

# **Dynamic Systems**

---

an introduction

Rico A. R. Picone  
Department of Mechanical Engineering  
Saint Martin's University

01 October 2019

Copyright © 2019 Rico A. R. Picone All Rights Reserved

# Contents

<b>I Modeling electromechanical systems</b>	<b>7</b>
<b>01 Introduction</b>	<b>9</b>
01.01 The systems approach . . . . .	10
01.02 State-determined systems . . . . .	12
01.03 Energy, power, and lumping . . . . .	14
01.04 Mechanical translational elements . . . . .	16
01.05 Mechanical rotational elements . . . . .	20
01.06 Electronic elements . . . . .	24
01.07 Generalized through- and across-variables . . . . .	28
01.08 Generalized one-port elements . . . . .	30
<b>02 Linear graph models</b>	<b>33</b>
02.01 Introduction to linear graphs . . . . .	34
02.02 Sign convention . . . . .	36
02.03 Element interconnection laws . . . . .	43
02.04 Systematic linear graph modeling . . . . .	46
<b>03 State-space models</b>	<b>49</b>
03.01 State variable system representation . . . . .	50
03.02 State and output equations . . . . .	52
03.03 Graphs to state-space I normal trees . . . . .	54
03.04 Graphs to state-space II the algorithm . . . . .	56
03.05 State-space model of a translational mechanical system . . . . .	60
03.06 State-space model of a rotational mechanical system . . . . .	63
03.07 Between state-space and io differential equations . . . . .	66

03.08 Exercises . . . . .	72
<b>04 Electromechanical systems</b>	<b>75</b>
04.01 Ideal transducers . . . . .	76
04.02 Modeling with transducers . . . . .	79
04.03 DC motors . . . . .	80
04.04 A real electromechanical system . . . . .	86
04.05 Steady DC motor performance . . . . .	90
04.06 Transient DC motor performance . . . . .	96
04.07 Driving motors . . . . .	106
04.08 Exercises for Chapter 04 . . . . .	109
<b>II System response</b>	<b>111</b>
<b>05 Superposition, stability, etc.</b>	<b>113</b>
05.01 Superposition, derivative, and integral properties . . . . .	114
05.02 Equilibrium and stability properties . . . . .	115
05.03 Vibration isolation table analysis . . . . .	118
<b>06 Transient response</b>	<b>125</b>
06.01 Characteristic transient responses . . . . .	126
06.02 First-order systems . . . . .	127
06.03 Second-order systems . . . . .	130
<b>07 State-space response</b>	<b>137</b>
07.01 Solving for the state-space response . . . . .	139
07.02 Linear algebraic eigenproblem . . . . .	142
07.03 Diagonalizing basis . . . . .	145
07.04 Simulating state-space response . . . . .	149
<b>III Modeling other systems</b>	<b>157</b>
<b>08 Fluid and thermal systems</b>	<b>159</b>
08.01 Fluid system elements . . . . .	160
08.02 Thermal system elements . . . . .	165
08.03 Fluid transducers . . . . .	170
08.04 State-space model of a hydroelectric dam . . . . .	172

<b>IV Transfer function modeling</b>	<b>177</b>
<b>09 Transfer functions</b>	<b>179</b>
09.01 Poles and zeros . . . . .	180
<b>10 Impedance-based modeling</b>	<b>185</b>
10.01 Input impedance and admittance . . . . .	186
10.02 Impedance with two-port elements . . . . .	190
10.03 Transfer functions via impedance . . . . .	192
10.04 Norton and Thevenin theorems . . . . .	195
10.05 The divider method . . . . .	198
<b>V Frequency domain analysis</b>	<b>205</b>
<b>11 Steady frequency domain</b>	<b>207</b>
11.01 Fourier series . . . . .	208
11.02 Fourier transform . . . . .	212
11.03 Frequency and impulse response . . . . .	216
11.04 Sinusoidal frequency response . . . . .	219
11.05 Bode plots . . . . .	222
11.06 Periodic frequency response . . . . .	232
11.07 Design problem . . . . .	234
11.08 Nonlinear fluid system example . . . . .	237
<b>VI Appendices</b>	<b>247</b>
<b>A Mathematics reference</b>	<b>249</b>
A.01 Quadratic forms . . . . .	250
A.02 Trigonometry . . . . .	251
A.03 Matrix inverses . . . . .	254
A.04 Laplace transforms . . . . .	255
<b>B Advanced topics</b>	<b>257</b>
B.01 Systems with repeated eigenvalues . . . . .	258
<b>C Summaries</b>	<b>261</b>
C.01 Summary of system representations . . . . .	262
C.02 Summary of one-port elements . . . . .	263
C.03 Laplace transforms . . . . .	264

---

C.04 Fourier transforms . . . . .	266
<b>D Bibliography</b>	<b>269</b>



**Part I**

**Modeling electromechanical  
systems**



## Introduction

Read [Rowell and Wormley \(1997\)](#).

## Lecture 01.01 The systems approach

systems  
engineering

Simon Ramo and Richard Booton, Jr.—the folks who brought us the intercontinental ballistic missile (ICBM) (thanks? ...I mean thanks. But, thanks?)—defined *systems engineering* to be

*the design of the whole as distinguished from the design of the parts.*  
*(Booton and Ramo, 1984)*

Like the ICBM, many modern technologies require this systems engineering design approach.

mathematical  
modeling

dynamic systems

system properties

system dynamics

electromechanical  
systems

A key aspect of the systems engineering design process is the *mathematical modeling* of the system. Most systems of interest to the mechanical engineer are *dynamic systems*: those that change with time. Enter the time-derivative. And differential equations.

Dynamic systems exhibit behavior that can be characterized through analysis and called the system's *properties*. A property of a dynamic system might be how long it takes for it to respond to a given input or which types of inputs would cause a damaging response. Clearly, such properties are of significant interest to the design process.

*System dynamics* is the study and practice of dynamic system analysis and design. We primarily focus on analysis, at first, since it is a prerequisite for design. One cannot make a thing do what one wants without first understanding how the thing works.

This course focuses on *electromechanical systems*: systems with an interface between electronics and mechanical subsystems. These are ubiquitous: manufacturing plants, power plants, vehicles, robots, consumer products, anything with a motor—all include electromechanical systems. In the next course, we will consider more types of systems (e.g. fluid and thermal) and their interactions.

Electromechanical systems analysis can proceed with initially separate modeling of the electronic and mechanical subsystems, then, through an unholy union, combining their equations and attempting a solution. This is fine for simple systems. However, many systems will require a systematic approach.

graphs  
graph theory

We adopt a systematic approach that draws *graphs* (á la *graph theory*) for electronic and mechanical systems that are intentionally analogous to electronic circuit diagrams. This allows us to use a single graphical diagram to express a system's composition and interconnections. Virtually every technique from electronic circuit analysis can be applied to these

representations. Elemental equations, Kirchhoff's laws, impedance—each will be generalized.

## Lecture 01.02 State-determined systems

<b>system boundary</b>	A <i>system</i> is defined to be a collection of objects and their relations contained within a <i>boundary</i> . The collection of those objects that are external to the system and yet interact with it is called the <i>environment</i> . <i>System variables</i> are variables that represent the behavior of the system, both those that are internal to the system and those that are external—that is, with the system's environment.
<b>environment</b>	
<b>system variables</b>	
<b>input variables</b>	
<b>input vector</b>	There are three important classes of system variable, all typically expressed as vector-valued functions of time $t$ , conventionally all expressed as column-vectors (and called “vectors” even though they’re vector-valued functions ... because nothing makes sense and we’re all going to die). Consider Figure 01.1 for the following definitions. <i>Input variables</i> are system variables that do not depend on the internal dynamics of the system; for a system with $r$ input variables, the “ <i>input vector</i> ” is a vector-valued function $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^r$ . The environment prescribes inputs, making them <i>independent variables</i> . Conversely, <i>output variables</i> are system variables of interest to the designer; for a system with $m$ output variables, the “ <i>output vector</i> ” is a vector-valued function $\mathbf{y} : \mathbb{R} \rightarrow \mathbb{R}^m$ . Outputs may or may not directly interact with the environment. Finally, a minimal set of variables that define the internal state of the system are defined as the <i>state variables</i> ; for a system with $n$ state variables, the “ <i>state vector</i> ” is a vector-valued function $\mathbf{x} : \mathbb{R} \rightarrow \mathbb{R}^n$ .
<b>output variables</b>	
<b>output vector</b>	
<b>state variables</b>	
<b>state vector</b>	
<b>state-determined</b>	We consider a very common class of system: those that are <i>state-determined</i> , which are those for which (Rowell and Wormley, 1997)

1. a mathematical description,

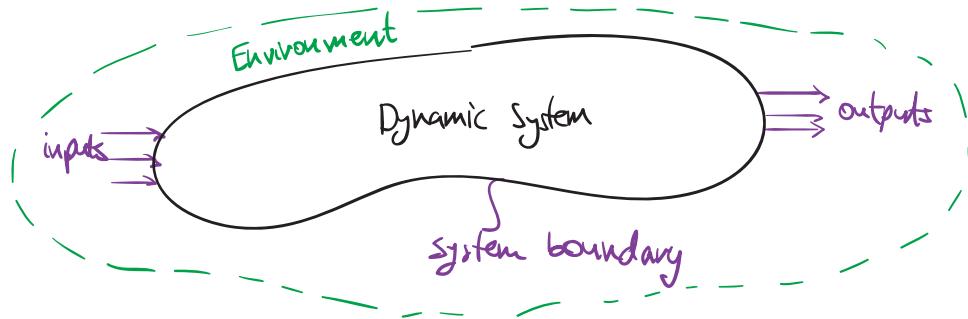


Figure 01.1: illustration of a system and its environment.

2. the state at time  $t_0$ , called the *initial condition*  $x(t)|_{t=t_0}$ , and
3. the input  $u$  for all time  $t \geq t_0$

**initial condition**

are necessary and sufficient conditions to determine  $x(t)$  (and therefore  $y(t)$ ) for all  $t \geq t_0$ .

The “mathematical description” of the system requires a set of primitive elements be assigned to represent its internal and external interactions. The equations derive from two key types of mathematical relationships:

1. the input-output behavior of each primitive element and
2. the topology of interconnections among elements.

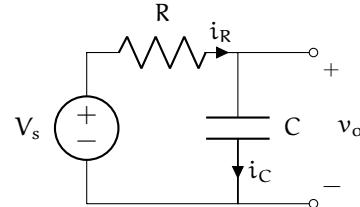
The former generate *elemental equations* and the latter, *continuity* or *compatibility equations*.

**elemental  
equations**  
**continuity  
equations**  
**compatibility  
equations**

#### Example 01.02-1 a state-determined system

In the RC circuit shown, let  $V_s$  be a source and  $v_o$  the voltage of interest. Identify

1. the system boundary,
2. the input vector,
3. the output vector,
4. a state vector,
5. an elemental equation,
6. and which equations might be continuity or compatibility equations.



## Lecture 01.03 Energy, power, and lumping

**law of energy conservation**  
**power**

The *law of energy conservation* states that, for an isolated system, the total energy remains constant. Let  $\mathcal{E} : \mathbb{R} \rightarrow \mathbb{R}$  be the function of time representing the total energy in a system and  $\mathcal{P} : \mathbb{R} \rightarrow \mathbb{R}$  be the function of time representing *power* into the system, defined as

$$\mathcal{P}(t) = \frac{d\mathcal{E}(t)}{dt}. \quad (01.1)$$

The energy in a system can change if it exchanges energy with its environment. We consider this exchange to occur through a finite number of *ports*, each of which can supply or remove energy (positive or negative power), as in Figure 01.2. This is expressed in an equation for power into a port  $\mathcal{P}_i$  and  $N$  ports as

We construct our systems such that they have no internal energy sources.

### 01.03.1 Lumping

We have assumed power enters a system via a finite number of ports. Similarly, we assume the energy *in* a system is stored in a finite number  $M$  of distinct elements with energy  $\mathcal{E}_i$  such that

$$\mathcal{E}(t) = \sum_{i=1}^M \mathcal{E}_i(t). \quad (01.2)$$

**energy storage elements  
abstraction**

We call these elements *energy storage elements*.

Considering a system to have a finite number of elements, as we have done, requires a specific kind of *abstraction* from real systems. A familiar example is the “point mass” of elementary mechanics. We say it interacts with its environment via specific connections called ports (maybe it’s attached to a spring element) and behaves a certain way in these interactions (for a mass element, Newton’s second law). We do not often encounter an object that behaves as if it has mass, but no volume. Yet, this is a useful abstraction for many problems.

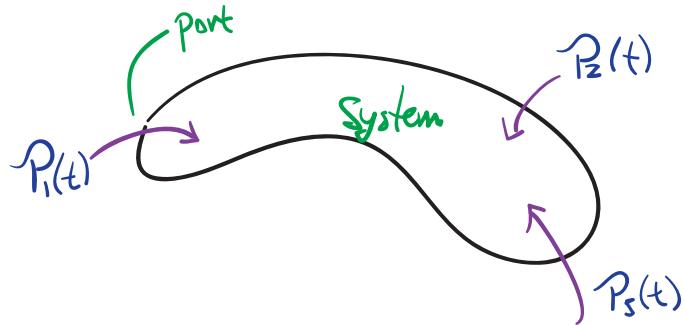


Figure 01.2: system ports

When we abstract like this, considering an object to be described fully as a discrete object with interaction ports, we are said to be *lumped-parameter modeling*. This is often contrasted with *distributed* or *continuous modeling*, which consider the element in greater detail. For instance, an object might be considered to be distributed through space and perhaps be flexible or behave as a fluid.

Determining if lumped-parameter modeling is proper for a given system is dependent on the type of insight one wants to achieve about the system. The system itself does not prescribe the proper modeling technique, but the desired understanding does. Every system is incredibly complex in its behavior, if one considers it at a fine-granularity. In this light, it is striking that simple models work *at all*. Nevertheless, lumping is highly effective for many analyses.

It is important to note that lumped-parameter models can be applied at different levels of granularity for the same system. *Finite element modeling* can use a large number of small lumped-parameter model elements to approximate a continuous model. Such applications are beyond the scope of this course.

**lumped-parameter modeling**  
**distributed modeling**  
**continuous modeling**

**finite element modeling**

## Lecture 01.04 Mechanical translational elements

**force  
velocity**

We now introduce a few lumped-parameter elements for mechanical systems in translational (i.e. straight-line) motion. Newton's laws of motion can be applied. Let a *force*  $f$  and *velocity*  $v$  be input to a port in a mechanical translational element. Since, for mechanical systems, the power into the element is

$$\mathcal{P}(t) = f(t)v(t) \quad (01.3)$$

**power-flow  
variables**

**ground  
work**

we call  $f$  and  $v$  the *power-flow variables* for mechanical translational systems. Some mechanical translational elements have two distinct locations between which its velocity is defined (e.g. the velocity across a spring's two ends) and other elements have just one (e.g. a point-mass), the velocity of which must have an inertial frame of reference. This is analogous to how a point in a circuit can be said to have a voltage—by which we mean “relative to ground.” In fact, we call this mechanical translational inertial reference *ground*.

Work done on the system over the time interval  $[0, T]$  is defined as

$$W \equiv \int_0^T \mathcal{P}(\tau) d\tau. \quad (01.4)$$

Therefore, the work done on a mechanical system is

$$W = \int_0^T f(\tau)v(\tau)d\tau. \quad (01.5)$$

**linear displacement**

The *linear displacement*  $x$  is

$$x(t) = \int_0^t v(\tau)d\tau + x(0). \quad (01.6)$$

**linear momentum**

Similarly, the *linear momentum* is

$$p(t) = \int_0^t f(\tau)d\tau + p(0). \quad (01.7)$$

**energy storage  
elements**  
**energy dissipative  
elements**  
**source elements**

We now consider two elements that can store energy, called *energy storage elements*; an element that can dissipate energy to a system's environment, called an *energy dissipative element*; and two elements that can supply power from outside a system, called *source elements*.

### 01.04.1 Translational springs

A *translational spring* is defined as an element for which the displacement  $x$  across it is a monotonic function of the force  $f$  through it. A *linear translational spring* is a spring for which Hooke's law applies; that is, for which

$$f(t) = kx(t), \quad (01.8)$$

where  $f$  is the force through the spring and  $x$  is the displacement across the spring, minus its unstretched length, and  $k$  is called the *spring constant* and is typically a function of the material properties and geometry of the element. This is called the element's *constitutive equation* because it constitutes what it means to be a spring.

Although there are many examples of nonlinear springs, we can often use a linear model for analysis in some operating regime. The *elemental equation* for a linear spring can be found by time-differentiating Equation 01.8 to obtain

We call this the elemental equation because it relates the element's power-flow variables  $f$  and  $v$ .

A spring stores energy as elastic potential energy, making it an *energy storage element*. The amount of energy it stores depends on the force it applies. For a linear spring,

$$\mathcal{E}(t) = \frac{1}{2k}f(t)^2. \quad (01.9)$$

### 01.04.2 Point-masses

A non-relativistic translational point-mass element with mass  $m$ , velocity  $v$  (relative to an inertial reference frame), and momentum  $p$  has the constitutive equation

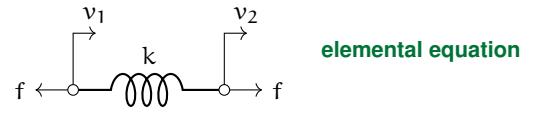
$$p = mv. \quad (01.10)$$

**translational spring**  
**linear translational spring**

**spring constant**

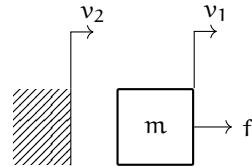
**constitutive equation**

**elemental equation**



**Figure 01.3:** schematic symbol for a spring with spring constant  $k$  and velocity drop  $v = v_1 - v_2$ .

Once again, time-differentiating the constitutive equation gives us the elemental equation:



which is just Newton's second law.

Point-masses can store energy (making them *energy storage elements*) in gravitational potential energy or, as will be much more useful in our analyses, in kinetic energy

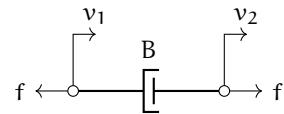
$$\mathcal{E}(t) = \frac{1}{2}mv^2. \quad (01.11)$$

### 01.04.3 Dampers

#### dampers

*Dampers* are defined as elements for which the force  $f$  through the element is a monotonic function of the velocity  $v$  across it. *Linear dampers* have constitutive equation (and, it turns out, elemental equation)

$$f = Bv \quad (01.12)$$



**Figure 01.5:** schematic symbol for a damper with damping coefficient  $B$  and velocity drop  $v = v_1 - v_2$ .

#### linear dampers

#### damping coefficient viscous damping

where  $B$  is called the *damping coefficient*. Linear damping is often called *viscous damping* because systems that push viscous fluid through small orifices or those that have lubricated sliding are well-approximated by this equation. For this reason, a damper is typically schematically depicted as a *dashpot*.

#### dry friction Coulomb friction drag

Linear damping is a reasonable approximation of lubricated sliding, but it is rather poor for *dry friction* or *Coulomb friction*, forces for which are not very velocity-dependent. Aerodynamic *drag* is quite velocity-dependent, but is rather nonlinear, often represented as

where  $c$  is called the drag constant.

Dampers dissipate energy from the system (typically to heat), making them *energy dissipative elements*.

#### 01.04.4 Force and velocity sources

An *ideal force source* is an element that provides arbitrary energy to a system via an independent (of the system) force. The corresponding velocity across the element depends on the system.

**ideal force source**

An *ideal velocity source* is an element that provides arbitrary energy to a system via an independent (of the system) velocity. The corresponding force through the element depends on the system.

**ideal velocity source**

## Lecture 01.05 Mechanical rotational elements

We now introduce a few lumped-parameter elements for mechanical systems in rotational motion. Newton's laws of motion, in their angular analogs, can be applied. Let a *torque*  $T$  and *angular velocity*  $\Omega$  be input to a port in a mechanical rotational element. Since, for mechanical rotational systems, the power into the element is

$$\mathcal{P}(t) = T(t)\Omega(t) \quad (01.13)$$

**power-flow variables**

we call  $T$  and  $\Omega$  the *power-flow variables* for mechanical rotational systems. Some mechanical rotational elements have two distinct locations between which its angular velocity is defined (e.g. the angular velocity across a spring's two ends) and other elements have just one (e.g. a rotational inertia), the velocity of which must have an inertial frame of reference. This is analogous to how a point in a circuit can be said to have a voltage—by which we mean “relative to ground.” In fact, we call this mechanical rotational inertial reference *ground*.

**ground work**

Work done on the system over the time interval  $[0, t_f]$  is defined as

$$W \equiv \int_0^{t_f} \mathcal{P}(\tau) d\tau. \quad (01.14)$$

Therefore, the work done on a mechanical system is

$$W = \int_0^{t_f} T(\tau)\Omega(\tau) d\tau. \quad (01.15)$$

**angular displacement**

The *angular displacement*  $\theta$  is

$$\theta(t) = \int_0^t \Omega(\tau) d\tau + \theta(0). \quad (01.16)$$

**angular momentum**

Similarly, the *angular momentum* is

$$h(t) = \int_0^t T(\tau)d\tau + h(0). \quad (01.17)$$

**energy storage elements**  
**energy dissipative elements**  
**source elements**

We now consider two elements that can store energy, called *energy storage elements*; an element that can dissipate energy to a system's environment, called an *energy dissipative element*; and two elements that can supply power from outside a system, called *source elements*.

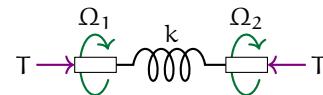
### 01.05.1 Rotational springs

A *rotational spring* is defined as an element for which the angular displacement  $\theta$  across it is a monotonic function of the torque  $T$  through it. A *linear rotational spring* is a rotational spring for which the angular form of Hooke's law applies; that is, for which

$$T(t) = k\theta(t), \quad (01.18)$$

where  $T$  is the torque through the spring and  $\theta$  is the angular displacement across the spring and  $k$  is called the *torsional spring constant* and is typically a function of the material properties and geometry of the element. This is called the element's *constitutive equation* because it constitutes what it means to be a rotational spring.

Although there are many examples of nonlinear springs, we can often use a linear model for analysis in some operating regime. The *elemental equation* for a linear spring can be found by time-differentiating Equation 01.18 to obtain



**Figure 01.6:** schematic symbol for a spring with spring constant  $k$  and angular velocity drop  $\Omega = \Omega_1 - \Omega_2$ .

**rotational spring**  
**linear rotational**  
**spring**

**torsional spring**  
**constant**  
**constitutive**  
**equation**

**elemental equation**

We call this the elemental equation because it relates the element's power-flow variables  $T$  and  $\Omega$ .

A rotational spring stores energy as elastic potential energy, making it an *energy storage element*. The amount of energy it stores depends on the torque it applies. For a linear rotational spring,

$$\mathcal{E}(t) = \frac{1}{2k}T(t)^2. \quad (01.19)$$

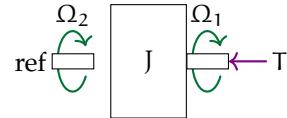
### 01.05.2 Moments of inertia

A *moment of inertia* element with moment of inertia  $J$ , angular velocity  $\Omega$  (relative to an inertial reference frame), and angular momentum  $h$  has the constitutive equation

$$h = J\Omega. \quad (01.20)$$

**moment of inertia**

Once again, time-differentiating the constitutive equation gives us the elemental equation:



which is just the angular version of Newton's second law.

**flywheel** Any rotating element with mass can be considered as a lumped-inertia element. The *flywheel* is the quintessential example. Flywheels store energy in their angular momentum, with the expression

$$\mathcal{E}(t) = \frac{1}{2} J \Omega^2, \quad (01.21)$$

making them (and all moments of inertia) *energy storage elements*.

### 01.05.3 Rotational dampers

**rotational dampers**

*Rotational dampers* are defined as elements for which the torque  $T$  through the element is a monotonic function of the angular velocity  $\Omega$  across it. *Linear rotational dampers* have constitutive equation (and, it turns out, elemental equation)

$$T = B\Omega \quad (01.22)$$

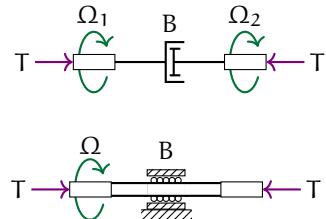
**torsional damping coefficient**

where  $B$  is called the *torsional damping coefficient*. Linear torsional damping is often called *torsional viscous damping* because systems that push viscous fluid through small orifices or those that have lubricated bearings are well-approximated by this equation. For this reason, a damper is typically schematically depicted as a *drag cup* or as a *bearing*, both of which are shown in Figure 01.8.

**torsional viscous damping**

**drag cup bearing**

**dry friction Coulomb friction**



**Figure 01.8:** schematic symbol for a drag cup (above) and bearing (below) with damping coefficient  $B$ . For the drag cup, the angular velocity drop is  $\Omega = \Omega_1 - \Omega_2$  and for the bearing,  $\Omega$  is reference is ground.

Linear damping is a reasonable approximation of lubricated sliding, but it is rather poor for *dry friction* or *Coulomb friction*, forces for which are not very velocity-dependent.

Rotational dampers dissipate energy from the system (typically to heat), making them *energy dissipative elements*.

#### 01.05.4 Torque and angular velocity sources

An *ideal torque source* is an element that provides arbitrary energy to a system via an independent (of the system) torque. The corresponding angular velocity across the element depends on the system.

**ideal torque source**

An *ideal angular velocity source* is an element that provides arbitrary energy to a system via an independent (of the system) angular velocity. The corresponding torque through the element depends on the system.

**ideal angular velocity source**

## Lecture 01.06 Electronic elements

We now review a few lumped-parameter elements for electronic systems.

**current voltage** Let a *current*  $i$  and *voltage*  $v$  be input to a port in an electronic element. Since, for electronic system, the power into the element is

$$P(t) = i(t)v(t) \quad (01.23)$$

**power-flow variables ground** we call  $i$  and  $v$  the *power-flow variables*. Voltage is always understood to be between two points in a circuit. If only one point is included, the voltage is implicitly relative to a reference voltage, called *ground*.

**magnetic flux linkage** The *magnetic flux linkage*  $\lambda$  is

$$\lambda(t) = \int_0^t v(\tau)d\tau + \lambda(0). \quad (01.24)$$

**charge** Similarly, the *charge* is

$$q(t) = \int_0^t i(\tau)d\tau + q(0). \quad (01.25)$$

**energy storage elements energy dissipative element source elements** We now consider two elements that can store energy, called *energy storage elements*; an element that can dissipate energy to a system's environment, called an *energy dissipative element*; and two elements that can supply power from outside a system, called *source elements*.

### 01.06.1 Capacitors

Capacitors have two terminal and are composed of two conductive surfaces separated by some distance. One surface has charge  $q$  and the other  $-q$ . A capacitor stores energy in an *electric field* between the surfaces.

**capacitance** Let a capacitor with voltage  $v$  across it and charge  $q$  be characterized by the parameter *capacitance*  $C$ , where the constitutive equation is

$$q = Cv. \quad (01.26)$$

**farad (F)** The capacitance has derived SI unit *farad* (F), where  $F = A \cdot s/V$ . A farad is actually quite a lot of capacitance. Most capacitors have capacitances best represented in  $\mu F$ ,  $nF$ , and  $pF$ .

The time-derivative of this equation yields the  $v$ - $i$  relationship (what we call the “elemental equation”) for capacitors.

$$\frac{dv}{dt} = \frac{1}{C} i \quad (01.27)$$

Capacitors allow us to build many new types of circuits: filtering, energy storage, resonant, blocking (blocks dc-component), and bypassing (draws ac-component to ground).

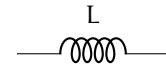
Capacitors come in a number of varieties, with those with the largest capacity (and least expensive) being *electrolytic* and most common being *ceramic*. There are two functional varieties of capacitors: *bipolar* and *polarized*, with circuit diagram symbols shown in Figure 01.9. Polarized capacitors can have voltage drop across in only one direction, from *anode* (+) to *cathode* (−)—otherwise they are damaged or may *explode*. Electrolytic capacitors are polarized and ceramic capacitors are bipolar.

So what if you need a high-capacitance bipolar capacitor? Here's a trick: place identical high-capacity polarized capacitors *cathode-to-cathode*. What results is effectively a bipolar capacitor with capacitance *half* that of one of the polarized capacitors.

**electrolytic capacitor**  
**ceramic capacitor**  
**bipolar capacitor**  
**polarized capacitor**  
**anode**  
**cathode**  
**explosion**  
**cathode-to-cathode**

### 01.06.2 Inductors

A *pure inductor* is defined as an element in which *flux linkage*  $\lambda$ —the integral of the voltage—across the inductor is a monotonic function of the current  $i$ . An *ideal inductor* is such that this monotonic function is linear, with slope called the *inductance*  $L$ ; i.e. the ideal constitutive equation is



**Figure 01.10:** inductor circuit diagram symbol.

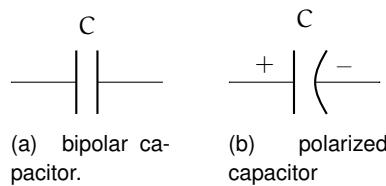
**pure inductor**  
**flux linkage**  $\lambda$   
  
**ideal inductor**  
**inductance**  $L$

$$\lambda = Li \quad (01.28)$$

The units of inductance are the SI derived unit *henry* (H). Most inductors have inductance best represented in mH or  $\mu$ H.

**henry (H)**

The elemental equation for an inductor is found by taking the time-derivative of the constitutive equation.



**Figure 01.9:** capacitor circuit diagram symbols.

$$\frac{di}{dt} = \frac{1}{L}v \quad (01.29)$$

Inductors store energy in a *magnetic field*. It is important to notice how inductors are, in a sense, the *opposite* of capacitors. A capacitor's current is proportional to the time rate of change of its voltage. An inductor's voltage is proportional to the time rate of change of its current.

Inductors are usually made of wire coiled into a number of turns. The geometry of the coil determines its inductance L.

**core** Often, a *core* material—such as iron and ferrite—is included by wrapping the wire around the core. This increases the inductance L.

Inductors are used extensively in radio-frequency (rf) circuits, which we won't discuss in this text. However, they play important roles in ac-dc conversion, filtering, and transformers—all of which we will consider extensively.

The circuit diagram for an inductor is shown in Figure 01.10.

### 01.06.3 Resistors

Resistors *dissipate* energy from the system, converting electrical energy to thermal energy (heat).

**constitutive equation** The *constitutive equation* for an ideal resistor is

$$v = iR. \quad (01.30)$$



**Figure 01.11:** resistor circuit diagram symbol.

This is already in terms of power variables, so it is also the *elemental equation*.

### 01.06.4 Sources

Sources (a.k.a. supplies) supply power to a circuit. There are two primary types: *voltage sources* and *current sources*.

#### 01.06.4.1 Ideal voltage sources

An ideal voltage source provides exactly the voltage a user specifies, independent of the circuit to which it is connected. All it must do in order to achieve this is to supply whatever current necessary.

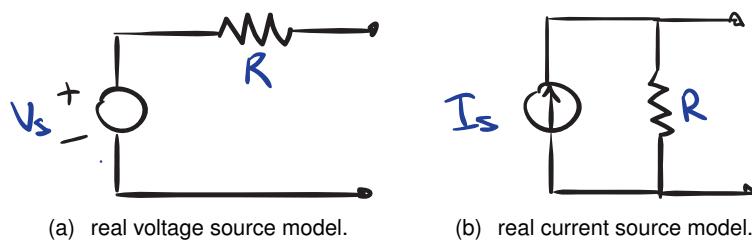
#### 01.06.4.2 Ideal current sources

An ideal current source provides exactly the current a user specifies, independent of the circuit to which it is connected. All it must do in order to achieve this is to supply whatever voltage necessary.

#### 01.06.4.3 Modeling real sources

No real source can produce infinite power. Some have feedback that controls the output within some finite power range. These types of sources can be approximated as ideal when operating within its specifications. Many voltage sources (e.g. batteries) do not have internal feedback controlling the voltage. When these sources are “loaded” (delivering power) they cannot maintain their nominal output, be that voltage or current. We model these types of sources as ideal sources in series or parallel with a resistor, as illustrated in [Figure 01.12](#).

Most manufacturers specify the nominal resistance of a source as the “output resistance.” A typical value is  $50 \Omega$ .



**Figure 01.12:** Models for power-limited “real” sources.

## Lecture 01.07 Generalized through- and across-variables

**energy domains** We have considered mechanical translational, mechanical rotational, and electronic systems—which we refer to as different *energy domains*. There are analogies among these systems that allow for generalizations of certain aspects of these systems. These generalizations will allow us to use a single framework for unifying the analysis of these (and other) dynamic systems.

There are two important classes of variables common to lumped-parameter dynamic systems: *across-variables* and *through-variables*.

**across-variable** An *across-variable* is one that makes reference to two nodes of a system element. For instance, the following are across-variables:

- 
- 
- 

**generalized across-variable through-variable** We denote a *generalized across-variable* as  $\mathcal{V}$ .

A *through-variable* is one that represents a quantity that passes through a system element. For instance, the following are through-variables:

- 
- 
- 

**generalized through-variable generalized integrated across-variable** We denote a *generalized through-variable* as  $\mathcal{F}$ .

The *generalized integrated across-variable*  $\mathcal{X}$  is

$$\mathcal{X} = \int_0^t \mathcal{V}(\tau) d\tau + \mathcal{X}(0). \quad (01.31)$$

**generalized integrated through-variable**

The *generalized integrated through-variable*  $\mathcal{H}$  is

$$\mathcal{H} = \int_0^t \mathcal{F}(\tau) d\tau + \mathcal{H}(0). \quad (01.32)$$

For mechanical and electronic systems, power  $\mathcal{P}$  passing through a lumped-parameter element is

$$\mathcal{P}(t) = \mathcal{F}(t)\mathcal{V}(t). \quad (01.33)$$

These generalized across- and through-variables are sometimes used in analysis. However, the key idea here is that there are two classes of power-flow variables: across and through. These two classes allow us to strengthen the sense in which we consider different dynamic systems to be analogous.

