Lecture 1: Introduction to Dynamical Systems

- General introduction
- Modeling by differential equations (inputs, states, outputs)
- Simulation of dynamical systems
- Equilibria and linearization
- System interconnections and block diagrams

Related Reading

[AM]: Chapters 2.1-2.3, 4.1-4.2, 5.4 and [F]: Chapters 2.1-2.4

Dynamical Systems

In a dynamical system the effects of action do not occur immediately.

This phenomenon can be observed in all systems surrounding us:

- Mechatronic systems

  Pushing gas pedal in car does only gradually increase the velocity
- Heating systems

  Temperature does not rise immediately when a heater is switched on
- Biological systems

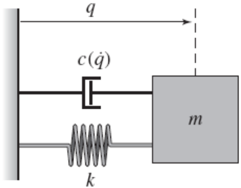
  A headache disappears slowly when medicine is taken
- Business systems: An investment leads to future gains or losses

Variables in a dynamical systems **evolve in time**. The time evolution depends on the **external excitation**, which itself changes over time.

Mathematical Models

One way to analyze the behavior of a dynamical system is by means of a **mathematical model**. Such models are often described by (ordinary or partial) **differential equations**.

Example: Mass-Spring-Damper System



Newton's second law leads to

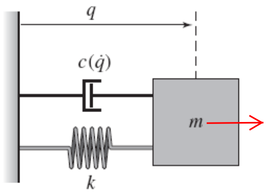
$$m\ddot{q} + c(\dot{q}) + kq = 0.$$

- q denotes the position of the mass (in a chosen coordinate system) and varies with time. $\dot{q}$ and $\ddot{q}$ are the velocity and the acceleration.
- kq is the spring restoring force (assumed to satisfy Hooke's law) and $c(\dot{q})$ is the friction force which can depend nonlinearly on the velocity.

Inputs

The previous system is said to be **autonomous** since it is not exposed to external influences. Non-autonomous systems do have external inputs.

Example: Mass-Spring-Damper System



With an external force u acting on the mass, we obtain

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

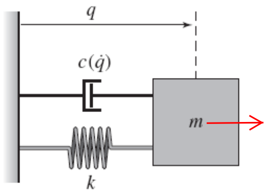
The external force u may as well vary with time. Depending on the circumstances it can be interpreted as follows:

- If we are allowed to manipulate u then it is called a **control input**.
- If u is generated by nature and cannot be influenced/changed by us then it is called a **disturbance input**.

Outputs

Often not all variables that appear in a model are of interest. We choose **outputs** in order to describe those quantities that get focus.

Example: Mass-Spring-Damper System



If we are only interested in the position of the mass, the output y is

$$m\ddot{q} + c(\dot{q}) + kq = u, \quad y = q.$$

Clearly the output y will vary in time, in response to the system and its input. Also outputs can have different interpretations:

- y is related to variables that can be measured (through sensors).
- y might just indicate a variable which we would like to monitor in order to investigate/analyze the properties of the system.

Interpretation of the Model

Let $u(t)$ be an **input signal** defined for $t \geq 0$ and an initial position q_0 as well as an initial velocity v_0. Solve the differential equation

$$m\ddot{q}(t) + c(\dot{q}(t)) + kq(t) = u(t) \;\; \text{with} \;\; q(0) = q_0, \; \dot{q}(0) = v_0.$$

If the nonlinear function $c(\cdot)$ has a continuous derivative, this differential equation with initial condition has a unique **solution** $q(t)$ and $\dot{q}(t)$ that is defined for $t \in [0, t_1)$ with some $t_1 > 0$.

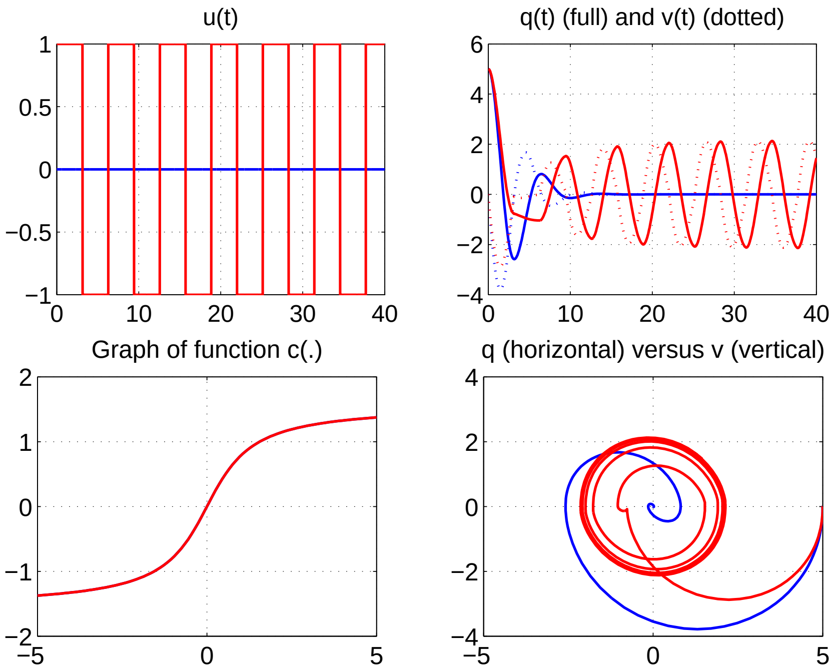
Warning and Reminder: Solutions might not exist for all $t \geq 0$.

