Introduction to modeling

Overview of mathematical models of dynamic systems

Zdeněk Hurák
September 20, 2020

In this introductory lecture we will discuss what we actually mean by modeling of dynamic systems in this course. Apparently, the topic is tremendously broad but keep in mind that our key ultimate motivation is to assist in a controller design. This will shape the whole course. We will see an overview of formats of mathematical models that we intend to use in the rest of the course. Some of them will be notoriosly known (state-space equations, transfer functions), some will be less familiar (differential-algebraic equations), multiports, bond graphs). This course has an ambition to teach you the skill (or even the art) of modeling of realistically complex systems, and in order to achieve it, it seems inevitable to follow the bottom-up approach, starting with simpler subsystems and components. We will see at the very beginning that following this reasonable approach turns out plain impossible when sticking to the widely popular the formalism of block diagrams (or signal flow models in the Simulink style). It will turn out that the interface between components and subsystems which describes exchange of energy rather then passing of signals is better suited for the component-based and multi-domain modeling. The powerful concept of energy-based modeling will be introduced in this introductory lecture and contrasted with signal-based modeling. The very key idea of the modeling methodology based on bond graphs will be introduced.

1 What do we mean by model and modeling?

Long introductory paragraphs can be written on the definitions of the concepts of system and model. We will skip these while directing a student to any of the many available texts (see the discussion of literature at the end of this text) and state (maybe too abruptly) that *system* is whatever you define it to be... Well, perhaps this was too abrupt, there are at least two atributes that must be specified:

- the *boundaries* of the system, which do not necessarily have to coincide with the physical borders of the object as in the case of flight dynamics where the flight is due to interaction of the aircraft and the bulk of air surrounding it,

- and the relevant *variables* (while studying flight dynamics, you are certainly not interested in knowing temperature distribution over the control surfaces).

In this course we are going to consider dynamic systems, that is, system which exhibit *inertia*, some sluggishness (not exactly a delay) in response to external stimuli.

Now, model of a system is our structured knowledge that helps forecast the behavior of the system. In our course we only consider mathematical models, that is, mathematical equations that relate the physical variables characterizing the system under study. Having specified that we are interested in dynamic systems, differential and/or algebraic equations seem to be the right tool for description of such systems, although these need not be explicitly written down but rather encoded in a graphical form or software code.

Towards the end of the course we are going to enter the discipline of *simulation* of dynamic systems. Suffice to say for now that it is a numerical experimentation with the mathematical model.

2 What is our motivation for modeling?

Modeling is a skill exercised by perhaps every engineer and scientist. The number of expert domains is huge, as well as the number of motivations of these experts. Starting our course, we need to identify our goal and keep it in mind throughout the course, otherwise we will be chasing everyone else's job and responsibility. Well then, we are control engineers and our ultimate goal is to design a control system. Hence, we only need a model for

1. approximate analysis of the system (to learn fundamental limitations of on achievable control performance)

2. design a control systems (feedback and/or feedforward loops, filters on measured variables, shapers for reference variables)

3. simulation of the response of the resulting system (original system plus our control system) to assess the performance.

Of course, in a well-managed project, control engineers have some interactions with designers of the system, therefore lessons learnt from some early control-centered analysis of mathematical models can induce some modification of the design (choice of more powerful motors, less noisy sensors, or even a complete redesign of mechanical configuration), but most often than not we are free from the design worries.

Furthemore, our models do not have to describe the reality perfectly! They only need to be just as accurate as we need to guarantee a requested closed-loop performance.

Models can also be used for other purposes than control design. Employing real-time simulation, models can be used for detection of some failure modes.

3 Classes of systems we are going to model

Although many of the concepts introduced in this course are quite versatile, we opt to focus the course on systems where the concept of *energy* makes sense. These include electronic, electrical, mechanical, electromechanical, hydraulic, magnetic, thermal, thermofluidic and perhaps some other technical systems.

Relying on the unifying concept of energy we could achieve even more then just step into the someone elses' domain of expertize, we could *connect* these domains within a single project. After all, these integration skills are typical for control engineers. To summarize, the modeling methods presented in this course will make it possible to model systems with these two attributes

- multidomain systems: mechanical, electrical, hydraulic, thermal, ...
- complex: composed of a large number of interacting subsystems and compoments.

To pick a single system perfectly demonstrating the above two atributes, consider the laboratory-scale pump storage power plant as located in our "Strojovna" lab with the schematics given in Fig. 1 and described in some detail in the corresponding section of the course website [4].

Apparently, the three major physical domains are included in this project: electrical (motors), mechanical (valves), hydraulic (pipes, tanks). Moreover, the system is no longer of a toy size.

4 Classes of systems we are not going to model

Unfortunately, many interesting systems encountered in the fields such as sociology, economy, and even biology are based on transfer of *information* rather then energy and we will not investigate these.

This is an itroductory course and inevitably we had to skip many modeling concepts. One of the key concepts that we do not cover at all is that of stochastic systems and their models. Talking about these would first need solid basis in probability and random processes, for which we do not have space in our course.

5 Modeling approaches presented in our course

As energy is the unifying phenomenon across physical domains and systems, our methods will be based on it in one way or another. We will study three

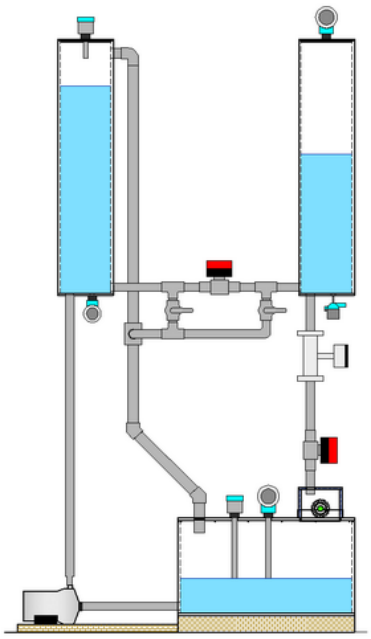


Figure 1: Laboratory-scale pump storage power plant as located in our "Strojovna" lab.

classes of methods. Studying them deeper, these could be eventually shown to be just different facets of the same basic modeling philosophy.