## Lecture 01.08 Generalized one-port elements

We can categorize the behavior of one-port elements—electronic, mechanical translational, and mechanical rotational—considered thus far. In the following sections, we consider two types of energy storage elements, dissipative elements, and source elements.

### 01.08.1 A-type energy storage elements

**A-type energy storage element**  
generalized capacitor

An element that stores energy as a function of its across-variable is called an *A-type energy storage element*. Sometimes we call it a *generalized capacitor* because a capacitor is an A-type energy storage element.

For generalized through-variable  $\mathcal{F}$ , across-variable  $\mathcal{V}$ , integrated through-variable  $\mathcal{H}$ , and integrated across-variable  $X$  the ideal, linear constitutive equation is

$$\mathcal{H} = C\mathcal{V} \quad (01.34)$$

**generalized capacitance  $C$**  for  $C \in \mathbb{R}$  called the *generalized capacitance*. Differentiating Equation 01.34 with respect to time, the elemental equation is

capacitors  
masses  
rotational inertia

A-type energy storage elements considered thus far are *capacitors*, *translational masses*, and *rotational moments of inertia*. As with generalized variables, the analogs among elements are more important than are generalized A-type energy storage elements.

### 01.08.2 T-type energy storage elements

**T-type energy storage element**  
generalized inductor

An element that stores energy as a function of its through-variable is called a *T-type energy storage element*. Sometimes we call it a *generalized inductor* because an inductor is a T-type energy storage element.

The ideal, linear constitutive equation is

$$X = L\mathcal{F} \quad (01.35)$$

**generalized inductance  $L$**  for  $L \in \mathbb{R}$  called the *generalized inductance*. Differentiating Equation 01.35 with respect to time, the elemental equation is

T-type energy storage elements considered thus far are *inductors*, *translational springs*, and *rotational springs*. As with generalized variables, the analogs among elements are more important than are generalized T-type energy storage elements.

**inductors**  
**translational**  
**springs**  
**rotational springs**

### 01.08.3 D-type energy dissipative elements

An element that dissipates energy from the system and has an algebraic relationship between its through-variable and its across-variable is called a *D-type energy dissipative element*. Sometimes we call it a *generalized resistor* because a resistor is a D-type energy dissipative element.

The ideal, linear constitutive and elemental equation is

$$\mathcal{V} = R\mathcal{F} \quad (01.36)$$

for  $R \in \mathbb{R}$  called the *generalized resistance*.

D-type energy dissipative elements considered thus far are *resistors*, *translational dampers*, and *rotational dampers*. As with generalized variables, the analogs among elements are more important than are generalized D-type energy dissipative elements.

**D-type energy**  
**dissipative element**  
**generalized**  
**resistor**

**generalized**  
**resistance R**  
**resistors**  
**translational**  
**dampers**  
**rotational dampers**

### 01.08.4 Sources

An *ideal through-variable source* is an element that provides arbitrary energy to a system via an independent (of the system) through-variable. The corresponding across-variable element depends on the system. Current, force, and torque sources are the through-variable sources considered thus far.

**ideal**  
**through-variable**  
**source**

An *ideal across-variable source* is an element that provides arbitrary energy to a system via an independent (of the system) across-variable. The corresponding through-variable through the element depends on the system. Voltage, translational velocity, and angular velocity are the across-variable sources considered thus far.

**ideal**  
**across-variable**  
**source**



02

---

## Linear graph models

## Lecture 02.01 Introduction to linear graphs

**linear graphs**  
**topology**

Engineers often use graphical techniques to aid in analysis and design. We will use *linear graphs* to represent the *topology* or structure of a system modeled as interconnected lumped elements.

This represents to us the essential structure of the system in a minimalist form. In this way, it is like Massimo Vignelli's famous 1972 New York subway system "map," which inspired widespread adoption of his style (see Figure 02.1).<sup>1</sup> Besides minimalism, the key idea in Vignelli subway

<sup>1</sup>Vignelli was a brilliant Minimalist designer of many products, from dishes to clothing, but he was most known for his graphic design. Great places to start studying Vignelli are



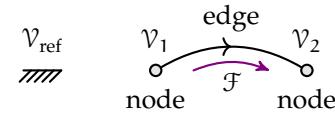
Figure 02.1: a modern New York subway map in the style of Vignelli (Jake Berman).

maps is that the details of the tunnels' paths are irrelevant and, in fact, distracting to the person attempting to get from one station to another.

In a similar way, a linear graph represents the system in a minimalist style, with only two types of objects:

1. A set of *edges*, each of which represents an energy port associated with a system element. Each edge is drawn as an oriented line segment "→".
2. A set of *nodes*, each of which represents a point of interconnection among system elements. Each node is drawn as a dot "o".

All edges begin and end at nodes. The nodes represent locations in the system where distinct across-variable values may be measured. For example, wires that connect elements are actually *nodes* at which voltage may be measured. Putting an edge together with nodes, we have [Figure 02.2](#).



**Figure 02.2:** an edge with nodes. The across variable is  $\mathcal{V} = \mathcal{V}_1 - \mathcal{V}_2$ .

It is important to note that linear graphs can represent *nonlinear* system elements—the name is a reference to the *lines* used.

It is common to choose a node of the graph as the *reference node*, to which all across-variables are referenced. Due to its similarity to the electronic *ground*, we often use these terms interchangeably.

**edges**

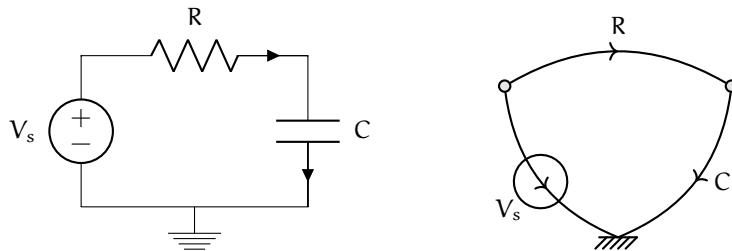
**nodes**

**nonlinearity**

**reference node**

**ground**

[Figure 02.3](#) shows how a linear graph can be constructed for a simple RC-circuit. Note that the wires become nodes, the elements become edges, and the reference node represents the circuit ground. In a similar manner, we will construct linear graphs of circuits, mechanical translational systems, and mechanical rotational systems.



**Figure 02.3:** an example of a linear graph representation of an RC-circuit.

the documentary *Design is One* (2012) and *The Vignelli Canon*.

## Lecture 02.02 Sign convention

**sign** The *sign* (positive or negative) of a variable is used to represent an orientation of its physical quantity. For instance,  $-3 \text{ m/s}$  could mean  $3 \text{ m/s}$  to the *right* or *left*. No one can say which is better (right is better). Deciding how the physical quantity corresponds to the sign of the variable is called *sign assignment*.

**sign assignment** When we use a *sign convention*, we make the assignment in a conventional manner. For instance, the sign convention for normal stress is that compression is negative and tension is positive.

**sign convention** Why use a sign convention? If we follow a convention when constructing a problem, we can use the convention's *interpretation* of the result. For complicated systems, this helps us keep things straight. Furthermore, if someone else attempts to understand our work, it is much easier to simply say "using the standard sign convention, ..." than explaining our own snowflake sign assignment. However, it is nonetheless true that we can assign signs arbitrarily.

**Vive la révolution!**

Arbitrary? *Vive la révolution!* But wait. If a *source* is present, we must observe some caution. A source typically *comes with its own convention*. For instance, if we hook up a power supply to the circuit with the + and - leads a certain way, unless we want to get very confused, we should probably accommodate that sign.

A sign convention for each of the energy domains we've considered follows.

### 02.02.1 Electronic systems

**passive sign convention** We use the *passive sign convention* of electrical engineering, defined below.

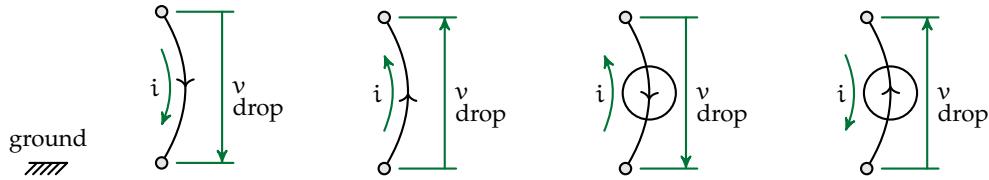
#### Definition 02.02.1: passive sign convention

Power flowing *in* to a component is considered to be *positive* and power flowing *out* of a component is considered *negative*.

**passive element** Because power  $\mathcal{P} = vi$ , this implies the current and voltage signs are prescribed by the convention. For *passive elements*, the electrical potential must drop in the direction of positive current flow. This means the assumed direction of voltage drop across a passive element must be the same as that of the current flow.

For *active elements*, which supply power to the circuit, the converse is true: the voltage drop and current flow must be in opposite directions. [Figure 02.4](#) illustrates the possible configurations.

**active element**



**Figure 02.4:** passive sign convention for electronic systems in terms of voltage  $v$  and current  $i$ . Passive elements are on the left, active on the right.

When drawing a linear graph of a circuit, for each passive element's edge, draw the arrow beside it pointing in the direction of assumed current flow and voltage drop.

The purpose of a sign convention is to help us *interpret* the signs of our results. For instance, if, at a given instant, a capacitor has voltage  $v_C = 3$  V and current  $i_C = -2$  A, we compute  $P_C = -6$  W and we know 6 W of power is flowing *from* the capacitor into the circuit.

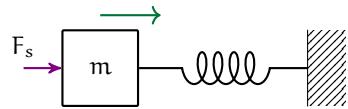
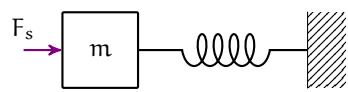
interpretation

For passive elements, there is no preferred direction of "assumed" voltage drop and current flow. If a voltage or current value discovered by performing a circuit analysis is positive, this means the "assumed" and "actual" directions are the same. For a negative value, the directions are opposite.

For active elements, choose the sign in accordance with the physical situation. For instance, if a positive terminal of a battery is connected to a certain terminal in a circuit, it ill behooves one to simply say "but Darling, I'm going to call that negative." It's positive whether you like it or not, Nancy.

### 02.02.2 Translational mechanical systems

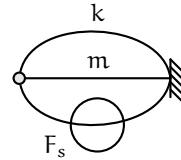
The following steps can be applied to any translational mechanical system. We introduce the convention with an inline example. Consider the simple mechanical system shown at right.



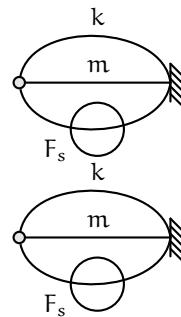
**coordinate arrow** Assign the sign by drawing a coordinate arrow, as shown at right. The direction of the arrow is arbitrary, however, if possible, assign the positive direction to match

the sources. If the problem allows, it is best practice to have all sources and the coordinate arrow in the same direction.

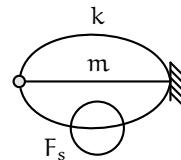
**draw linear graph without arrows** There are two nodes with distinct velocities: ground and the mass, as shown at right. The mass node is always drawn to ground. The spring connects between the mass and ground. Finally, the force source connects to the mass, where it is applied, and also connects to ground, which is impervious to it.



**assign spring and damper directions** On each spring and damper element, define the positive velocity drop and edge arrow to be *in the direction of the coordinate arrow*.



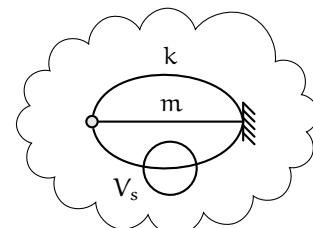
**assign mass directions** On each mass element, define the positive velocity drop and edge arrow to be *toward ground*. Sometimes we dash the latter half of the mass edge in to signify that it is “virtually” connected to ground.



**assign force source directions** On each force source element, define the positive direction as follows.

(ideal) If the force source has the *same* definition of positive as your coordinate arrows, draw it *toward the node of application*.

(if needed) If the force source has the *opposite* definition of positive as your coordinate arrow, draw it *away from the node of application*.



**assign velocity source directions** On each velocity source element, define the positive direction as follows.

(ideal) If the velocity source has the *same* definition of positive as your coordinate arrows, draw it *away from the node of application*.

(if needed) If the velocity source has the *opposite* definition of positive as your coordinate arrow, draw it *toward the node of application*.

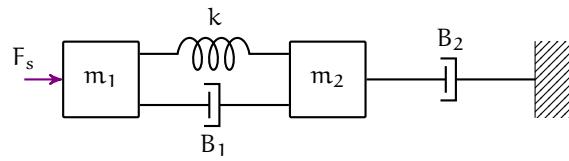
This convention yields the interpretations of Table 02.1.

**Table 02.1:** interpretation of the translational mechanical system sign convention.

	force $f$		velocity $v$	
	positive +	negative -	positive +	negative -
m	force <i>in</i> direction of the coordinate arrow	force <i>opposite</i> the direction of the coordinate arrow	velocity <i>in</i> the coordinate arrow direction	velocity <i>opposite</i> the coordinate arrow direction
k	<i>compressive</i> force	<i>tensile</i> force	velocity drops <i>in</i> the coordinate arrow direction	velocity drops <i>opposite</i> the coordinate arrow direction
B	<i>compressive</i> force	<i>tensile</i> force	velocity drops <i>in</i> the coordinate arrow direction	velocity drops <i>opposite</i> the coordinate arrow direction

**Example 02.02-1 translational mechanical sign convention**

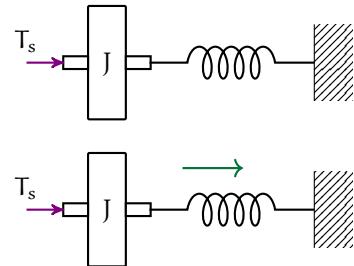
For the system shown, draw a linear graph and assign signs according to the sign convention.



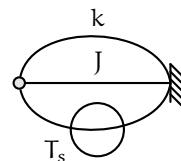
### 02.02.3 Rotational mechanical systems

The following steps can be applied to any rotational mechanical system. We introduce the convention with an inline example. Consider the simple system shown at right.

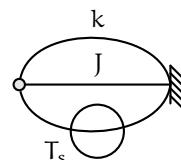
**coordinate arrow** Assign the sign by drawing a coordinate arrow, as shown at right. The direction of the arrow is arbitrary, however, if possible, assign the positive direction to match the sources. If the problem allows, it is best practice to have all sources and the coordinate arrow in the same direction. The right-hand rule is always implied.



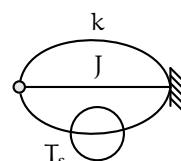
**draw linear graph without arrows** There are two nodes with distinct velocities: ground and the inertia, as shown at right. The inertia node is always drawn to ground. The spring connects between the inertia and ground. Finally, the torque source connects to the mass, where it is applied, and also connects to ground, which is impervious to it.



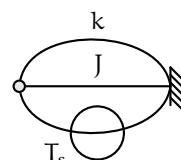
**assign spring and damper directions** On each inline spring and damper element, define the positive velocity drop and edge arrow to be *in the direction of the coordinate arrow*. Springs and dampers that aren't inline typically connect to ground, toward which edge arrows should point.



**assign inertia directions** On each inertia element, define the positive angular velocity drop and edge arrow to be *toward ground*. Sometimes we dash the latter half of the inertia edge to signify that it is “virtually” connected to ground.



**assign torque source directions** On each torque source element, define the positive direction as follows.



(ideal) If the torque source has the *same* definition of positive as your coordinate arrows,

**Table 02.2:** interpretation of the  
tational mechanical system sign convention.

torque $\tau$		angular velocity $\Omega$	
positive +	negative -	positive +	negative -
J torque <i>in</i> direction of the coordinate arrow	torque <i>opposite</i> the direction of the coordinate arrow	velocity <i>in</i> the coordinate arrow direction	velocity <i>opposite</i> the coordinate arrow direction
k wring!	wrong!	wrong!	velocity drops <i>in</i> the coordinate arrow direction
			velocity drops <i>opposite</i> the coordinate arrow direction
B wring!	wrong!	wrong!	velocity drops <i>in</i> the coordinate arrow direction
			velocity drops <i>opposite</i> the coordinate arrow direction

draw it *toward the node of application*.

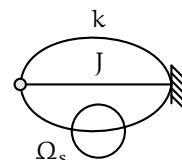
(if needed) If the torque source has the *opposite* definition of positive as your coordinate arrow, draw it *away from the node of application*.

**assign angular velocity source directions** On each angular velocity source element, define the positive direction as follows.

(ideal) If the source has the *same* definition of positive as your coordinate arrows, draw it *away from the node of application*.

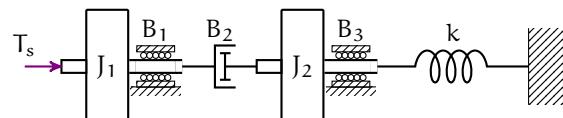
(if needed) If the source has the *opposite* definition of positive as your coordinate arrow, draw it *toward the node of application*.

This convention yields the interpretations of Table 02.2.



**Example 02.02-2 rotational mechanical sign convention**

For the system shown, draw a linear graph and assign signs according to the sign convention.



## Lecture 02.03 Element interconnection laws

The interconnections among elements constrain across- and through-variable relationships. The first element interconnection law requires the concept of a *contour* “”: a closed path that does not self-intersect superimposed over the linear graph. The first interconnection law is called the *continuity law*.

**contour****continuity law**

### Definition 02.03.1: continuity law

The sum of the through-variables that flow on *into* a contour on a linear graph is zero, or, in terms of generalized through-variables  $\mathcal{F}_i$  for  $N$  elements with through variables defined as positive into the contour,

$$\sum_{i=1}^N \mathcal{F}_i = 0. \quad (02.1)$$

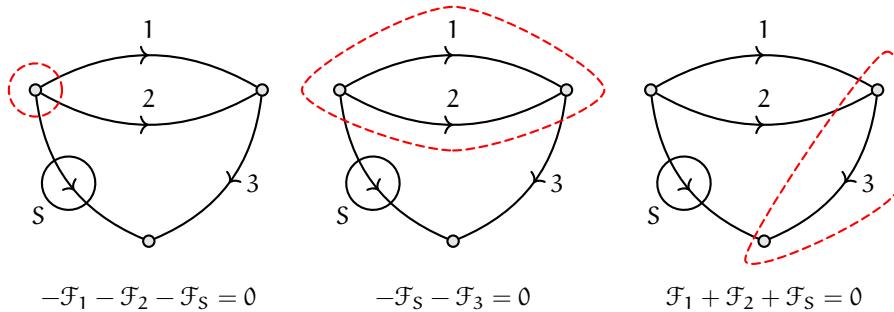
Contours can enclose any number of nodes and edges, as illustrated in Figure 02.5. Kirchhoff's current law (KCL) is the special case of the continuity law for electronic systems.

**KCL**

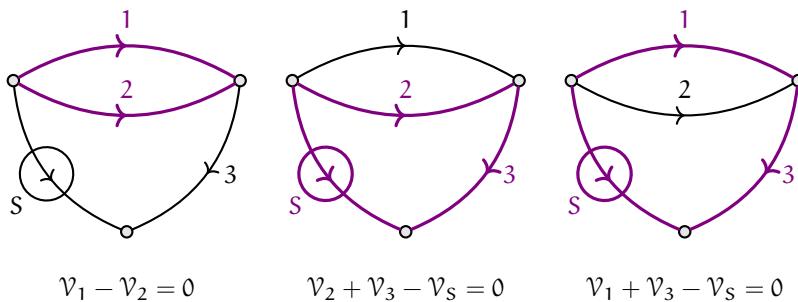
The second interconnection law we consider requires the concept of a *loop* “”: a continuous series of edges that begin and end at the same node, not reusing any edges.<sup>2</sup> The second interconnection law is called the

**loop**

<sup>2</sup>Technically, we need not restrict the definition to series that do not reuse edges for purposes of the compatibility law, but these loops are superfluous and we exclude them here.



**Figure 02.5:** illustration of different contours, denoted with red dashed lines “”, contours for which the continuity law applies, as shown below each graph.



**Figure 02.6:** illustration of different loops, denoted with violet edges “,” loops for which the compatibility law applies.

**compatibility law** *compatibility law.*

#### Definition 02.03.2: compatibility law

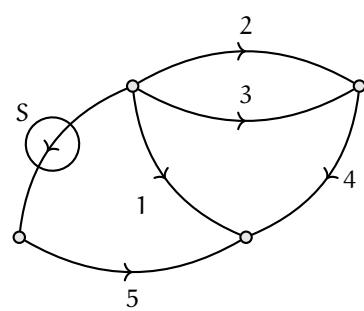
The sum of the across-variable drops on edges around any closed loop on a linear graph is zero, or, in terms of generalized across variables  $\mathcal{V}_i$  for  $N$  elements in a loop,

$$\sum_{i=1}^N \mathcal{V}_i = 0. \quad (02.2)$$

**KVL** A loop can be “inner” or “outer,” as shown in Figure 02.6. Kirchhoff’s voltage law (KVL) is the special case of the compatibility law for electronic systems.

#### Example 02.03-1 element interconnection laws

For the system shown, (a) write three unique continuity and three unique compatibility equations. Moreover, (b) write a continuity equation solved for  $\mathcal{F}_4$  in terms of  $\mathcal{F}_S$  and  $\mathcal{F}_1$ . Finally, (c) write a compatibility equation solved for  $\mathcal{V}_5$  in terms of  $\mathcal{V}_S$ ,  $\mathcal{V}_3$ , and  $\mathcal{V}_4$ .



## Lecture 02.04 Systematic linear graph modeling

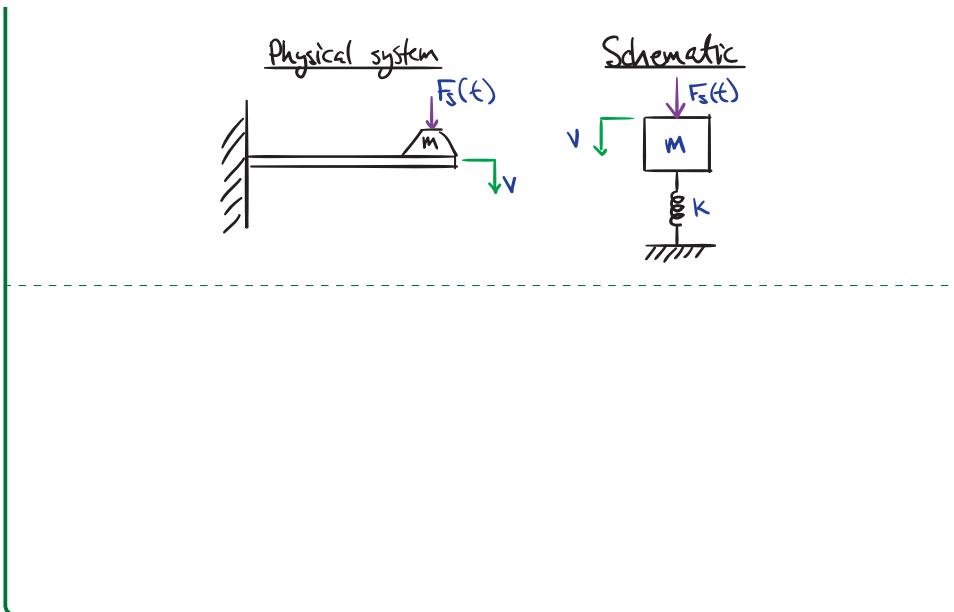
**system graph** A *system graph* is a representation of a physical system as a set of interconnected linear graph elements. The construction of a system graph requires a number of engineering decisions. In general, we can use the following procedure.

1. Define the system boundary and analyze the physical system to determine the essential features that must be included in the model, especially:
  - a) inputs,
  - b) outputs,
  - c) energy domains, and
  - d) key elements.
2. Form a schematic model of the physical system and assign schematic signs according to the sign convention of [Lecture 02.02](#).
3. Determine the necessary lumped-parameter elements representing the system's
  - a) energy sources,
  - b) energy storage, and
  - c) energy dissipation.
4. Identify the across-variables that define the linear graph nodes and draw a set of nodes.
5. Determine appropriate nodes for each lumped element and include each element in the graph.
6. Assign linear graph signs according to the sign convention of [Lecture 02.02](#).

The first three of these steps are the hardest. Considerable physical insight is required to construct an effective model. Often it is helpful—if not necessary—to have experimental results to guide the process.

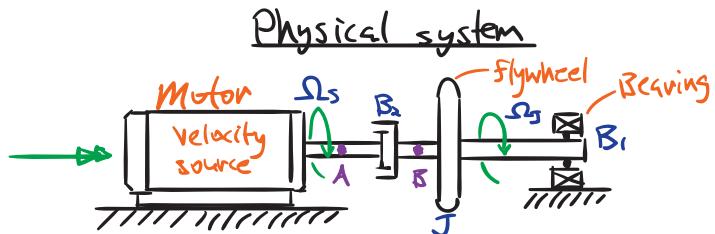
Example 02.04-1 linear graph model of translational mechanical system

For the system shown, develop a linear graph model.



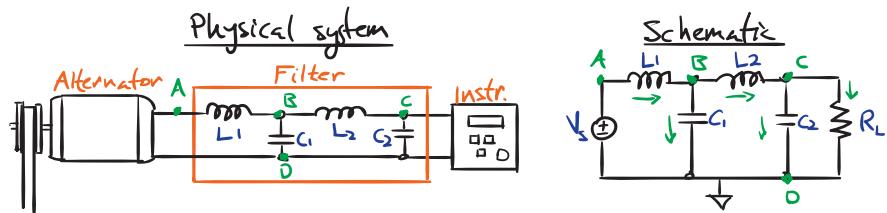
Example 02.04-2 linear graph model of rotational mechanical system

For the system shown, develop a linear graph model.



## Example 02.04-3 linear graph model of electronic system

For the system shown, develop a linear graph model.



## State-space models

## Lecture 03.01 State variable system representation

**state variables**

*State variables*, typically denoted  $x_i$ , are members of a minimal set of variables that completely expresses the *state* (or status) of a system. All variables in the system can be expressed algebraically in terms of state variables and *input variables*, typically denoted  $u_i$ .

**state**

A *state-determined system model* is a system for which

1. a mathematical description in terms of  $n$  state variables  $x_i$ ,
2. initial conditions  $x_i(t_0)$ , and
3. inputs  $u_i(t)$  for  $t \geq t_0$

**system order**

are sufficient conditions to determine  $x_i(t)$  for all  $t \geq t_0$ . We call  $n$  the *system order*.

**output variables**

**state vector**

The state, input, and *output variables* are all functions of time. Typically, we construct vector-valued functions of time for each. The so-called *state vector*  $x$  is actually a vector-valued function of time  $x : \mathbb{R} \rightarrow \mathbb{R}^n$ . The  $i$ th value of  $x$  is a state variable denoted  $x_i$ .

**input vector**

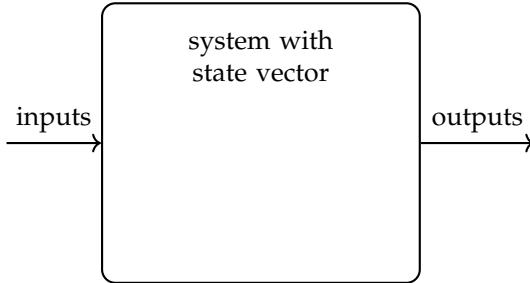
Similarly, the so-called *input vector*  $u$  is actually a vector-valued function of time  $u : \mathbb{R} \rightarrow \mathbb{R}^r$ , where  $r$  is the number of *inputs*. The  $i$ th value of  $u$  is an input variable denoted  $u_i$ .

**output vector**

Finally, the so-called *output vector*  $y$  is actually a vector-valued function of time  $y : \mathbb{R} \rightarrow \mathbb{R}^m$ , where  $m$  is the number of *outputs*. The  $i$ th value of  $y$  is an output variable denoted  $y_i$ .

Most systems encountered in engineering practice can be modeled as state-determined. For these systems, the number of state variables  $n$  is equal to the number of *independent energy storage elements*.

**independent energy storage elements**



**Figure 03.1:** block diagram of a system with input  $u$ , state  $x$ , and output  $y$ .

Since to know the state vector  $x$  is to know everything about the state, the energy stored in each element can be determined from  $x$ . Therefore, the time-derivative  $dx/dt$  describes the *power flow*.

power flow

The choice of state variables represented by  $x$  is not unique. In fact, any basis transformation yields another valid state vector. This is because, despite a vector's *components* changing when its basis is changed, a "symmetric" change also occurs to its *basis vectors*. This means *a vector is a coordinate-independent object*, and the same goes for vector-valued functions. This is not to say that there aren't invalid choices for a state vector. There are. But if a valid state vector is given in one basis, any basis transformation yields a valid state vector.

One aspect of the state vector *is* invariant, however: it must always be a vector-valued function in  $\mathbb{R}^n$ . Our method of analysis will yield a special basis for our state vectors. Some methods yield rather unnatural state variables (e.g. the third time-derivative of the voltage across a capacitor), but ours will yield natural state variables (e.g. the voltage across a capacitor or the force through a spring).

## Lecture 03.02 State and output equations

The state  $x$ , input  $u$ , and output  $y$  vectors interact through two equations:

$$\frac{dx}{dt} = f(x, u, t) \quad (03.1a)$$

$$y = g(x, u, t) \quad (03.1b)$$

where  $f$  and  $g$  are vector-valued functions that depend on the system.

**state-space model**

Together, they comprise what is called a *state-space model* of a system.

**state equation**

Let's not glide past these equations, which will be our dear friends for the rest of our analytic lives. The first equation (03.1a) is called the *state equation*. Given state and input vectors at a moment in time, it's function  $f$  describes, *how the state is changing* (i.e.  $dx/dt$ ). Clearly, the state equation is a vector differential equation, which is equivalent to a system of first-order differential equations.<sup>1</sup>

In accordance with the definition of a state-determined system from Lecture 03.01, given an initial condition  $x(t_0)$  and input  $u$ , the state  $x$  is determined for all  $t \geq t_0$ . The state-space model is precisely the “mathematical model” described in the definition of a state-determined system. Determining the state requires the solution—analytic or numerical—of the vector differential equation.

The second equation (03.1b) is *algebraic*. It expresses how the output  $y$  can be constructed from the state  $x$  and input  $u$ . This means we must first solve the state equation (03.1a). Since the output  $y$  is a vector of variables of interest, the output equation is constructed in two steps: (1) define the output variables and (2) write them in terms of the state variables  $x_i$  and input variables  $u_j$ .

Just because we know that, for a state-determined system, there exists a solution to Equation 03.1a, doesn't mean we know how to find it. In general,  $f : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R} \rightarrow \mathbb{R}^n$  and  $g : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R} \rightarrow \mathbb{R}^m$  can be nonlinear functions.<sup>2</sup> We don't know how to solve most nonlinear state equations analytically. An additional complication can arise when, in addition to states and inputs, system parameters are themselves time-varying (note the explicit time  $t$  argument of  $f$  and  $g$ ). Fortunately, often a linear model is sufficiently effective.<sup>3</sup>

<sup>1</sup>We'll learn how to solve such systems both analytically and numerically in later chapters.

<sup>2</sup>Technically, since  $x$  and  $u$  are themselves functions,  $f$  and  $g$  are *functionals*.

<sup>3</sup>A later lecture will describe the process of deriving a “linearized” model from a nonlinear one.

A *linear, time-invariant* (LTI) system has state-space model

**linear,  
time-invariant**

$$\frac{dx}{dt} = Ax + Bu \quad (03.2a)$$

$$y = Cx + Du \quad (03.2b)$$

where

- A is an matrix that describes how the changes the ,
- B is an matrix that describes how the changes the ,
- C is an matrix that describes how the contributes to the , and
- D is an matrix that describes how the contributes to the .

In the next two lectures, we will learn how to derive a state-space model—for linear systems, how to find A, B, C, and D—for a system *from its linear graph*. This is the link between the linear graph model and the state-space model.

## Lecture 03.03 Graphs to state-space I normal trees

**normal tree  
subgraph**

Before we introduce the algorithm for constructing the state-space model in [Lecture 03.04](#), we introduce the first step from the system graph to the state-space model: the *normal tree*. It is a *subgraph* of the system's linear graph.

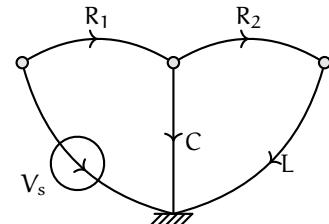
In the following, we will consider a connected graph with  $E$  edges, of which  $S$  are sources. There are  $2E - S$  unknown across- and through-variables, so that's how many equations we need. We have  $E - S$  elemental equations and for the rest we will write continuity and compatibility equations.  $N$  is the number of nodes.

The following rules must be respected.

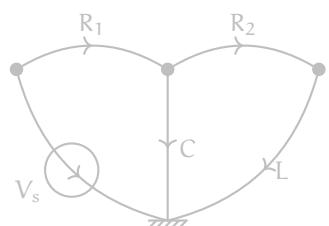
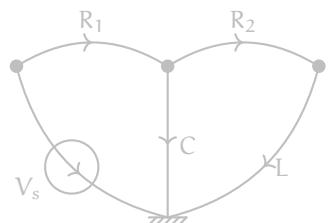
- R1. There can be no loops.
- R2. Every node must be connected.

Form a normal tree with the following steps. For an inline example, we will construct a normal tree from the linear graph for an electronic system, shown at right.

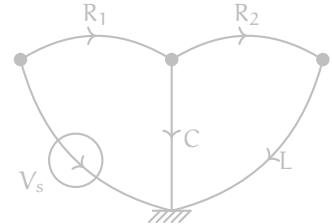
1. Include all nodes.