Since the values for $q(0)$ and $\dot{q}(0)$ specify the solution of the differential equation (for a fixed input) uniquely, we call

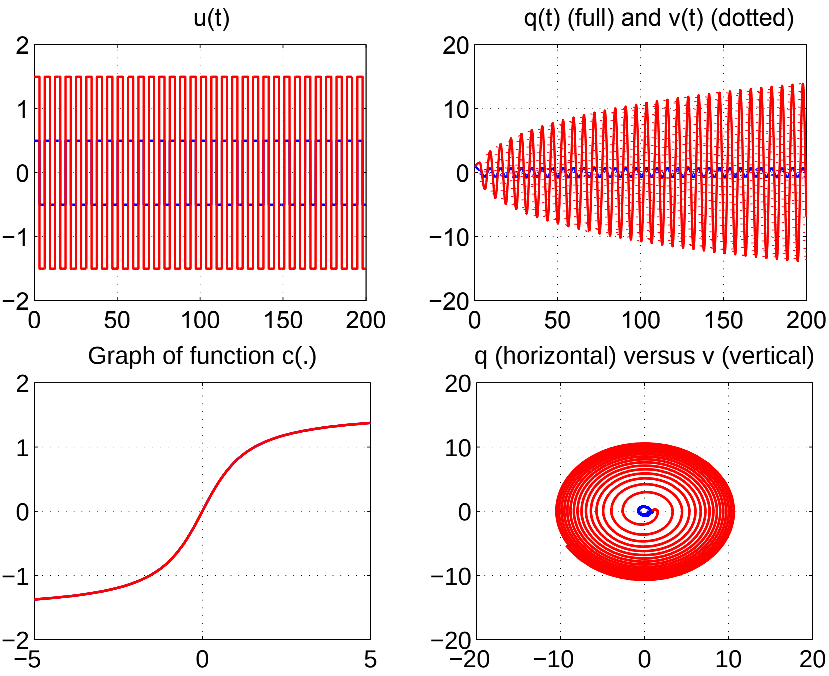
$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} \;\; \text{and} \;\; x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} q(t) \\ \dot{q}(t) \end{pmatrix}$$

the **state-vector** and **state-response** of the system. The system's **output (response)** is then just given by $y(t) = q(t)$.