1. graphical technique of *bond graphs*,
2. software technique of *object-oriented modeling* (also equation-based modeling), in particular the popular Modelica language (or recently also Simscape language by The Mathworks),
3. analytical techniques of Lagrange and Hamilton.

6 Modeling approaches not presented in our course

Related to stochastic systems is the discipline of *systems identification*, which can also be regarded as a modeling methodology. Based on the collected measurements of the inputs and outputs, a mathematical mode is build which best relates the two data sets. We are not going to study this either. Besides the already expressed capacity arguments, system identification represents a methodology which is in some sense orthogonal to our approach. It is certainly wise to regard system identification as a complementary technique, not the key line of attack.

7 Mathematical models of dynamic systems

7.1 Causal models (also assignment-based, signal-based)

We start the survey of formats of mathematical models of dynamic systems by a couple of fairly popular formats which we collectively label as "causal" or "signal-based" or "assignment-based". It will only be explained later in this lecture what we mean by these adjectives.

7.1.1 State equations

Our focus in this course is on dynamical systems, hence differential equations are the key mathematical vehicle here. Among these, the first-order explicit ordinary differential equations are certainly the most studied and you have surely encountered these under the name of a *state equations*. These equations are generally nonlinear and possibly also time varying

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t), u(t), t). \tag{1}$$

Recall that in order to specify a unique solution, the initial conditions, that is, the value of the state variable $x(t)$ at the initional time t_0 needs to be specified

$$x(t_0) = x_0. \tag{2}$$

This can also represent a situation in which $x(t)$ denotes an n-tuple of variables, that is, $x(t) \in \mathbb{R}^n$, $u(t)$ is an m-tuple of inputs, that is, $u \in \mathbb{R}^m$, and f is a vector-valued function. It is a good practice to use a bold face for vector variables and therefore we write

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t). \tag{3}$$

The $x(t)$ variable is called *state* variable. This name reflects the fact that in order to predict the system's response to some external stimulus $u(t)$, we only need to know the value of $x(t)$ right at the moment t_0 of application of the external signal. No past values are needed to determine the response of the system uniquely.

Speaking about existence and uniqueness of a solution, we omit here a lengthy discussion of the underlying mathematical issues, which were certainly included in some previous mathematical course. In a few words, the $f()$ function above needs to be sufficiently "smooth", namely, it must satisfy so-called Lipschitz conditions with respect to x. Suffice to say that for physically motivated equations the existence and uniqueness of the solution are automatically guaranteed, at least on a finite time inteval (see *finite escape time*).

The current value of $x(t_0)$ now includes all the information to determine the future values of $x(t)$ in presence of the input $u()$ (on the interval $[t_0, t]$). Compare with, say, the second-order differential equation expressing Newton's law

$$m\ddot{x}(t) = F(t). \tag{4}$$

In order the predict how the trajectory $x(t)$ of the system evolves in time in response to a change in the force $F(t)$ applied at some time t_0, knowledge of the initial position $x(t_0)$ is not enough! We also need to know the value of its derivative at the initial time, that is, the initial velocity $\dot{x}(t_0)$. It is generally possible to transform (we say "realize") a given differential equation into a state-space format. We can do this by introduction a new state variable—velocity in our example:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}_{\boldsymbol{x(t)}} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \underbrace{F(t)}_{u(t)}. \tag{5}$$

The state equation is often accompanied by the *output equation*, which in the vector form is

$$\boldsymbol{y}(t) = \mathbf{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \tag{6}$$

which only shows how some of the state variables are scaled and combined in order to create the output variables. For instance, the output variables can include just some subset of physically measurable state variables. You are certainly already familiar with the linear(ized) version

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \mathbf{A}(t)\boldsymbol{x}(t) + \mathbf{B}(t)\boldsymbol{u}(t), \\ \boldsymbol{y}(t) &= \mathbf{C}(t)\boldsymbol{x}(t) + \mathbf{D}(t)\boldsymbol{u}(t). \end{aligned} \tag{7}$$

Most often the matrices involved in the model were independent of time

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t), \\ \boldsymbol{y}(t) &= \mathbf{C}\boldsymbol{x}(t) + \mathbf{D}\boldsymbol{u}(t). \end{aligned} \tag{8}$$

in which case we talk about an LTI (linear time invariant) system.

Frequently uses is the following compact way of writing down a state-space LTI model (named somehow, say, $\boldsymbol{G}$)

$$\boldsymbol{G} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right]. \tag{9}$$

Make sure you do understand, what makes a linear system linear... Make sure you are familiar with the relationship between the size of the matrices and the dimension of the state vector and number of input and output variables.

Before we show an example of a state-space model, we first introduce another format of a model, which generalizes the state-space model.

7.1.2 DAE models, also singular or descriptor models

Strictly speaking, the state-space model formed by (1) and 6) is a combination of a differential equation and some nonlinear equation. But in a standard state-space model the solution of the state equation does not depend on the output equation at all. However, this situation can be generalized in a way that solution of the differential equation does depend on the solution of the nondifferential (algebraic) equation. Such equations are called differential-algebraic equations (DAE). Generally,

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \tag{10}$$

$$\mathbf{0} = \mathbf{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{11}$$

There is a caveat here. Now the initial value $\boldsymbol{x}(t_0)$ needs to be consistent with the output equation! In other words, the set of variables $\boldsymbol{x}(t)$ is now redundant. That is why $\boldsymbol{x}$ is called a *semi-state* variable.