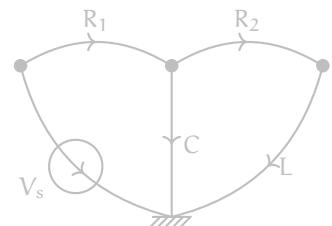
2. Include all across-variable sources.



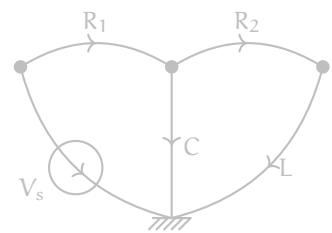
3. Select as many as possible A-type elements.



4. Select as many as possible D-type elements.



5. Select as many as possible T-type elements.



We call those edges in the normal tree its *branches* and those not, the *links*.

*A-type elements not in* and *T-type elements in* the normal tree are called *dependent energy storage elements*. All other A- and T-types are *independent energy storage elements*. The energy in these can be independently controlled.

In order to avoid an artificial excess in state variables and construct what is called a *controllable* model, whenever *A-types in series* (sharing one node) or *T-types in parallel* (sharing two nodes) appear, we should combine them to form equivalent elements in accordance with the formulas

$$C_e = \frac{1}{\sum_i 1/C_i} \quad \text{or} \quad (03.3a)$$

$$L_e = \frac{1}{\sum_i 1/L_i}. \quad (03.3b)$$

There are special names for power-flow variables associated with an element, depending on whether the element is a branch or link. *Primary variables* are: *across*-variables on branches and *through*-variables on links. *Secondary variables* are: *through*-variables on branches and *across*-variables on links.

**branches**  
**links**  
**dependent energy storage elements**  
**independent energy storage elements**  
**controllable**  
**A-types in series**  
**T-types in parallel**

**Primary variables**  
**Secondary variables**

## Lecture 03.04 Graphs to state-space II the algorithm

At long last, we consider an algorithm to generate a state-space model from a linear graph model. In the following, we will consider a connected graph with  $E$  edges, of which  $S$  are sources (split between through-variable sources  $S_T$  and across  $S_A$ ). There are  $2E - S$  unknown across- and through-variables, so that's how many equations we need. We have  $E - S$  elemental equations and for the rest we will write continuity and compatibility equations.  $N$  is the number of nodes.

- Derive  $2E - S$  independent differential and algebraic equations from elemental, continuity, and compatibility equations.

**normal tree**

**primary variables**

**secondary**

**variables**

**state variables**

**state vector**

**input vector**

**output vector**

**elemental equation**

**continuity equation**

**compatibility**

**equation**

- Draw a *normal tree*.
- Identify *primary* and *secondary variables*.
- Select the *state variables* to be *across-variables* on A-type branches and *through-variables* on T-type links.
- Define the *state vector*  $\mathbf{x}$ , *input vector*  $\mathbf{u}$ , and *output vector*  $\mathbf{y}$ .
- Write an *elemental equation* for each passive element.<sup>4</sup>
- Write a *continuity equation* for each passive branch by drawing a contour intersecting that and no other branch. Solve each for the secondary through-variable associated with that branch.<sup>5</sup>
- Write a *compatibility equation* for each passive link by temporarily “including” it in the tree and finding the compatibility equation for the resulting loop. Solve each for the secondary across-variable associated with that link.<sup>6</sup>

- Eliminate variables that are not state or input variables and their derivatives. The following procedure is recommended.

- Eliminate all secondary variables by substitution into the elemental equations of the continuity and compatibility equations.
- Reduce the resulting set of equations to  $n$  (system order) in state and input variables, only. If not elimination, use linear algebra.
- Write the result in standard form ([Equation 03.1a](#) or [Equation 03.2a](#)).

<sup>4</sup>There will be  $E - S$  elemental equations.

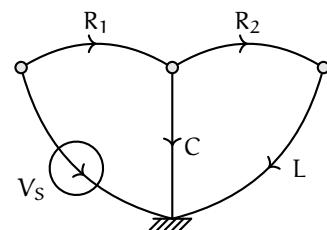
<sup>5</sup>There will be  $N - 1 - S_A$  independent continuity equations.

<sup>6</sup>There will be  $E - N + 1 - S_T$  independent compatibility equations.

- d) Express the output variables in terms of state and input variables, using any of the elemental, continuity, or compatibility equations.
- e) Write the result in standard form (Equation 03.1b or Equation 03.2b).

### Example 03.04-1 circuit state-space model

For the electronic system shown, find a state-space model with outputs  $i_L$ ,  $I_s$ , and  $v_{R_2}$ .





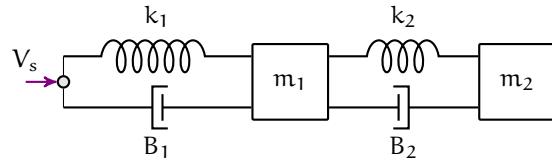


## Lecture 03.05 State-space model of a translational mechanical system

Let's try an example of a higher-order translational mechanical system.

Example 03.05-1 state-space model of a translational mechanical system

For the translational mechanical system shown, find a state-space model with outputs the spring forces and mass momenta.





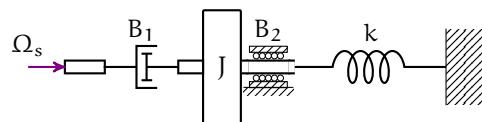


## Lecture 03.06 State-space model of a rotational mechanical system

Let's try an example of a rotational mechanical system.

Example 03.06-1 state-space model of a rotational mechanical system

For the rotational mechanical system shown, find a state-space model with outputs the spring torque and moment of inertia angular momentum.







## Lecture 03.07 Bridge between state-space and input-output differential equations and, oh yeah, transfer functions

**Laplace transform transfer function** The *Laplace transform*  $\mathcal{L}$  is cool af. It is used to solve differential equations and define the *transfer function*  $H$ : you know, just another awesome dynamic system representation. For now, we'll use it as a bridge between state-space and input/output differential equation representations, merely waving at transfer functions as we pass through. Later, transfer functions will be considered extensively.

**transfer function definition**

### 03.07.1 Transfer functions

Let a system have an input  $u$  and an output  $y$ . Let the Laplace transform of each be denoted  $U$  and  $Y$ , both functions of complex Laplace transform variable  $s$ . A *transfer function*  $H$  is defined as the ratio of the Laplace transform of the output over the input:

$$H(s) = \frac{Y(s)}{U(s)}. \quad (03.4)$$

The transfer function is exceedingly useful in many types of analysis. One of its most powerful aspects is that it gives us access to thinking about systems as operating on an input  $u$  and yielding an output  $y$ .

### 03.07.2 Bridging transfer functions and io differential equations

Consider a dynamic system described by the *input-output differential equation*—with variable  $y$  representing the *output*, dependent variable time  $t$ , variable  $u$  representing the *input*, constant coefficients  $a_i, b_j$ , order  $n$ , and  $m \leq n$  for  $n \in \mathbb{N}_0$ —as:

$$\begin{aligned} \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y &= \\ b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + \cdots + b_1 \frac{du}{dt} + b_0 u. \end{aligned} \quad (03.5)$$

**Laplace transform**

The *Laplace transform*  $\mathcal{L}$  of Equation 03.5 yields something interesting

(assuming zero initial conditions):

$$\begin{aligned}
 & \mathcal{L} \left( \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y \right) = \\
 & \mathcal{L} \left( b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + \cdots + b_1 \frac{du}{dt} + b_0 u \right) \Rightarrow \\
 & \mathcal{L} \left( \frac{d^n y}{dt^n} \right) + a_{n-1} \mathcal{L} \left( \frac{d^{n-1} y}{dt^{n-1}} \right) + \cdots + a_1 \mathcal{L} \left( \frac{dy}{dt} \right) + a_0 \mathcal{L} (y) = \\
 & b_m \mathcal{L} \left( \frac{d^m u}{dt^m} \right) + b_{m-1} \mathcal{L} \left( \frac{d^{m-1} u}{dt^{m-1}} \right) + \cdots + b_1 \mathcal{L} \left( \frac{du}{dt} \right) + b_0 \mathcal{L} (u) \Rightarrow \\
 & s^n Y + a_{n-1} s^{n-1} Y + \cdots + a_1 s Y + a_0 Y = \\
 & b_m s^m U + b_{m-1} s^{m-1} U + \cdots + b_1 s U + b_0 U.
 \end{aligned}$$

Solving for  $Y$ ,

The inverse Laplace transform  $\mathcal{L}^{-1}$  of  $Y$  is the *forced response*. However, this is not our primary concern; rather, we are interested to solve for the transfer function  $H$  as the ratio of the output transform  $Y$  to the input transform  $U$ , i.e. **forced response**

$$H(s) \equiv \frac{Y(s)}{U(s)} \quad (03.6)$$

$$= \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}. \quad (03.7)$$

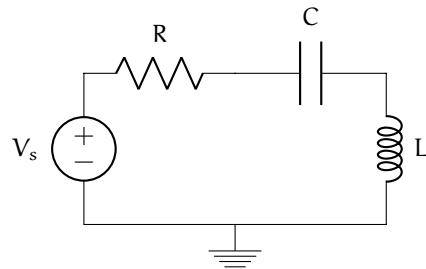
Exactly the reverse procedure, then, can be used to convert a given transfer function to an input-output differential equation.

**Example 03.07-1 A circuit transfer function**

The circuit shown has input-output differential equation

$$L \frac{d^2v_L}{dt^2} + R \frac{dv_L}{dt} + \frac{1}{C} v_L = L \frac{d^2V_s}{dt^2}.$$

What is the transfer function from  $V_s$  to  $v_L$ ?



### 03.07.3 Bridging transfer functions and state-space models

Given a system in the standard form of a state equation,

$$\frac{dx}{dt} = Ax + Bu,$$

we take the Laplace transform to yield, assuming zero initial conditions,

which can be solved for the state:

where  $I$  is the identity matrix with the same dimension as that of  $A$ . The standard form of the output equation yields the output solution

$$Y = Hu, \quad (03.9)$$

**matrix transfer function**

where we define the *matrix transfer function*  $H$  to be

The element  $H_{ij}$  is a transfer function from the  $j$ th input  $U_j$  to the  $i$ th output  $Y_i$ .

The reverse procedure of deriving a state-space model from a transfer function is what is called a *state-space realization*, which is not a unique operation (there are different realizations for a single transfer function) and is not considered here.

state-space  
realization

Example 03.07-2 Matrix transfer function from state-space

Given the linear state-space model

$$\begin{aligned}\dot{x} &= \begin{bmatrix} -3 & 4 \\ -1 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u,\end{aligned}$$

derive the matrix transfer function.

**Example 03.07-3 state-space to io differential equations**

For the following state-space model, derived in Example 03.04-1,  
derive the io differential equations for each output variable:

$$\begin{aligned}\frac{dx}{dt} &= \begin{bmatrix} -\frac{1}{R_1 C} & -\frac{1}{C} \\ 1/L & -R_2/L \end{bmatrix} x + \begin{bmatrix} \frac{1}{R_1 C} \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & 1 \\ -1/R_1 & 0 \\ 0 & R_2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/R_1 \\ 0 \end{bmatrix} u.\end{aligned}$$

The output variables are  $i_L$ ,  $I_S$ , and  $v_{R_2}$ .

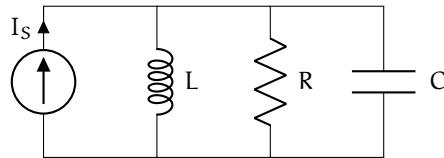


## Lecture 03.08 Exercises

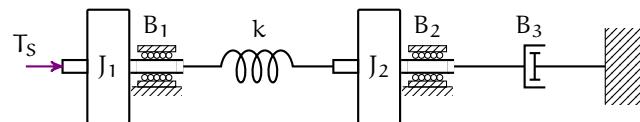
### Exercise 03.1 Linear graph models

Draw necessary sign coordinate arrows, a *linear graph*, a *normal tree*, and identify *state variables* and *system order* for each of the following schematics.

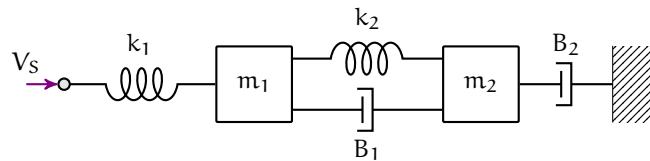
- a. electronic system, current source



- b. rotational mechanical system, torque source



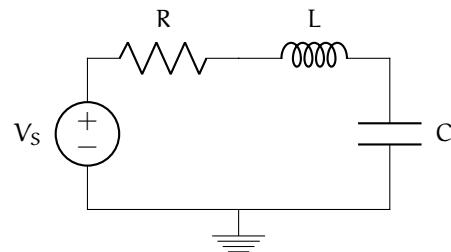
- c. translational mechanical system, velocity source



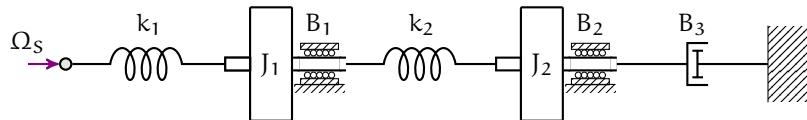
### Exercise 03.2 Linear graph models

Draw necessary sign coordinate arrows, a *linear graph*, a *normal tree*, and identify *state variables* and *system order* for each of the following schematics.

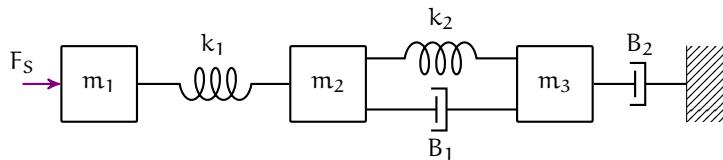
- a. electronic system, voltage source



- b. rotational mechanical system, angular velocity source

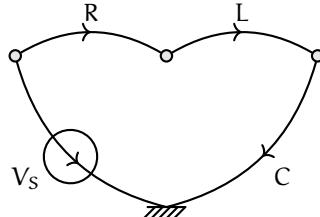


- c. translational mechanical system, force source



### Exercise 03.3 State-space model from a linear graph model

Use the following linear graph for a circuit to answer the questions below, which are the steps to determining a state-space model of the circuit. Use the sign convention from the diagram.  $V_S$  is a voltage source.

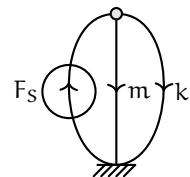


- Determine the *normal tree*, *state variables*, *system order*, *state vector*, *input vector*, and *output vector* for the outputs  $i_R$  and  $v_C$ .
- Write the required *elemental*, *continuity*, and *compatibility equations*.
- Solve for the *state equation* in standard form.
- Solve for the *output equation* in standard form.

### Exercise 03.4 State-space model from a linear graph model

Use the following linear graph for a mechanical translational system to answer the questions below, which are the steps to determining a state-space model from the linear graph.

Use the sign convention from the diagram.  $F_S$  is a force source. Let the outputs be  $v_m$  and  $f_k$ .



- a. Determine the *normal tree*, *state variables*, *system order*, *state vector*, *input vector*, and *output vector*.
- b. Write the required *elemental*, *continuity*, and *compatibility equations*.
- c. Solve for the *state equation* in standard form.
- d. Solve for the *output equation* in standard form.

## Electromechanical systems

## Lecture 04.01 Ideal transducers

**two-port elements  
transducers**

**motor**

*Two-port* system elements can model *transducers*—elements that transfer energy between two energy domains or change its form within an energy domain. The quintessential example, which we will consider in detail, is the *motor*, which converts electrical energy to mechanical energy. However, many other system elements can be considered transducers, and we'll consider a few in this lecture.

Each of the two ports has a through- and an across-variable. We use the convention that the power *into* each port ( $\mathcal{P}_1$  and  $\mathcal{P}_2$ ) is positive, which has implications for the signs of the power flow variables  $\mathcal{F}_1$ ,  $\mathcal{F}_2$ ,  $\mathcal{V}_1$ , and  $\mathcal{V}_2$ . For an two-port element to transfer power, we have

**transformer ratio**

We define the *transformer ratio* TF to be

$$\text{TF} \equiv \frac{\mathcal{V}_1}{\mathcal{V}_2} = -\frac{\mathcal{F}_2}{\mathcal{F}_1}. \quad (04.1)$$

**gyrator modulus**

Furthermore, we define the *gyrator modulus* GY to be

$$\text{GY} \equiv \frac{\mathcal{V}_1}{\mathcal{F}_2} = -\frac{\mathcal{V}_2}{\mathcal{F}_1}. \quad (04.2)$$

**ideal transducer**

For an *ideal transducer*—one that is linear, time-invariant, and without power loss—we have only two nontrivial solutions:<sup>1</sup>

$$\begin{aligned} \mathcal{V}_2 &= \mathcal{V}_1/\text{TF} & \text{or} & & \mathcal{V}_2 &= -\text{GY}\mathcal{F}_1 \\ \mathcal{F}_2 &= -\text{TF}\mathcal{F}_1 & & & \mathcal{F}_2 &= \mathcal{V}_1/\text{GY}. \end{aligned}$$

**transformer  
gyrator**

For a given element, if the solution with TF is a good model, we call that element a *transformer*. If the GY solution is a good model, we call it a *gyrator*.

<sup>1</sup>For an explanation of *why* that is the case, see Rowell and Wormley (1997).

**Example 04.01-1 DC motor**

Consider a DC motor with rotor radius  $r$ , number of coil turns  $N$ , background field  $B$ , and rotor length  $\ell$ . The torque  $T$  of a DC motor is related to its coil current  $i$  by the relation

$$T = -2rNB\ell i.$$

1. Determine if DC motors are transformers or gyrators.
2. Find TF or GY.
3. Derive the relation between the voltage  $v$  and the angular velocity  $\Omega$  across the motor using the assumption that it is an ideal transducer.

**Example 04.01-2 gears**

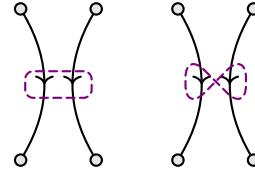
Consider two gears with radii  $r_1$  and  $r_2$  and number of teeth  $n_1$  and  $n_2$ .

1. Determine the power flow variables for gears.
2. Write two independent equations relating the power flow variables.
3. Determine if gears are transformers or gyrators.
4. Find TF or GY.



## Lecture 04.02 Modeling with transducers

We now develop both linear graph and state-space models of systems that include transducers. Linear graphs of two-port ideal transducer elements are drawn as shown in [Figure 04.1](#). Once again, we use the sign convention that power into an element is positive. Often, the edges are drawn toward ground nodes, which are always different when the transducer acts between different energy domains. Transducers may or may not be sufficiently modeled by ideal transducers. For instance, we may need to consider the moment of inertia associated with a gear. When this is the case, additional elements can be connected in parallel and in series with the two-port element nodes. DC motors—another example—are typically not modeled with an ideal transducer, alone, because the windings have both resistance and inductance.



**Figure 04.1:** two-port ideal linear graph elements of a transformer (left) and a gyrator (right).

### 04.02.1 State-space modeling with transducers

We present a method for constructing a state-space model of systems containing transducer elements. This procedure begins, as before, with the construction of the normal tree. The following rules must be respected.

- R1. There can be no loops.
- R2. Every node must be connected.
- R3. Of a transformer's two edges, exactly one is included.
- R4. Of a gyrator's two edges, either both are or neither is included.

Form a normal tree with the following steps.

1. Include all nodes.
2. Include all across-variable sources.
3. Include as many as possible A-type elements.
4. Include transducer edges, minimizing the number of T-types in the tree.
5. Include as many as possible D-type elements.
6. Include as many as possible T-type elements.

The state and output equations can be derived as before, but with the following caveat: each two-port element requires two elemental equations.

## Lecture 04.03 DC motors

DC motors are commonly used in mechanical engineering designs as an actuator. Products such as pumps, fans, conveyors, and robots use DC motors to convert electrical energy to mechanical (rotational) energy.

DC motors first emerged in the mid-19th century as the first device to produce useful mechanical work from electrical power.<sup>2</sup> One of fathers of the DC motor, the Benedictine priest Ányos Jedlik, invented the key facets of the motor: the *stator*, the *rotor*, and the *commutator*. Roughly speaking, for a typical brushed DC motor, current flowing through the wire windings of the stator produces a magnetic field that turns the rotor, which has windings of its own; the commutator mechanically switches the direction of current flow through the windings to yield continuous electromagnetic torque.

We will begin our study of DC motors with a review of a key physical phenomenon: the mechanical force on a charged particle moving in a magnetic field.

### 04.03.1 Lorentz force

Consider a charged particle moving through a background magnetic field. The *Lorentz force* is the (mechanical!) force on the particle, which depends on the velocity of the particle, the background magnetic field, and the background electric field. Charge flowing through a straight, stationary<sup>3</sup> wire with current  $i$  in a uniform background magnetic field  $B$  is subject to the cumulative effect of the Lorentz force on each charge. Let the straight wire's length and orientation in the  $B$ -field be described by the vector  $\ell$ , which should be chosen to be in the direction of positive current flow. It can be shown that the resultant force  $f$  on the wire is

$$f = i\ell \times B \quad (04.3)$$

as shown in Figure 04.2.

With a curved wire, then, we could take infinitesimal sections  $d\ell$  and integrate along the wire's path:

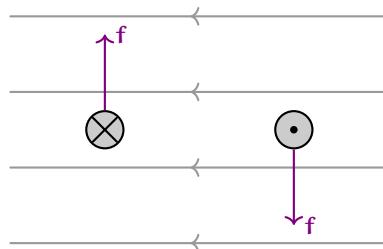
$$f = i \int d\ell \times B. \quad (04.4)$$

---

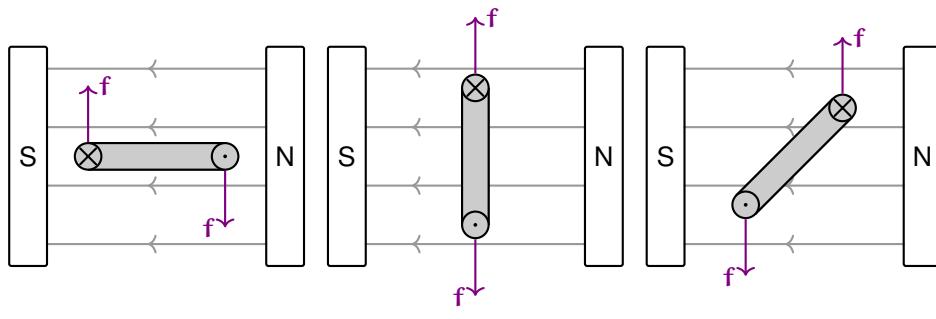
<sup>2</sup>See a decent history [here](#).

<sup>3</sup>The equations here assume a stationary wire. In a DC motor, the wire is moving, which creates additional effects, but the Lorentz force is still present.

stator  
rotor  
commutator



**Figure 04.2:** the forces  $f$  on two wires in a magnetic field  $B$ . The wire on the left has current flowing *into* the board, that on the right has current flowing *out of* the board. The cross-product right-hand-rule applies.



(a) Orientation A: horizontal.

(b) Orientation B: vertical.

(c) Orientation C: mixed.

**Figure 04.3:** axial section view of a simple DC motor with permanent magnets.

DC motors take advantage of this electromechanical phenomenon by driving current through cleverly arranged wires to generate torque on a shaft.

#### 04.03.2 Permanent magnet DC motors

In order to take advantage of the Lorentz force, first a uniform background magnetic field  $B$  is required. Some DC motors, called *permanent magnet DC motors* (PMDC motors) generate this field with two stationary permanent magnets arranged as shown in Figure 04.3. The magnets are affixed to the “stationary” part of the motor called the *stator*.

**PMDC motors**

**stator**

Now consider a rigidly supported wire with current  $i$  passing through the field such that much of its length is perpendicular to the magnetic field. Consider the resultant forces on these perpendicular sections of wire for different wire configurations, as illustrated in Figure 04.3. We have torque!

**armature** But note that it changes direction for different armature orientations, which will need to be addressed in a moment. Note that we can wind this wire—which we call the *armature*—multiple times around the loop to increase the torque. The rotating bit of the motor that supports the armature is called the *rotor*, which includes the shaft.

**not #winning** The trouble is, if we connect our armature up to a circuit—which is usually located alongside the stator, i.e. *not rotating*—the wire will wrap about itself, which is *not #winning*. But we’re tricky af so let’s consider just cutting that wire and rigidly connecting it to a disk—called a *commutator*—with two conductive regions, one for each terminal of the armature. The commutator will rotate with the armature, but it provides smooth contacts along the perimeter of the disk.

**brushes** We can then connect the driving circuit to these contacts via *brushes*: conductive blocks pressed against the commutator on opposite sides such that they remain in contact (conducting current) yet allow the commutator to slide easily, as shown in Figure 04.4. Brushes are typically made from carbon and wear out over time. This is partially mitigated by spring-loading, but eventually the brushes must be replaced, nonetheless.

So brushes solve the “wire wrapping” problem, but do they have an effect on the “torque flipping” issue? Yes! When the armature passes through its vertical orientation, *current reverses direction through the armature*. So whenever the perpendicular section of wire is on the right, current flows in the same direction, regardless of to which side of the armature it belongs.

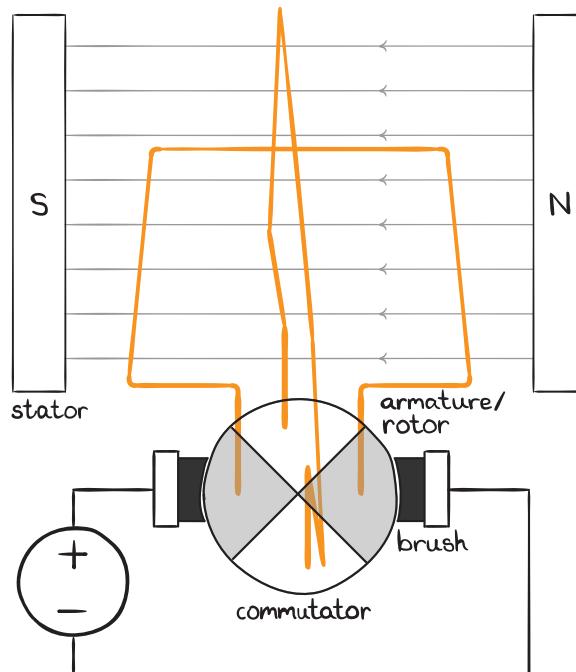
Finally, is there a way to overcome the limitation of torque variation with different armature angles? Yes: if there are several different armature windings at different angles and correspondingly the commutator is split into several conductive contact pairs (one for each armature winding), a relatively continuous torque results! Real PMDC motors use this technique.

#### 04.03.3 Wound stator DC motors

**wound stator DC motors** *Wound stator DC motors* operate very similarly to PMDC motors, but generate their background field with two stationary coils in place of the permanent magnets, above. These electromagnets require a current of their own, which is usually provided through the same circuitry that supplies the armature current (DC motors typically have only two terminals).

Three common configurations of the electrical connection of these are shown in Figure 04.5. These define the following three DC motor types.

**shunt DC motors** **shunt** The *shunt DC motor* has its stator and rotor windings connected in



**Figure 04.4:** illustration of brushes, commutator, and two armatures.

parallel. These are the most common wound stator DC motors and their speeds can be easily controlled without feedback, but they have very low starting torque.

**series** The *series DC motor* has stator and rotor windings connected in series. These have high starting torque—so high, in fact, that it is not advisable to start these motors without a load—but their speeds are not as easily controlled without feedback.

**series DC motors**

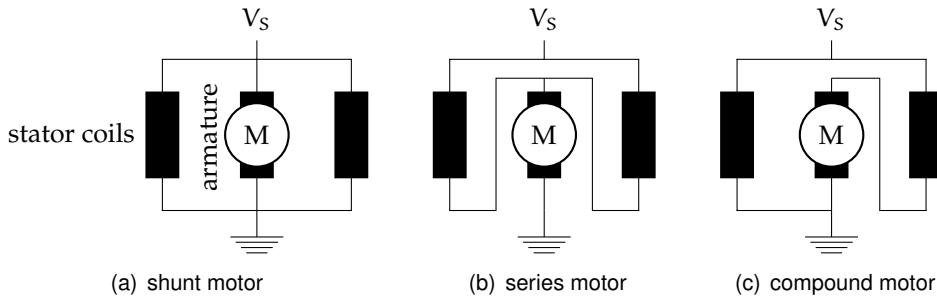
**compound** The *compound DC motor* has stator and rotor windings connected in both series and parallel. These can exhibit characteristics that mix advantages and disadvantages of shunt and series DC motors.

**compound DC motors**

#### 04.03.4 Brushless DC motors

There is yet another type of DC motor: *brushless (BLDC)*. Brushless DC motors work on principles more similar to AC motors, but require complex solid-state switching that must be precisely timed. As their name implies,

**brushless (BLDC) motors**



**Figure 04.5:** connections for shunt, series, and compound DC motors.

these motors do not require brushes. A brushless DC motor mathematical model is not presented here, but a nice introduction is given by [Baldursson \(2005\)](#).

The brushed DC motor is still widely used, despite its limitations, which include relatively frequent maintenance to replace brushes that wear out or clean/replace commutators. Other disadvantages of brushed DC motors include their relatively large size, relatively large rotor inertia, heat generated by the windings of the stator and/or rotor, and arcing that creates electronic interference for nearby electronics. Reasons they are still widely used include that they are inexpensive (about half the cost of brushless DC motors), don't require (but often still use) complex driving circuits, are easy to model, and are easily driven at different speeds; for these reasons, an additional reason emerges: they're relatively easy to design with!

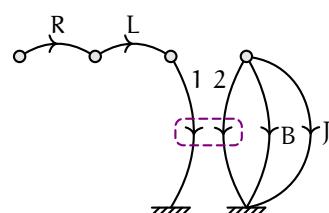
### 04.03.5 A PMDC motor model

We have already explored a model for a PMDC motor in [Example 04.01-1](#), which yielded elemental equations

$$T_2 = -TFi_1 \quad \text{and} \quad (04.5)$$

$$\Omega_3 \equiv v_1/T E \quad (04.6)$$

where  $TF$  is the motor constant. That model assumed neither armature resistance nor inductance were present—that is, it was an ideal transformer model. A linear graph of a much better model for a DC motor is shown in Figure 04.6. This model



**Figure 04.6:** a better brushed DC motor model.

includes a resistor  $R$  and inductor  $L$  in series with an ideal transducer. On the mechanical side, the rotor inertia  $J$  and internal bearing damping  $B$  are included. The tail ends of  $R$  and  $2$  should be connected to external electrical and mechanical subgraphs, respectively.

#### 04.03.6 Motor constants

The *motor torque constant*  $K_t$  and *back-emf voltage constant*  $K_v$  are related to the transformer ratio  $TF$  derived above to characterize a brushed DC motor's response. If expressed in a set of consistent units—say, SI units— $K_t$  and  $K_v$  have the same numerical value and are equivalent to  $TF$ . Precisely, with consistent units,  $TF = K_v = K_t$ .

However, manufacturers usually use weird units like oz-in/A and V/krpm. If they are given in anything but SI units, we recommend converting to SI for analysis.

Once in SI, we will have something like (for  $x \in \mathbb{R}$ ):

**motor torque  
constant**  
**back-emf voltage  
constant**

So if we are given either  $K_t$  or  $K_v$ , the unknown constant can be found (in SI units) by converting the known constant to SI.<sup>4</sup>

#### 04.03.7 Animations

There are some great animations of DC motor operating principles and construction. I've included the url of my favorite, along with some bonus animations for other important types of motors we don't have time to discuss, here.

- Brushed DC motors: [youtu.be/LAtPHANEfQo](https://youtu.be/LAtPHANEfQo)
- Brushless DC motors: [youtu.be/bCEiOnuODac](https://youtu.be/bCEiOnuODac)
- AC (asynchronous) induction motors: [youtu.be/AQqyGNOP\\_3o](https://youtu.be/AQqyGNOP_3o)
- AC synchronous motors: [youtu.be/Vk2jDXxZIhs](https://youtu.be/Vk2jDXxZIhs)
- Stepper motors: [youtu.be/eyqwLiowZiU](https://youtu.be/eyqwLiowZiU)

---

<sup>4</sup>One more note. When given a torque constant, the unit "oz" means "ounce-force," which is the mass in regular (mass) ounces multiplied by the gravitational acceleration  $g$ .

## Lecture 04.04 Modeling a real electromechanical system

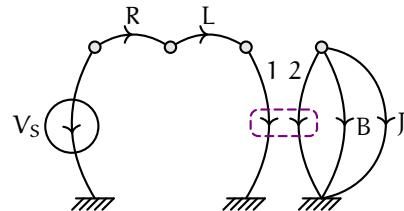
We now model the electromechanical systems from the laboratory, shown in [Figure 04.7](#). The system includes a brushed DC motor (*Electrocraft 23SMDC-LCSS servomotor from Servo Systems*), two shafts, a shaft coupler, two bearings, and a flywheel. The motor's datasheet specifications are given in [Table 04.1](#). The mechanical subsystem's inertia is dominated by the stainless steel flywheel with  $J_f = 0.324 \cdot 10^{-3} \text{ kg}\cdot\text{m}^2$ . The bearing damping  $B_b$  is the most difficult parameter to determine. Let's begin with the assumption that the combined bearing damping is  $B_b = 20 \cdot 10^{-6} \text{ N}\cdot\text{m}/(\text{rad}/\text{s})$ .