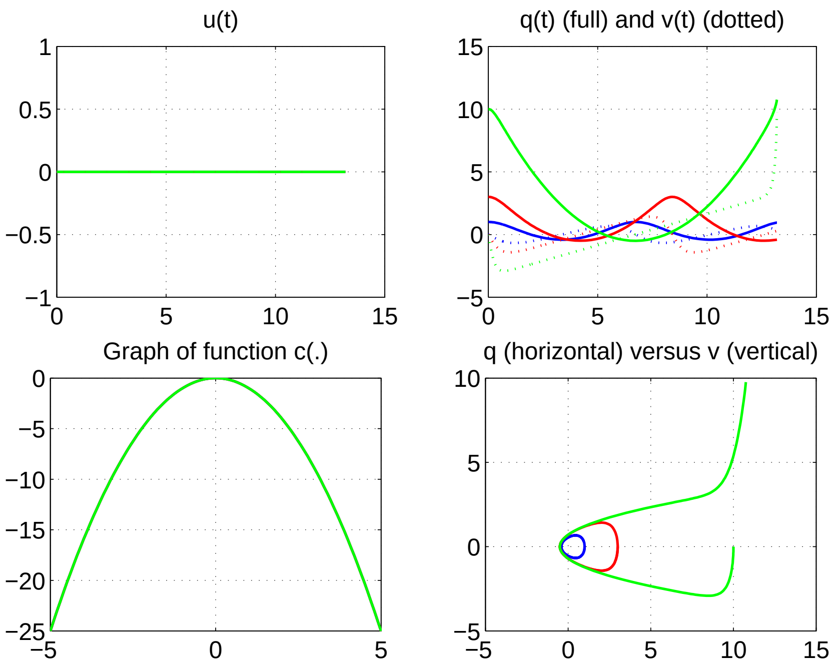
Demo: Free and Driven Response



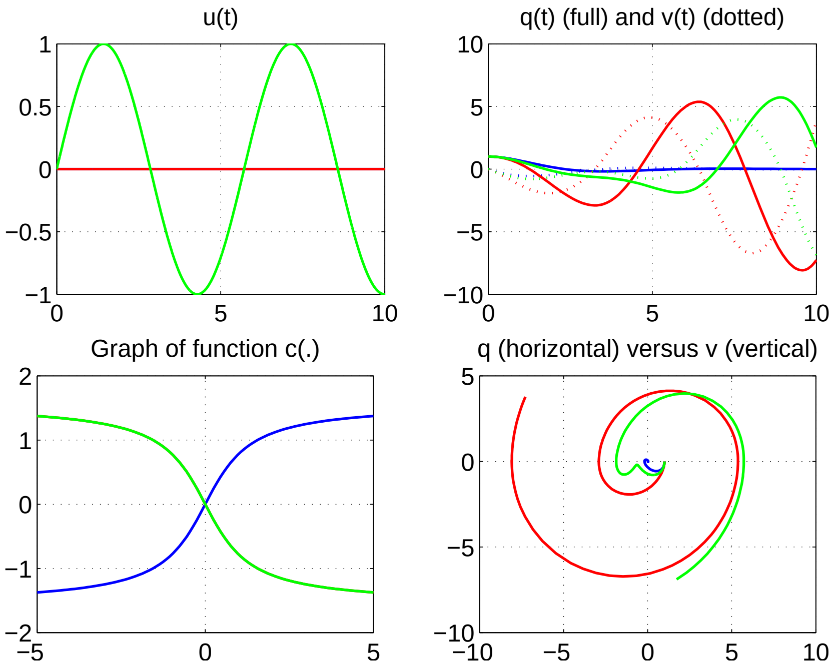
Demo: Effect of Nonlinearity



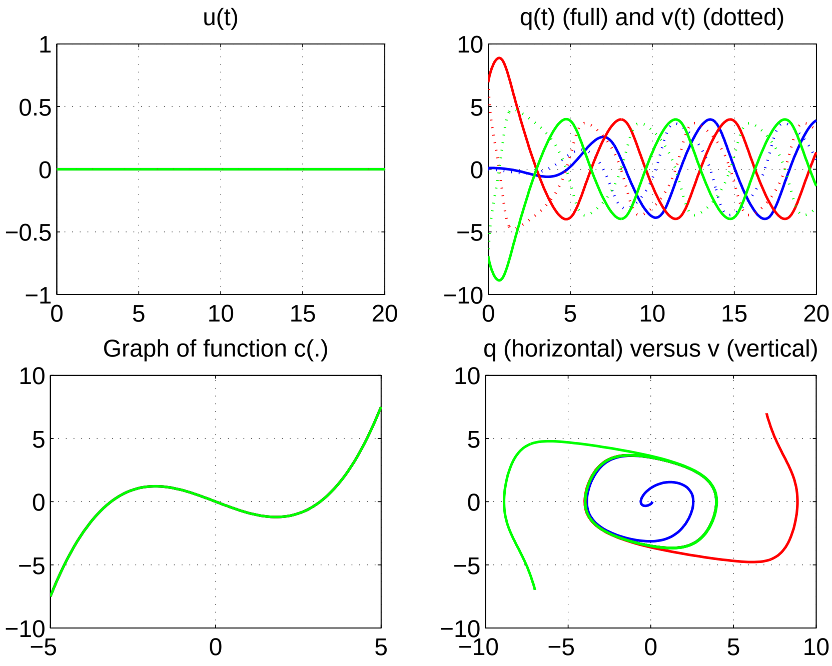
Demo: Finite Escape Time



Demo: Stability and Instability



Demo: Stable Limit Cycle



Interpretation of the Model

If we write down a system like

$$m\ddot{q} + c(\dot{q}) + kq = u, \quad y = q,$$

we are actually interested in its **behavior**, the set of all input-, state- and output-signals $u(t)$, $(q(t), \dot{q}(t))$ and $y(t)$ that satisfy these laws.

- Signals are functions of time and also called **trajectories**.
- The laws to-be-satisfied are often described by differential equations.
- Different system descriptions (laws) can describe the same behavior. We then call the system (descriptions) **equivalent**.

In the so-called **behavioral approach** to dynamical systems all this is developed in a mathematically precise fashion.

Sample Questions in Control

For a system like

$$m\ddot{q} + c(\dot{q}) + kq = u, \quad y = q$$

we are e.g. interested in the following issues:

- If there is no external excitation how does the state/output behave?
- Can we steer the system from one position to another?
- Is it possible to determine the input from knowledge of the output?
- Can we modify the system in order to create a desired behavior?

Some of these (yet rough) questions are related to **analyzing** the system for its properties, while others involve **synthesizing** control inputs or **modify the system laws** in order to change the system's behavior.

Control is the field to answer such questions systematically.

From Second-Order to First-Order Models

The system description

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

involves the first and second derivative of q. If $m \neq 0$ it is called a **second order** differential equation.

If we introduce the state-variables $x_1 = q$ and $x_2 = \dot{q}$ we conclude

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

$$\dot{x}_2 = \ddot{q} = -\frac{k}{m}q - \frac{c(\dot{q})}{m} + \frac{1}{m}u = -\frac{k}{m}x_1 - \frac{1}{m}c(x_2) + \frac{1}{m}u.$$

This can be written more compactly as

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -\frac{k}{m}x_1 - \frac{1}{m}c(x_2) + \frac{1}{m}u \end{pmatrix} = f(x_1, x_2, u).$$

This is a **first-order** and two-dimensional vector differential equation.

From High-Order to First-Order Models

Suppose r is a general real nonlinear function with $n+1$ arguments.

The system

$$q^{(n)} + r(q^{(n-1)}, q^{(n-2)}, \ldots, \dot{q}, q, u) = 0$$

is equivalent to

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{pmatrix} = \begin{pmatrix} x_2 \\ x_3 \\ \vdots \\ x_n \\ -r(x_n, x_{n-1}, \ldots, x_2, x_1, u) \end{pmatrix} = f(x, u).$$

Proof. Introduce $x_1 = q$, $x_2 = \dot{q}$, $\ldots$, $x_{n-1} = q^{(n-2)}$, $x_n = q^{(n-1)}$, substitute the variables and write down the resulting vector equation.

Non-Uniqueness

It is important to note that there is no unique way to do this. Understanding this issue will be an essential topic later.

The system

$$q^{(n)} + r(q^{(n-1)}, q^{(n-2)}, \ldots, \dot{q}, q, u) = 0$$

is equivalent to

$$\dot{z} = \begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_{n-1} \\ \dot{z}_n \end{pmatrix} = \begin{pmatrix} -r(z_1, z_2, \ldots, z_{n-1}, z_n, u) \\ z_1 \\ z_2 \\ \vdots \\ z_{n-1} \end{pmatrix} = \hat{f}(z, u).$$

Proof. Introduce $z_1 = q^{(n-1)}$, $z_2 = q^{(n-2)}$, $\ldots$, $z_{n-1} = \dot{q}$, $z_n = q$ and proceed as before.