The model could be made even more general by considering implicit differential equation.

$$\mathbf{0} = \mathbf{F}\left(\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t}, \boldsymbol{x}(t), \boldsymbol{u}(t), t\right), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{12}$$

The model can, however, be also found in a more restricted form of a *descriptor system*

$$\mathbf{E}\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t), \tag{13}$$

$$\boldsymbol{y}(t) = \mathbf{C}\boldsymbol{x}(t) + \mathbf{D}\boldsymbol{u}(t), \tag{14}$$

where the matrix $\mathbf{E}$ is singular (or close to singular). If it were not, one could easily get the standard state space model just by multiplying both sides of the differential equation by the inverse of $\mathbf{E}$.

To demonstrate the concepts, let us explore one example of an electronic circuit in Fig.2.

The first five equations are just models of the individual components, the capacitor and the inductor described in the format displaying the derivatives of state variables:

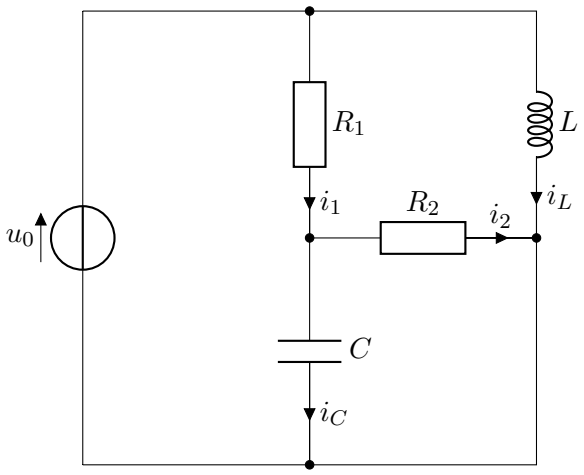


Figure 2: Circuit diagram of an electrical RLC circuit.

$$u_o = 10, \tag{15}$$
$$u_1(t) - R_1 i_1(t) = 0, \tag{16}$$
$$u_2(t) - R_2 i_2(t) = 0, \tag{17}$$
$$i_C(t) - C\frac{\mathrm{d}u_C}{\mathrm{d}t}(t) = 0, \tag{18}$$
$$u_L(t) - L\frac{\mathrm{d}i_L}{\mathrm{d}t}(t) = 0. \tag{19}$$

The other three equations come out from a direct application of Kirchhoff's Voltage Law:

$$u_0 - u_1(t) - u_C(t) = 0, \tag{20}$$
$$u_L(t) - u_1(t) - u_2(t) = 0, \tag{21}$$
$$u_C(t) - u_2(t) = 0. \tag{22}$$

The remaining equations are obtained by application of Kirchhoff's Current Law. There are three nodes, but we only need two equations (to have 10 equations in total for 10 variables). The explanation is that the currents through the three nodes are not completely independent, in other words, one of the three equations is necessarily redundant (linearly dependent) and need not be considered. It is customary to regard the ground node as the redundant guy. The remaining equations are thus

$$i_0 - i_1(t) - i_L(t) = 0, \tag{23}$$
$$i_1(t) - i_2(t) - i_C(t) = 0. \tag{24}$$

What we obtained is a set of differential and algebraic equations, that is, an implicit DAE model. In principle we could stop here since there are both analytical and numerical tools to handle these as the Matlab code below demonstrates.

```
L = 0.0015;       % [H]
C = 1e-6;         % [F]
R1 = 100;         % [Ohm]
R2 = 20;          % [Ohm]
U0 = 10;          % [V]

E = zeros(9,9);
E(1,8) = C;
E(2,7) = L;

A = zeros(9,9);
A(1,9) = 1;
A(2,6) = 1;
A(3,2) = 1;
A(3,3) = -R1;
A(4,4) = 1;
A(4,5) = -R2;
A(5,2) = -1;
A(5,8) = -1;
A(6,2) = -1;
A(6,4) = -1;
A(6,6) = 1;
A(7,4) = -1;
A(7,8) = 1;
A(8,1) = 1;
A(8,3) = -1;
A(8,7) = -1;
A(9,3) = 1;
A(9,5) = -1;
A(9,9) = -1;

B = zeros(9,1);
B(5,1) = 1;

C = zeros(1,9);
C(1,5) = 1;

D = 0;

Gdae = dss(A,B,C,D,E);
set(Gdae,'InputName','u_0')
set(Gdae,'OutputName','i_2')
set(Gdae,'StateName',{'i_0','u_1','i_1','u_2','i_2',...
'u_L','i_L','u_c','i_c'})
```

The `dss()` constructor is used for this purpose. Note also that it is a good practice to label the inputs, outputs and the state variables.

We may want to turn this DAE model into the more familiar state-space format, if possible. In order to do that, the equations just need to be modified and reordered so that their left-hand sides can be immediately computed using the right-hand sides. Substituting from one equation to another we finally eliminate all but two variables—the state variables u_C and i_L, which define our desired state-space model. We will not discuss any systematic procedure

for such conversion in this introductory lecture since this will be a topic for one of the future lectures (the one on simulation of DAE systems). Instead, we will state the model directly here

$$\begin{bmatrix} \frac{\mathrm{d}u_C(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}i_L(t)}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} -\frac{R_1+R_2}{R_1R_2+C} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_C(t) \\ i_L(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1C} \\ \frac{1}{L} \end{bmatrix} u_0 \tag{25}$$

$$i_2(t) = \begin{bmatrix} \frac{1}{R_2} & 0 \end{bmatrix} \begin{bmatrix} u_C(t) \\ i_L(t) \end{bmatrix} + 0u_0. \tag{26}$$

Our state-space model can be easily encoded in Matlab using the `ss()` constructor:

```
A = [-(R1+R2)/(R1*R2*C), 0; 0, 0];
B = [1/(R1*C); 1/L];
C = [1/R2, 0];
D = 0;

Gss = ss(A,B,C,D);

set(Gss,'InputName','u_o')
set(Gss,'OutputName','i_2')
set(Gss,'StateName',{'u_C','i_L'})
```

Whether using one or the other model, you can now start analyzing the responses to input signal, stability and so on. You will learn soon during such experiments that the set of tools for state-space models is much wider then for DAEs.