**Figure 04.7:** electromechanical systems from the lab.

### 04.04.1 Linear graph model

A linear graph model is in order. An ideal voltage source drives the motor<sup>6</sup>—modeled as an ideal transducer with armature resistance  $R$  and inductance  $L$ , given in [Table 04.1](#). The ideal transducer's rotational mechanical side (2) is connected to a moment of inertia  $J = J_m + J_f = 0.381 \cdot 10^{-3} \text{ kg}\cdot\text{m}^2$ , dominated by the flywheel,<sup>7</sup> and damping  $B$ , which is the parallel combination of the internal motor damping of [Table 04.1](#) and the bearing damping  $B_d$ , to yield  $B = 26.9 \cdot 10^{-6} \text{ N}\cdot\text{m}/\text{s}^2$ . We choose to ignore the flexibility of the coupler. Problem [04.08.1](#) considers the same system but does not ignore the coupler's flexibility. In general, shaft couplers



**Figure 04.8:** a linear graph model of the electromechanical systems of [Figure 04.7](#).

<sup>5</sup>Load applied at one inch from bearing.

<sup>6</sup>Often we can model our motor-driving source as ideal within an operating range. See [Lecture 04.07](#) for more details.

<sup>7</sup>This is the sum of the inertia of the flywheel  $J_f = 0.324 \cdot 10^{-3} \text{ kg}\cdot\text{m}^2$  and the rotor  $J_m = 0.0565 \cdot 10^{-3} \text{ kg}\cdot\text{m}^2$ . It might be worthwhile combining this with the inertia from the shaft and coupler to obtain a more accurate value, but the difference is likely negligible.

**Table 04.1:** datasheet specifications for the *Electrocraft 23SMDC-LCSS* servomotor from *Servo Systems*. This is the motor used in the lab.

	parameter	specification	SI conversion
general	continuous stall torque	55 oz-in	0.388 N-m
	peak torque $T_{\max}$	400 oz-in	2.82 N-m
	max terminal voltage	60 V <sub>dc</sub>	60 V <sub>dc</sub>
	max operating speed $\Omega_{\max}$	6000 rpm	628 rad/s
mechanical	rotor inertia $J_m$	0.008 oz-in/s <sup>2</sup>	$56.5 \cdot 10^{-6}$ N-m/s <sup>2</sup>
	damping constant $B_m$	0.25 oz-in/krpm	$16.9 \cdot 10^{-6}$ N-m/(rad/s)
	thermal resistance	4 C/W	4 K/W
	max armature temp	155 C	428 K
	max friction torque	3 oz-in	0.0212 N-m
	max radial load <sup>5</sup>	10 lb	44.5 N
	weight (motor only)	3.5 lb	15.6 N
electrical	torque constant $K_t$	13.7 oz-in/A	0.097 N-m/A
	voltage constant $K_v$	10.2 V/krpm	0.097 V/(rad/s)
	terminal resistance	1.6 Ω	1.6 Ω
	electrical time constant	2.6 ms	$2.6 \cdot 10^{-3}$ s
	mechanical time constant	8.9 ms	$8.9 \cdot 10^{-3}$ s
	max continuous current	4 A	4 A
	armature inductance	4.1 mH	$4.1 \cdot 10^{-3}$ H
	max peak current	34 A	34 A

have significant flexibility and, depending on the application, this may require consideration in the dynamic model.

#### 04.04.2 State-space model

The normal tree can be constructed by the procedure from [Lecture 04.02](#). The voltage source  $V_S$  is first included, followed by  $J$ . Then exactly one edge of the ideal transducer must be selected, minimizing the number of T-types in the tree. We don't really have a choice, in this case, because selecting edge 2 would create a loop, so we must select edge 1. Next,  $R$  is included. No more elements can be included without creating a loop, so we are finished.

We are now prepared to determine variables. The state variables are across variables of A-type tree branches and through variables of T-type

links—so  $\Omega_J$  and  $i_L$ , and the system is second-order ( $n = 2$ ). Clearly, the system's input is the voltage source  $V_S$ . We are interested in all the variables for the analysis in [Lecture 04.05](#), so we choose them all for our outputs. In summary, then, the state, input, and output vectors are:

$$\mathbf{x} = \begin{bmatrix} \Omega_J \\ i_L \end{bmatrix}, \mathbf{u} = [V_s], \text{ and}$$

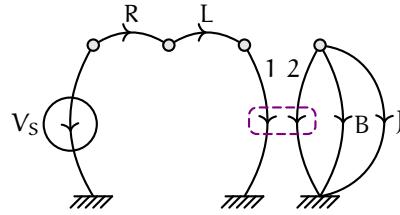
$$\mathbf{y} = [\Omega_J \quad T_J \quad v_L \quad i_L \quad \Omega_B \quad T_B \quad v_R \quad i_R \quad v_1 \quad i_1 \quad \Omega_2 \quad T_2 \quad V_s \quad I_s]^\top.$$

Let's write some equations! Elemental are up first.

J	
L	
B	

R	
1	
2	

Now, continuity and compatibility equations are developed by summing through-variables into contours. The three required contours—one for each of R, 1, and J—can be drawn on [Figure 04.9](#). The three compatibility equations—one for each of L, 2, and B—are found by “temporarily including” those links in the tree and summing across-variables around the loops created. Let's write the equations.



**Figure 04.9:** the linear graph model for drawing contours.

branch	continuity equation	link	compatibility equation
R		L	
1		2	
J		B	

All that remains to form the state-space model is to eliminate variables that are neither states nor inputs from the elemental, continuity, and

compatibility equations. Eliminating secondary variables by substituting the continuity and compatibility equations into the elemental equations, the following results.

$$\begin{array}{c|c} J & R \\ \hline L & 1 \\ B & 2 \end{array}$$

The last four equations allow us to eliminate the remaining undesirable variables to obtain the state model in the standard form<sup>8</sup>

$$\frac{dx}{dt} = Ax + Bu \quad (04.7a)$$

$$y = Cx + Du \quad (04.7b)$$

where

$$A = \begin{bmatrix} -B/J & TF/J \\ -TF/L & -R/L \end{bmatrix}, \quad (04.7c)$$

$$B = \begin{bmatrix} 0 \\ 1/L \end{bmatrix}, \quad (04.7d)$$

$$C = \begin{bmatrix} 1 & -B & -TF & 0 & 1 & B & 0 & 0 & TF & 0 & 1 & 0 & 0 & 0 \\ 0 & TF & -R & 1 & 0 & 0 & R & 1 & 0 & 1 & 0 & -TF & 0 & 1 \end{bmatrix}^\top, \text{ and} \quad (04.7e)$$

$$D = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]^\top. \quad (04.7f)$$

---

<sup>8</sup> Here is the `rnd` file for use with Statum ([statum.camerondevine.me](http://statum.camerondevine.me)) to derive the state-space model from the elemental, continuity, and compatibility equations:

`ricopic.one/dynamic_systems/source/motor_model.rnd`

Note that the “constraint equations” are the continuity and compatibility equations solved for primary variables.

## Lecture 04.05 DC motor performance in steady-state

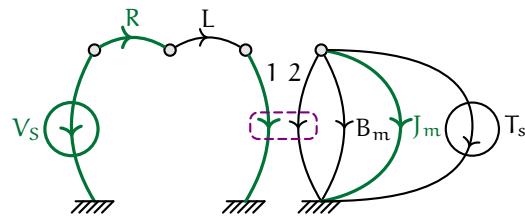
**motor curve**

Brushed DC motor performance has several aspects, but most of them revolve around the so-called *motor curve*: for a given motor voltage, its steady-state speed versus a constant torque applied to the load. The test setup for drawing such a curve requires a calibrated, controllable torque source applied to the motor shaft.

**brake**  
**magnetic particle**  
**brake**

A *brake* is typically used. A voltage-controlled *magnetic particle brake* is ideal.<sup>9</sup>

We will gain a deep understanding of DC motor performance characteristics only by tarrying with this situation. Therefore, we begin by modeling it in 04.05.1 and analyzing its performance in 04.05.2.



**Figure 04.10:** a linear graph model of the motor from Lecture 04.04 in a test-configuration with a brake modeled by  $T_s$ .

### 04.05.1 Modeling the test system

Including a torque source  $T_s$  on the load changes the model only slightly, as shown in Figure 04.10. Note that the mechanical subsystem is reduced to only the motor, since during such a test the load and bearings would be detrimental (it is a test for the *motor*, after all). Invariant are the normal tree, state variables, and most of the derivation of the state equations.

The input vector becomes

$$\mathbf{u} = \begin{bmatrix} V_s \\ T_s \end{bmatrix}. \quad (04.8)$$

The continuity equation for the inertia becomes  $T_{J_m} = -T_2 - T_{B_m} - T_s$  (the torque specifically *opposes* motion, to which we assign the positive

<sup>9</sup>See, for instance [here](#) or [here](#).

direction) and the state model's matrices B and D change, such that<sup>10</sup>

$$A = \begin{bmatrix} -B_m/J_m & TF/J_m \\ -TF/L & -R/L \end{bmatrix}, \quad (04.9a)$$

$$B = \begin{bmatrix} 0 & -1/J_m \\ 1/L & 0 \end{bmatrix} \quad (04.9b)$$

$$C = \begin{bmatrix} 1 & -B_m & -TF & 0 & 1 & B_m & 0 & 0 & TF & 0 & 1 & 0 & 0 & 0 \\ 0 & TF & -R & 1 & 0 & 0 & R & 1 & 0 & 1 & 0 & -TF & 0 & 1 \end{bmatrix}^\top, \quad (04.9c)$$

$$D = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top. \quad (04.9d)$$

#### 04.05.2 Steady-state performance analysis

Let's begin by defining the system parameters.

```
Kt_spec = 13.7; % oz-in/A ... torque constant from spec
Kv_spec = 10.2; % V/krpm ... voltage constant from spec
Tmax_spec = 2.82; % N-m ... max (stall) torque from spec
Omax_spec = 628; % rad/s ... max speed (no load) from spec
N_oz = 0.278013851; % N/oz
m_in = 0.0254; % m/in
Kt_si = Kt_spec*N_oz*m_in; % N-m/A
rads_krpm = 1e3*2*pi/60; % (rad/s)/krpm
Kv_si = Kv_spec/rads_krpm; % V/(rad/s)
Jm = 56.5e-6; % kg-m^2 ... inertia of rotor
Bm = 16.9e-6; % N-m/s^2 ... motor damping coef
R = 1.6; % Ohm ... armature resistance
L = 4.1e-3; % H ... armature inductance
TF = Kv_si; % N-m/A ... trans ratio/motor constant
```

Let's investigate what happens in steady-state  $\bar{x}$ . The system is stationary when  $\dot{x} = 0$  and  $u = \bar{u}$  (stationary),<sup>11</sup> so

$$\begin{aligned} 0 &= A\bar{x} + B\bar{u} \Rightarrow \\ \bar{x} &= -A^{-1}B\bar{u}. \end{aligned} \quad (04.10)$$

<sup>10</sup> Here is the rnd file for use with [statemodelrnd.camerondevine.me/](http://statemodelrnd.camerondevine.me/) to derive the state-space model from the elemental, continuity, and compatibility equations.

<sup>11</sup> A stationary input  $\bar{u}$  is required for a stationary state if the input has any effect on the state; that is, if B is nonzero.

Let's compute our steady-state solution for a constant voltage input  $V_s(t) = \bar{V}$  and braking torque  $T_s(t) = \bar{T}$ . We use a symbolic approach to gain insight.

```
syms B_ J_ TF_ L_ R_ Vs_ Ts_ % using underscore for syms

a_ = [-B_/J_, TF_/J_; -TF_/L_, -R_/L_];
b_ = [0, -1/J_; 1/L_, 0];
u_ = [Vs_; Ts_];

M1_ = -inv(a_) * b_ % matrix -A^-1 B
den_ = TF_^2 + B_*R_; % common den
M2_ = M1_.*den_; % factor
xs_ = M1_*u_ % full ss sol
xs_2_ = M2_*u_; % naughty factorless ss sol
```

```
M1_ =
[ TF_/(TF_^2 + B_*R_), -R_/(TF_^2 + B_*R_) ]
[ B_/(TF_^2 + B_*R_), TF_/(TF_^2 + B_*R_) ]
```

```
xs_ =
(TF_*Vs_)/(TF_^2 + B_*R_) - (R_*Ts_)/(TF_^2 + B_*R_)
(B_*Vs_)/(TF_^2 + B_*R_) + (TF_*Ts_)/(TF_^2 + B_*R_)
```

```
eig(a_)
```

```
ans =
-((B_^2*L_^2 - 2*B_*J_*L_*R_ + J_^2*R_^2 - 4*J_*L_*TF_^2)^(1/2) + B_*L_ + J_*R_)
-(B_*L_ - (B_^2*L_^2 - 2*B_*J_*L_*R_ + J_^2*R_^2 - 4*J_*L_*TF_^2)^(1/2) + J_*R_)
```

A little more human-readably, using the fact that  $\Omega_2 = \Omega_J$  and  $i_1 = i_L$ , and using bars to denote steady-state values,

$$\bar{\Omega}_2 = \frac{1}{TF^2 + B_m R} (TF\bar{V}_s - R\bar{T}_s) \quad (04.11)$$

$$\bar{i}_1 = \frac{1}{TF^2 + B_m R} (B\bar{V}_s + T\bar{F}_s) \quad (04.12)$$

Let's focus on the first of these, the relationship between  $\overline{\Omega_2}$  and  $\overline{T_s}$ . For given  $\overline{V_s}$ , there is a linearly decreasing relationship between  $\overline{\Omega_2}$  and  $\overline{T_s}$ . This is precisely the *motor curve*. But it's one of a few curves plotted versus  $\overline{T_s}$ . Other common curves are current  $\overline{i_1}$ , mechanical braking power  $\mathcal{P}_{\text{brk}} = \overline{T_s} \overline{\Omega_s}$ , and efficiency  $\varepsilon$ . The efficiency is defined as the ratio of the braking power to the voltage source power  $\mathcal{P}_{\text{src}} = \overline{I_s} \overline{V_s}$ ; i.e.

$$\varepsilon = \mathcal{P}_{\text{brk}} / \mathcal{P}_{\text{src}}. \quad (04.13)$$

We already have expressions for  $\overline{\Omega_2}$  and  $\overline{i_1}$  in terms of  $\overline{T_s}$ , but we must still derive them for  $\mathcal{P}_{\text{brk}}$  and  $\varepsilon$ . For  $\mathcal{P}_{\text{brk}}$ , we must express  $\overline{\Omega_s}$  in terms of known quantities. From the linear graph, it is obvious that  $\overline{\Omega_s} = \overline{\Omega_2}$ . Therefore,

$$\mathcal{P}_{\text{brk}} = \overline{T_s} \overline{\Omega_2}. \quad (04.14)$$

Now for  $\varepsilon$ . We have the unknown source current  $\overline{I_s}$ . However, from the linear graph, it is obvious that  $\overline{I_s} = \overline{i_1}$ . Therefore,

$$\varepsilon = \frac{\overline{T_s} \overline{\Omega_2}}{\overline{i_1} \overline{V_s}}. \quad (04.15)$$

Let's compute these quantities for our parameters.

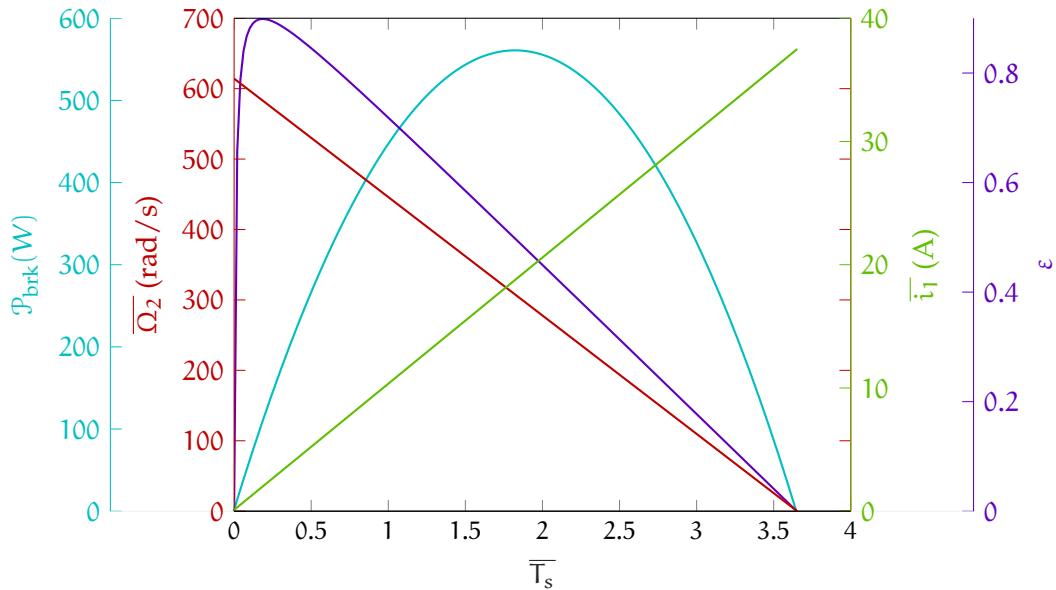
```

Vs = 60; % V ... max used, which is common
Tmax = TF/R*Vs; % N-m ... occurs when Omega_J = 0
Ts_a = linspace(0,Tmax,180); % N-m ... braking torques
O2_a = 1/(TF^2 + Bm*R)*(TF*Vs-R*Ts_a); % rad/s ... ss speed
i1_a = 1/(TF^2 + Bm*R)*(Bm*Vs+TF*Ts_a);
Pbrk_a = Ts_a.*O2_a; % W ... braking power
eff_a = Pbrk_a./(i1_a*Vs);

```

Now let's plot them! The output is shown in Figure 04.11.

There are some key quantities that can be read from the graph and found analytically. The most important are the maximum speed  $\overline{\Omega_{2\max}}$ , which occurs at zero torque, and maximum torque  $\overline{T_{s\max}}$ , which occurs at zero speed. Another is that the maximum mechanical power (output) occurs at  $\overline{T_{s\max}}/2$ . Finally, the maximum efficiency occurs at relatively low torque and high speed, which is typical for the following reason: the two



**Figure 04.11:** motor curves derived from the model.

energy-dissipative elements, the resistor and the damper, trade-off as being the dominant effect at the peak, and the resistor tends to dominate. That is, at high speed/voltage and low torque/current, the damper dominates dissipation; at low speed/voltage and high torque/current, the resistor dominates dissipation. It is very common for a motor's resistance to dominate the damping, as in our case.

Let's examine the maximum speed and torque.

```
Omax = O2_a(1) % rad/s ... occurs when T_s = 0
Tmax % N-m ... already computed and occurs when Omega_2 = 0
```

Omax =

614.2479

Tmax =

3.6526

Comparing these to the values given in the spec sheet, we see we're pretty good, but there's a bit of a discrepancy in the max torque.

```
Omax_spec  
Tmax_spec  
disp(sprintf('percent error for speed: %0.3g', ...  
            (Omax-Omax_spec)/Omax_spec*100))  
disp(sprintf('percent error for torque: %0.3g', ...  
            (Tmax-Tmax_spec)/Tmax_spec*100))
```

```
Omax_spec =  
  
628  
  
Tmax_spec =  
  
2.8200  
  
percent error for speed: -2.19  
percent error for torque: 29.5
```

We should investigate further, but what we will find is that these values are fairly sensitive to  $T_F$ ,  $B$ , and  $R$ . In our case, it is likely that the given value for  $R$  is a bit low. It is given as  $1.6 \Omega$ , but it is probably closer to  $2 \Omega$ . However, the datasheet for this motor was not clear about whether the maximum speed and torque values were derived from a full motor curve fit or if they were the only points measured. The former is best for estimating dynamic model parameters like  $R$  and  $B$ , but the latter is occasionally sufficient.

## Lecture 04.06 Transient DC motor performance

Let's begin by defining the system parameters.

```
Kt_spec = 13.7; % oz-in/A ... torque constant from spec
Kv_spec = 10.2; % V/kgpm ... voltage constant from spec
Tmax_spec = 2.82; % N-m ... max (stall) torque from spec
Omax_spec = 628; % rad/s ... max speed (no load) from spec
N_oz = 0.278013851; % N/oz
m_in = 0.0254; % m/in
Kt_si = Kt_spec*N_oz*m_in; % N-m/A
rads_krpm = 1e3*2*pi/60; % (rad/s)/krpm
Kv_si = Kv_spec/rads_krpm; % V/(rad/s)
d = 2.5*m_in; % m ... flywheel diameter
thick = 1*m_in; % m ... flywheel thickness
vol = pi*(d/2)^2*thick; % flywheel volume
rho = 8000; % kg/m^3 ... flywheel density (304 stainless)
m = rho*vol; % kg ... flywheel mass
Jf = 1/2*m*(d/2)^2; % kg-m^2 ... inertia of flywheel
Jr = 56.5e-6; % kg-m^2 ... inertia of rotor
J = Jf+Jr; % kg-m^2 ... total inertia
Bm = 16.9e-6; % N-m/s^2 ... motor damping coef
Bd = 20e-6; % N-m/s^2 ... bearing damping coef
B = Bm + Bd; % N-m/s^2 ... total damping coef
R = 1.6; % Ohm ... armature resistance
L = 4.1e-3; % H ... armature inductance
TF = Kv_si; % N-m/A ... trans ratio/motor constant
```

The state-space model was derived in [Lecture 04.04](#). First, we construct the A, B, C, and D matrices (a, b, c, and d). Then we define a *MATLAB LTI system model* using the `ss` command.

```
a = [-B/J,TF/J;-TF/L,-R/L];
b = [0;1/L];
c = [1,0;-B,TF;-TF,-R;0,1;1,0;B,0;...
      0,R;0,1;TF,0;0,1;1,0;0,-TF;0,0;0,1];
d = [0; 0; 1; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0];
sys = ss(a,b,c,d);
```

### 04.06.1 Simulating the step response

The step input is widely used to characterize the transient response of a system. *MATLAB's step function* conveniently simulates the step response

of an LTI system model.

```
[ys_a,t_a] = step(sys);
disp([t_a(1:6),ys_a(1:6,1:4)]) % print a little
```

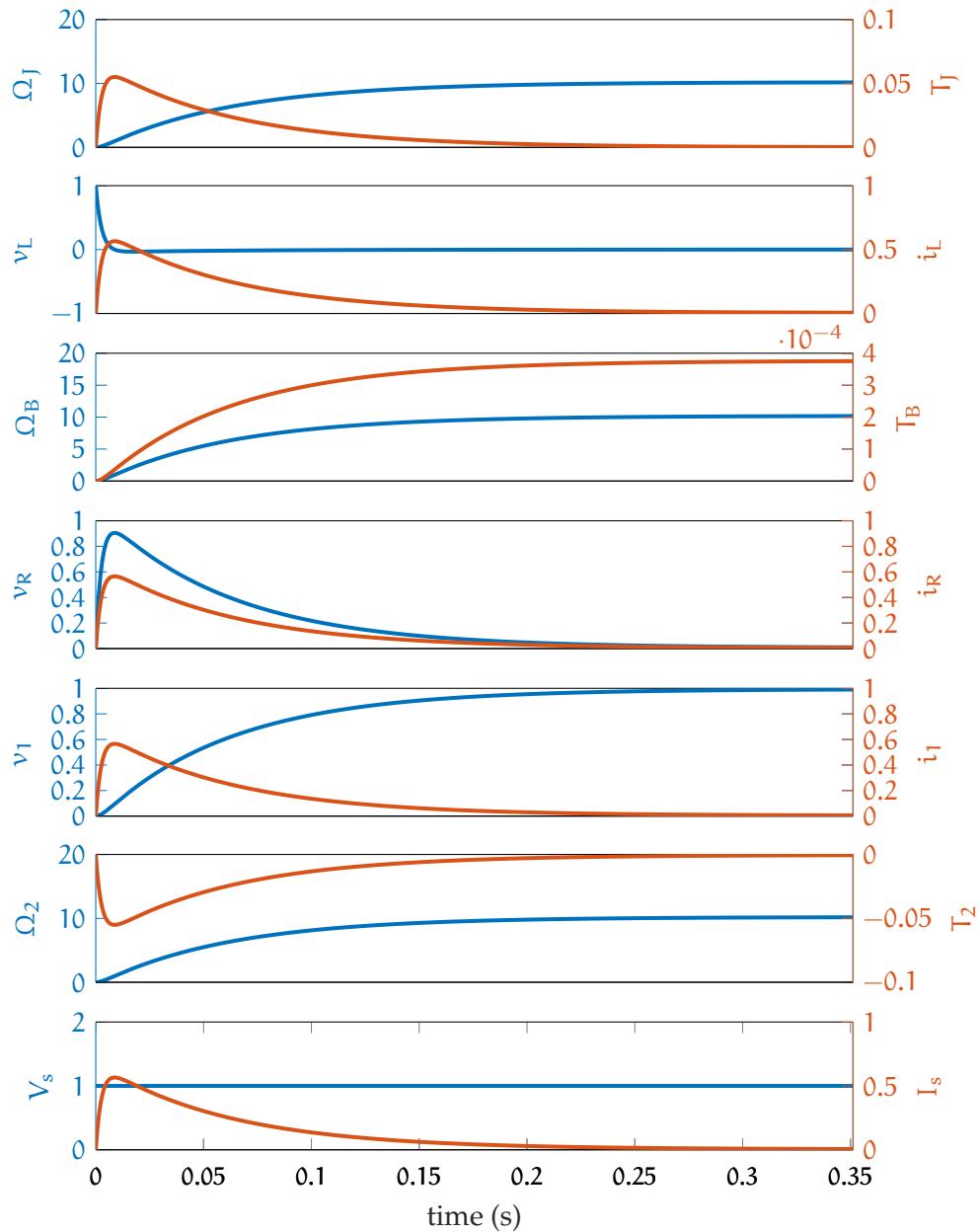
0	0	0	1.0000	0
0.0002	0.0018	0.0056	0.9082	0.0573
0.0005	0.0071	0.0106	0.8245	0.1093
0.0007	0.0155	0.0152	0.7482	0.1565
0.0010	0.0267	0.0194	0.6786	0.1993
0.0012	0.0405	0.0232	0.6151	0.2381

The vector  $t_a$  contains values of time and array  $ys_a$  contains a vector of time-series values for each output. If one would like the output for a step input  $k u_s(t)$  (scaled unit step  $u_s(t)$ ), by the principle of superposition for linear systems, one can scale the output by  $k$ . The outputs are plotted in Figure 04.12.

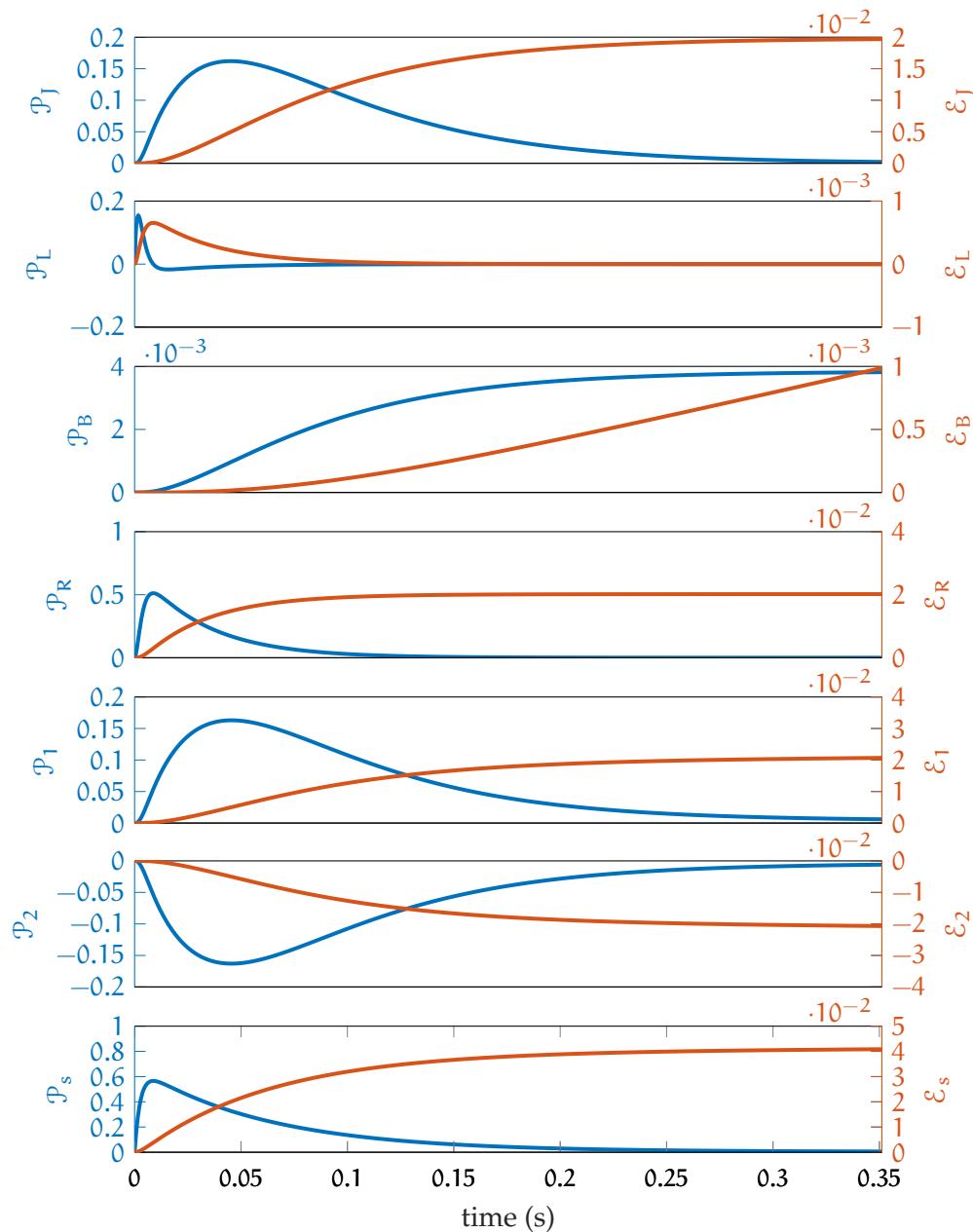
It is also interesting to inspect the power flow and energy associated with each element. Since we have simulated both the across and the through variable for each element, we can compute the instantaneous power by simply taking the product of them at each time step. Moreover, we can cumulatively compute the energy contribution of that power for each element. For energy storage elements, this is the change in energy stored or supplied; for energy dissipative elements, this is the change in energy dissipated; for source elements, this is the energy supplied or absorbed. The results are plotted in Figure 04.13.

```
P = NaN*ones(size(ys_a,1),size(ys_a,2)/2);
E = NaN*ones(size(P));
j = 0;
for i = 1:2:size(ys_a,2)
    j = j+1;
    P(:,j) = ys_a(:,i).*ys_a(:,i+1);
    E(:,j) = cumtrapz(t_a,P(:,j));
end
disp('power:');
disp(P(1:6,1:4)) % print a little
disp('energy change:')
disp(E(1:6,1:4)) % print a little
```

power:	0	0	0
--------	---	---	---



**Figure 04.12:** unit step responses for across- (left axes) and through-variables (right axes). Units are as follows: voltage is in V, current is in A, angular velocity is in rad/s, and torque is in N-m. and.



**Figure 04.13:** power flow (left axes) and energy storage/dissipation/transformation (right axes) for a unit step response. The unit of power is W and the unit of energy is J.

```

    0.0000    0.0520    0.0000    0.0052
    0.0001    0.0901    0.0000    0.0191
    0.0002    0.1171    0.0000    0.0392
    0.0005    0.1352    0.0000    0.0635
    0.0009    0.1465    0.0000    0.0907
energy change:
1.0e-03 *
    0        0        0        0
    0.0000   0.0064   0.0000   0.0006
    0.0000   0.0239   0.0000   0.0036
    0.0001   0.0494   0.0000   0.0108
    0.0001   0.0805   0.0000   0.0235
    0.0003   0.1152   0.0000   0.0425

```

#### 04.06.2 Estimating parameters from the step response

Often, our model has a couple parameters we don't know well from the specifications, but must attempt to measure. For the system under consideration, perhaps the two parameters most interesting to measure are the dominant time constant and the transformer ratio TF (most important). In this section, we explore how one might estimate them from a measured step response. Other parameters in the system could be similarly estimated.

By way of the transfer function, the state-space model can be transformed into input-output differential equations.

```

syms B_ J_ TF_ L_ R_ Vs_ s % using underscore for syms

a_ = [-B_/J_, TF_/J_; -TF_/L_, -R_/L_];
b_ = [0; 1/L_];

(s*eye(2)-a_)^-1*b_

```

```

ans =
    TF_/(TF_^2 + B_*R_ + B_*L_*s + J_*R_*s + J_*L_*s^2)
(B_ + J_*s)/(TF_^2 + B_*R_ + B_*L_*s + J_*R_*s + J_*L_*s^2)

```

The differential equation for  $\Omega_J$  is

$$\frac{d^2\Omega_J}{dt^2} + \left(\frac{R}{L} + \frac{B}{J}\right) \frac{d\Omega_J}{dt} + \frac{TF^2 + BR}{JL} \Omega_J = \frac{TF}{JL} V_s. \quad (04.16)$$

The corresponding characteristic equation is

$$\lambda^2 + \left(\frac{R}{L} + \frac{B}{J}\right) \lambda + \frac{TF^2 + BR}{JL} = 0 \quad (04.17)$$

which has solution

$$\lambda_{1,2} = -\frac{1}{2} \left( \frac{R}{L} + \frac{B}{J} \right) \pm \frac{1}{2} \sqrt{\left( \frac{R}{L} + \frac{B}{J} \right)^2 - 4 \frac{TF^2 + BR}{JL}}. \quad (04.18)$$

For a step input  $V_s(t) = \bar{V}_s$ ,  $\Omega_J(0) = d\Omega_J(0)/dt = 0$ , and distinct roots  $\lambda_1$  and  $\lambda_2$ , the solution is

$$\Omega_J(t) = \bar{V}_s \frac{TF}{TF^2 + BR} \left( 1 - \frac{1}{\lambda_2 - \lambda_1} (\lambda_2 e^{\lambda_1 t} - \lambda_1 e^{\lambda_2 t}) \right) \quad (04.19)$$

Let's compute  $\lambda_1$  and  $\lambda_2$ .

```
lambda12 = -1/2*(R/L+B/J) + ...
[1, -1]*1/2*sqrt((R/L+B/J)^2 - 4*(TF^2+BR)/(J*L))
```

```
lambda12 =
-16.3467 -373.9941
```

Both values are *real*, so we expect not an oscillation, but a decay to a final value. However, that decay occurs with two different time constants:  $\tau_1 = -1/\lambda_1$  and  $\tau_2 = -1/\lambda_2$ .