State Equations From Linear Input-Output ODEs

Example 1: One of the states is made equal to the output.

$$\dddot{y}(t) + a_2\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = b_0 u(t)$$

The state variable definitions $x_1(t) = y(t)$, $x_2(t) = \dot{x}_1(t) = \dot{y}(t)$, $x_3(t) = \dot{x}_2(t) = \ddot{y}(t)$ leads to:

$$\begin{aligned}
\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{pmatrix} &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ b_0 \end{pmatrix} u(t) \\
y(t) &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} x(t).
\end{aligned}$$

State Equations From Linear Input-Output ODEs

Example 2: Derivatives of the input appear.

$$\dddot{y}(t) + a_2\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = b_2\ddot{u}(t) + b_1\dot{u}(t) + b_0 u(t)$$

Introduce variable $z(t)$ such that

$$\begin{aligned} u(t) &= \left(\frac{d^3}{dt^3} + a_2\frac{d^2}{dt^2} + a_1\frac{d}{dt} + a_0\right) z(t) \\ y(t) &= \left(b_2\frac{d^2}{dt^2} + b_1\frac{d}{dt} + b_0\right) z(t). \end{aligned}$$

Then using the state variable definitions $x_1(t) = z(t)$, $x_2(t) = \dot{z}(t)$, $x_3(t) = \ddot{z}(t)$ leads to:

$$\begin{aligned} \dot{x}(t) &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u(t) \\ y(t) &= \begin{pmatrix} b_0 & b_1 & b_2 \end{pmatrix} x(t). \end{aligned}$$

Generic System Description in Control

Physical modeling often leads to a system of higher-order differential equations. As see above, these can typically be equivalently written as a first-order vector differential equation

$$\dot{x} = f(x,u) \quad \text{and} \quad y = h(x,u).$$

Here $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ are smooth mappings.

This compact system description forms the starting point of control.

More explicitly these equations read as

$$\begin{array}{rclrcl} \dot{x}_1 & = & f_1(x_1,\ldots,x_n,u_1,\ldots,u_m) & y_1 & = & h_1(x_1,\ldots,x_n,u_1,\ldots,u_m) \\ & \vdots & & & \vdots & \\ \dot{x}_n & = & f_n(x_1,\ldots,x_n,u_1,\ldots,u_m) & y_k & = & h_k(x_1,\ldots,x_n,u_1,\ldots,u_m) \end{array}$$

Note that these functions could as well depend explicitly on time!

System Response and Differential Equations

With a control input $u(t)$ defined for $t \geq 0$ and an initial state $x_0 \in \mathbb{R}^n$, the state-response is obtained by solving the differential equation

$$\dot{x}(t) = f(x(t), u(t)) \;\; \text{with initial condition} \;\; x(0) = x_0.$$

The output response is just defined by $y(t) = h(x(t), u(t))$ for $t \geq 0$.

- Existence and uniqueness of the solution requires hypothesis on the function $f(x, u)$. That's why it is assumed to be smooth.
- The system is **time-invariant**. Roughly, the behavior does not change under time-shifts. That's why the initial time can be chosen to be 0.
- **Warning:** Even for "nice" systems it can happen that the solution does not exist for all times $t \geq 0$. It is then only defined for $t \in [0, t_+)$ and one can prove that the state escapes: $\|x(t)\| \to \infty$ for $t \to t_+$.

Generic Description of Linear Systems

If the functions $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ are **linear** they can be represented as

$$f(x, u) = Ax + Bu \quad \text{and} \quad h(x, u) = Cx + Du.$$

with matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{k \times n}$, $D \in \mathbb{R}^{k \times m}$.

Hence a generic **linear (time-invariant) system** is described as

$$\dot{x} = Ax + Bu \quad \text{and} \quad y = Cx + Du.$$

This is the system description which will be considered in this course.

The behavior of many dynamical systems in engineering can be (approximately) modeled by linear systems. On the other hand, physical modeling often results in descriptions with nonlinear dynamics.

How can we justify that this course discusses mainly linear systems?

Motivation for Generic System Description

The described compact and uniform but rather abstract notation for the description of a dynamical system has several substantial advantages.

- It can be applied to models that describe dynamical systems in totally **different physical domains**.
- A common system description allows the development of a **uniform set of tools** for addressing the questions raised above.

  Example: Simulation of a system with an ode-solver or with Simulink.

Creating such models might not be an easy task and often requires collaboration with colleagues with expertise in the respective fields.

The involved abstraction requires some training. But it turns systems and control into an **interdisciplinary science**.

Simulation

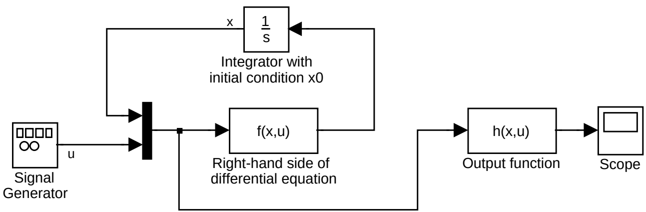
A system description in terms of the differential and output equation

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \quad y(t) = h(x(t), u(t))$$

allows the numerical computation of the response (to an input and an initial condition) by so-called ode-solvers (such as `ode45` or `ode15s` in Matlab). By integrating over time, the description is equivalent to

$$x(t) = x_0 + \int_0^t f(x(\tau), u(\tau))\, d\tau, \quad y(t) = h(x(t), u(t)).$$

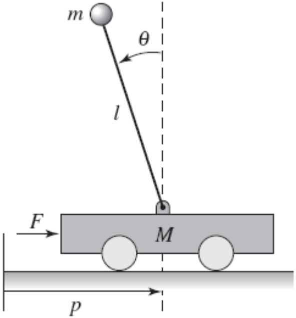
This leads to the general diagram for a Simulink simulation:



Example: Segway (Cart-Pendulum)

(a) Segway

(b) Saturn rocket



(c) Cart–pendulum system

Modeling with Lagrange technique ([F] p.30-31) leads to

$$\begin{aligned} (M+m)\ddot{p} - ml\cos(\theta)\ddot{\theta} + c\dot{p} + ml\sin(\theta)\dot{\theta}^2 &= F \\ -ml\cos(\theta)\ddot{p} + ml^2\ddot{\theta} + \gamma\dot{\theta} - mgl\sin(\theta) &= 0 \end{aligned}$$

where, next to the parameters defined in the figure, c and γ are the coefficients describing the viscous friction of the cart and the pendulum.