We will have more to say on software in one of the next lectures.

7.1.3 Second-order model

Whereas the electrical circuits discussed above are very conveniently described by first-order differential equations, the mechanical systems are very naturally described by second-order differential equations. Recall that the second Newton's law determines the second derivative of position. Although these second-order equations could be in principle converted to first-order state-space models, it is sometimes preferrable not to do so. There may be two reasons. First, numerical issues. Second, the structure is then lost.

Consider a mechanical system consisting of multiple masses interconnected by springs and dampers. The natural model describing deviations x_i of the individual masses (indexed by i) from their equilibrium positions is

$$\mathbf{M}\ddot{\boldsymbol{x}}(t) + \mathbf{D}\dot{\boldsymbol{x}}(t) + \mathbf{K}\boldsymbol{x}(t) = \mathbf{B}\boldsymbol{u}(t), \tag{27}$$

where $\boldsymbol{x}(t) = \left[x_1(t), x_2(t), \ldots, x_n(t)\right]^{\mathrm{T}}$ is no longer a state vector but the vector consisting only of the positional variables (no velocities), the matrix $\mathbf{M}$ is called a generalized mass matrix, $\mathbf{D}$ is a damping matrix and $\mathbf{K}$ is a stifness

matrix. The vector input $\boldsymbol{u}$ corresponds to forces applied to individual masses. This model can possibly be accompanied by the output equation

$$\boldsymbol{y} = \mathbf{C}_x \boldsymbol{x}(t) + \mathbf{C}_v \dot{\boldsymbol{x}}(t). \tag{28}$$

In order to verify our mastering of the concepts, let us consider an example as in Fig. 3

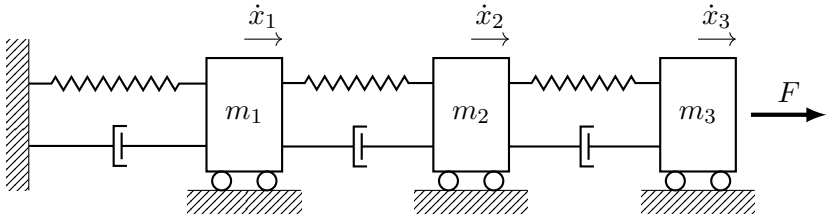


Figure 3: Multiple masses, springs and dampers. Could be a real system or just an approximation to a flexible structure.

The Newton equations for the three masses are

$$m_1\ddot{x}_1(t) = -k_1x_1(t) - d_1\dot{x}_1(t) + k_2(x_2(t) - x_1(t)) + d_2(\dot{x}_2(t) - \dot{x}_1(t)), \tag{29}$$

$$m_2\ddot{x}_2(t) = -k_2(x_2(t) - x_1(t)) - d_2(\dot{x}_2(t) - \dot{x}_1(t)) + k_3(x_3(t) - x_2(t)) + d_3(\dot{x}_3(t) - \dot{x}_2(t)), \tag{30}$$

$$m_3\ddot{x}_3(t) = -k_3(x_3(t) - x_2(t)) - d_3(\dot{x}_3(t) - \dot{x}_2(t)) + F(t). \tag{31}$$

where the ks are the stiffness constants for the three springs and ds are the coefficients for the three dampers. We can immediately rewrite this into the format of M, D and K matrices

$$\begin{bmatrix} m_1 & & \\ & m_2 & \\ & & m_3 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \end{bmatrix} + \begin{bmatrix} (k_1+k_2) & -k_2 & \\ -k_2 & (k_2+k_3) & -k_3 \\ & -k_3 & k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} (d_1+d_2) & -d_2 & \\ -d_2 & (d_2+d_3) & -d_3 \\ & -d_3 & d_3 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \\ 1 \end{bmatrix} F(t). \tag{32}$$

The procedure for conversion to state-space model is fairly straightforward and students are asked to check they are able to reproduce it:

$$\begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \ddot{\boldsymbol{x}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \dot{\boldsymbol{x}}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{B} \end{bmatrix} u(t). \tag{33}$$

Observing that the process of creating an equivalent state-space description includes numerical inversion of the mass matrix, it is then easily understood that numerical troubles might be expected for large-scale systems. Similarly,

the structure enjoyed by the matrices $\mathbf{M}$, $\mathbf{D}$ and $\mathbf{K}$ does not easily transform to the structure of the corresponding state-space matrices. We will not elaborate more on this in our course. This was just to inform you that there are reasons for keeping the models in the format of second-order differential equations. The analysis and control synthesis techniques have been developed for these models.

However artificial these systems consisting of multiple massess, springs and dampers might look, they actually consitute a reasonable model of fairly common mechanical systems—flexible mechanical structures. Noble examples are flexible cantilevers for atomic force microsopes (AFM) or flexible wing aircraft. One approach how to model these systems is to discretize them spatially, which leads exactly to the model described above. The dimensions of the coresponding matrices can be, however, huge. Hundreds or easily hundred thousands...