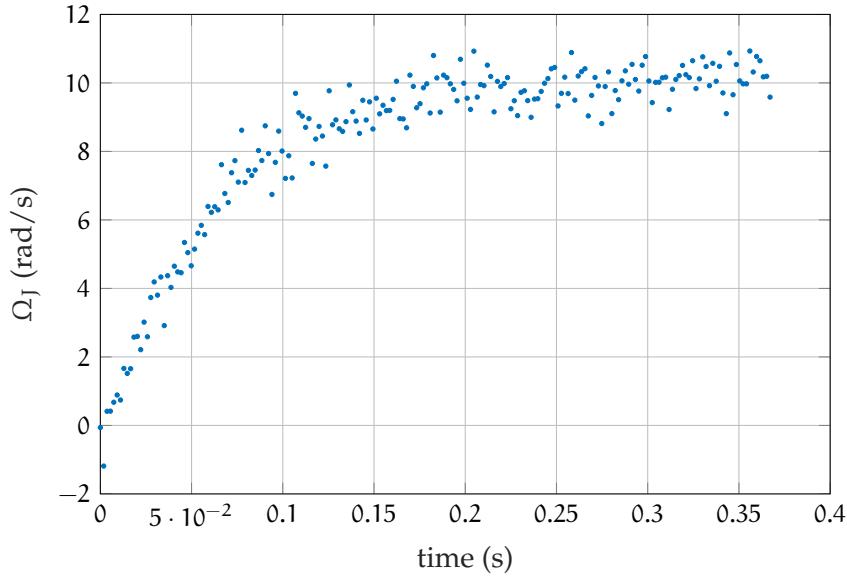
```
tau12 = -1./lambda12
disp(['ratio: ', num2str(tau12(1)/tau12(2))])
```

```
tau12 =
0.0612    0.0027
ratio: 22.8788
```

So second decays much faster than the first. That's good news for our estimation project because we can easily ignore the step response's first  $5\tau_2 \approx 0.0134$  s and assume the rest is decaying at  $\tau_1$ , which we call the *dominant time constant* and which we would like to estimate.

Let's generate some fake response data to get the idea. We'll layer on some Gaussian noise with `randn` to be more realistic. The data is plotted in [Figure 04.14](#).

```
t_data = linspace(0, -6/lambda12(1), 200);
O_fun = @(t) TF/(TF^2+BR)*...
```



**Figure 04.14:** unit step response “data.”

```
(1-1/(lambda12(2)-lambda12(1))*...
lambda12(2)*exp(lambda12(1)*t)-...
lambda12(1)*exp(lambda12(2)*t));
rng(2);
O_data = O_fun(t_data) + .5*randn(size(t_data));
```

Let's trim the data to eliminate the time interval corresponding to the first five of the “fast” time constant  $\tau_2$ .

```
[t_5,i_5] = min(abs(t_data-(-5/lambda12(2)))); % delete
t_data_trunc = t_data((i_5+1):end);
O_data_trunc = O_data((i_5+1):end);
```

We need want to take the natural logarithm of the data so we can perform a linear regression to estimate the “experimental” slow time constant  $\tilde{\tau}_1$ . We must first estimate the steady-state value  $\Omega_{J\infty}$  (which we'll also need). We don't want to just take the last value in the array due to its noisiness. The data goes for six slow time constants, so averaging the data for the last time constant is a good estimate.

```
[t_ss,i_ss] = ...
    min(abs(t_data_trunc-(-5/lambdal2(1)))); % start here
O_data_ss = O_data_trunc((i_ss+1):end);
mu_O_ss = mean(O_data_ss)
S_mu_O_ss = std(O_data_ss) / sqrt(length(O_data_ss))
```

```
mu_O_ss =
    10.1801
S_mu_O_ss =
    0.0763
```

Let's use this result to transform the data into its linear form.

```
O_lin = log(-(O_data_trunc-mu_O_ss));
O_lin_complex = find(imag(O_lin)>0);
disp(['number of complex values: ',...
    num2str(length(O_lin_complex))])
```

```
| number of complex values: 33
```

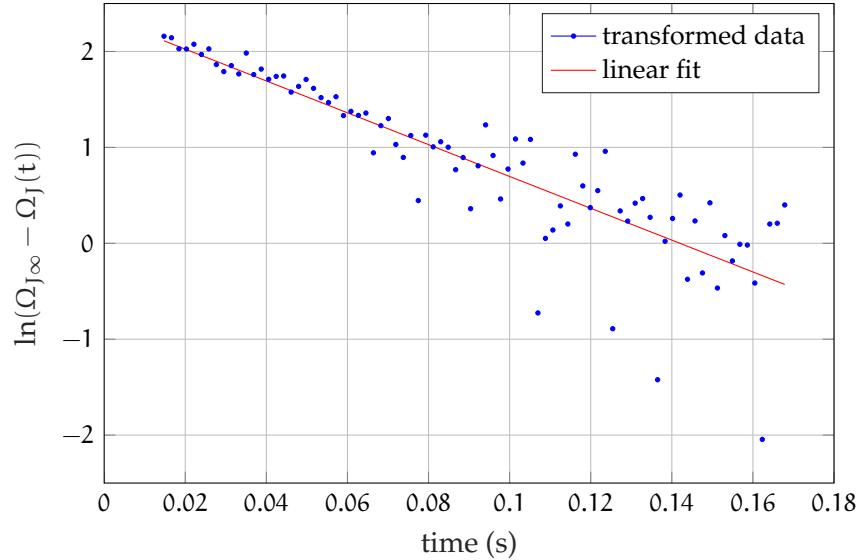
Now we have encountered a problem. The noisiness of the data makes some of our points wander into negative-land. Logarithms of negative numbers are complex. Naive approaches like just taking real parts, excluding complex values, or coercing complex values to  $-\infty$  all have the issue of biasing the data.

There are a lot of approaches we could take. The best approaches include nonlinear regression and discrete filtering to smooth the data (e.g. `filtfilt`).

We opt for an easier approach: we find the index at which the time series first transgresses the boundary and exclude the data beyond the previous index.

```
i_bad = O_lin_complex(1);
t_lin_trunc = t_data_trunc(1:i_bad-1);
O_lin_trunc = O_lin(1:i_bad-1);
```

This is plotted in [Figure 04.15](#) along with the linear regression least-squares fit, computed below.



**Figure 04.15:** transformed angular velocity “data” with a linear fit.

```

pf = polyfit(t_lin_trunc,O_lin_trunc,1);
O_lin_fit = polyval(pf,t_lin_trunc);
tau_1_est = -1/pf(1)

```

```

tau_1_est =
0.0603

```

So our estimate for  $\tau_1$  is  $\tilde{\tau}_1 = 60.3$  ms. Recall that our analytic expression for  $\tau_1$  is known in terms of other parameters. Similarly, the steady-state value of  $\Omega_J$ , which has already been estimated to be  $\Omega_{J\infty} = 10.18$  (i.e.  $\mu_0_{ss}$ ). This occurs when the time-derivatives of  $\Omega_J$  are zero. From the solution for  $\Omega_J$  (or its differential equation), for constant  $V_s(t) = \bar{V}_s$ , this occurs when

$$\Omega_{J\infty} = \frac{\text{TF}}{\text{TF}^2 + \text{BR}} \bar{V}_s. \quad (04.20)$$

An analytic expression for TF can be found by solving Equation 04.20, which yields

$$\text{TF} = \bar{V}_s \pm \frac{1}{2\tilde{\Omega}_{J\infty}} \sqrt{\bar{V}_s^2 - 4\text{BR}\tilde{\Omega}_{J\infty}^2} \quad (04.21)$$

We choose the solution closer to the *a priori* (spec) value of 0.0974.

$$\boxed{\text{TF\_est} = (1 + (-4 * \text{B} * \text{R} * \mu_{\text{O}} * \text{ss}^2 + 1^2)^{1/2}) / (2 * \mu_{\text{O}} * \text{ss})}$$

$$\boxed{\begin{aligned}\text{TF\_est} = \\ 0.0976\end{aligned}}$$

This estimate  $\tilde{T}_F = 0.0976$  is very close to the value given in the specification sheet *because we constructed it to be so*. Real measurements would probably yield an estimate further from the specification, which is why we would estimate it.

## Lecture 04.07 Driving motors

The DC motor requires DC electrical power provided by a circuit called the “driving” circuit. For industrial motors at least, these circuits must provide significant power, and for this reason a separate (from the control circuit) power supply is often used. There is a quick-and-dirty way to drive a DC motor at variable speed: since its angular velocity is reliably proportional to its voltage, place a potentiometer in series with the power supply and motor. However, this has disadvantages that include the power being limited and dissipated at high potentiometer resistance (low speed). For most applications, we will need either a current (or power) amplifier or—more likely—a microcontroller and an integrated circuit to produce a pulse-width modulation driving signal.

### 04.07.0.1 Pulse-width modulation

*Pulse-width modulation* (PWM) is a technique used to deliver an effectively variable signal to a load (in this case a motor) without a truly variable power source. A pulse of full source amplitude is repeated at a high frequency (e.g. 20 kHz), delivering a signal that is effectively averaged by the load dynamics such that its effects on the load are nearly continuous. The fraction of the period that the signal is high (on) is called the *duty cycle*  $\delta$ . The following figure shows a PWM signal  $v(t)$  and its average  $\bar{v}(t)$  with a few parameter definitions.

The mean of any periodic signal can be computed with the integral

$$\bar{v}(t) = \frac{1}{T} \int_0^T v(t),$$

which is easily evaluated for a PWM signal:

$$\bar{v}(t) = \frac{Aw}{T} = A\delta.$$

This result shows that if a PWM signal is delivered to a load, such as a DC motor, that is relatively unaffected by high-frequency signals, the effective signal will be simply the product of the source amplitude  $A$  and the duty cycle  $\delta$ . The duty cycle can have values from 0 to 1, so the effective DC signal produced varies linearly  $\delta$  from 0 to  $A$ .

#### 04.07.0.2 PWM with a microcontroller and integrated circuit

A microcontroller such as the myRIO or Arduino can easily produce a PWM signal; however, this signal is typically *low-power* and cannot drive even small DC motors. Therefore it is common to include a special kind of integrated circuit (IC) that uses the microcontroller's low-power PWM signal to gate a high-power DC source signal for delivery to the motor. We use a connectorized printed circuit board (PCB, e.g. a PC motherboard)—the [Pololu motor driver carrier](#)—that includes on it a [STMicroelectronics VNH5019](#) H-bridge motor driver *integrated circuit* (IC, i.e. a microchip).

**H-bridge circuits** We want to drive DC motors at different effective voltages *and* different directions. An H-bridge circuit allows us to reverse the direction of the PWM signal delivered to the motor. The following is a diagram of the H-bridge circuit.

The switches S1-S4 are typically instantiated with MOSFET transistors. As shown in the figure below, during the high duration of the PWM pulse, either S1 and S4 (a) or S2 and S3 (b) are closed and the others are open.

- (a) motor driven one direction
- (b) motor driven the opposite direction

Recall that a DC motor can be modeled as a resistor and inductor in series with an electro-mechanical transformer. The inductance of the windings make it an “inductive” load, which presents the following challenge. We can't rapidly change the current flow through an inductor without a huge spike in voltage, and the switches do just that, leading to switch damage. Therefore, during the low or “off” duration of the PWM signal, S1-S4 cannot all be simply opened. There are actually a few options for switch positions that allow the current to continue to flow without inductive “kickback.”

What's up with the diodes? Technically, they could be used to deal with the kickback, but they typically are not because they dissipate power. However, they are used to do just that to ease the transition between switch flips, which are never quite simultaneous.

#### 04.07.1 Motor curves

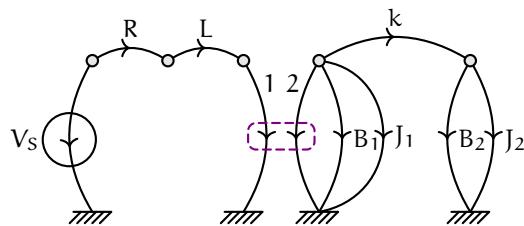
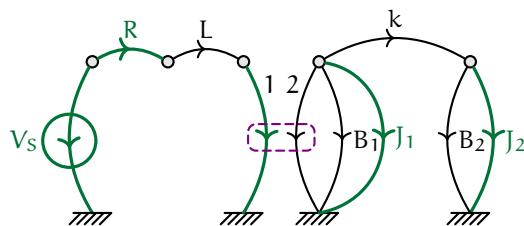
Motors are often characterized by three *steady-state* curves:

1. a torque  $T$  versus angular velocity  $\Omega$  curve;

2. an angular velocity  $\Omega$  versus voltage  $v$  curve, which has slope  $1/k_m$ ; and
3. a torque  $T$  versus current  $i$  curve, which has slope  $-k_m$ .

We will develop our own motor curves for the DC motor in the lab by simultaneously measuring  $v$ ,  $i$ , and  $\Omega$ . Unfortunately, we will not be measuring  $T$  directly, and so we will be unable to measure all these curves directly; however, we will be able to infer them based on the (reasonable, but not perfect) assumption that the motor has no power losses. In the end, they should look something like the following (using our usual sign convention).

In order to construct such curves, we will measure  $v$ ,  $i$ , and  $\Omega$ . The following sections describe the measurement process.

**Figure 04.16:** a linear graph model of the electromechanical system.**Figure 04.17:** a linear graph model of the electromechanical system.

## 04.08 Exercises for Chapter 04

### 04.08.1 Motor driving load with flexible coupler

This is similar to the model from the chapter, but includes the flexibility of the shaft coupler. An ideal voltage source drives the motor—modeled as an ideal transducer with armature resistance  $R$  and inductance  $L$ , given by the manufacturer in [Table 04.1](#). The ideal transducer's rotational mechanical side (2) is connected to a moment of inertia  $J_1 = 0.0565 \cdot 10^{-3} \text{ kg}\cdot\text{m}^2$ , as given in [Table 04.1](#), modeling the rotor inertia, dominated by the flywheel, and damping  $B$ , which is the parallel combination of the internal motor damping of [Table 04.1](#) and the bearing damping  $B_d$ , to yield  $B = 26.9 \cdot 10^{-6} \text{ N}\cdot\text{m}/\text{s}^2$ .<sup>12</sup>

### 04.08.2 Motor driving load with flexible coupler

A linear graph model is in order. An ideal voltage source drives the motor—modeled as an ideal transducer with armature resistance  $R$  and

<sup>12</sup>This is the sum of the inertia of the flywheel  $J_f = 0.324 \cdot 10^{-3} \text{ kg}\cdot\text{m}^2$ . It might be worthwhile combining this with the inertia from the shaft and coupler to obtain a more accurate value, but the difference is likely negligible.

inductance  $L$ , given by the manufacturer in [Table 04.1](#). The ideal transducer's rotational mechanical side (2) is connected to a moment of inertia  $J_1 = 0.0565 \cdot 10^{-3} \text{ kg}\cdot\text{m}^2$ , as given in [Table 04.1](#), modeling the rotor inertia. dominated by the flywheel, and damping  $B$ , which is the parallel combination of the internal motor damping of [Table 04.1](#) and the bearing damping  $B_d$ , to yield  $B = 26.9 \cdot 10^{-6} \text{ N}\cdot\text{m}/\text{s}^2$ .<sup>13</sup>

---

<sup>13</sup>This is the sum of the inertia of the flywheel  $J_f = 0.324 \cdot 10^{-3} \text{ kg}\cdot\text{m}^2$ . It might be worthwhile combining this with the inertia from the shaft and coupler to obtain a more accurate value, but the difference is likely negligible.

## **Part II**

# **System response**



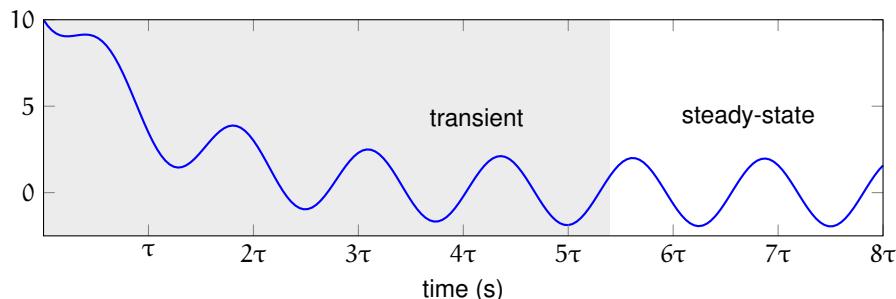
## Superposition, stability, and other LTI system properties

In this chapter, we will extend our understanding of linear, time-invariant (LTI) system properties. We must keep in mind a few important definitions.

The *transient response* of a system is its response during the initial time-interval during which the initial conditions dominate. The *steady-state response* of a system is its remaining response, which occurs after the transient response. Figure 05.1 illustrates these definitions.

The *free response* of a system is the response of the system to initial conditions—*without forcing* (i.e. the specific solution of the io ODE with the forcing function identically zero). This is closely related to, but distinct from, the transient response, which is the free response *plus* an additional term. This additional term is the *forced response*: the response of the system to a forcing function—*without initial conditions* (i.e. the specific solution of the io ODE with the initial conditions identically zero).

**transient response**  
**steady-state response**  
**free response**  
**forced response**



**Figure 05.1:** transient and steady-state responses. Note that the transition is not precisely defined. (Figure adapted from *Electronics: an introduction* by Picone.)

## Lecture 05.01 Superposition, derivative, and integral properties of LTI systems

**superposition  
linear,  
time-invariant (LTI)  
systems**

From the principle of *superposition*, *linear*, *time invariant* (LTI) system responses to both initial conditions and nonzero forcing can be obtained by summing the free response  $y_{fr}$  and forced response  $y_{fo}$ :

$$y(t) = y_{fr}(t) + y_{fo}(t).$$

Moreover, superposition says that any linear combination of inputs yields a corresponding linear combination of outputs. That is, we can find the response of a system to each input, separately, then linearly combine (scale and sum) the results according to the original linear combination. That is, for inputs  $u_1$  and  $u_2$  and constants  $a_1, a_2 \in \mathbb{R}$ , a forcing function

would yield output

where  $y_1$  and  $y_2$  are the outputs for inputs  $u_1$  and  $u_2$ , respectively.

This powerful principle allows us to construct solutions to complex forcing functions by decomposing the problem. It also allows us to make extensive use of existing solutions to common inputs.

There are two more LTI system properties worth noting here. Let a system have input  $u_1$  and corresponding output  $y_1$ . If the system is then given input  $u_2(t) = \dot{u}_1(t)$ , the corresponding output is

Similarly, if the same system is then given input  $u_3(t) = \int_0^t u_1(\tau) d\tau$ , the corresponding output is

**derivative property  
integral property**

These are sometimes called the *derivative* and *integral properties* of LTI systems.

## Lecture 05.02 Equilibrium and stability properties of LTI systems

For a system with LTI state-space model  $\dot{x} = Ax + Bu$ ,  $y = Cx + Du$ , the model is in *equilibrium* state  $x_0$  if all time-derivatives of the state vector ( $\dot{x}$ ,  $\ddot{x}$ ,  $\ddot{\ddot{x}}$ ,  $\dots$ ) are zero 0. This implies  $Ax_0 + Bu = 0$ . For constant input  $\bar{u}$ , this implies

**equilibrium**

If  $A$  is invertible,<sup>1</sup> as is often the case, there is a unique solution for a single *equilibrium state*:

**equilibrium**

If  $x$  is perturbed from an equilibrium state  $x_0$ , the response  $x(t)$  can:

1. asymptotically return to  $x_0$
2. diverge from  $x_0$
3. remain perturbed or oscillate about  $x_0$  with a constant amplitude

A *phase portrait* is a parametric plot of state variable *trajectories*, with time implicit. Phase portraits are exceptionally useful for understanding nonlinear systems, but they also give us a nice way to understand stability, as in [Figure 05.2](#).

**phase portrait**  
**trajectories**

These definitions of stability can be interpreted in terms of the free response of a system, as described, below.

### 05.02.1 Stability defined by the free response

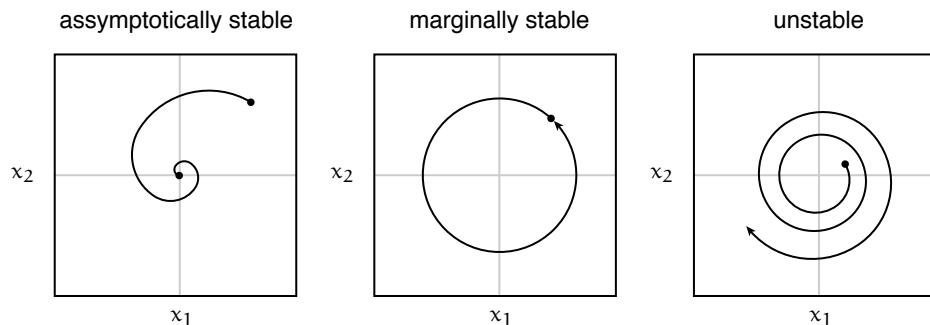
Using the concept of the free response (no inputs, just initial conditions), we define the following types of stability for LTI systems [Nise \(2015\)](#).

1. An LTI system is *asymptotically stable* if the free response approaches an equilibrium state as time approaches infinity.
2. An LTI system is *unstable* if the free response grows without bound as time approaches infinity.

**asymptotic stability**

**instability**

<sup>1</sup>If  $A$  is not invertible, the system has at least one eigenvalue equal to zero, which yields an *equilibrium subspace* equal to an offset (by  $B\bar{u}$ ) of the null space of the state space  $\mathbb{R}^n$ .



**Figure 05.2:** a phase-portrait demonstration of (left) asymptotic stability, (center) marginal stability, and (right) instability for a second-order system.

#### marginal stability

3. An LTI system is *marginally stable* if the free response neither decays nor grows but remains constant or oscillates as time approaches infinity.

These statements imply that the free response alone governs stability. Recall that the free response  $y_{fr}$  of a system with characteristic equation roots  $\lambda_i$  with multiplicity  $m_i$ , for constants  $C_i$ , is

Each term will either decay to zero, remain constant, or increase without bound—depending on the sign of the *real part* of the corresponding root of the characteristic equation  $\text{Re}(\lambda_i)$ .

In other words, for an LTI system, the following statements hold.

1. An LTI system is *asymptotically stable* if, for all  $\lambda_i$ ,  $\text{Re}(\lambda_i) < 0$ .
2. An LTI system is *unstable* if, for any  $\lambda_i$ ,  $\text{Re}(\lambda_i) > 0$ .
3. An LTI system is *marginally stable* if, for all  $\lambda_i$ ,  $\text{Re}(\lambda_i) \leq 0$  and at least one  $\text{Re}(\lambda_i) = 0$ .

#### 05.02.2 Stability defined by the forced response

An alternate formulation of the stability definitions above is called the *bounded-input bounded-output* (BIBO) definition of stability, and states the following Nise (2015).

#### BIBO stability

1. A system is *BIBO stable* if every bounded input yields a bounded output. **BIBO stable**
2. A system is *BIBO unstable* if any bounded input yields an unbounded output. **BIBO unstable**

In terms of BIBO stability, *marginal stability*, then, means that a system has a bounded response to some inputs and an unbounded response to others. For instance, a second-order undamped system response to a sinusoidal input at the natural frequency is unbounded, whereas every other input yields a bounded output.

Although we focus on the definitions of stability in terms of the free response, it is good to understand BIBO stability, as well.

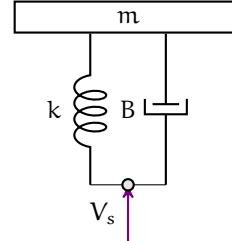
**BIBO marginal stability**

## Lecture 05.03 Vibration isolation table analysis

In this example, we exercise many of the methods for modeling and analysis explored thus far.

Given the vibration isolation table model in Figure 05.3—with  $m = 320 \text{ kg}$ ,  $k = 16000 \text{ N/m}$ , and  $B = 1200 \text{ N}\cdot\text{m/s}$ —derive:

1. a linear graph model,
2. a state-space model,
3. the equilibrium state  $x_0$  for the unit step input,
4. a transfer function model,
5. an input-output differential equation model with input  $V_s$  and output  $v_m$ ,
6. a solution for  $v_m(t)$  for a unit step input  $V_s(t) = 1 \text{ m/s}$  for  $t \geq 0$ ,
7. the system's stability.



**Figure 05.3:** a vibration isolation table schematic with input velocity  $V_s$ .

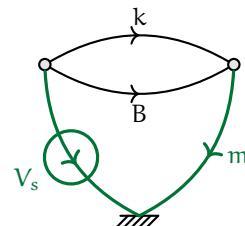
### 05.03.1 Linear graph and state-space models

The linear graph and normal tree are shown in Figure 05.4. The state variables are the velocity of the mass  $v_m$  and the force through the spring  $f_k$  and the system order is  $n = 2$ . The input, state, and output vectors are

$$\mathbf{u} = [V_s] \quad \mathbf{x} = \begin{bmatrix} v_m \\ f_k \end{bmatrix} \quad \mathbf{y} = [v_m]. \quad (05.1)$$

The elemental equations are as follows.

$$\begin{array}{l|l} m & \dot{v}_m = \frac{1}{m} f_m \\ k & \dot{f}_k = kv_k \\ B & f_B = Bv_B \end{array}$$



**Figure 05.4:** linear graph of the isolation table.

The continuity and compatibility equations are as follows.

branch	continuity equation	link	compatibility equation
$m$	$f_m = f_k + f_B$	$k$	$v_k = V_s - v_m$
$B$		$B$	$v_B = V_s - v_m$

The state equation can be found by substituting the continuity and compatibility equations into the elemental equations, and eliminating  $f_B$ , to yield

$$\dot{x} = \begin{bmatrix} -B/m & 1/m \\ -k & 0 \end{bmatrix} x + \begin{bmatrix} B/m \\ k \end{bmatrix} u \quad (05.2a)$$

$$y = [1 \ 0] x + [0] u. \quad (05.2b)$$

### 05.03.2 Equilibrium

Let's check to see if  $A$  is invertible by trying to compute its inverse:

$$A^{-1} = \begin{bmatrix} -B/m & 1/m \\ -k & 0 \end{bmatrix}^{-1} \quad (05.3)$$

$$= \frac{1}{k/m} \begin{bmatrix} 0 & -1/m \\ k & -B/m \end{bmatrix} \quad (05.4)$$

So it has an inverse, after all! Let's use this to compute the equilibrium state:

$$x_0 = -A^{-1} B \bar{u} \quad (05.5)$$

$$= \frac{-m}{k} \begin{bmatrix} 0 & -1/m \\ k & -B/m \end{bmatrix} \begin{bmatrix} B/m \\ k \end{bmatrix} [1] \quad (05.6)$$

$$= \frac{-m}{k} \begin{bmatrix} -k/m \\ 0 \end{bmatrix} \quad (05.7)$$

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (05.8)$$

So the system is in equilibrium with  $v_m = 1$  m/s and  $f_k = 0$  N. Since  $v_m$  is also our output, we expect 1 m/s to be our steady-state output value.

### 05.03.3 Transfer function model

The transfer function  $H(s) = V_m(s)/V_s(s)$  will be used as a bridge to the input-output differential equation. The transfer function can be found from the usual formula, from [Lecture 03.07](#),

$$H(s) = C(sI - A)^{-1} B + D. \quad (05.9)$$

Let's first compute  $(sI - A)^{-1}$ :<sup>2</sup>

$$(sI - A)^{-1} = \left( \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} -B/m & 1/m \\ -k & 0 \end{bmatrix} \right)^{-1} \quad (05.10a)$$

$$= \begin{bmatrix} s + B/m & -1/m \\ k & s \end{bmatrix}^{-1} \quad (05.10b)$$

$$= \frac{1}{(s + B/m)(s) - (-1/m)(k)} \begin{bmatrix} s & 1/m \\ -k & s + B/m \end{bmatrix} \quad (05.10c)$$

$$= \frac{1}{s^2 + (B/m)s + k/m} \begin{bmatrix} s & 1/m \\ -k & s + B/m \end{bmatrix} \quad (05.10d)$$

Now we're ready to compute the entirety of  $H$ :

$$H(s) = \frac{1}{s^2 + (B/m)s + k/m} [1 \ 0] \begin{bmatrix} s & 1/m \\ -k & s + B/m \end{bmatrix} \begin{bmatrix} B/m \\ k \end{bmatrix} + [0] \quad (05.11a)$$

$$= \frac{1}{s^2 + (B/m)s + k/m} [s \ 1/m] \begin{bmatrix} B/m \\ k \end{bmatrix} \quad (05.11b)$$

$$= \frac{(B/m)s + k/m}{s^2 + (B/m)s + k/m}. \quad (05.11c)$$

#### 05.03.4 Input-output differential equation

The input-output differential equation can be found from the reverse of the procedure in [Lecture 03.07](#). Beginning from the transfer function,

$$\frac{V_m}{V_s} = \frac{(B/m)s + k/m}{s^2 + (B/m)s + k/m} \Rightarrow \quad (05.12a)$$

$$(s^2 + (B/m)s + k/m) V_m = ((B/m)s + k/m) V_s \Rightarrow \quad (05.12b)$$

$$\dot{v}_m + (B/m)\dot{v}_m + (k/m)v_m = (B/m)\dot{V}_s + (k/m)V_s. \quad (05.12c)$$

#### 05.03.5 Step response

The step response is found using superposition and the derivative property of LTI systems. The forcing function  $f(t) = (B/m)\dot{V}_s + (k/m)V_s$  is composed of two terms, one of which has a derivative of the input  $V_s$ . Rather than attempting to solve the entire problem at once, we choose to find the response for a forcing function  $f(t) = 1$  (for  $t \geq 0$ )—that is, the unit step response—and use superposition and the derivative property of LTI systems to calculate the composite response.

---

<sup>2</sup>See ([Rowell and Wormley, 1997](#), Sec. A.4.3) for details on the matrix inverse.

#### 05.03.5.1 Unit step response

The characteristic equation of [Equation 05.12c](#) is

$$\lambda^2 + (B/m)\lambda + k/m = 0 \Rightarrow \quad (05.13a)$$

$$= -\frac{B}{2m} \pm \frac{\sqrt{B^2 - 4mk}}{2m} \Rightarrow \quad (05.13b)$$

$$\lambda_{1,2} = -1.875 \pm j6.818. \quad (05.13c)$$

The roots are complex, so the system will have a damped sinusoidal step response. Let  $\sigma = -1.875$  and  $\omega = 6.818$  such that  $\lambda_{1,2} = \sigma \pm j\omega$ . The homogeneous solution is

$$v_{m_h}(t) = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t}. \quad (05.14)$$

In this form,  $C_1$  and  $C_2$  are complex. It is somewhat easier to deal with

$$v_{m_h}(t) = C_1 e^{\sigma t} e^{j\omega t} + C_2 e^{\sigma t} e^{-j\omega t} \quad (05.15a)$$

$$= e^{\sigma t} (C_1 \cos \omega t + jC_1 \sin \omega t + C_2 \cos \omega t - jC_2 \sin \omega t) \quad (05.15b)$$

$$= e^{\sigma t} ((C_1 + C_2) \cos \omega t + j(C_1 - C_2) \sin \omega t). \quad (05.15c)$$

Let  $C_3 = C_1 + C_2$  and  $C_4 = j(C_1 - C_2)$  such that

$$v_{m_h}(t) = e^{\sigma t} (C_3 \cos \omega t + C_4 \sin \omega t). \quad (05.16)$$

This is a decaying (because  $\sigma < 0$ ) sinusoid. A nice aspect of this new form is that  $C_3$  and  $C_4$  are real.

Now, the particular solution can be found by assuming a solution of the form  $v_{m_p}(t) = K$  for  $t \geq 0$ . Substituting this into [Equation 05.12c](#) (with forcing  $f(t) = 1$ , we attempt to find a solution for  $K$  (that is, *determine it*):

$$(k/m)K = 1 \Rightarrow K = m/k. \quad (05.17)$$

Therefore,  $v_{m_p}(t) = m/k$  is a solution, and therefore the general solution is

$$v_{m_{step}}(t) = v_{m_h}(t) + v_{m_p}(t) \quad (05.18a)$$

$$= e^{\sigma t} (C_3 \cos \omega t + C_4 \sin \omega t) + m/k. \quad (05.18b)$$

This leaves the specific solution, to be found applying the initial conditions (assumed to be zero). Before we do so, however, we need the time-derivative of the  $v_{m_{step}}$ :

$$\dot{v}_{m_{step}}(t) = e^{\sigma t} ((C_3 \sigma + C_4 \omega) \cos(\omega t) + (C_4 \sigma - C_3 \omega) \sin(\omega t)). \quad (05.19)$$

Now, applying the initial conditions,

$$v_{m_{\text{step}}}(0) = 0 \Rightarrow \quad (05.20\text{a})$$

$$C_3 = -m/k \quad (05.20\text{b})$$

$$\dot{v}_{m_{\text{step}}}(0) = 0 \Rightarrow 0 = C_3\sigma + C_4\omega \Rightarrow \quad (05.20\text{c})$$

$$C_4 = \frac{\sigma}{\omega} \cdot \frac{m}{k}. \quad (05.20\text{d})$$

It's good form to re-write this as a single sinusoid:

$$v_{m_{\text{step}}}(t) = v_{m_h}(t) + v_{m_p}(t) \quad (05.21\text{a})$$

$$= A_1 e^{\sigma t} \cos(\omega t + \psi_1) + m/k \quad (05.21\text{b})$$

where we have used [A.02.11](#) to find

$$A_1 = \sqrt{C_3^2 + C_4^2} \quad (05.22\text{a})$$

$$\psi_1 = -\arctan(C_4/C_3). \quad (05.22\text{b})$$

#### 05.03.5.2 Superposition and the derivative property

Recall that the actual forcing function is a linear combination of the input and its time-derivative. Therefore, it is expedient to re-write the time-derivative of the unit step response:

$$\dot{v}_{m_{\text{step}}}(t) = A_1 e^{\sigma t} (\sigma \cos(\omega t + \psi_1) - \omega \sin(\omega t + \psi_1)) \quad (05.23\text{a})$$

$$= A_1 A_2 e^{\sigma t} \cos(\omega t + \psi_1 + \psi_2) \quad (05.23\text{b})$$

where

$$A_2 = \sqrt{\sigma^2 + \omega^2} \quad (05.23\text{c})$$

$$\psi_2 = -\arctan(-\omega/\sigma). \quad (05.23\text{d})$$

Finally, applying superposition and the derivative rule of LTI systems,

$$v_m(t) = (B/m)\dot{v}_{m_{\text{step}}}(t) + (k/m)v_{m_{\text{step}}} \quad (05.24\text{a})$$

$$= \frac{B}{m} A_1 A_2 e^{\sigma t} \cos(\omega t + \psi_1 + \psi_2) + \frac{k}{m} A_1 e^{\sigma t} \cos(\omega t + \psi_1) + 1. \quad (05.24\text{b})$$

This is the solution!