Example: First-Order System Description

More insightful description by using vector- and matrix-notation:

$$\begin{pmatrix} (M+m) & -ml\cos(\theta) \\ -ml\cos(\theta) & ml^2 \end{pmatrix} \begin{pmatrix} \ddot{p} \\ \ddot{\theta} \end{pmatrix} + \begin{pmatrix} c\dot{p} + ml\sin(\theta)\dot{\theta}^2 \\ \gamma\dot{\theta} - mgl\sin(\theta) \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}.$$

Introduce the abbreviations

$$U(\theta) = \begin{pmatrix} (M+m) & -ml\cos(\theta) \\ -ml\cos(\theta) & ml^2 \end{pmatrix}, \;\; v(\theta,\dot{p},\dot{\theta}) = \begin{pmatrix} c\dot{p} + ml\sin(\theta)\dot{\theta}^2 \\ \gamma\dot{\theta} - mgl\sin(\theta) \end{pmatrix}.$$

Then the system can be described as

$$U(\theta)\begin{pmatrix} \ddot{p} \\ \ddot{\theta} \end{pmatrix} + v(\theta,\dot{p},\dot{\theta}) = \begin{pmatrix} F \\ 0 \end{pmatrix}.$$

Since the matrix $U(\theta)$ is always invertible (why?) this is the same as

$$\begin{pmatrix} \ddot{p} \\ \ddot{\theta} \end{pmatrix} = -U(\theta)^{-1}v(\theta,\dot{p},\dot{\theta}) + U(\theta)^{-1}\begin{pmatrix} F \\ 0 \end{pmatrix} = \begin{pmatrix} w_1(p,\theta,\dot{p},\dot{\theta},F) \\ w_2(p,\theta,\dot{p},\dot{\theta},F) \end{pmatrix}$$

with yet another abbreviation of the vector on the right-hand side.

Example: First-Order System Description

Finally introduce the states and the input

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} p \\ \theta \end{pmatrix}, \quad \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} \dot{p} \\ \dot{\theta} \end{pmatrix} \quad \text{and} \quad u = F.$$

The system is then described as

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \\ w_1(x_1, x_2, x_3, x_4, u) \\ w_2(x_1, x_2, x_3, x_4, u) \end{pmatrix} = \begin{pmatrix} f_1(x_1, x_2, x_3, x_4, u) \\ f_2(x_1, x_2, x_3, x_4, u) \\ f_3(x_1, x_2, x_3, x_4, u) \\ f_4(x_1, x_2, x_3, x_4, u) \end{pmatrix}.$$

The given derivation defines the function $f(x, u)$ via intermediate abbreviations. This is much less error-prone than explicit derivations as e.g. given in [AM] p.37, in particular for more complicated situations.

However, let us stress again we just obtained another representation of the system behavior as defined by the equations on slide 24.

Equilibria

All pairs of vectors $(x_e, u_e) \in \mathbb{R}^n \times \mathbb{R}^m$ which satisfy

$$\dot{x}_e = 0 = f(x_e, u_e)$$

are called **equilibria** of the system $\dot{x} = f(x, u)$.

If the system is driven with the constant control input $u(t) = u_e$ and if the state is initialized as $x_e(0) = x_e$, then the state-response is given by

$$x(t) = x_e \;\text{ and thus }\; \dot{x}_e(t) = 0 \;\text{ for all }\; t \geq t_0.$$

If starting at an equilibrium the state-trajectory stays there.

Equilibria are in general **not** unique. Not even for linear systems.

Often (but by no means always!) our interest is in trajectories that do not deviate too much from equilibria.

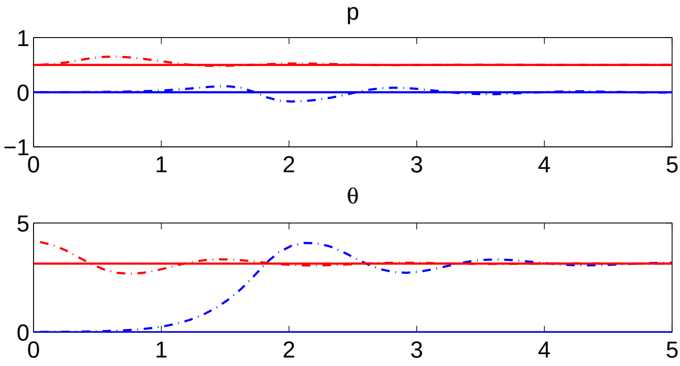
Example: Segway

The equilibria of the system on slide 24 are determined by

$$0 = F \quad \text{and} \quad mgl\sin(\theta) = 0.$$

Solutions are $\theta_e = 0$ (upright position) and $\theta_e = \pi$ (downright position), while p_e is arbitrary.

Four simulations, including perturbation of initial condition (dash-dotted):



Linearization

Recall that the first-order Taylor expansions of f and h are given by

$$\begin{aligned} f(x,u) &\approx f(x_e,u_e) + \underbrace{\partial_x f(x_e,u_e)}_{A}\underbrace{(x-x_e)}_{x_\Delta} + \underbrace{\partial_u f(x_e,u_e)}_{B}\underbrace{(u-u_e)}_{u_\Delta} \\ h(x,u) &\approx h(x_e,u_e) + \underbrace{\partial_x h(x_e,u_e)}_{C}\underbrace{(x-x_e)}_{x_\Delta} + \underbrace{\partial_u h(x_e,u_e)}_{D}\underbrace{(u-u_e)}_{u_\Delta}. \end{aligned}$$

These are good approximation if $x \approx x_e$ and $u \approx u_e$.