Another instance of second-order systems may be found in the field of robotics. The matrices are then dependend on the the state variables. Now, instead of x, a different symbol is used — q, which makes it possible to include both translational and rotational coordinates. q_is are called generalized coordinates. We will have more to say on this later. The second-order model for a robotic manipular is frequently found in the format

$$\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}(t) + \mathbf{D}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}(t) + \mathbf{K}(\boldsymbol{q}) = \mathbf{B}\boldsymbol{u}(t), \tag{34}$$

More on this later in this course when we are exploring the classical analytical techniques known under the name of Lagrangian modeling.

7.1.4 Systems of n-th order, polynomial matrix fractions

We started with first-order systems, and then proceeded with second-order systems. It is obvious that n-th order systems (for $n > 2$) must be mentioned as well. There are perhaps not many analytical tools for general nonlinear differential equations of order n but the situation is different in the linear case.

Denote as usual the independent (input) variable by u and the dependent (output) variable as y. The standard linear ordinary differential equation of order n is

$$\begin{aligned} a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \ldots + a_1 \dot{y}(t) + a_0 y(t) \\ = b_m u^{(m)}(t) + b_{m-1} u^{(m-1)}(t) + \ldots + b_1 \dot{u}(t) + b_0 u(t). \end{aligned} \tag{35}$$

Applying Laplace transform while assuming zero initial conditions, the differential equation is transformed into an algebraic equation, in particular a polynomial equation

$$(a_n s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0) Y(s) = (b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 + b_0) U(s). \tag{36}$$

The last step to express how $Y(s)$ depends on $U(s)$ is to perform a *formal* inversion of the polynomial $a(s)$. The intended inversion is indeed just formal, resulting in the popular fraction of two polynomials, well known as *transfer function*

$$Y(s) = \underbrace{\frac{(b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 + b_0)}{(a_n s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0)}}_{G(s)} U(s). \tag{37}$$

In Matlab, the constructor `tf()` can be used to create an object of class the transfer function.

This last step easily extends to the scenario with several inputs and several outputs, where several n-the linear differential equations are available. Applying Laplace transform, the set of ODEs is transformed into a set of polynomial equations. These can be written in a matrix format

$$\mathbf{A}(s)\boldsymbol{Y}(s) = \mathbf{B}(s)\boldsymbol{U}(s), \tag{38}$$

where $\boldsymbol{Y}(s)$ and $\boldsymbol{U}(s)$ are vectors of the transformed outputs and inputs, respectively, and $\mathbf{A}(s)$ and $\mathbf{B}(s)$ are now matrix polynomials (or polynomial matrices).

Now, what constitutes the right analogy to the transfer function in the MIMO case? A matrix of transfer function? Actually no. The natural extension is just a polynomial matrix fraction. That is, a fraction of two polynomial matrices

$$\boldsymbol{Y}(s) = \underbrace{\mathbf{A}^{-1}(s)\mathbf{B}(s)}_{\mathbf{G}(s)} \boldsymbol{U}(s). \tag{39}$$

Note that as matrices are not commutative, it matters whether the multiplication by the inverse of $\mathbf{A}(s)$ is done from the left or from the right. In the above, the model is in the form of a left matrix fraction. More on this in the graduate course on Linear course given currently by Petr Hušek.

We would like to mention here that this particular approach to linear systems based on fractions of polynomials and polynomial matrices has major part of its roots in the research of professor Vladimír Kučera and professor Michael Šebek and a number of their colleagues at Academy of Sciences of the Czech Republic. As a product of this research, Polynomial toolbox was created, which contains a set of functions for polynomial matrices. Entering a polynomial matrix is as easy as writing

```
>> A = [s^2+1, s; 5*s^3+6*s^2+3, 1]

A =

     1 + s^2              s
     3 + 6s^2 + 5s^3      1
```

7.1.5 Spatially distributed systems—partial differential equations

The modeling methodology introduced in this course will primarily focus on *lumped systems*. We can hardly feed our ambitions to step systematically into the discipline of modeling spatially distributed systems within this course. This would force us to study the mathematically advanced topic of *partial differential equations* (PDE), which is a broad subject of study on its own. Nonetheless, we will be able to use our tools for lumped systems to get approximate (typically low-order) models for spatially distributed systems.

As an example, consider the standard *transmission line* model (probably you have discussed this in some previous course on electric circuits). The full PDE model is

$$\frac{\partial u(x,t)}{\partial x} = -Ri(x,t) - L\frac{\partial i(x,t)}{\partial t}, \tag{40}$$

$$\frac{\partial i(x,t)}{\partial x} = -Gu(x,t) - C\frac{\partial u(x,t)}{\partial t}, \tag{41}$$

where R, L, G and C are resistance, inductance, admittance and capacitance, respectively, related to a unit lenght of the transmission line.

In order to analyze the system, one would have to work with the two PDEs, arriving to the well-known wave equation after a while. We can, however, get a very useful qualitative results (and usable quantitative outcomes as well) if we approximate the continuum model by the one in Fig. 4. For that we could use the concepts presented in this course.

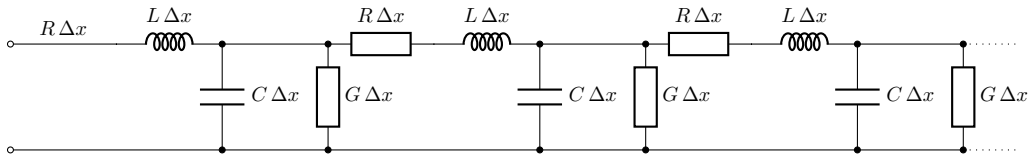


Figure 4: A few sections for a finite-dimensional model of a transmission line.