It's worth plotting the response. Begin by defining the system parameters.

```
m = 320; % kg ... mass
k = 16000; % N/m ... spring constant
B = 1200; % N-m/s ... damping coefficient
```

Now define the secondary parameters.

```
lambda = -B/(2*m)+[-1,1]*sqrt(B^2-4*m*k)/(2*m);
sigma = real(lambda(1));
omega = imag(lambda(2));
K = m/k;
C3 = -m/k;
C4 = sigma/omega*m/k;
A1 = sqrt(C3^2+C4^2);
psi1 = -atan2(C4,C3);
A2 = sqrt(sigma^2+omega^2);
psi2 = -atan2(-omega,sigma);
```

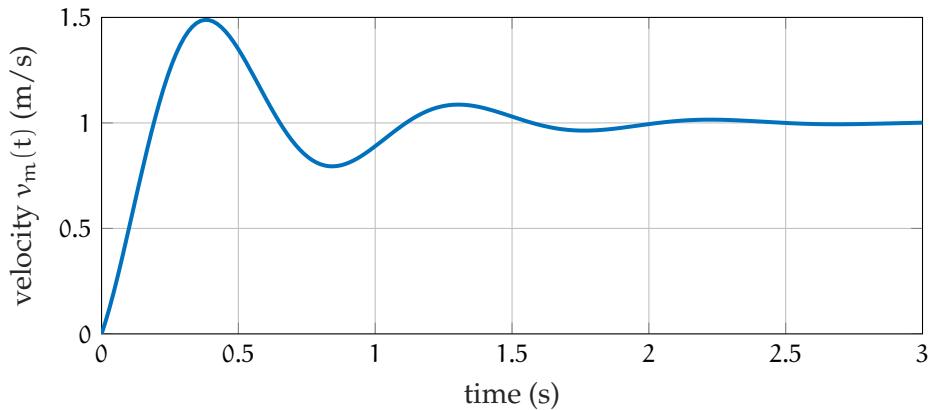
Finally, the solution for  $v_m(t)$  can be defined as an anonymous function.

```
vm = @(t) ...
    A1*A2*B/m*exp(sigma*t).*cos(omega*t+psi1+psi2)+...
    A1*k/m*exp(sigma*t).*cos(omega*t+psi1)+...
    1;
```

Now, plot over the first few seconds. The results are shown in Figure 05.5.

```
t_a = linspace(0,3,200);
h = figure;
p = plot(t_a,vm(t_a), 'linewidth',1.5);
xlabel('time (s)')
ylabel('velocity $v_m(t)$ (m/s)',...
    'interpreter','latex');
grid on
hgsave(h, 'figures/temp');
```

Note that the steady-state output value agrees with that predicted by the equilibrium analysis, above.



**Figure 05.5:** vibration table step response  $v_m(t)$ .

#### 05.03.6 Stability

We have learned what we need in order to analyze the system's stability. The roots of the characteristic equation were  $\lambda_{1,2} = -1.875 \pm j6.818$ , which clearly *all* have negative real parts, and therefore the system is *asymptotically stable*.

## Qualities of transient response

In this chapter, we explore the qualities of transient response—the response of the system in the interval during which initial conditions dominate.

We focus on characterizing first- and second-order linear systems; not because they're easiest (they are), but because nonlinear systems can be *linearized* about an *operating point* and because higher-order linear system responses are just *sums of first- and second-order responses*, making "everything look first- and second-order." Well, many things, at least.

In this chapter, we primarily consider systems represented by *single-input, single-output (SISO)* ordinary differential equations (also called io ODEs)—with variable  $y$  representing the *output*, dependent variable time  $t$ , variable  $u$  representing the *input, forcing function*  $f$ , constant coefficients  $a_i, b_j$ , order  $n$ , and  $m \leq n$  for  $n \in \mathbb{N}_0$ —of the form

$$\frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = f, \text{ where} \quad (06.1a)$$

$$f \equiv b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + \cdots + b_1 \frac{du}{dt} + b_0 u. \quad (06.1b)$$

**linearization**  
**operating point**

**SISO**

**forcing function**

Note that the forcing function  $f$  is related to but distinct from the input  $u$ . This terminology proves rather important.

## Lecture 06.01 Characteristic transient responses

**characteristic  
response  
singularity  
functions**

A system's *characteristic responses* are responses to specific forcing functions—called the *singularity functions*. The reasons these are “characteristic” are:

1. the singularity functions model commonly interesting system inputs (e.g. a sudden change in the input), and so they can be said to *characterize inputs*, and
2. the ways in which the system responds to these specific functions reveal aspects of the system (e.g. how quickly it responds), so these responses can be said to *characterize systems*.

**superposition  
differentiation**

Now, one may object that [Equation 06.1b](#) shows that a forcing function needn't look anything like an input due to its being composed of a sum of scaled copies of the input and its derivatives. Yes, but given two key properties of linear, time-invariant (LTI) systems—*superposition* and the *differentiation* property—, knowing a system's response  $y_1$  to a forcing function  $f_1$  allows us to construct its response to that input (that is,  $y_2$  for input  $u_2 = f_1$ ) as

I know.

There are three singularity functions, which are now defined as piecewise functions of time  $t$ .

**unit impulse  
Dirac delta**

First, the *unit impulse* or *Dirac delta* function<sup>1</sup>  $\delta$  is defined as zero everywhere except at  $t = 0$ , when it is undefined, and has unity as its integral over all time. When  $\delta$  is scaled (e.g.  $5\delta$ ), its integral scales by the same factor. This strange little beast models a sudden “spike” in the input.

**unit step**

Second, the *unit step* function  $u_s$  is defined as zero for  $t \leq 0$  and unity for  $t > 0$ . It models a sudden change in the input. Scaling it scales the amount of change. Often, this is considered to be the gold-standard for characterizing the transient response of a system.

**unit ramp**

Third, the *unit ramp* function  $u_r$  is defined as zero for  $t \leq 0$  and  $t$  for  $t > 0$ —that is, it is linearly increasing with unity slope. It models a steadily increasing input and is probably the least useful of the singularity functions. Scaling it scales the rate of steady change.

<sup>1</sup>Technically,  $\delta$  is a distribution, not a function, but we use the common, confusing, comfortably couched terminology.

## Lecture 06.02 First-order systems in transient response

First order systems have input-output differential equations of the form

$$\tau \frac{dy}{dt} + y = b_1 \frac{du}{dt} + b_0 u \quad (06.2)$$

with  $\tau \in \mathbb{R}$  called the *time constant* of the system. Systems with a single energy storage element—such as those with electrical or thermal capacitance—can be modeled as first-order.

The characteristic equation yields a single root  $\lambda = -1/\tau$ , so the *homogeneous solution*  $y_h$ , for constant  $\kappa \in \mathbb{R}$ , is

time constant

homogeneous solution

### 06.02.1 Free response

The *free response*  $y_{fr}$  of a system is its response to initial conditions and no forcing ( $f(t) = 0$ ). This is useful for two reasons:

1. perturbations of the system from equilibrium result in free response and
2. from superposition, the free response can be added to a forced response to find the specific response:  $y(t) = y_{fr}(t) + y_{fo}(t)$ . This allows us to use tables of solutions like [Table 06.1](#) to construct solutions for systems with nonzero initial conditions with forcing.

free response  $y_{fr}$ 

The free response is found by applying initial conditions to the homogeneous solution. With initial condition  $y(0)$ , the free response is

$$y_{fr}(t) = y(0) e^{-t/\tau}, \quad (06.3)$$

which begins at  $y(0)$  and decays exponentially to zero.

### 06.02.2 Step response

In what follows, we develop *forced response*  $y_{fo}$  solutions, which are the *specific solution* responses of systems to given inputs and *zero initial conditions*: all initial conditions set to zero.

forced response  $y_{fo}$   
zero initial conditions

If we consider the common situation that  $b_1 = 0$  and  $u(t) = Ku_s(t)$  for some  $K \in \mathbb{R}$ , the solution to [Equation 06.2](#) is

The non-steady term is simply a constant scaling of a decaying exponential.

A plot of the step response is shown in Figure 06.1. As with the free response, within  $5\tau$  the transient response is less than 1% of the difference between  $y(0)$  and steady-state.

### 06.02.3 Impulse and ramp responses

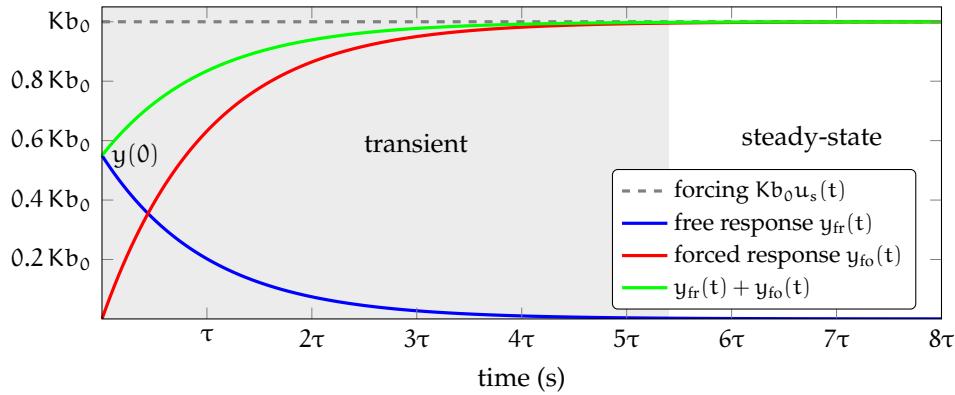
The response to all three singularity inputs are included in Table 06.1. These can be combined with the free response of Equation 06.3 using superposition. Results could be described as *bitchin'*.

#### Example 06.02-1 RC-circuit response the easy way

Consider a parallel RC-circuit with input current  $I_S(t) = 2u_s(t)$  A, initial capacitor voltage  $v_C(0) = 3$  V, resistance  $R = 1000 \Omega$ , and capacitance  $C = 1 \text{ mF}$ . Proceeding with the usual analysis would produce the io differential equation

$$C \frac{dv_C}{dt} + v_C/R = I_S.$$

Use Table 06.1 to find  $v_C(t)$ .



**Figure 06.1:** free and forced responses and their sum for a first order system with input  $u(t) = Kb_0 u_s(t)$ , initial condition  $y(0)$ , and  $b_1 = 0$ .

**Table 06.1:** first-order system characteristic and total forced responses for singularity inputs. The relevant differential equation is of the standard form  $\tau\dot{y} + y = f$ .

$u(t)$	characteristic response $f(t) = u(t)$	total forced response $y_{fo}$ for $t \geq 0$ $f(t) = b_1\dot{u} + b_0u$
$\delta(t)$	$\frac{1}{\tau}e^{-t/\tau}$	$\frac{b_1}{\tau}\delta(t) + \left(\frac{b_0}{\tau} - \frac{b_1}{\tau^2}\right)e^{-t/\tau}$
$u_s(t)$	$1 - e^{-t/\tau}$	$b_0 - \left(b_0 - \frac{b_1}{\tau}\right)e^{-t/\tau}$
$u_r(t)$	$t - \tau(1 - e^{-t/\tau})$	$b_0t + (b_1 - b_0\tau)(1 - e^{-t/\tau})$

## Lecture 06.03 Second-order systems in transient response

Second-order systems have input-output differential equations of the form

$$\frac{d^2y}{dt^2} + 2\zeta\omega_n \frac{dy}{dt} + \omega_n^2 y = f(t) \quad (06.4)$$

**natural frequency**  $\omega_n$  where  $\omega_n$  is called the *natural frequency*,  $\zeta$  is called the (dimensionless) *damping ratio*, and  $f$  is a forcing function that depends on the input  $u$  as

$$f(t) = b_2 \frac{d^2u}{dt^2} + b_1 \frac{du}{dt} + b_0 u. \quad (06.5)$$

Systems with two energy storage elements—such as those with an inertial element and a spring-like element—can be modeled as second-order.

For distinct roots ( $\lambda_1 \neq \lambda_2$ ), the homogeneous solution is, for some real constants  $\kappa_1$  and  $\kappa_2$ ,

$$y_h(t) = \kappa_1 e^{\lambda_1 t} + \kappa_2 e^{\lambda_2 t} \quad (06.6)$$

where

$$\lambda_1, \lambda_2 = -\zeta\omega_n \pm \omega_n \sqrt{\zeta^2 - 1}. \quad (06.7)$$

### 06.03.1 Free response

The free response  $y_{fr}$  is found by applying initial conditions to the homogeneous solution. With initial conditions  $y(0)$  and  $\dot{y}(0) = 0$ , the free response is

$$y_{fr}(t) = y(0) \frac{1}{\lambda_2 - \lambda_1} (\lambda_2 e^{\lambda_1 t} - \lambda_1 e^{\lambda_2 t}). \quad (06.8)$$

There are five possibilities for the location of the roots  $\lambda_1$  and  $\lambda_2$ , all determined by the value of  $\zeta$ .

**boom ↗**  $\zeta \in (-\infty, 0)$ : **unstable** This case is very undesirable because it means our system is unstable and, given any nonzero input or output, will *explode* ↗ to infinity.

$\zeta = 0$ : **undamped** An undamped system will oscillate forever if perturbed from zero output.

$\zeta \in (0, 1)$ : **underdamped** Roughly speaking, underdamped systems oscillate, but not forever. Let's consider the form of the solution for initial conditions and no forcing. The roots of the characteristic equation are

$$\lambda_1, \lambda_2 = -\zeta\omega_n \pm j\omega_n \sqrt{1 - \zeta^2} = -\zeta\omega_n \pm j\omega_d \quad (06.9)$$

where the *damped natural frequency*  $\omega_d$  is defined as

$$\omega_d \equiv \omega_n \sqrt{1 - \zeta^2}. \quad (06.10)$$

**damped natural frequency**  $\omega_d$

From Equation (06.8) for the free response, using Euler's formulas to write in terms of trigonometric functions, and the initial conditions  $y(0)$  and  $\dot{y}(0) = 0$ , we have

$$y_{fr}(t) = y(0) \frac{e^{-\zeta\omega_n t}}{\sqrt{1 - \zeta^2}} \cos(\omega_d t + \psi) \quad (06.11)$$

where the phase  $\psi$  is

$$\psi = -\arctan \frac{\zeta}{\sqrt{1 - \zeta^2}}. \quad (06.12)$$

This is an oscillation that decays to the value it oscillates about,  $y(t)|_{t \rightarrow \infty} = 0$ . So any perturbation of an underdamped system will result in a decaying oscillation about equilibrium.

$\zeta = 1$ : **critically damped** In this case, the roots of the characteristic equation are equal:

$$\lambda_1 = \lambda_2 = -\omega_n \quad (06.13)$$

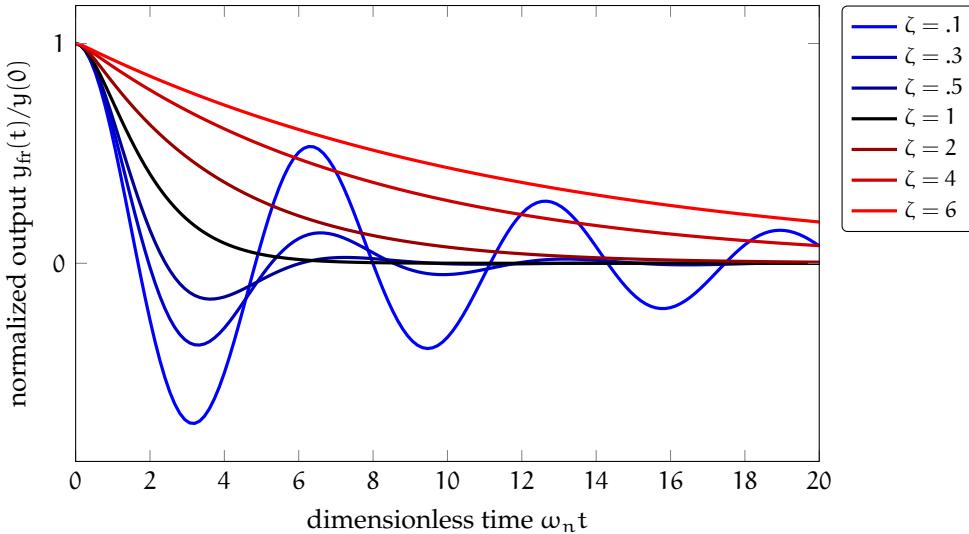
So we must modify Equation 06.6 with a factor of  $t$  for the homogeneous solution. The free response for initial conditions  $y(0)$  and  $\dot{y}(0) = 0$ , we have

$$y_{fr}(t) = y(0) (1 + \omega_n t) e^{-\omega_n t}. \quad (06.14)$$

This decays without oscillation, but just barely.

$\zeta \in (1, \infty)$ : **overdamped** Here the roots of the characteristic equation are distinct and real. From Equation (06.8) with free response to initial conditions  $y(0)$  and  $\dot{y}(0) = 0$ , we have the sum of two decaying real exponentials. This response will neither overshoot nor oscillate—like the critically damped case—but with even less gusto.

Figure 06.2 displays the free response results. Note that a small damping ratio results in overshooting and oscillation about the equilibrium value. In contrast, large damping ratio results in neither overshoot nor oscillation. However, both small and large damping ratios yield responses that take longer durations to approach equilibrium than damping ratios near unity. In terms of system performances, there are tradeoffs on either side of  $\zeta = 1$ . Slightly less than one yields faster responses that overshoot a small amount. Slightly greater than one yields slower responses less prone to oscillation.



**Figure 06.2:** free response  $y_{fr}(t)$  of a second-order system with initial conditions  $y(0)$  and  $\dot{y}(0) = 0$  for different values of  $\zeta$ . Underdamped, critically damped, and overdamped cases are displayed.

### 06.03.2 Step response

Second-order systems are subjected to a variety of forcing functions  $f$ . In this lecture, we examine a common one: step forcing. In what follows, we develop *forced response*  $y_{fo}$  solutions.

Unit step forcing of the form  $f(t) = u_s(t)$ , where  $u_s$  is the unit step function, models abrupt changes to the input. The solution is found by applying zero initial conditions ( $y(0) = 0$  and  $\dot{y}(0) = 0$ ) to the general solution. If the roots of the characteristic equation  $\lambda_1$  and  $\lambda_2$  are distinct, the forced response is

$$y_{fo}(t) = \frac{1}{\omega_n^2} \left( 1 - \frac{1}{\lambda_2 - \lambda_1} (\lambda_2 e^{\lambda_1 t} - \lambda_1 e^{\lambda_2 t}) \right) \quad (06.15)$$

where

$$\lambda_1, \lambda_2 = -\zeta \omega_n \pm \omega_n \sqrt{\zeta^2 - 1}. \quad (06.16)$$

Once again, there are five possibilities for the location of the roots of the characteristic equation  $\lambda_1$  and  $\lambda_2$ , all determined by the value of  $\zeta$ . However, there are three stable cases: underdamped, critically damped, and overdamped.

$\zeta \in (0, 1)$  **underdamped** In this case, the roots are distinct and complex:

$$\lambda_1, \lambda_2 = -\zeta\omega_n \pm j\omega_d. \quad (06.17)$$

From Equation 06.15, the forced step response is

$$y_{fo}(t) = \frac{1}{\omega_n^2} \left( 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \cos(\omega_d t + \psi) \right) \quad (06.18)$$

where the phase  $\psi$  is

$$\psi = -\arctan \frac{\zeta}{\sqrt{1-\zeta^2}}. \quad (06.19)$$

This response overshoots, oscillates about, and decays to  $1/\omega_n^2$ .

$\zeta = 1$  **critically damped** The roots are equal and real:

$$\lambda_1, \lambda_2 = -\omega_n \quad (06.20)$$

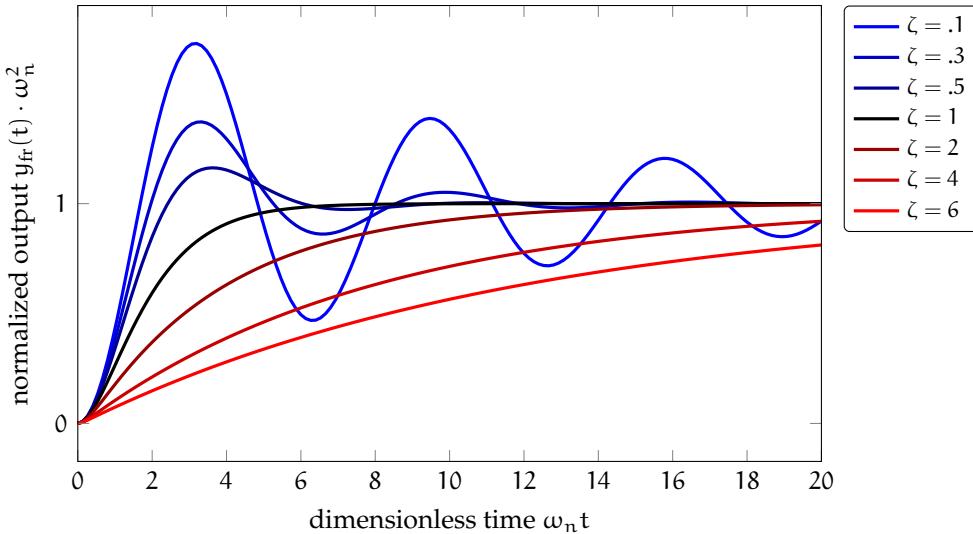
so the forced step of Equation 06.15 must be modified; it reduces to

$$y_{fo}(t) = \frac{1}{\omega_n^2} (1 - e^{-\omega_n t} (1 + \omega_n t)). \quad (06.21)$$

This response neither oscillates nor overshoots its steady-state of  $\frac{1}{\omega_n^2}$ , but just barely.

$\zeta \in (1, \infty)$  **overdamped** In this case, the roots are distinct and real, given by Equation 06.16. The forced step given by Equation 06.15 is the sum of two decaying real exponentials. These responses neither overshoot nor oscillate about their steady-state of  $1/\omega_n^2$ . With increasing  $\zeta$ , approach to steady-state slows.

Figure 06.3 displays the forced step response results. These responses are inverted versions of the free responses of 06.03.1. Note that a small damping ratio results in overshooting and oscillation about the steady-state value. In contrast, large damping ratio results in neither overshoot nor oscillation. However, both small and large damping ratios yield responses that take longer durations to approach equilibrium than damping ratios near unity. For this reason, the damping ratio of a system should be close to  $\zeta = 1$ . There are tradeoffs on either side of one. Slightly less yields faster responses that overshoot a small amount. Slightly greater than one yields slower responses less prone to oscillation.



**Figure 06.3:** forced step response  $y_{fo}(t)$  of a second-order system for different values of  $\zeta$ . Underdamped, critically damped, and overdamped cases are displayed.

### 06.03.3 Impulse and ramp responses

The response to all three singularity inputs are included in Table 06.2. These can be combined with the free response of Equation 06.3 using superposition.

### 06.03.4 An example with superposition

The results of the above are powerful not so much in themselves, but when they are wielded with the principle of superposition, as the following example shows.

Example 06.03-1 MRFM cantilever beam with initial condition and forcing

In magnetic resonance force microscopy (MRFM), the primary detector is a small cantilever beam with a magnetic tip. Model the beam as a spring-mass-damper system with mass  $m = 6 \text{ pg}$ ,<sup>a</sup> spring constant  $k = 15 \text{ mN/m}$ , and damping coefficient  $B = 37.7 \cdot 10^{-15} \text{ N}\cdot\text{s}/\text{m}$ . Magnetic input forces on the beam  $u(t)$  are applied directly to the

**Table 06.2:** responses of system  $\frac{d^2y}{dt^2} + 2\zeta\omega_n \frac{dy}{dt} + \omega_n^2 y = f$  to different singularity forcing. Note that  $\tau_1 = -1/\lambda_1$ ,  $\tau_2 = 1/\lambda_2$ , and  $\psi = -\arctan(\zeta/\sqrt{1-\zeta^2})$ .

damping	$f(t)$	characteristic response
$0 \leq \zeta < 1$	$\delta(t)$	$\frac{e^{-\zeta\omega_n t}}{\omega_n \sqrt{1-\zeta^2}} \sin(\omega_d t)$ $u_s(t) \quad \frac{1}{\omega_n^2} \left( 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \cos(\omega_d t + \psi) \right)$ $u_r(t) \quad \frac{1}{\omega_n^2} \left( t + \frac{e^{-\zeta\omega_n t}}{\omega_n} \left( 2\zeta \cos \omega_d t + \frac{2\zeta^2 - 1}{\sqrt{1-\zeta^2}} \sin \omega_d t \right) - \frac{2\zeta}{\omega_n} \right)$
	$\zeta = 1$	$\delta(t) \quad te^{-\omega_n t}$ $u_s(t) \quad \frac{1}{\omega_n^2} (1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t})$ $u_r(t) \quad \frac{1}{\omega_n^2} \left( t + \frac{2}{\omega_n} e^{-\omega_n t} + te^{-\omega_n t} - \frac{2}{\omega_n} \right)$
	$\zeta > 1$	$\delta(t) \quad \frac{1}{2\omega_n \sqrt{\zeta^2 - 1}} (e^{-t/\tau_1} - e^{-t/\tau_2})$ $u_s(t) \quad \frac{1}{\omega_n^2} \left( 1 - \frac{\omega_n}{2\sqrt{\zeta^2 - 1}} (\tau_1 e^{-t/\tau_1} - \tau_2 e^{-t/\tau_2}) \right)$ $u_r(t) \quad \frac{1}{\omega_n^2} \left( t - \frac{2\zeta}{\omega_n} + \frac{\omega_n}{2\sqrt{\zeta^2 - 1}} (\tau_1^2 e^{-t/\tau_1} - \tau_2^2 e^{-t/\tau_2}) \right)$

magnetic tip. That is, a Newtonian force-analysis yields the input-output ODE

$$m\ddot{y} + B\dot{y} + ky = u,$$

where  $y$  models the motion of the tip.

1. What is the natural frequency  $\omega_n$ ?
2. What is the damping ratio  $\zeta$ ?
3. Find the free response for initial conditions  $y(0) = 10$  nm and  $\dot{y}(0) = 0$ .
4. Find the impulse (forced) response for input  $u(t) = 3\delta(t)$ .

5. Find the total response for the initial condition and forcing, from above, are *both* applied.

---

<sup>a</sup>One pg =  $10^{-12}$ g =  $10^{-15}$ kg.

---

## State-space response

Recall that, for a state-space model, the state  $x$ , input  $u$ , and output  $y$  vectors interact through two equations:

$$\frac{dx}{dt} = f(x, u, t) \quad (07.1a)$$

$$y = g(x, u, t) \quad (07.1b)$$

where  $f$  and  $g$  are vector-valued functions that depend on the system. Together, they comprise what is called a *state-space model* of a system.

**state-space model**

In accordance with the definition of a state-determined system, given an initial condition  $x(t_0)$  and input  $u$ , the state  $x$  is determined for all  $t \geq t_0$ . Determining the state response requires the solution—analytic or numerical—of the vector differential equation [Equation 07.1a](#).

The second equation [\(07.1b\)](#) is *algebraic*. It expresses how the output  $y$  can be constructed from the state  $x$  and input  $u$ . This means we must first solve the state equation [\(07.1a\)](#) for  $x$ , then the output  $y$  is given by [Equation 07.1b](#).

Just because we know that, for a state-determined system, there *exists* a solution to [Equation 07.1a](#), doesn't mean we know how to find it. In general,  $f : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R} \rightarrow \mathbb{R}^n$  and  $g : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R} \rightarrow \mathbb{R}^m$  can be nonlinear functions.<sup>1</sup> We don't know how to solve most nonlinear state equations analytically. An additional complication can arise when, in addition to states and inputs, system parameters are themselves time-varying (note the explicit time  $t$  argument of  $f$  and  $g$ ). Fortunately, often a linear, time-invariant (LTI) model is sufficient.

---

<sup>1</sup>Technically, since  $x$  and  $u$  are themselves functions,  $f$  and  $g$  are *functionals*.

Recall that an LTI state-space model is of the form

$$\frac{dx}{dt} = Ax + Bu \quad (07.2a)$$

$$y = Cx + Du, \quad (07.2b)$$

where A, B, C, and D are constant matrices containing system lumped-parameters such as mass or inductance. See [Chapter 03](#) for details on the derivation of such models.

In this chapter, we learn to solve [Equation 07.2a](#) for the state response and substitute the result into [Equation 07.2b](#) for the output response. First, we learn an analytic solution technique. Afterward, we learn simple software tools for numerical solution techniques.

## Lecture 07.01 Solving for the state-space response

In this lecture, we solve the state equation for the *state response*  $x(t)$  and substitute this into the output equation for the *output response*  $y(t)$ .

**state response**  
**output response**

### 07.01.1 State response

The state equation can be solved by a synthesis of familiar techniques, as follows. First, we rearrange:

$$\frac{dx}{dt} - Ax = Bu. \quad (07.3)$$

An integrating factor would be clutch, but what should it be? It looks analogous to a scalar ODE that would use the natural exponential  $\exp(-at)$  (for positive constant  $a$ ), but we have a vector ODE. We need a matrix-version of the exponential. Recall that a series definition of the scalar exponential function  $\exp : \mathbb{C} \rightarrow \mathbb{C}$  is

We define the *matrix exponential*  $\exp : \mathbb{C}^n \times \mathbb{C}^n \rightarrow \mathbb{C}^n \times \mathbb{C}^n$  (we use the same symbol) to be, for  $n \times n$  complex matrix  $Z$ ,

$$\exp Z = \sum_{k=0}^{\infty} \frac{1}{k!} Z^k. \quad (07.4)$$

because why not? For the hell of it, let's see if the matrix exponential

$$\exp(-At) \quad (07.5)$$

works as an integrating factor, if for no other reason than it was constructed to be a sort of matrix-analog of  $\exp(-at)$ , which would work for the scalar case. Premultiplying (07.3) on both sides:

Rearranging and integrating over the interval  $(0, t)$ ,

**state response  
solution  
state transition  
matrix**

This last expression can be solved for  $x$ , the *state response solution*. Before we do this, however, let's define the matrix function called the *state transition matrix*  $\Phi$  to be the matrix-valued function

$$\Phi(t) = \exp(At), \quad (07.6)$$

Substituting  $\Phi$  and solving,

$$x = \Phi(t)x(0) + \Phi(t) \int_0^t \Phi(-\tau)Bu(\tau)d\tau \quad (07.7a)$$

$$= \Phi(t)x(0) + \int_0^t \Phi(t-\tau)Bu(\tau)d\tau. \quad (07.7b)$$

**free response  
forced response**

Note that the first term of each version of Equation 07.7 is the *free response* (due to initial conditions) and the second term is the *forced response* (due to inputs).

### 07.01.2 State transition matrix

The state transition matrix  $\Phi$  introduced in Equation 07.6 wound up being a key aspect of the response, which is why we call it that. We used two of its properties (in matrix exponential form) during that derivation: the *initial-value property*

**initial-value  
property**

$$\Phi(0) = I \quad (\text{where } I \text{ is the identity matrix})$$

**inverse property** and the *inverse*

$$\Phi^{-1}(t) = \Phi(-t). \quad (07.8)$$

**bootstrapping  
property**

There is a third property that might be called the *bootstrapping property*: for time intervals  $\Delta t_i$ ,

$$\Phi(\Delta t_1 + \Delta t_2 + \dots) = \Phi(\Delta t_1)\Phi(\Delta t_2)\dots. \quad (07.9)$$

This allows one to compute the state transition matrix<sup>2</sup> *incrementally*, from one previously computed.

A final property we'll consider is the special-case of a *diagonal*  $A$  with diagonal elements  $a_{11}, a_{22}, \dots, a_{nn}$ , which yields a diagonal state transition matrix

**diagonal property**

The last property turns out to be quite convenient for *deriving*  $\Phi$  for a given system, as we will see in [Lecture 07.03](#). For now, we must rely on the definition of  $\Phi$  from [Equation 07.6](#) and the series definition of the matrix exponential from [Equation 07.4](#). This requires us to derive the first several terms of the series solution and attempt to divine the corresponding scalar exponential series, a rather tedious task. Other than to familiarize ourselves with the definition through exercises, we prefer the derivation method of [Lecture 07.03](#).