Suppose that $f(x_e,u_e) = 0$. The **linearization** of $\dot{x} = f(x,u)$, $y = h(x,u)$ at the equilibrium (x_e,u_e) is

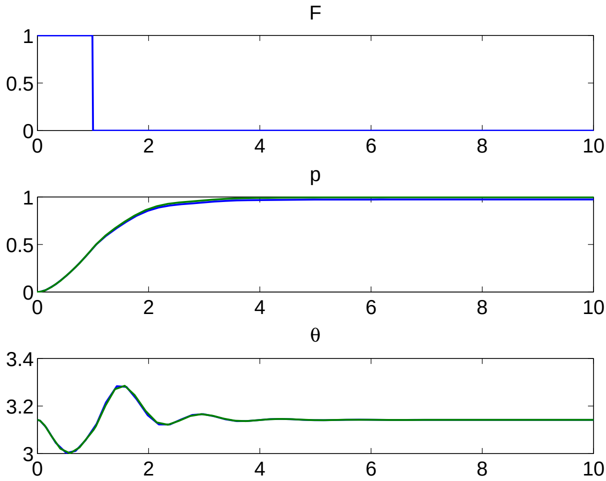
$$\dot{x}_\Delta = Ax_\Delta + Bu_\Delta, \quad y_\Delta = Cx_\Delta + Du_\Delta.$$

Let $u(t) \approx u_e$ and $x(0) \approx x_e$ have the nonlinear response $y(t) \approx y_e$.

If one drives the linearization with $u_\Delta(t) = u(t) - u_e$ and $x_\Delta(0) \approx 0$ then **it is our hope** that $y_\Delta(t) + y_e$ approximates $y(t)$.

Example: Segway

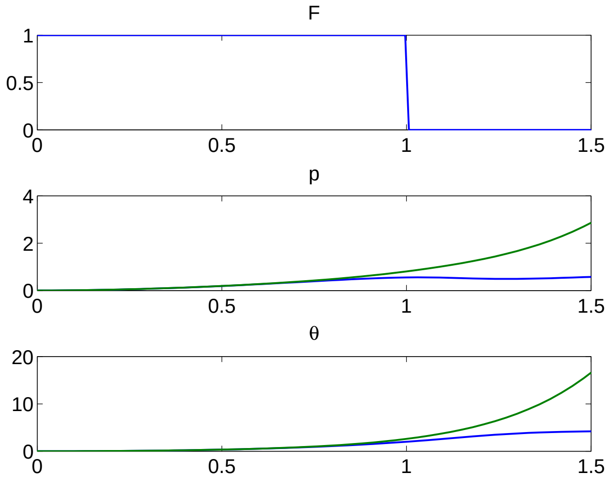
Nonlinear (blue) versus linearized (green) simulation at $(p_e, \theta_e) = (0, \pi)$:



Good match.

Example: Segway

Nonlinear (blue) versus linearized (green) simulation at $(p_e, \theta_e) = (0, 0)$:



Huge deviation in the long run.

System Interconnections and Modularity

One of the most important concepts in modeling dynamical systems is **modularity**. Complex models are built by interconnecting simple system components in a hierarchial manner.

Dynamical systems are **interconnected** by equating signals.

- Allows to use software libraries with standard system components and interfaces between different physical domains.
- Many concrete modeling packages are available (Matlab, Modelica).
- Even for complex models, the hierarchical (nested) structure renders bookkeeping manageable.
- Individual components can be adapted while preserving the overall interconnection structure.

Example: Mass-Spring-Damper System

The simple mass-spring-damper system $m\ddot{q}+c(\dot{q})+kq = u$ does actually result from Newton's laws as follows:

$$F = \dot{p} \ \text{ with } \ p = mv.$$

The total force acting on the mass (assumed to be constant) is

$$F = F_1 + F_2 + F_3$$

with the externally exerted force

$$F_1 = u,$$

the friction force (assumed to depend on velocity)

$$F_2 = c(v),$$

and the spring restoring force (assumed to depend linearly on elongation)

$$F_3 = -kq.$$

Example: Pendulum on Cart

The cart force is generated by the torque T of an electric motor.

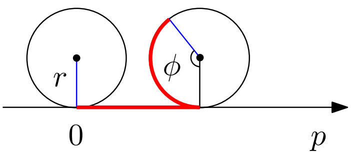
If the voltage V is applied to the motor, the winding current I generates a torque T (with constants k_e, k_m, L, R) according to

$$V - V_{\rm emf} = L\dot{I} + RI \quad \text{and} \quad T = k_m I.$$

The incurred rotation with speed $\dot{\phi}$ creates the back-emf $V_{\rm emf} = k_e\dot{\phi}$.

Cart-position and the angular position of the (mass-less) wheel are coupled as

$$p = r\phi.$$



We clearly also have $T = Fr$.

Interconnection equations: $V_{\rm emf} = k_e\dot{\phi}$, $p = r\phi$, $T = Fr$.

Example: Pendulum on Cart

We can eliminate the variables T, F, ϕ, $\dot{\phi}$ and obtain the following description of the interconnection:

$$\begin{aligned} L\dot{I} + RI + \frac{k_e}{r}\dot{p} &= V \\ (M+m)\ddot{p} - ml\cos(\theta)\ddot{\theta} + c\dot{p} + ml\sin(\theta)\dot{\theta}^2 - \frac{k_m}{r}I &= 0 \\ -ml\cos(\theta)\ddot{p} + ml^2\ddot{\theta} + \gamma\dot{\theta} - mgl\sin(\theta) &= 0. \end{aligned}$$