7.2 Acausal models (also equation-based, energy-based)

Why did we call the formats in the previous section as *causal* or *signal-based* or *assignment based*? Because we got stuck to an input-output viewpoint, whether we wanted it or no: one variable (possibly a vector variable) is an

input to the system and another variable (or a set of variables) represents a response, which we call an output. Remember that when we write

$$y(t) = g(x, u, t), \tag{42}$$

we actually mean (using the computer programming style)

$$y(t) := g(x, u, t), \tag{43}$$

or even better

$$y(t) \leftarrow g(x, u, t). \tag{44}$$

That is, take the inputs (and possibly the states if there are any), do some manipulation with them and *assign* them to the output. Hence, assignment-based modeling. Equivalently, in a graphical format of popular block diagrams (Simulink style)

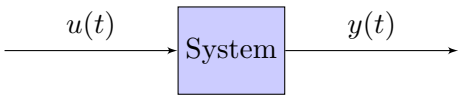


Figure 5: *Signal flow* model of a system (aka *block diagram*).

Why do we need to study anything else? Or do we? Well, the major shortcoming of assignment-based (or signal-based) models is that they do not allow building a model of a more complex system by interconnection of subsystems and components. Consider just a simple DC rotary motor with a permanent magnet. Finding a model is a well-mastered task. Then consider a mechanical load on the motor shaft, its mechanical model relating the input torque and the velocity is even simpler. Say, both models are in Simulink format. Now connect the two models together. Is it possible to obtain something like Fig.6 suggests?

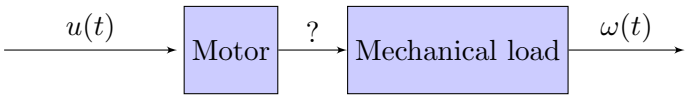
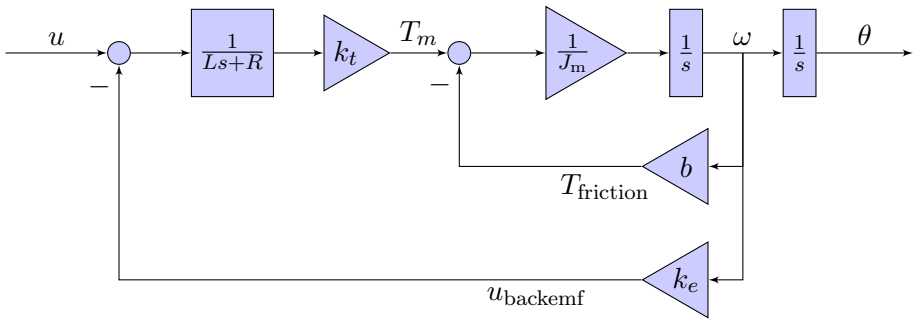


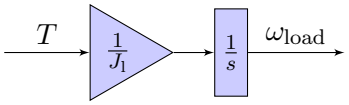
Figure 6: Block diagram for an interconnection of a motor and a load.

The situation is detailed in Fig. 7, where the motor and the mechanical motor are modeled using block diagrams in Simulink style.

The answer to the question preceeding Fig. 6 is negative! Apparently, unless the moment of inertia is negligible, the load will also influence the angular rate of the motor. Obviously we would expect some kind of a mutual coupling of the two blocks. But how do we represent this coupling? In this particular case we know that the resulting model inherits the structure of the model of the motor but instead of J_m we need to consider the total load as "seen" by the motor, that is $J_\mathrm{m} + J_\mathrm{l}$.



(a) Block diagram of a DC motor with a permanent magnet.



(b) Block diagram of a rotational load.

Figure 7: Block diagrams for a DC motor with permanent magnet and a load.

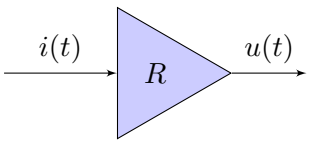
Another viewpoint at the source of the troubles here is that with block diagrams (signal flows) we have to specify the inputs and the outputs first, and only then we can start building the model. This has a major consequence on our capability or actually incapability to create libraries of basic elements. Consider just a single resistor. What is its mathematical model? The key phenomenon is that a resistor delivers a linear relationship between the voltage and the current due to Ohm's law. We can write

$$u(t) = Ri(t) \tag{45}$$

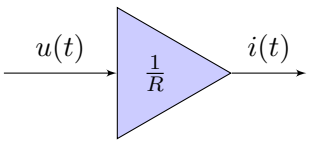
but we can also equally well write

$$i(t) = \frac{1}{R}u(t). \tag{46}$$

Which one of the two models is correct? Note that both will have different implementation in Simulink as the Fig. 8 shows.



(a) Current as the input to the model.



(b) Voltage as the input to the model.

Figure 8: Two different models of a resistor, depending on the causality.

Of course, both models are correct, nonetheless, a choice of the model depends on the circumstances—whether it is the current or the voltage that is enforce on the resistor. We had to specify the cause and the consequence, hence a causal model.

7.2.1 Bond graphs

Let's finally start exploring the alternatives. The solution to avoid the troubles induced by causality is to *avoid specifying the inputs and outputs*, hence *acausal modeling*. Equivalently, the interactions among the models will be realized not via signals but via energy exchange. Hence our model of a motor and a load should look like in Fig. 9

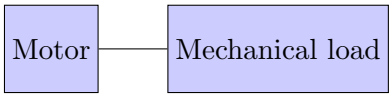


Figure 9: Exchange of energy between a motor and a load.