**deriving  $\Phi$**

### 07.01.3 Output response

The output response  $y(t)$  requires little additional solution: assuming we have solved for the state response  $x(t)$ , the output is given in the output equation [Equation 07.2b](#). Through direct substitution, we find the *output response solution*

**output response solution**

$$y(t) = Cx(t) + Du(t) \quad (07.10a)$$

$$= C\Phi(t)x(0) + C \int_0^t \Phi(t-\tau)Bu(\tau)d\tau + Du(t). \quad (07.10b)$$

---

<sup>2</sup>As is common, we refer to it as the “state transition matrix at a certain time,” but, technically, it’s the *image* of the state transition matrix (which is actually a matrix-valued function) at a certain time. It is good to occasionally acknowledge the violence we do to math.

## Lecture 07.02 Linear algebraic eigenproblem

- eigenproblem** The linear algebraic *eigenproblem* can be simply stated. For  $n \times n$  real matrix  $A$ ,  $n \times 1$  complex vector  $m$ , and  $\lambda \in \mathbb{C}$ ,  $m$  is defined as an *eigenvector* of  $A$  if and only if it is nonzero and

$$Am = \lambda m \quad (07.11)$$

- eigenvalue** for some  $\lambda$ , which is called the corresponding *eigenvalue*. That is,  $m$  is an eigenvector of  $A$  if its linear transformation by  $A$  is equivalent to its scaling; i.e. an eigenvector of  $A$  is a vector of which  $A$  changes the length, but not the direction.

Since a matrix can have several eigenvectors and corresponding eigenvalues, we typically index them with a subscript; e.g.  $m_i$  pairs with  $\lambda_i$ .

### 07.02.1 Solving for eigenvalues

Equation 07.11 can be rearranged:

$$(\lambda I - A)m = 0. \quad (07.12)$$

For a nontrivial solution for  $m$ ,

$$\det(\lambda I - A) = 0, \quad (07.13)$$

- characteristic equation eigenvalues** which has as its left-hand-side a polynomial in  $\lambda$  and is called the *characteristic equation*. We define *eigenvalues* to be the roots of the characteristic equation.

#### Box 07.1 eigenvalues and roots of the characteristic equation

If  $A$  is taken to be the linear state-space representation  $A$ , and the state-space model is converted to an input-output differential equation, the resulting ODE's "characteristic equation" would be identical to this matrix characteristic equation. Therefore, everything we already understand about the roots of the "characteristic equation" of an i/o ODE—especially that they govern the transient response and stability of a system—holds for a system's  $A$ -matrix eigenvalues.

Here we consider only the case of  $n$  distinct eigenvalues. For eigenvalues of (algebraic) multiplicity greater than one (i.e. repeated roots), see the discussion of Lecture B.01.

## 07.02.2 Solving for eigenvectors

Each eigenvalue  $\lambda_i$  has a corresponding eigenvector  $m_i$ . Substituting each  $\lambda_i$  into [Equation 07.12](#), one can solve for a corresponding eigenvector. It's important to note that an eigenvector is unique within a scaling factor. That is, if  $m_i$  is an eigenvector corresponding to  $\lambda_i$ , so is  $3m_i$ .<sup>3</sup>

Example 07.02-1 eigenproblem for a  $2 \times 2$  matrix

Let

$$A = \begin{bmatrix} 2 & -4 \\ -1 & -1 \end{bmatrix}.$$

Find the eigenvalues and eigenvectors of  $A$ .

<sup>3</sup>Also of note is that  $\lambda_i$  and  $m_i$  can be complex.



## Lecture 07.03 Diagonalizing basis

It is useful to transform a system's state vector  $\mathbf{x}$  into a special basis that *diagonalizes*—leaves nonzero components along only the diagonal—the system's  $A$ -matrix. For systems with  $n$  distinct eigenvalues, to which we limit ourselves in this discussion,<sup>4</sup> this is always possible. In diagonalized form, it will be relatively easy to solve for the state transition matrix  $\Phi$ .

**diagonalizes**

### 07.03.1 Changing basis in the state equation

As with all basis transformations, the basis transformation we seek can be written

$$\mathbf{x} = \mathbf{P}\mathbf{x}' \Rightarrow \mathbf{x}' = \mathbf{P}^{-1}\mathbf{x}, \quad (07.14)$$

where  $\mathbf{P}$  is the *transformation matrix*,  $\mathbf{x}$  is a representation of the state vector in the original basis, and  $\mathbf{x}'$  is a representation of the state vector in the new basis.<sup>5</sup>

**transformation matrix**

Substituting this transformation into the standard linear state-model equations yields the model

$$\dot{\mathbf{x}}' = \underbrace{\mathbf{P}^{-1} \mathbf{A} \mathbf{P} \mathbf{x}'}_{\mathbf{A}'} + \underbrace{\mathbf{P}^{-1} \mathbf{B} \mathbf{u}}_{\mathbf{B}'}, \quad (07.15a)$$

$$\mathbf{y} = \underbrace{\mathbf{C} \mathbf{P} \mathbf{x}'}_{\mathbf{C}'} + \underbrace{\mathbf{D} \mathbf{u}}_{\mathbf{D}'}. \quad (07.15b)$$

### 07.03.2 Modal and eigenvalue matrices

Let a state equation have matrix  $A$  with  $n$  distinct eigenvalues ( $\lambda_i$ ) and eigenvectors ( $\mathbf{m}_i$ ). Let the *eigenvalue matrix*  $\Lambda$  be defined as

**eigenvalue matrix**

Furthermore, let the *modal matrix*  $M$  be defined as

**modal matrix**

<sup>4</sup>See Lecture B.01 for general considerations.

<sup>5</sup>We are being a bit fast-and-loose with terminology here: a vector is an object that does not change under basis transformation, only its components and basis vectors do. However, we use the common notational and terminological abuses.

$$M = \begin{bmatrix} & & & \\ | & | & \cdots & | \\ m_1 & m_2 & \cdots & m_n \\ | & | & & | \end{bmatrix} \quad (07.16)$$

### 07.03.3 Diagonalization of the state equation

Let the modal matrix  $M$  be the transformation matrix for our state-model. Then<sup>6</sup>  $x' = M^{-1}x$ .

The state equation becomes

$$\dot{x}' = M^{-1}Ax' + M^{-1}Bu. \quad (07.17)$$

The eigenproblem implies that

**diagonalized** That is,  $A' = \Lambda$ ! Recall that  $\Lambda$  is diagonal; therefore, we have *diagonalized* the state-space model. In full-form, the diagonalized model is

$$\dot{x}' = \underbrace{\Lambda}_{A'} x' + \underbrace{M^{-1}B}_{B'} u \quad (07.18a)$$

$$y = \underbrace{CM}_{C'} x' + \underbrace{D}_{D'} u. \quad (07.18b)$$

### 07.03.4 Computing the state transition matrix

Recall our definition of the state transition matrix  $\Phi(t) = e^{\Lambda t}$ . Directly applying this to the diagonalized system of Equation 07.18,

$$\Phi'(t) = e^{\Lambda t} \quad (07.19a)$$

$$= \begin{bmatrix} e^{\lambda_1 t} & & & 0 \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ 0 & & & e^{\lambda_n t} \end{bmatrix}. \quad (07.19b)$$

**diagonal property** In this last equality, we have used the *diagonal property* of the state transition

<sup>6</sup>As long as there are  $n$  distinct eigenvalues,  $M$  is invertible.

matrix, defined in 07.01.2.

Recall that the free response solution to the state equation is given by the initial condition and state transition matrix, so

$$\mathbf{x}'_{\text{fr}}(t) = \Phi'(t)\mathbf{x}'(0) \quad (07.20a)$$

$$= x'_1(0)e^{\lambda_1 t} + x'_2(0)e^{\lambda_2 t} + \cdots + x'_n(0)e^{\lambda_n t} \quad (07.20b)$$

where the initial conditions are  $\mathbf{x}'(0) = M^{-1}\mathbf{x}(0)$ . We have completely decoupled each state's free response, one of the remarkable qualities of the diagonalized system.

At this point, one could simply solve the diagonalized system for  $\mathbf{x}'(t)$ , then convert the solution to the original basis with  $\mathbf{x}(t) = M\mathbf{x}'(t)$ .

Sometimes, we might prefer to transform the diagonalized-basis state transition matrix into the original basis. The following is a derivation of that transformation.

Beginning with the free response solution in the diagonalized-basis and transforming the equation into the original basis, we find an expression for the original state transition matrix, as follows.

This last expression is just the free response solution in the original basis, so we can identify

$$\Phi(t) = M\Phi'(t)M^{-1}. \quad (07.21)$$

This is a powerful result. Equation 07.21 is the preferred method of deriving the state transition matrix for a given system. The eigenvalues give  $\Phi'$  and the eigenvectors give  $M$ .

#### Example 07.03-1 state free response

For the state equation

$$\dot{\mathbf{x}} = \begin{bmatrix} -2 & 2 \\ 2 & -3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} \mathbf{u}$$

find the state's free response to initial condition  $\mathbf{x}(0) = [2 \ -1]^\top$ .



## Lecture 07.04 Simulating state-space response

Ahem.<sup>7</sup>

For many nonlinear models, numerical solution of the state equation is required. For linear models, we can always solve them analytically using the methods of this chapter. However, due to its convenience, we will often want to use numerical techniques even when analytic ones are available.

Matlab has several built-in and *Control Systems Toolbox* functions for analyzing state-space system models, especially *linear* models. We'll explore a few, here.

Consider, for instance, a linear state model with the following A, B, C, and D matrices:

$$A = \begin{bmatrix} -3 & 4 & 5 \\ 0 & -2 & 3 \\ 0 & -6 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (07.22a)$$

```
A = [-3,4,5;0,-2,3;0,-6,1];
B = [1;0;1];
C = [1,0,0;0,-1,0];
D = [0;0];
```

For a step input  $u(t) = 3u_s(t)$  and initial state  $x(0) = [1 \ 2 \ 3]^\top$ , let's compare analytic and numerical solutions for the output response  $y(t)$ .

```
u = @(t) 3*ones(size(t)); % for t>=0
x_0 = [1; 2; 3];
```

### 07.04.1 Analytic solution

For an analytic solution, we'll use a rearranged version of Equation 07.10.<sup>8</sup>

---

<sup>7</sup>The source of this lecture can be downloaded as a Matlab m-file at [http://ricopic.one/dynamic\\_systems/source/simulating\\_state\\_space\\_response.m](http://ricopic.one/dynamic_systems/source/simulating_state_space_response.m).

<sup>8</sup>Although we call this the "analytic" solution, we are not solving for a detailed symbolic expression, although we \*could\*. In fact, Equation 07.23 \*is\* the analytic solution and what follows is an attempt to represent it graphically.

$$\mathbf{y}(t) = \mathbf{C}\Phi(t)\mathbf{x}(0) + \mathbf{C}\Phi(t) \int_0^t \Phi(-\tau) \mathbf{B}\mathbf{u}(\tau) d\tau + \mathbf{D}\mathbf{u}(t). \quad (07.23a)$$

First, we need the state transition matrix  $\Phi(t)$ , so we consider the eigenproblem.

```
[M, L] = eig(A)
```

M =

1.0000 + 0.0000i	0.7522 + 0.0000i	0.7522 + 0.0000i
0.0000 + 0.0000i	0.3717 + 0.0810i	0.3717 - 0.0810i
0.0000 + 0.0000i	0.0787 + 0.5322i	0.0787 - 0.5322i

L =

-3.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i
0.0000 + 0.0000i	-0.5000 + 3.9686i	0.0000 + 0.0000i
0.0000 + 0.0000i	0.0000 + 0.0000i	-0.5000 - 3.9686i

Note that, when assigning its output to two variables M and L, the eig function returns the modal matrix to M and the eigenvalue matrix to L. The modal matrix of eigenvectors M has each column (eigenvector) normalized to unity. Also notice that M and L are *complex*. The imaginary parts of two eigenvalues and their corresponding eigenvectors are significant. Finally, since the *real* parts of all eigenvalues are *negative*, the system is stable.

The “diagonal”-basis state transition matrix  $\Phi'(t)$  is simply

$$\Phi'(t) = e^{\Lambda t}. \quad (07.24)$$

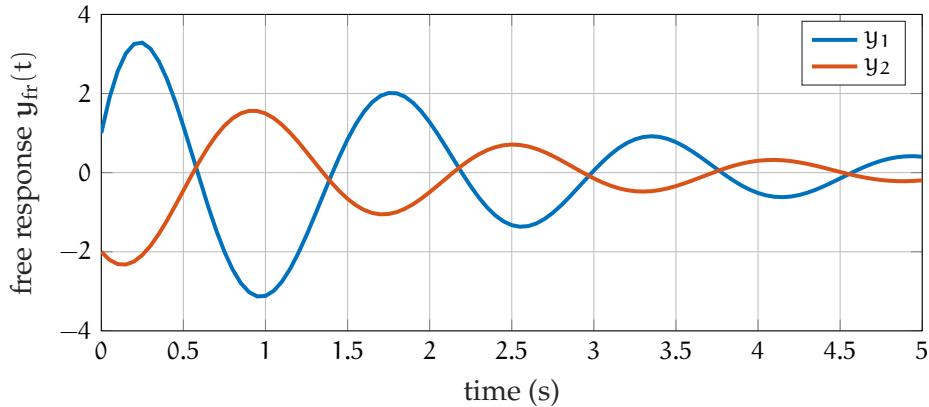
Let's define this as an “anonymous” function.

```
Phi_p = @(t) diag(diag(exp(L*t))); % diags to get diagonal mat
```

The original-basis state transition matrix  $\Phi(t)$  is, from Equation 07.21,

$$\Phi(t) = M\Phi'(t)M^{-1}. \quad (07.25)$$

```
M_inv = M^-1; % compute just once, not on every call
Phi = @(t) M*Phi_p(t)*M_inv;
```

Figure 07.1: free response  $y_{fr}$ .

#### 07.04.1.1 Free response

The free response is relatively straightforward to compute.

```
t_a = 0:.05:5; % simulation time
y_fr = NaN*ones(size(C,1),length(t_a)); % initialize
for i = 1:length(t_a)
    y_fr(:,i) = C*Phi(t_a(i))*x_0;
end
y_fr(:,1:3) % first three columns
```

```
ans =
1.0000 - 0.0000i   1.8922 - 0.0000i   2.5646 - 0.0000i
-2.0000 + 0.0000i  -2.2030 + 0.0000i  -2.3105 + 0.0000i
```

A time array  $t_a$  was defined such that  $\Phi$  could be evaluated. The first three columns of  $y_{fr}$  are printed for the first three moments in time. Note how there's a "hanging chad" of imaginary components. Before we realize them, let's make sure they're negligibly tiny.

```
max(max(abs(imag(y_fr))))
y_fr = real(y_fr);
```

```
ans =
5.2907e-16
```

The results are plotted in Figure 07.1. As we might expect from the eigenvalues, the free responses of both outputs oscillate and decay.

#### 07.04.1.2 Forced response

Now, there is the matter of integration in Equation 07.23. Since Matlab does not excel in symbolic manipulation, we have chosen to avoid attempting to write the solution, symbolically.<sup>9</sup> For this reason, we choose a simple numerical (trapezoidal) approximation of the integral using the `trapz` function.

First, the integrand can be evaluated over the simulation interval.

```
integrand_a = NaN*ones(size(C, 2), length(t_a)); % initialize
for i = 1:length(t_a)
    tau = t_a(i);
    integrand_a(:, i) = Phi(-tau)*B*u(tau);
end
```

Now, numerically integrate.

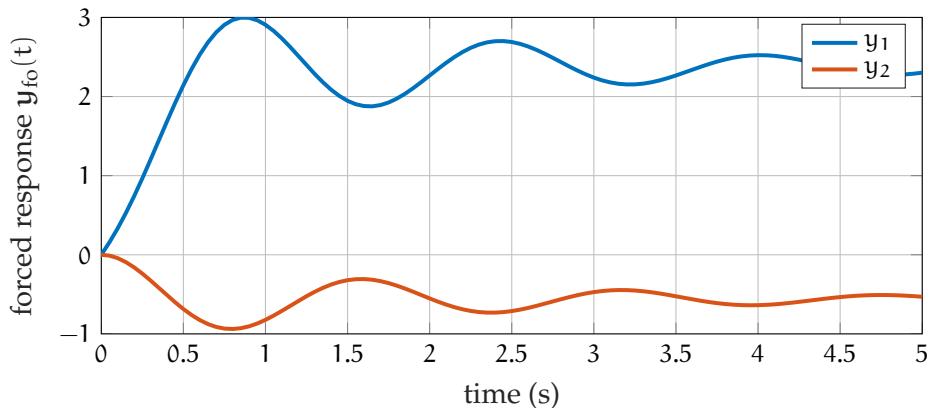
```
integral_a = zeros(size(integrand_a));
for i = 2:length(t_a)
    i_up = i; % upper limit of integration
    integral_a(:, i) = ... % transposes for trapz
        trapz(t_a(1:i_up)', integrand_a(:, 1:i_up)')';
end
```

Now, evaluate the forced response at each time.

```
y_f0 = NaN*ones(size(C, 1), length(t_a)); % initialize
for i = 1:length(t_a)
    y_f0(:, i) = C*Phi(t_a(i))*integral_a(:, i);
end
y_f0(:, 1:3) % first three columns
```

```
ans =
0.0000 + 0.0000i  0.1583 - 0.0000i  0.3342 - 0.0000i
0.0000 + 0.0000i  -0.0109 + 0.0000i  -0.0426 + 0.0000i
```

<sup>9</sup>Mathematica or SageMath would be preferable for this.

**Figure 07.2:** forced response  $y_{fo}$ .

```
max(max(abs(imag(y_fo))))  
y_fo = real(y_fo);
```

```
ans =  
2.1409e-16
```

The forced response is shown in Figure 07.2, which shows damped oscillations.

#### 07.04.1.3 Total response

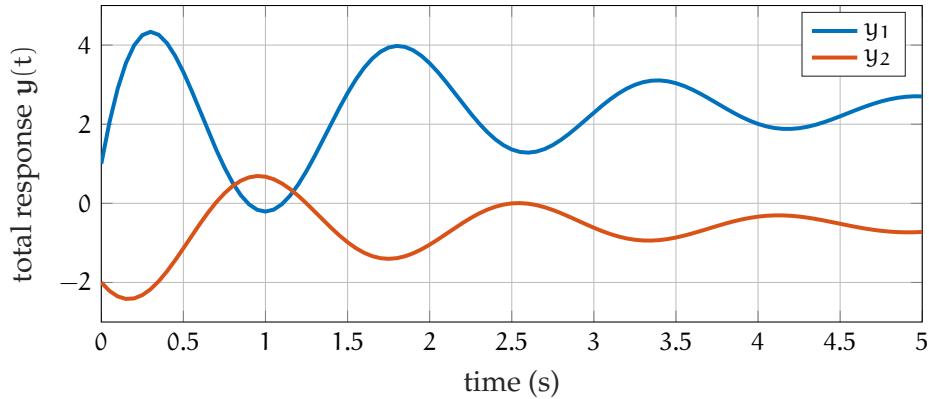
The total response is found from the sum of the free and forced responses:  $y(t) = y_{fr} + y_{fo}$ . We can simply sum the arrays.

```
y_t = y_fr + y_fo;
```

The result is plotted in Figure 07.3.

#### 07.04.2 Numerical solution

The numerical solution of the state equations is rather simple using Matlab's `ss` and `step` or `lsim` commands, as we show, here. First, we define an `ss` model object—a special kind of object that encodes a state-space model.

Figure 07.3: total response  $y$ .

```
sys = ss(A, B, C, D);
```

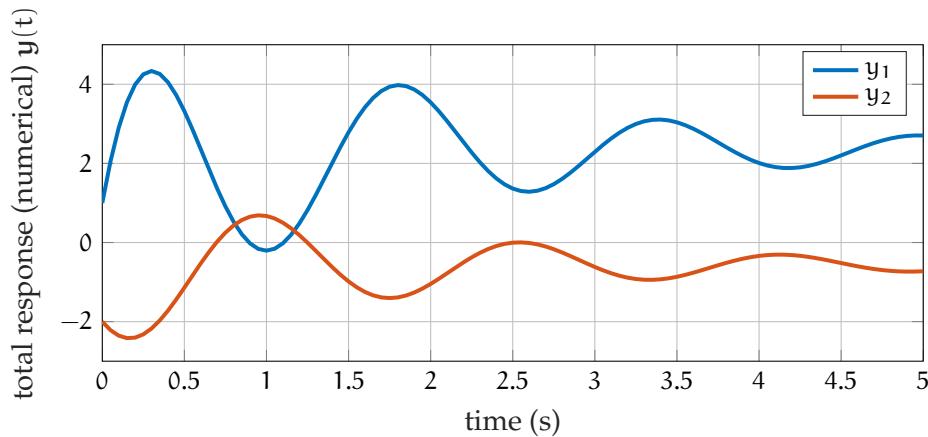
At this point, using the `step` function would be the easiest way to solve for the step response. However, we choose the more-general `lsim` for demonstration purposes.

```
y_t_num = lsim(sys, u(t_a), t_a, x_0);
```

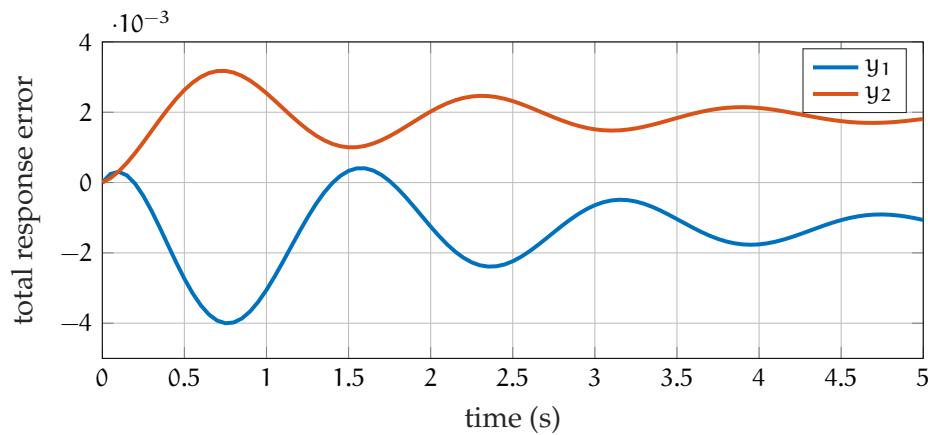
This total solution is shown in Figure 07.4.

```
d_y = y_t - y_t_num';
```

Figure 07.5 shows a plot of the differences between the analytic total solution  $y_t$  and the numerical  $y_t\_num$  for each output. Note that calling this “error” is a bit presumptuous, given that we used numerical integration in the analytic solution. If a more accurate method is desired, working out the solution, symbolically, is the best.



**Figure 07.4:** total response  $y$  from `lsim`.



**Figure 07.5:** total response error  $y_t - y_{t\_num}$ .

## Exercises

### Exercise 07.1

Let a system have the following state and output equation matrices:

$$A = \begin{bmatrix} -3 & 0 \\ 1 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad D = [0].$$

For this system, answer the following imperatives.

- Find the *eigenvalue matrix*  $\Lambda$  and comment on the stability of the system (justify your comment). Use the convention that  $\lambda_1 \leq \lambda_2$  and order  $\Lambda$  accordingly.
- Find the eigenvectors and the *modal matrix*  $M$ .
- Find the *state transition matrix*  $\Phi(t)$ . Hint: first find the “diagonalized” state transition matrix  $\Phi'(t)$ .
- Using the state transition matrix, find the *output homogeneous solution* for initial condition

$$x(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

### Exercise 07.2

Let a system have the following state and output equation matrices:

$$A = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D = [0].$$

For this system, answer the following imperatives.

- Find the *eigenvalue matrix*  $\Lambda$  and comment on the stability of the system (justify your comment). Use the convention that  $\lambda_1 \leq \lambda_2$  and order  $\Lambda$  accordingly.
- Find the eigenvectors and the *modal matrix*  $M$ .
- Find the *state transition matrix*  $\Phi(t)$ . Hint: first find the “diagonalized” state transition matrix  $\Phi'(t)$ .
- Using the state transition matrix, find the *output homogeneous solution* for initial condition

$$x(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

## **Part III**

# **Modeling other systems**



## Lumped-parameter modeling fluid and thermal systems

We now consider the *lumped-parameter modeling* of *fluid systems* and *thermal systems*. The linear graph-based, state-space modeling techniques of [Chapters 02 to 04](#) are called back up to service for this purpose. Recall that this method defines several types of discrete elements in an energy domain—in [Chapters 02 and 03](#), the electrical and mechanical energy domains. Also recall from [Chapter 04](#) that energy transducing elements allow energy to flow among domains. In this chapter, we introduce fluid and thermal energy domains and discrete and transducing elements associated therewith.

**lumped-parameter  
modeling  
fluid systems  
thermal systems**

The analogs between the mechanical and electrical systems from [Chapter 02](#) are expanded to include fluid and thermal systems. This generalization allows us to include, in addition to electromechanical systems, inter-domain systems including electrical, mechanical, fluid, and thermal systems.

This chapter begins by defining discrete lumped-parameter elements for fluid and thermal systems. We then categorize these into energy source, energy storage (A-type and T-type), and energy dissipative (D-type) elements, allowing us to immediately construct linear graphs and normal trees in the manner of [Chapter 02](#). Then we can directly apply the methods of [Chapter 03](#) to construct state-space models of systems that include fluid and thermal elements.

## Lecture 08.01 Fluid system elements

Detailed *distributed* models of fluids, such as the Navier-Stokes equations, are necessary for understanding many aspects of fluid systems and for guiding their design (e.g. a pump or an underwater vehicle). However, a great many fluid systems are networks of pipes, tanks, pumps, valves, orifices, and elevation changes—and at this *system-level*, a different approach is required.

As with electrical and mechanical systems, we can describe fluid systems as consisting of discrete lumped-parameter elements. The dynamic models that can be developed from considering these elements are often precisely the right granularity for system-level design.

We now introduce a few lumped-parameter elements for modeling fluid systems. Let a *volumetric flowrate*  $Q$  and *pressure drop*  $P$  be input to a port in a fluid element. Since, for fluid systems, the power into the element is

$$\mathcal{P}(t) = Q(t)P(t) \quad (08.1)$$

**power-flow variables** we call  $Q$  and  $P$  the *power-flow variables* for fluid systems. A fluid element has two distinct locations between which its pressure drop is defined. We call a reference pressure *ground*.

**work** Work done on the system over the time interval  $[0, T]$  is defined as

$$W \equiv \int_0^T \mathcal{P}(\tau) d\tau. \quad (08.2)$$

Therefore, the work done on a fluid system is

$$W = \int_0^T Q(\tau)P(\tau)d\tau. \quad (08.3)$$

**pressure momentum** The *pressure momentum*  $\Gamma$  is

$$\Gamma(t) = \int_0^t P(\tau)d\tau + \Gamma(0). \quad (08.4)$$

**volume** Similarly, the *volume* is

$$V(t) = \int_0^t Q(\tau)d\tau + V(0). \quad (08.5)$$

**energy storage elements** We now consider two elements that can store energy, called *energy storage elements*; an element that can dissipate energy to a system's environment, called an *energy dissipative element*; and two elements that can supply power from outside a system, called *source elements*.

### 08.01.1 Fluid inertances

When fluid flows through a pipe, it has a momentum associated with it. The more mass (fluid density by its volume) moving in one direction and the faster it moves, the more momentum. This is stored kinetic energy. The discrete element we now introduce models this aspect of fluid systems.

A *fluid inertance* is defined as an element for which the pressure momentum  $\Gamma$  across it is a monotonic function of the volumetric flowrate  $Q$  through it. A *linear inertance* is such that

$$\Gamma(t) = IQ(t), \quad (08.6)$$

where  $I$  is called the *inertance* and is typically a function of pipe geometry and fluid properties. This is called the element's *constitutive equation* because it constitutes what it means to be an inertance.

Although there are nonlinear inertances, we can often use a linear model for analysis in some operating regime. The *elemental equation* for a linear inertance can be found by time-differentiating Equation 08.6 to obtain

**fluid inertance**

**linear inertance**

**inertance**  
**constitutive**  
**equation**

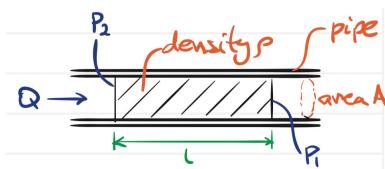
**elemental equation**

We call this the elemental equation because it relates the element's power-flow variables  $Q$  and  $P$ .

An inertance stores energy as kinetic energy, making it an *energy storage element*. The amount of energy it stores depends on the volumetric flowrate it contains. For a linear inertance,

$$\mathcal{E}(t) = \frac{1}{2}IQ(t)^2. \quad (08.7)$$

The inertance  $I$  for a uniform pipe can be derived, as follows, with reference to the sectioned pipe of Figure 08.1. For an incompressible fluid flowing through a pipe of uniform area  $A$  and length  $L$ , with uniform velocity profile (a convenient fiction), an element of fluid obeys Newton's second law, from which several interesting



**Figure 08.1:** a section of pipe for deriving its inertance.

equalities can be derived:

$$\begin{aligned}
 F &= m \frac{dv}{dt} \Rightarrow \\
 \frac{F}{A} &= \frac{m}{A} \frac{dv}{dt} \Rightarrow \\
 P &= \frac{\rho AL}{A} \frac{dv}{dt} \\
 &= \rho L \frac{d}{dt} \left( \frac{Q}{A} \right) \\
 &= \frac{\rho L}{A} \frac{dQ}{dt} \Rightarrow \\
 \frac{dQ}{dt} &= \underbrace{\frac{A}{\rho L}}_{I/I} P.
 \end{aligned}$$

From this last equality, it is clear that, for a uniform pipe and the assumptions, above,

$$I = \frac{\rho L}{A}. \quad (08.8)$$

Clearly, *long, thin* pipes will have more inertance. In fact, we often ignore inertance in modeling a pipe, unless it is relatively long and thin.

### 08.01.2 Fluid capacitors

When fluid is stored in tanks or in pressure vessels, it stores potential energy via its pressure drop  $P$ . For instance, a tank with a column of fluid will have a pressure drop associated with the height of the column. This is analogous to how an electronic capacitor stores its energy via its voltage. For this reason, we call such fluid elements *fluid capacitors*.

A linear fluid capacitor with capacitance  $C$ , pressure drop  $P$ , and volume  $V$  has the constitutive equation

$$V = CP. \quad (08.9)$$

Once again, time-differentiating the constitutive equation gives us the elemental equation:

Fluid capacitors can store energy (making them *energy storage elements*) in fluid potential energy, which, for a linear capacitor is

$$\mathcal{E}(t) = \frac{1}{2}CP^2. \quad (08.10)$$

### 08.01.3 Fluid resistors

*Fluid resistors* are defined as elements for which the volumetric flowrate  $Q$  through the element is a monotonic function of the pressure drop  $P$  across it. *Linear fluid resistors* have constitutive equation (and, it turns out, elemental equation)

$$Q = \frac{1}{R}P \quad (08.11)$$

where  $R$  is called the *fluid resistance*.

Fluid resistors dissipate energy from the system (to heat), making them *energy dissipative elements*.

**fluid resistors**

**linear fluid resistors**

**fluid resistance**

### 08.01.4 Flowrate and pressure drop sources

Fluid sources include pumps, runoff, etc.

An *ideal volumetric flowrate source* is an element that provides arbitrary energy to a system via an independent (of the system) volumetric flowrate. The corresponding pressure drop across the element depends on the system.

**ideal volumetric flowrate source**

An *ideal pressure drop source* is an element that provides arbitrary energy to a system via an independent (of the system) pressure drop. The corresponding volumetric flowrate through the element depends on the system.

**ideal pressure drop source**

Real sources, usually pumps, cannot be ideal sources, but in some instances can approximate them. More typical is to include a fluid resistor in tandem with an ideal source, as we did with electrical resistors for real electrical sources.

### 08.01.5 Generalized element and variable types

In keeping with the definitions of Chapter 01, pressure  $P$  is an *across-variable* and flowrate  $Q$  is a *through-variable*.

**across-variable**  
**through-variable**

Consequently, the fluid capacitor is considered an *A-type* energy storage element. Similarly, the fluid inertance is a *T-type* energy storage element.

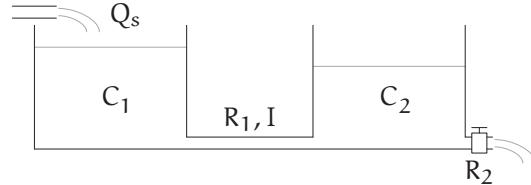
**A-type**  
**T-type**

**D-type** Clearly, a fluid resistor is a *D-type* energy dissipative element.

Pressure sources are, then, across-variable sources and volumetric flowrate sources are through-variable sources.

#### Example 08.01-1 fluid system graph

Use the schematic to draw a linear graph of the system.



## Lecture 08.02 Thermal system elements

Systems in which heat flow is of interest are called *thermal systems*. For instance, heat generated by an engine or a server farm flows through several bodies via the three modes of heat transfer: *conduction*, *convection*, and *radiation*. This is, of course, a dynamic process.

A detailed model would require a spatial continuum. However, we are often concerned with, say, the maximum temperature an engine will reach for different speeds or the maximum density of a server farm while avoiding overheating. Or, more precisely, *how* a given heat generation affects the temperature response of system components.

As with electrical, mechanical, and fluid systems, we can describe thermal systems as consisting of discrete lumped-parameter elements. The dynamic models that can be developed from considering these elements are often precisely the right granularity for system-level design.

We now introduce a few lumped-parameter elements for modeling thermal systems. Let a *heat flow rate*  $q$  (SI units W) and *temperature*  $T$  (SI units K or C) be input to a port in a thermal element. There are three structural differences between thermal systems and the other types we've considered. We are confronted with the *first*, here, when we consider that heat power is typically *not* considered to be the product of two variables; rather, the heat flow rate  $q$  is *already power*:

$$\mathcal{P}(t) = q(t). \quad (08.12)$$

A thermal element has two distinct locations between which its temperature drop is defined. We call a reference temperature *ground*.