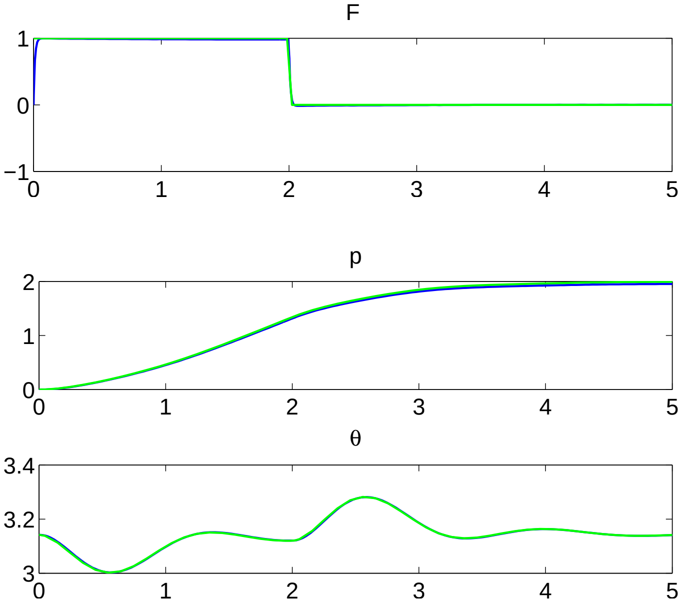
Note that the coupling is often called two-sided, since the dynamics of the interconnected system influence **each other**. If $k_e = 0$ then the coupling is one-sided.

The effect of the motor dynamics depends on the motor constants. It is not straightforward to predict the impact onto the dynamic responses.

Let us hence only analyze, by simulation, the influence of “large” and “small” values of L.

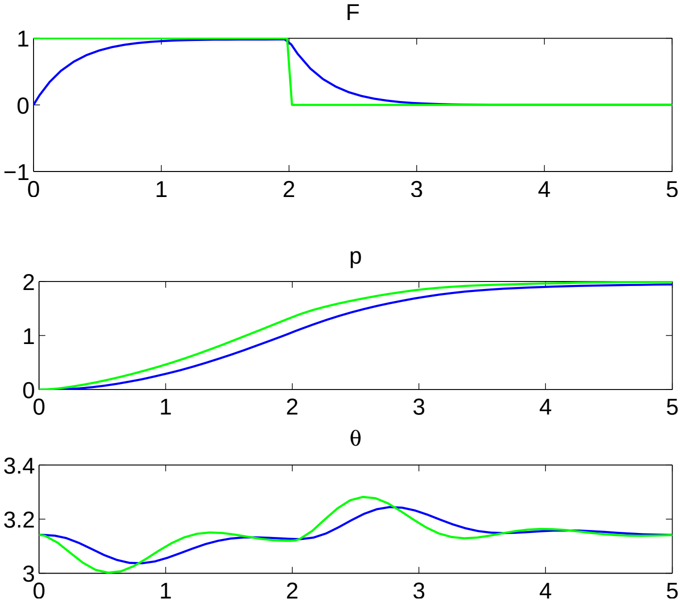
Example: Pendulum on Cart

Small L leads to fast motor dynamics:



Example: Pendulum on Cart

Larger L slows the motor dynamics down:



Series Interconnection

The **series interconnection** of the systems

$$\dot{x}_1 = f_1(x_1, u_1),\ y_1 = h_1(x_1, u_1),\ \ \dot{x}_2 = f_2(x_2, u_2),\ y_2 = h_2(x_2, u_2)$$

is obtained if the output of the first system enters the second system as an input: $u_2 = y_1$. (The vector dimensions must be equal.)

The **series interconnection** is then described by

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} f_1(x_1, u_1) \\ f_2(x_2, h_1(x_1, u_1)) \end{pmatrix}, \quad y_2 = h_2(x_2, h_1(x_1, u_1)).$$

Linear systems:

$$\begin{array}{rcl} \dot{x}_1 & = & A_1 x_1 + B_1 u_1, \\ y_1 & = & C_1 x_1 + D_1 u_1 \end{array} \quad \begin{array}{rcl} \dot{x}_2 & = & A_2 x_2 + B_2 u_2, \\ y_2 & = & C_2 x_2 + D_2 u_2 \end{array} \quad \xi = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\dot{\xi} = \begin{pmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{pmatrix} \xi + \begin{pmatrix} B_1 \\ B_2 D_1 \end{pmatrix} u_1,\ \ y_2 = \begin{pmatrix} D_2 C_1 & C_2 \end{pmatrix} \xi + D_2 D_1 u_1$$

Parallel Interconnection

The **parallel interconnection** of the systems

$$\dot{x}_1 = f_1(x_1, u_1),\ y_1 = h_1(x_1, u_1),\ \ \dot{x}_2 = f_2(x_2, u_2),\ y_2 = h_2(x_2, u_2)$$

is obtained by having them enter a common input and summing the outputs: $u_1 = u_2 = u$ and $y = y_1 + y_2$. (Dimensions must fit.)

The **parallel interconnection** is then described by

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} f_1(x_1, u) \\ f_2(x_2, u) \end{pmatrix}, \quad y = h_1(x_1, u) + h_2(x_2, u).$$

Linear systems:

$$\begin{array}{rcl} \dot{x}_1 & = & A_1 x_1 + B_1 u_1, \\ y_1 & = & C_1 x_1 + D_1 u_1 \end{array} \quad \begin{array}{rcl} \dot{x}_2 & = & A_2 x_2 + B_2 u_2, \\ y_2 & = & C_2 x_2 + D_2 u_2 \end{array} \quad \xi = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\dot{\xi} = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \xi + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u, \quad y = \begin{pmatrix} C_1 & C_2 \end{pmatrix} \xi + (D_1 + D_2) u$$

Matlab's Control System Toolbox

Matlab's Control System Toolbox has so-called `ss` objects which are used to work with linear state-space systems as follows:

- System definition: `sys1=ss(A1,B1,C1,D1)`, `sys2=ss(A2,B2,C2,D2)`
- Series interconnection: `ser=sys1*sys2`
- Parallel interconnection: `par=sys1+sys2`
- Simulation: `y=lsim(sys,u,T,x0)`
- Extraction of defining matrices: `[A,B,C,D]=ssdata(sys)`

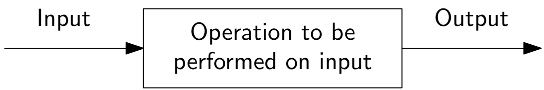
Note that the (overloaded) operations for the series and parallel interconnections remind us about the way how matrices act as mappings. Internally these operations are based on the above derived formulas.