The link between the two subsystems must now be interpretted as a (bidirectional) exchange of energy between the two systems, flow of energy, power. We call it a power bond and the above diagram is our first instance of a *bond graph*. We can immediately demonstrate its convenience by extending the system with a model of a source of (electrical) energy, say, the battery, see Fig. 10

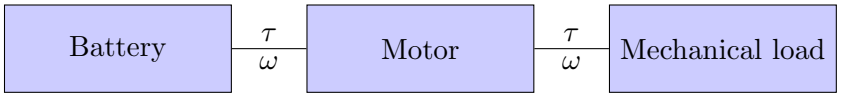
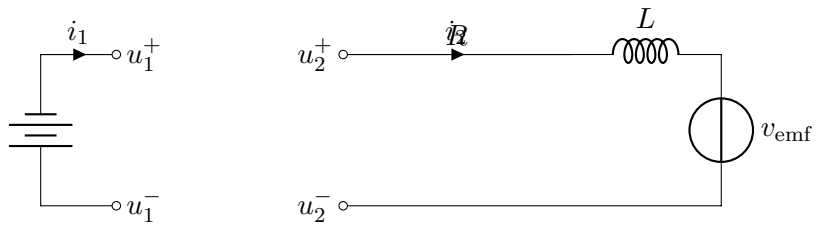


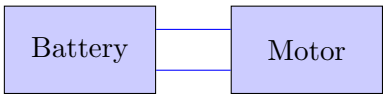
Figure 10: Bond graph for a motor and a load and a battery.

A noteworthy feature of the above model is that the two interfaces above are realized in two different physical domans—electrical and mechanical, and yet both of them can be uniformly quantitavely characterized in the same units—Watts! Energy (or power) seems to be a perfect universal currency. All our modeling in this course will be aimed at developing techniques for depicting the flow of energy in the system.

Let's stay for a while with the electrical bond (the one between the battery and the motor). As aspiring members of electrical engineering community we have already encountered another formalism for expressing this interaction—the one-port, two-port and in general *multiport* framework, see Fig. 11a. Both the batery and the motor are described here as one-ports. Each port is characterized by a voltage across the two pins and the current entering through one pin and leaving through the other. The interconnection of the two one-pins then amounts to connecting the corresponding pins as in Fig. 11b.



(a) A battery and a motor as one-ports. Each port characterized by a voltage across the pins and the current entering one pin and leaving the other.



(b) Interconnection of a battery and a motor as an interconnection of two one-ports. The two lines need to be literally interpretted as wires; the power bond is characterized by the voltage across the two wires and current flowing through one corrent from one system to another and back through the other wire.

Figure 11: Electrical interconnection of a battery and a motor.

As a matter of fact, the (multi)port formalism is what we are after. In other physical domains, however, there are no pins, no plus and minus leads. Therefore, graphically we will depict the existence of an exchange of energy by a single undirected line.

Taking further inspiration from the electrical engineering, the product of the two variables that characterized the port—the voltage $v = v^+ - v^-$ and the current i—yields a power $\mathcal{P}$ (in W). Can we find a similar pair of variables in mechanical domain? Yes. In the rotational mechanical domain the two variables are *torque* τ (in N m) and *angular velocity* ω (in $\mathrm{rad\,s^{-1}}$). You can verify that the product of the two variables indeed yields a power. We will keep track of these two variables associated with a bond by writing their names on the two sides of the bar denoting the bond as in Fig. 12

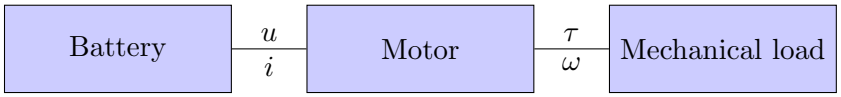


Figure 12: Bond graph for an interconnection motor and a load with the labels for the two associated variables.

Table 1: Analogy of variables for bond graphs (Maxwell).

Physical domain	Generalized force (or effort) e or gen. effort	Gen. velocity $\dot{q}$ or gen. flow f
Mech. rotation	Torque τ [N m]	Angular rate ω [rad s^{-1}]
Mech. translation	Force F [N]	Velocity v [m s^{-1}]
Electrical	Voltage u [V]	Current i [A]
Hydraulic	Pressure P [Pa]	Volumetric flow Q [m^3 s^{-1}]
Thermal	Absolute temperature θ [K]	Entropy flux $\dot{S}$ [J s^{-1} K^{-1}]
Thermal flow	Specific enthalpy h [J kg^{-1}]	Mass flow $\dot{m}$ [kg s^{-1}]

7.2.2 Maxwell's analogy

The pairs of variables for these two and a few more physical domains that we will visit in this course are summarized in Tab. 1.

What was the key used to classify the voltage as analogous to force, that is, as an instance of a generalized effort? And how come that electric current is declared as analogous to angular velocity, that is, an instance of a generalized velocity (also flow)? As a matter of fact, the analogy that we are going to use (called Maxwell's analogy) is just one possible way but a very clever one. Consider now a general bond graph with two subsystems in Fig. 13.

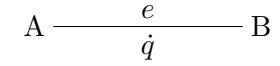


Figure 13: Simple bond graph depicting an energy exchange between subsystems A and B.

The e variable classifies as a *generalized force* (also *generalized effort*) as it is seen identically from the perspective of both A and B subsystems. The variable $\dot{q}$ (or f) classifies as a *generalized velocity* (also *generalized flow*) if it depends on whether it is viewed from the perspective of A or B. Hence, the property used to classify the variables into generalized efforts and generalized velocities (or flows) is *symmetry*. As an example, consider a twin wire connecting two electrical components: voltage is identical from the perspective of both components. The current is viewed as entering one component and leaving the other. Similarly, the rope connecting two movable masses, say, two humans playing the *tug-of-war* game: when one (or both) competitors are pulling, both persons feel the same force effect in their palms. However, while one person is moving backwards (the winner), the other one is moving forwards (the looser), hence the assymetry in velocities. Some more discussion can be found in [1].

Maxwell classification even makes nice correspondence with a very fundamental classification of variables into *extensive* and *intensive* as done in thermodynamics. This further encourages us to pay attention to energy as

the central object in modeling.

Note that for thermals system, a bit more convenient choice of variables is the (absolute) temperature T [K] and the heat flow rate $\dot{q}$ [W], which, however, do not yield the power as their product! Hence, the resulting model is not exactly a bond graph but rather a pseudo-bond graph. More on this later.

