(0), u(0)), u(1)) \\ x(3) &= \quad f(x(2), u(2)) = f(f(f(x(0), u(0)), u(1)), u(2)) \\ &\vdots \end{aligned}$$

Simulation

While solving a difference equation amounts to a recursive calculation, solving differential equations

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

is a more difficult task. Due to this complexity more often the calculation of the solution of a differential equation is done numerically, approximating it via a difference equation.

Euler approximation

- Discretise the continuous time axis $\mathbb{T} = \mathbb{R}$ taking sampling times

$$\ldots, -h, 0, h, 2h, \ldots, k\Delta, \ldots, \quad k \in \mathbb{Z}$$

where $h > 0$ is the sampling interval.

- Approximate the derivative $\dot{x}(t)$ at each sampling time as

$$\dot{x}(kh) = \lim_{h \to 0} \frac{x(kh + h) - x(kh)}{h} \approx \frac{x((k+1)h) - x(kh)}{h}$$

- Obtain the discrete-time Euler approximation of the forced nonlinear differential equation

$$x((k+1)h) \approx x(kh) + h \cdot f(x(kh), u(kh))$$

Simulation

The solution to

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

can be approximated via the discretized model

$$x((k+1)h) = x(kh) + h \cdot f(x(kh), u(kh)) =: F(x(kh), u(kh))$$

At the sampling times

$$\begin{aligned} x(h) = &\quad x(0) + h \cdot f(x(0), u(0)) = F(x(0), u(0)) \\ x(2h) = &\quad x(h) + h \cdot f(x(h), u(h)) = F(x(h), u(h)) = F(F(x(0), u(0), u(h)) \\ \cdots &\quad \cdots \end{aligned}$$

Comments:
- The smaller the sampling time h, the better is the approximation
- The system $x((k+1)h) = F(x(kh), u(kh))$ is a discrete-time model:
rename $\hat{x}(k) = x(kh), \hat{u}(k) = u(kh)$ to obtain the difference equation

$$\hat{x}(k+1) = F(\hat{x}(k), \hat{u}(k))$$

Normalization and scaling

- Normalization and scaling amounts to introduce dimension-free variables and scale them to obtain models with a reduced number of parameters. This allows for faster and more accurate simulations.
- Normalization and scaling defines new variables by dividing the independent variables by the chosen normalization unit.

Example Normalization and scaling of the spring-mass system

$$m\ddot{q}(t) + kq(t) = u(t)$$

where, to fix the ideas, t is measured in sec and q in m.
Define the dimension-free independent variables (time and position)

$$z = \frac{q}{\ell}, \quad \tau = \omega_0 t$$

where ℓ (m) and ω_0 (sec^{-1}) are the chosen spatial and temporal length scale.

Normalization and scaling

Example (cont'd) Normalization and scaling of spring-mass system

In the chosen rescaled spatial *and* temporal variables, we have

$$z(\tau) := \left. \frac{q(t)}{\ell} \right|_{t=\frac{\tau}{\omega_0}}$$

Observe that by the chain rule

$$\begin{aligned} \frac{d}{d\tau} z(\tau) &= \left. \frac{\dot{q}(t)}{\ell} \right|_{t=\frac{\tau}{\omega_0}} \frac{1}{\omega_0} \\ \frac{d^2}{d\tau^2} z(\tau) &= \left. \frac{\ddot{q}(t)}{\ell} \right|_{t=\frac{\tau}{\omega_0}} \frac{1}{\omega_0^2} \end{aligned}$$

The original model $m\ddot{q}(t) + kq(t) = F(t)$ can be written as

$$m\omega_0^2 \left. \frac{\ddot{q}(t)}{\ell} \right|_{t=\frac{\tau}{\omega_0}} \frac{1}{\omega_0^2} + k \left. \frac{q(t)}{\ell} \right|_{t=\frac{\tau}{\omega_0}} = \left. \frac{F(t)}{\ell} \right|_{t=\frac{\tau}{\omega_0}}$$

or, in the scaled variables $$m\omega_0^2 \frac{d^2}{d\tau^2} z(\tau) + kz(\tau) = \left. \frac{F(t)}{\ell} \right|_{t=\frac{\tau}{\omega_0}}$$

Normalization and scaling

Example (cont'd) Normalization and scaling of spring-mass system

Dividing both sides of of $m\omega_0^2 \frac{d^2}{d\tau^2} z(\tau) + kz(\tau) = \left.\frac{F(t)}{\ell}\right|_{t=\frac{\tau}{\omega_0}}$ by $m\omega_0^2$, we obtain

$$\frac{d^2}{d\tau^2} z(\tau) + \frac{k}{m} \frac{1}{\omega_0^2} z(\tau) = \left.\frac{F(t)}{\ell}\right|_{t=\frac{\tau}{\omega_0}} \frac{1}{m\omega_0^2}$$

Define

$$\omega_0 = \sqrt{\frac{k}{m}} \qquad \text{where } [\omega_0^2] = \frac{[k]}{[m]} = \frac{\mathrm{kg}\,\mathrm{sec}^{-2}}{\mathrm{kg}} = \mathrm{sec}^{-2}$$

and the new input variable

$$u(\tau) = \left.\frac{F(t)}{\ell}\right|_{t=\frac{\tau}{\omega_0}} \frac{1}{m\omega_0^2}$$

to obtain the scaled and normalized model

$$\frac{d^2 z(\tau)}{d\tau^2} + z(\tau) = u(\tau)$$

Note that τ, $z(\tau)$, $u(\tau)$ are all dimension-free variables

Normalization and scaling

Example (cont'd) Normalization and scaling of spring-mass system

We can simulate the model

$$\frac{d^2 z(\tau)}{d\tau^2} + z(\tau) = u(\tau)$$

without using the actual values of the parameters, which usually lead to faster and more accurate simulations.

Once the solution $z(\tau)$ is computed, one can recover the actual physical position $q(t)$ by the relation

$$q(t) = \ell \, z(\tau)\big|_{\tau=\omega_0 t}$$

obtained inverting $z(\tau) = \left.\frac{q(t)}{\ell}\right|_{t=\frac{\tau}{\omega_0}}$.

Similarly one can obtain the actual values of the velocity $\dot{q}(t)$, the acceleration $\ddot{q}(t)$ and the force $F(t)$.

Next lecture

Next lecture: Tuesday 16 April 2024, 9:00-11:00

Slides, textbook and reader available on Brightspace