You should be actively exploring the possibilities of the toolbox.

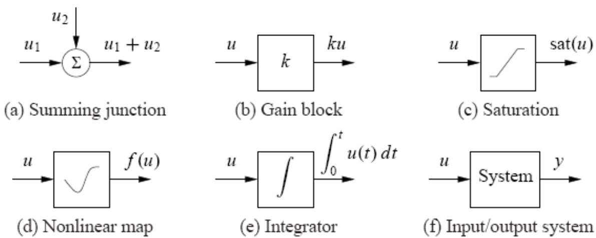
Block-Diagrams

The Simulink description is a particularly simple example of a **block-diagram** as used for visualization purposes in control.

A block one (vector-valued) input and output is indicated as



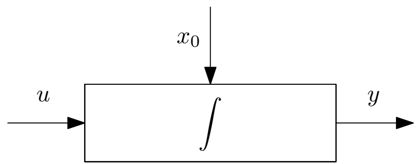
Static blocks are maps from $\mathbb{R}$ into $\mathbb{R}$ or from $\mathbb{R}^p$ into $\mathbb{R}^q$. **Dynamic blocks** map an input signal into an output signal. Examples:



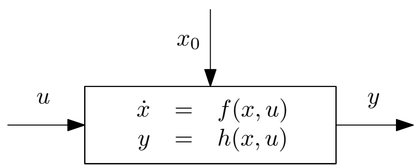
Block-Diagrams

Dynamic blocks described by differential equations require the specification of an **initial condition** such that the output is uniquely defined by the input. Such an initial condition can be viewed as an extra input.

That's why, for an integrator, one sometimes uses the notation



or, for a general system, the block description

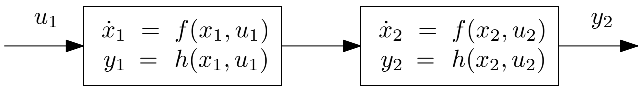


Block-Diagrams

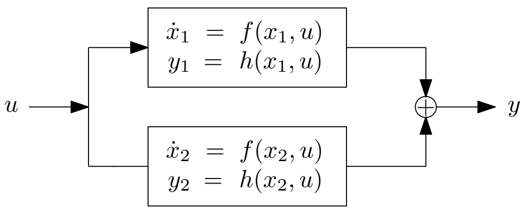
The **block-diagram** is obtained by interconnecting individual blocks (by equating signals). Hence such diagrams (should) always correspond in a precise fashion to algebraic equations.

Generic examples

Series interconnection:



Parallel interconnection:

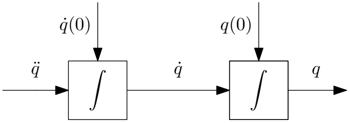


Example: Mass-Spring-Damper System

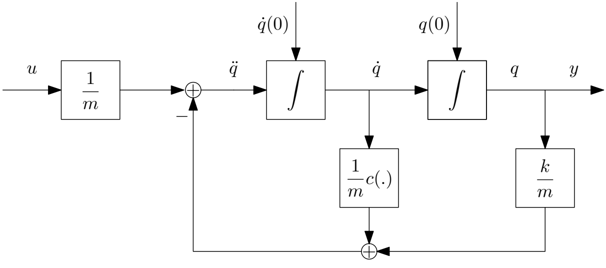
Differential equation and output equation:

$$\ddot{q} = \frac{1}{m}u - \frac{1}{m}c(\dot{q}) - \frac{k}{m}q, \quad y = q$$

Start from chain of integrators

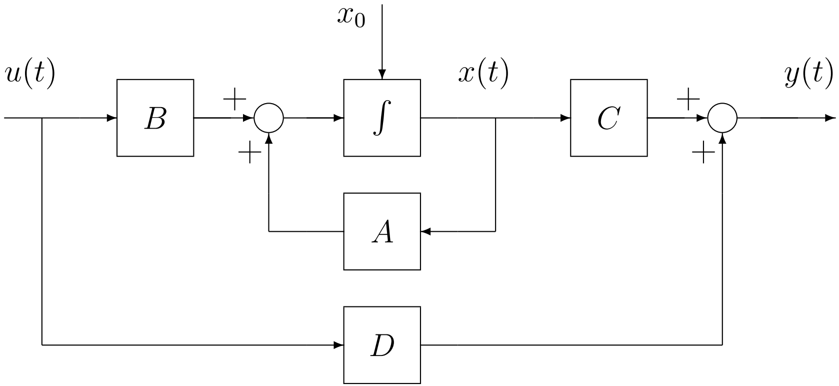


and obtain block-diagram with equations of motion:



Example: State-Space System

The standard linear state-space system can be depicted as



However, there is no need for such an explicit implementation in Simulink, since one can just use either a `State-Space` block (defined by A, B, C, D) or an `LTI Systems` block (defined by an `ss`-object).

Covered in Lecture 1

- Models of dynamical systems
  differential equations, inputs, outputs, state, reduction to first order, linear systems, compact notation, Simulink simulation
- Linearization
- System interconnections
  modularity in modeling, interconnection=equating signals
  feedback equations, relation to block diagrams
- Aspects of qualitative analysis of dynamic responses
  nonzero initial condition, external input signal, equilibria, stability, limit cycles, finite escape time