In order to make sure that we got the concepts right, let us consider another example: diesel motor with a turbocharger. The three major subsystems are the motor, the compressor and the turbine. The bond graph is in Fig. 14.

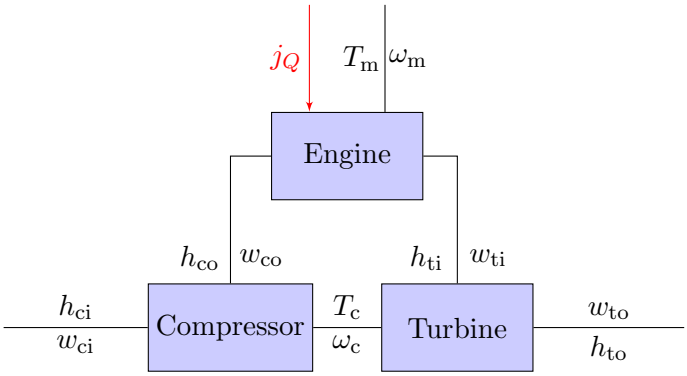


Figure 14: Bond graph for a turbocharger.

The description of the function can be found elsewhere, here we focus on the energy interactions. The diesel motor has three conventional power bonds (or ports). It exchanges a mechanical power with the shaft (the torque T_m and the angular velocity ω_m). It also exchanges power with the compressor through its air intake (the enthalpy h_co and the mass flow w_co) and with the turbine through the exhaust pipe (the enthalpy h_ti and the mass flow w_ti). Finally, there is also an energy income from the chemical energy of the fuel. This is not exactly a power bond with any outside subsystem since this is not a two-directional interaction. Similar analysis can be done for the remaining two subsystems, the names of the constituting variables are self-explaining.

7.2.3 Across-through analogy

A competitor to the Maxwell's analogy is the so-called *across-through* analogy. Sooner or later you will encounter it because it is quite popular, especially in the domain of electromechanical systems. Its major weakness is that it cannot be extended beyond these two physical domains. The variables are classified according to the way how they can be measured. Voltage and velocities can be measured *across* two pins or flanges, respectively. Current and force are classified as analogous since in order to measure them, the sensor must somehow be in the path of the energy, hence the name *through* variable. Disregarding the vagueness of these concepts (for instance, velocity can also be measured

by integrating the acelleration, current can also be measured by Hall's sensor, which is not in the path), there is one major deficiency: an extension to hydraulic domain hardly makes sense. As the electric current is a (rate of) flow of charge, it is naturally seen as analogous to volumetric fluid flow. But then volumetric fluid flow is analogous to force. It then hardly contributes to developing a good intuition if we now declare (hydraulic) pressure as equivalent to velocity! And yet you can encounter this analogy very often. For example, software tools like Modelica and Simscape that we are going to introduce later in this course use it. As long as this is restricted to mechanical and electrical domains, this is acceptable and in fact offers some advantages. We will discuss this in due course.

8 Literature

Major portion of this course was based on [1], which is an official literature for a graduate course on MIT [6]. Notheless, this introductory lecture constitutes a compilation of materials found elsewhere, hence it is not possible to map it directly to any chapter in the book. Restricted to the book [1], the chapters 4 and 7 can serve the partial purpose of providing an overview of formats of mathematical models.

The final part of the lecture, which brough some new stuff, was based on Sections 2.1 and 2.2 from [1]. It is strongly advisable to go through the text.

Another classical books on modeling oriented towards control design is [9]. Both books are biased towards so-called bond graph modeling.

In a similar vein, the monograph [2] is written, although it is a bit dated (but the fundamentals do not change). A nice thing is that a (poor) photocopy is available on its author website at ETH. In fact, the introductory example with an RLC circuit was taken from another book by the same author [3] oriented towards simulation.

A recommendable Czech text is a set of lecture notes [8].

Much inspiration for the this lecture was taken from the first two chapters of [5]. Unfortunately, the book is not easily available either in libraries or in bookstores. Some instructions on how to order it are on the course website.

A very fine discussion of the topic of analogies among physical variables in diverse domains can be found in [7].

References

[1] Forbes T. Brown. *Engineering System Dynamics: A Unified Graph-Centered Approach, Second Edition.* CRC Press, 2nd edition, August 2006.

[2] François E. Cellier and Jurgen Greifeneder. *Continuous System Modeling.* Springer, 1st edition, May 1991.

[3] François E. Cellier and Ernesto Kofman. *Continuous System Simulation*. Springer, softcover reprint of hardcover 1st ed. 2006 edition, October 2010.

[4] Loi Do, Krištof Pučejdl, and Zdeněk Hurák. Laboratory-scale pumped-storage power plant modeling—Assignment of a semestral project. `https://moodle.fel.cvut.cz/mod/page/view.php?id=157157`.

[5] Olaf Egeland and Jan T. Gravdahl. *Modeling and simulation for automatic control*. Marine Cybernetics, 2002.

[6] Neville Hogan. MIT OpenCourseWare: Modeling and Simulation of Dynamic Systems (2.141), Fall 2006. `http://ocw.mit.edu/courses/mechanical-engineering/2-141-modeling-and-simulation-of-dynamic-systems-fall-2006/`.

[7] Neville Hogan and Peter Breedveld. The Physical Basis of Analogies in Physical System Models. In *The mechatronics handbook*. CRC Press, 2002.

[8] Petr Horáček. *Systémy a modely*. Nakladatelství ČVUT, Praha, 2000.

[9] Dean C. Karnopp, Donald L. Margolis, and Ronald C. Rosenberg. *System Dynamics: Modeling and Simulation of Mechatronic Systems*. Wiley, 4th edition, January 2006.