The *heat energy*  $H$  of a system with initial heat  $H(0)$  is

$$H(t) = \int_0^t \mathcal{P}(\tau) d\tau + H(0). \quad (08.13)$$

We now consider an element that can store energy, called an *energy storage element*; an element that resists power flow; and two elements that can supply power from outside a system, called *source elements*. The *second difference* is that there is only one type of energy storage element in the thermal domain.

### 08.02.1 Thermal capacitors

When heat is stored in an object, it stores potential energy via its temperature  $T$ . This is analogous to how an electronic capacitor stores its energy via its voltage. For this reason, we call such thermal elements *thermal capacitors*.

**thermal systems**

**conduction**  
**convection**  
**radiation**

**heat flow rate**  
**temperature**

**first difference**

**ground**  
**heat**

**energy storage element**  
**source elements**  
**second difference**

**thermal capacitors**

A linear thermal capacitor with thermal capacitance  $C$  (SI units J/K), temperature  $T$ , and heat  $H$  has the constitutive equation

$$H = CT. \quad (08.14)$$

Once again, time-differentiating the constitutive equation gives us the elemental equation:

**extensive property**  
**specific heat capacity**  
**intensive property**

The thermal capacitance  $C$  is an *extensive property*—that is, it depends on the amount of its substance. This is opposed to the *specific heat capacity*  $c$  (units J/K/kg), an *intensive property*: one that is independent of the amount of its substance. These quantities are related for an object of mass  $m$  by the equation

$$C = mc. \quad (08.15)$$

**thermal resistors**

### 08.02.2 Thermal resistors

*Thermal resistors* are defined as elements for which the heat flowrate  $q$  through the element is a monotonic function of the temperature drop  $T$  across it. *Linear thermal resistors* have constitutive equation (and, it turns out, elemental equation)

$$q = \frac{1}{R}T \quad (08.16)$$

**thermal resistance**

where  $R$  is called the *thermal resistance*.

**third difference**

Thermal resistors do *not* dissipate energy from the system, which is the *third* difference between thermal and other energy domains we've considered. After all, the other “resistive” elements all dissipate energy *to heat*. Rather than dissipate energy, they simply impede its flow.

All three modes of heat transfer are modeled by thermal resistors, but only two of them are well-approximated as linear for a significant range of temperature.

**conduction** Heat conduction is the transfer of heat through an object's microscopic particle interaction.<sup>1</sup> It is characterized by a thermal

---

<sup>1</sup>We use the term “object” loosely, here, to mean a grouping of continuous matter in any phase.

resistance

$$R = \frac{L}{\rho A}, \quad (08.17)$$

where  $L$  is the length of the object *in the direction of heat transfer*,  $A$  is the transverse cross-sectional area, and  $\rho$  is the material's *thermal conductivity* (SI units  $\text{W}/\text{K}/\text{m}$ ).<sup>2</sup>

**convection** Heat convection is the transfer of heat via *fluid advection*: the bulk motion of a fluid. It is characterized by a thermal resistance

$$R = \frac{1}{hA}, \quad (08.18)$$

where  $h$  is the *convection heat transfer coefficient* (SI units  $\text{W}/\text{m}^2/\text{K}$ ) and  $A$  is the area of fluid-object contact (SI units  $\text{m}^2$ ). The convection heat transfer coefficient  $h$  is highly and nonlinearly dependent on the velocity of the fluid. Furthermore, the geometry of the objects and the fluid composition affect  $h$ .

**radiation** Radiative heat transfer is electromagnetic radiation emitted from one body and absorbed by another. For  $T_1$  the absolute temperature of a "hot" body,  $T_2$  the absolute temperature of a "cold" body,  $\epsilon$  the *effective emissivity/absorptivity*,<sup>3</sup> and  $A$  the area of the exposed surfaces, the heat transfer is characterized by

$$q = \epsilon \sigma A (T_1^4 - T_2^4), \quad (08.19)$$

where  $\sigma$  is the *Stefan-Boltzmann constant*

$$\sigma = 5.67 \cdot 10^{-8} \frac{\text{W}}{\text{m}^2 \text{K}^4}. \quad (08.20)$$

Clearly, this heat transfer is highly nonlinear. Linearization of this heat transfer is problematic because the temperature difference  $T$  between the bodies does not appear in the expression. For many system models, radiative heat transfer is assumed negligible. We must be cautious with this assumption, however, especially when high operating temperatures are anticipated.

<sup>2</sup>Thermal resistance can also be defined as an intensive property  $\rho^{-1}$ , the reciprocal of the thermal conductivity. Due to our lumped-parameter perspective, we choose the extensive definition.

<sup>3</sup>The parameter  $\epsilon$  is taken to be the combined "gray body" emissivity/absorptivity. Consult a heat transfer text for details.

thermal conductivity  
fluid advection

convection heat transfer coefficient

effective emissivity/absorptivity

Stefan-Boltzmann constant

### 08.02.3 Heat flow rate and temperature sources

Thermal sources include many physical processes—almost everything generates heat!

**ideal heat flow rate source**

An *ideal heat flow rate source* is an element that provides arbitrary heat flow rate  $Q_s$  to a system, independent of the temperature across it, which depends on the system.

**ideal temperature source**

An *ideal temperature source* is an element that provides arbitrary temperature  $T_s$  to a system, independent of the heat flow rate through it, which depends on the system.

### 08.02.4 Generalized element and variable types

**across-variable**  
**through-variable**

In keeping with the definitions of Chapter 01, temperature  $T$  is an *across-variable* and heat flow rate  $q$  is a *through-variable*.

**A-type**  
**D-type**

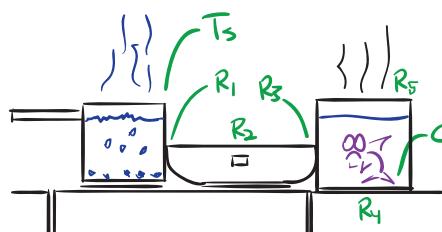
Consequently, the thermal capacitor is considered an *A-type* energy storage element. A thermal resistor is considered to be a *D-type* energy dissipative element, although it does not actually dissipate energy. It does, however, *resist* its flow and relates its across- and through-variables *algebraically*, both signature characteristics of D-type elements.

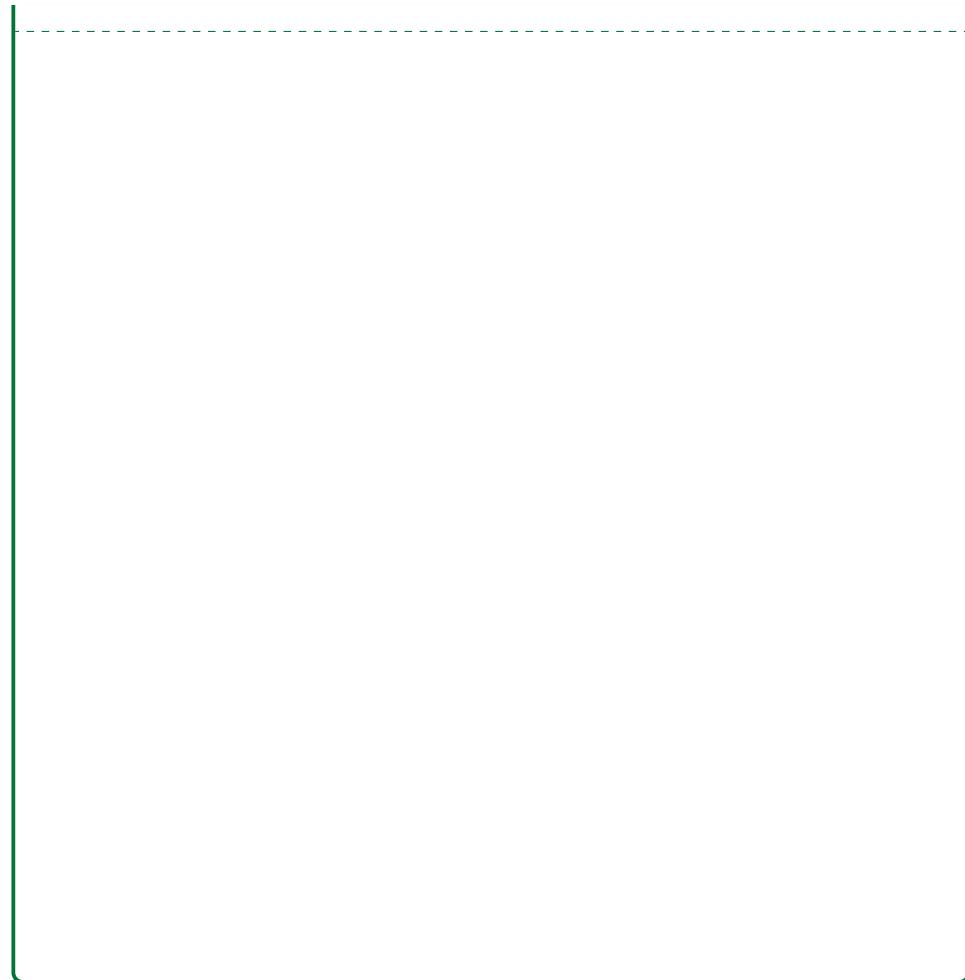
Temperature sources are, then, across-variable sources and heat flow rate sources are through-variable sources.

#### Example 08.02-1 thermal system graph

Careless Carlton left a large pot of water boiling on the stove. Worse, a cast-iron pan is bumped so that it is in solid contact with the pot *and* his glass fish tank, which was carelessly left next to the stove.

Draw a linear graph of the sad situation to determine what considerations determine if Careless Carlton's fish goes from winner to dinner.





## Lecture 08.03 Fluid transducers

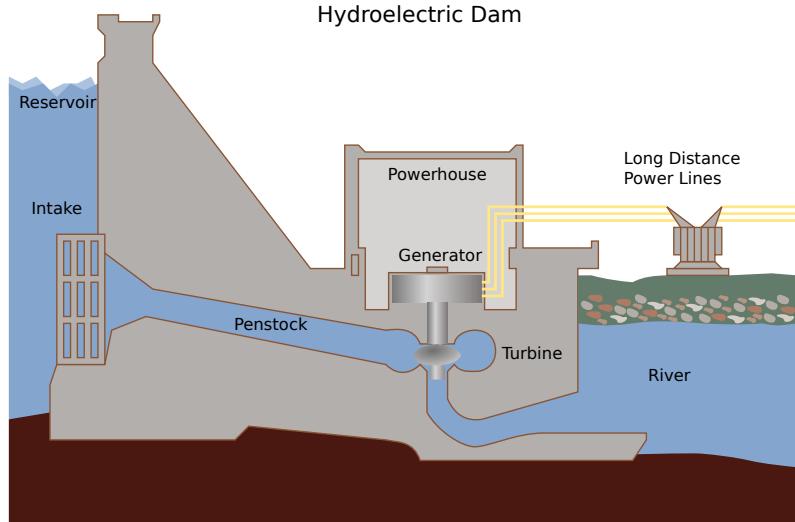
Although thermal systems often exchange energy with other energy domains, it is much more common to consider those systems that interact with thermal systems to be generating or sinking heat (often modeled as a *dependent source*) than to see a proper transducer.

Fluid systems, on the other hand, very naturally interact with mechanical systems. For instance, piston-cylinder mechanisms, propellers, turbines, and impellers (backward turbines) are just a few energy transducing elements.

These systems are often driven by motors (e.g. a pump's impeller) or drive generators (e.g. a dam's turbine). Therefore, it is common to require a fluid-electromechanical dynamic model.

### Example 08.03-1 microhydroelectric power generation

Dams, even small, "micro" dams, generate hydroelectric power by directing water through turbines, which rotate, creating mechanical power, and drive electric generators, generating electric power. For large-scale dams, the flowrate is regulated such that an AC generator produces a nice 60 Hz. However, a microhydroelectric generator typically cannot expect well-regulated flowrates, so sometimes they use a brushed DC generator (brush replacement being the primary drawback). Assuming a microhydroelectric dam can be set up in a manner similar to a large-scale dam, draw a linear graph model from the following schematic.



Schematic of a hydroelectric dam (*Authority and Tomia, 2018*).

## Lecture 08.04 State-space model of a hydroelectric dam

Consider the microhydroelectric dam of [Example 08.03-1](#). We derived the linear graph of [Figure 08.2](#). In this lecture, we will derive a state-space model for the system—specifically, a state equation.

### 08.04.1 Normal tree, order, and variables

**normal tree** Now, we define a *normal tree* by overlaying it on the system graph in [Figure 08.2](#). There are six independent energy storage elements, making it a sixth-order ( $n = 6$ ) system. We define the state vector to be

$$\mathbf{x} = [P_{C_1} \ P_{C_2} \ Q_{L_1} \ \Omega_J \ i_{L_2} \ v_{C_3}]^\top. \quad (08.21)$$

The input vector is defined as  $\mathbf{u} = [Q_s \ P_{s1} \ P_{s2}]^\top$ .

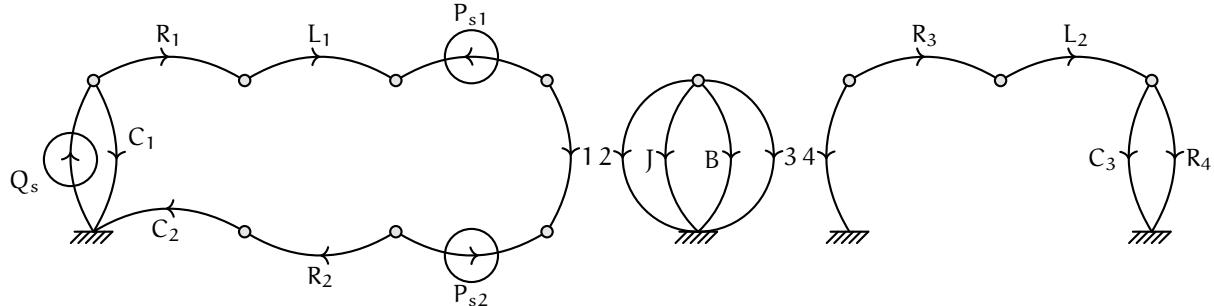
### 08.04.2 Elemental equations

Yet to be encountered is a turbine's transduction. A simple model is that the torque  $T_2$  is proportional to the flowrate  $Q_1$ , which are both through-  
**transformer** variables, making it a *transformer*, so

$$T_2 = -\alpha Q_1 \quad \text{and} \quad \Omega_2 = \frac{1}{\alpha} P_1, \quad (08.22)$$

**transformer ratio** where  $\alpha$  is the *transformer ratio*.

The other elemental equations have been previously encountered and are listed, below.



**Figure 08.2:** a linear graph for a microhydroelectric dam.

el.	elemental eq.	el.	elemental eq.	el.	elemental eq.
$C_1$	$\frac{dP_{C_1}}{dt} = \frac{1}{C_1} Q_{C_1}$	$C_3$	$\frac{dv_{C_3}}{dt} = \frac{1}{C_3} i_{C_3}$	$B$	$\Omega_B = \frac{1}{B} T_B$
$C_2$	$\frac{dP_{C_2}}{dt} = \frac{1}{C_2} Q_{C_2}$	$R_1$	$P_{R_1} = Q_{R_1} R_1$	$3$	$i_4 = \frac{-1}{k_m} T_3$
$L_1$	$\frac{dQ_{L_1}}{dt} = \frac{1}{L_1} P_{L_1}$	$R_2$	$P_{R_2} = Q_{R_2} R_2$	$4$	$v_4 = k_m \Omega_3$
$J$	$\frac{d\Omega_J}{dt} = \frac{1}{J} T_J$	$1$	$T_2 = -\alpha Q_1$	$R_3$	$v_{R_3} = i_{R_3} R_3$
$L_2$	$\frac{di_{L_2}}{dt} = \frac{1}{L_2} v_{L_2}$	$2$	$\Omega_2 = \frac{1}{\alpha} P_1$	$R_4$	$i_{R_4} = \frac{1}{R_4} v_{R_4}$

### 08.04.3 Continuity and compatibility equations

Continuity and compatibility equations can be found in the usual way—by drawing contours and temporarily creating loops by including links in the normal tree. We proceed by drawing a table of all elements and writing a continuity equation for each branch of the normal tree and a compatibility equation for each link.

el.	eq.	el.	eq.	el.	eq.
$C_1$	$Q_{C_1} = Q_s - Q_{L_1}$	$C_3$	$i_{C_3} = i_{L_2} - i_{R_4}$	$B$	$\Omega_B = \Omega_J$
$C_2$	$Q_{C_2} = Q_{L_1}$	$R_1$	$Q_{R_1} = Q_{L_1}$	$3$	$\Omega_3 = \Omega_J$
$L_1$	$P_{L_1} = -P_{R_1} + P_{C_1} - P_{C_2} +$ $-P_{R_2} + P_{s2} - P_1 + P_{s1}$	$R_2$	$Q_{R_2} = Q_{L_1}$	$4$	$i_4 = -i_{L_2}$
$J$	$T_J = -T_2 - T_B - T_3$	$1$	$Q_1 = Q_{L_1}$	$R_3$	$i_{R_3} = i_{L_2}$
$L_2$	$v_{L_2} = -v_{R_3} + v_4 - v_{C_3}$	$2$	$\Omega_2 = \Omega_J$	$R_4$	$v_{R_4} = v_{C_3}$

### 08.04.4 State equation

The system of equations composed of the elemental, continuity, and compatibility equations can be reduced to the state equation. There is a substantial amount of algebra required to eliminate those variables that are neither state nor input variables. Therefore, we use the Mathematica package *StateMint* (Devine and Picone, 2018). The resulting system model

is:

$$\frac{dx}{dt} = Ax + Bu,$$

$$A = \begin{bmatrix} 0 & 0 & -1/C_1 & 0 & 0 & 0 \\ 0 & 0 & 1/C_2 & 0 & 0 & 0 \\ 1/L_1 & -1/L_1 & -(R_1 + R_2)/L_1 & -\alpha/L_1 & 0 & 0 \\ 0 & 0 & \alpha/J & -B/J & -k_m/J & 0 \\ 0 & 0 & 0 & k_m/L_2 & -R_3/L_2 & -1/L_2 \\ 0 & 0 & 0 & 0 & 1/C_3 & -1/(R_4 C_3) \end{bmatrix},$$

$$B = \begin{bmatrix} 1/C_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1/L_1 & 1/L_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

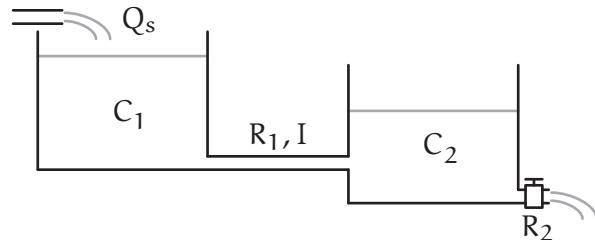
The rub is estimating all these parameters.

The Mathematica notebook use above can be found in the [source repository](#) for this text.

## Exercises

### Exercise 08.1

Draw a linear graph of the fluid system with schematic below.

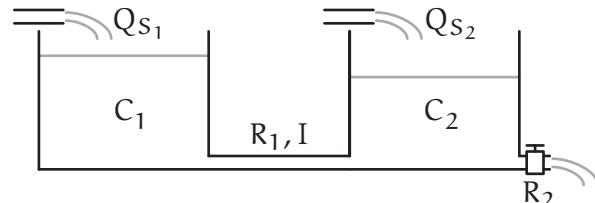


### Solution 08.1

a. blah

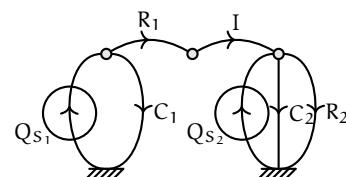
### Exercise 08.2

Draw a linear graph of the fluid system with schematic below.



### Solution 08.2

A linear graph representation is shown below.





## **Part IV**

# **Transfer function modeling**



## Transfer functions

In [Section 03.07](#), we briefly introduced the transfer function, a very important dynamic system representation to which we now turn our attention. We repeat the definition here.

Let a system have an input  $u$  and an output  $y$ . Let the Laplace transform of each be denoted  $U$  and  $Y$ , both functions of complex Laplace transform variable  $s$ . A *transfer function*  $H$  is defined as the ratio of the Laplace transform of the output over the input:

$$H(s) = \frac{Y(s)}{U(s)}. \quad (09.1)$$

[transfer function definition](#)

The transfer function is exceedingly useful in many types of analysis. One of its most powerful aspects is that it gives us access to thinking about systems as operating on an input  $u$  and yielding an output  $y$ .

As we learned in [Section 03.07](#), one can easily convert a state-space model to a (matrix) transfer function model with the formula

$$H(s) = C(sI - A)^{-1}B + D. \quad (09.2)$$

We also learned that a transfer function and an io ODE are related via the Laplace transform. The similarities are rather easy to spot, so io ODEs and transfer functions can be converted to each other via inspection.

poles  
zeros

## Lecture 09.01 Poles and zeros

Two important types of objects defined from a transfer function  $H$  can be used to characterize a system's behavior: *poles* and *zeros*.

### Definition 09.01.1: poles

Let a system have transfer function  $H$ . Its *poles* are values of  $s$  for which

$$|H(s)| \rightarrow \infty.$$

A transfer function written as a ratio has poles wherever the denominator is zero; that is,  $s$  for which<sup>1</sup>

### Definition 09.01.2: zeros

Let a system have transfer function  $H$ . Its *zeros* are values of  $s$  for which

$$|H(s)| \rightarrow 0.$$

A transfer function written as a ratio has zeros wherever the numerator is zero; that is,  $s$  for which<sup>2</sup>

Given a transfer function  $H$  with  $n$  poles  $p_i$  and  $v$  zeros  $z_j$ , we can write, for  $K \in \mathbb{R}$ ,

<sup>1</sup>It is common to use this as the definition of a pole, which allows us to talk of "pole-zero cancellation." Occasionally we will use this terminology.

<sup>2</sup>It is common to use this as the definition of a zero, which allows us to talk of "pole-zero cancellation." Occasionally we will use this terminology.

Poles and zeros can define a single-input, single-output (SISO) system's dynamic model, within a constant.

Recall that, even for multiple-input, multiple-output (MIMO) state-space models, the denominator of every transfer function is the corresponding system's *characteristic equation*—the roots of which dominate the system's response and are equal to its *eigenvalues*. It is not time to observe a crucial identity.

**Corollary 09.01.3: poles = eigenvalues = char. eq. roots**

A system's *poles* equal its *eigenvalues* equal its *characteristic equation roots*.

Therefore, everything we know about a system's eigenvalues and characteristic equation roots is true for a system's poles. This includes that they characterize a system's response (especially its free response) and stability.

### 09.01.1 Pole-zero plots and stability

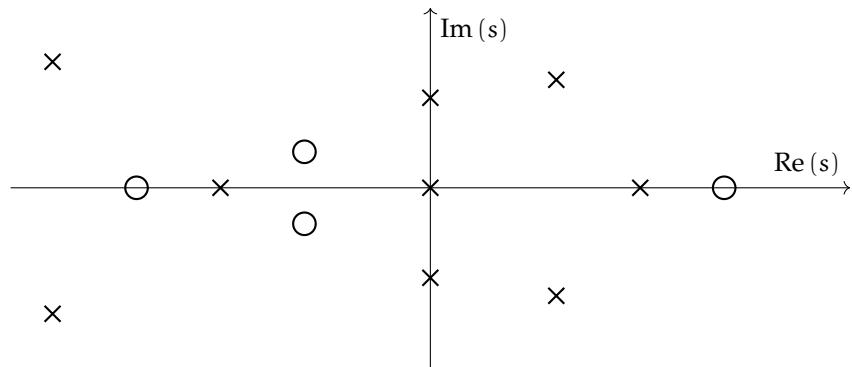
The complex-valued poles and zeros dominate system behavior via their values and value-relationships. Often, we construct a *pole-zero plot*—a plot in the complex plane of a system's poles and zeros—such as that of Figure 09.1.

**pole-zero plot**

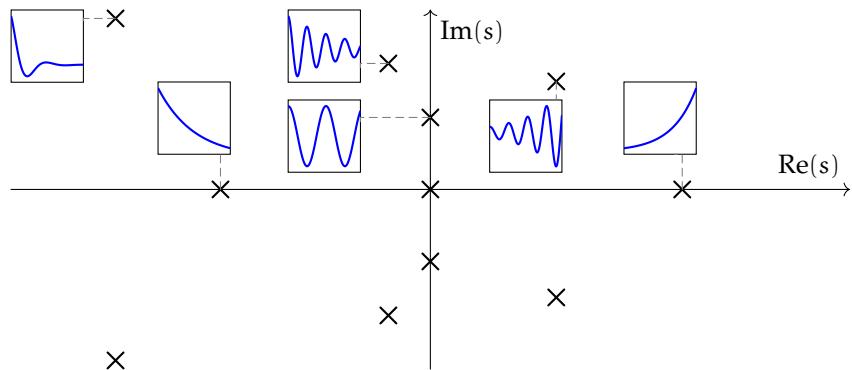
From our identification of poles with eigenvalues and roots of the characteristic equation, we can recognize that each pole contributes an exponential response that oscillates if it is complex. There are three stability contribution possibilities for each pole  $p_i$ :

- $\operatorname{Re}(p_i) < 0$ : a stable, decaying contribution;
- $\operatorname{Re}(p_i) = 0$ : a marginally stable, neither decaying nor growing contribution; and
- $\operatorname{Re}(p_i) > 0$ : an unstable, growing contribution.

This is explored graphically in Figure 09.2.



**Figure 09.1:** a pole-zero plot for a system with nine poles and four zeros. In this example, six of the poles are complex-conjugate pairs and three are real. Three are in the right half-plane, making the system unstable. One zero is in the right half-plane, making the system “minimum phase.”



**Figure 09.2:** free response contributions from poles at different locations. Complex poles contribute oscillating free responses, whereas real poles do not. Left half-plane poles contribute stable responses that decay. Right half-plane poles contribute unstable responses that grow. Imaginary-axis poles contribute marginal stability.

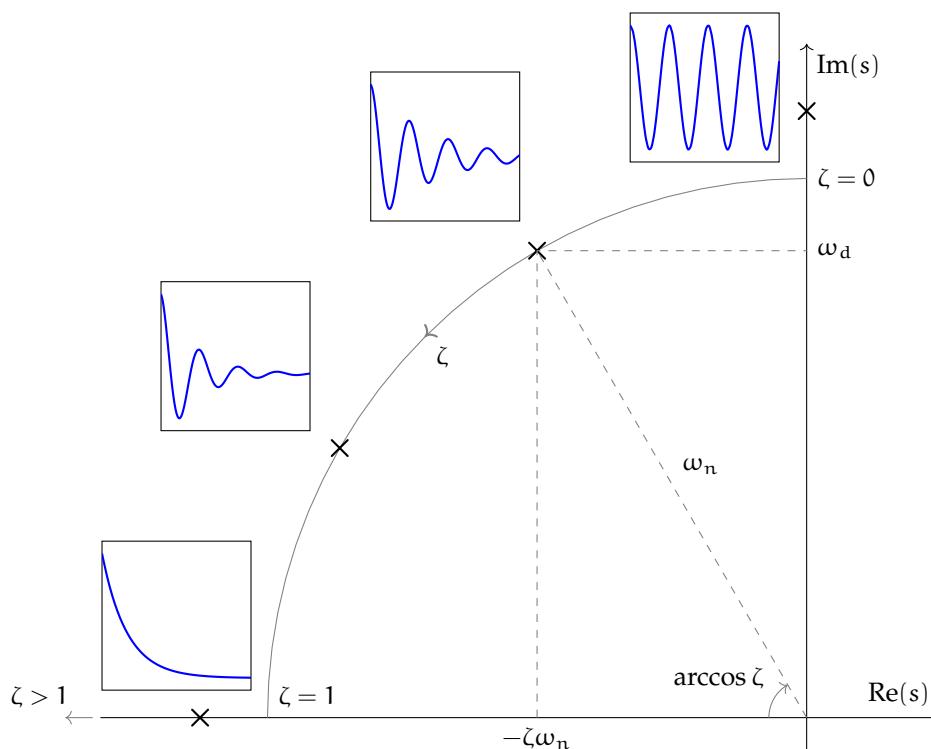
Of course, we must not forget that a *system's* stability is spoiled with a single unstable pole.

It can be shown that complex poles and zeros always arise as conjugate pairs. A consequence of this is that the pole-zero plot is always *symmetric about the real axis*.

real-axis symmetry

### 09.01.2 Second-order systems

Second-order response is characterized by a damping ratio  $\zeta$  and natural frequency  $\omega_n$ . These parameters have clear complex-plane “geometric” interpretations, as shown in [Figure 09.3](#). Pole locations are interpreted geometrically in accordance with their relation to rays of constant damping from the origin and circles of constant natural frequency, centered about the origin.



**Figure 09.3:** second-order free response contributions from poles at different locations, characterized by the damping ratio  $\zeta$  and natural frequency  $\omega_n$ . Constant damping occurs along rays from the origin. Constant natural frequency occurs along arcs of constant radius, centered at the origin.

---

## Impedance-based modeling

## Lecture 10.01 Input impedance and admittance

We now introduce a generalization of the familiar impedance and admittance of electrical circuit analysis, in which system behavior can be expressed algebraically instead of differentially. We begin with generalized input impedance.

Consider a system with a source, as shown in Figure 10.1. The source can be either an across- or a through-variable source. The ideal source specifies either  $V_{in}$  or  $F_{in}$ , and the other variable depends on the system.

**input impedance**      Let a source variables have Laplace transforms  $V_{in}(s)$  and  $F_{in}(s)$ . We define the system's *input impedance*  $Z$  and *input admittance*  $Y$  to be the Laplace-domain ratios  
**input admittance**

$$Z(s) = \frac{V_{in}(s)}{F_{in}(s)} \quad \text{and} \quad Y(s) = \frac{F_{in}(s)}{V_{in}(s)}. \quad (10.1)$$

Clearly,

### transfer functions

Both  $Z$  and  $Y$  can be considered *transfer functions*: for a through-variable source  $F_{in}$ , the impedance  $Z$  is the transfer function to across-variable  $V_{in}$ ; for an across-variable source  $V_{in}$ , the admittance  $Y$  is the transfer function to through-variable  $F_{in}$ . Often, however, we use the more common impedance  $Z$  to characterize systems with either type of source.

### system properties

Note that  $Z$  and  $Y$  are *system properties*, not properties of the source. An impedance or admittance can characterize a system of interconnected elements, or a system of a single element, as the next section explores.

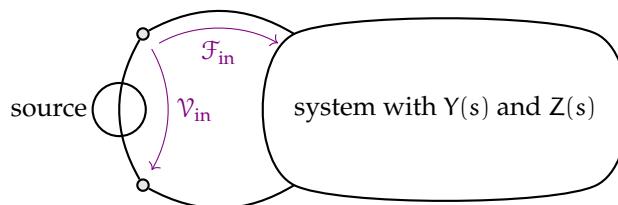


Figure 10.1:

### 10.01.1 Impedance of ideal passive elements

The impedance and admittance of a single, ideal, one-port element is defined from the Laplace transform of its elemental equation.

**Generalized capacitors** A *generalized capacitor* has elemental equation

$$\frac{d\mathcal{V}_C(t)}{dt} = \frac{1}{C}\mathcal{F}_C(t), \quad (10.2)$$

the Laplace transform of which is

$$s\mathcal{V}_C(s) = \frac{1}{C}\mathcal{F}_C(s), \quad (10.3)$$

which can be solved for impedance  $Z_C = \mathcal{V}_C/\mathcal{F}_C$  and admittance  $Y_C = \mathcal{F}_C/\mathcal{V}_C$ :

generalized capacitor

**Generalized inductors** A *generalized inductor* has elemental equation

$$\frac{d\mathcal{F}_L(t)}{dt} = \frac{1}{L}\mathcal{V}_L(t), \quad (10.4)$$

the Laplace transform of which is

$$s\mathcal{F}_L(s) = \frac{1}{L}\mathcal{V}_L(s), \quad (10.5)$$

which can be solved for impedance  $Z_L = \mathcal{V}_L/\mathcal{F}_L$  and admittance  $Y_L = \mathcal{F}_L/\mathcal{V}_L$ :

generalized inductor

**Generalized resistors** A *generalized resistor* has elemental equation

$$\mathcal{V}_R(t) = \mathcal{F}_R(t)R, \quad (10.6)$$

the Laplace transform of which is

$$\mathcal{V}_R(s) = \mathcal{F}_R(s)R, \quad (10.7)$$

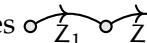
which can be solved for impedance  $Z_R = \mathcal{V}_R/\mathcal{F}_R$  and admittance  $Y_R = \mathcal{F}_R/\mathcal{V}_R$ :

generalized resistor

For a summary of the impedance of one-port elements, see Table C.1.

### 10.01.2 Impedance of interconnected elements

As with electrical circuits, impedances of linear graphs of interconnected elements can be combined in two primary ways: in parallel or in series.

**series** Elements sharing the same through-variable are said to be in *series* connection. N elements connected in series  have equivalent impedance Z and admittance Y:

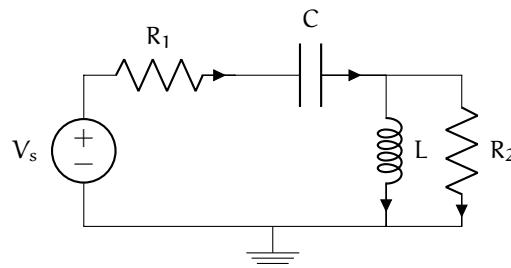
$$Z(s) = \sum_{i=1}^N Z_i(s) \quad \text{and} \quad Y(s) = 1 / \left( \sum_{i=1}^N 1/Y_i(s) \right) \quad (10.8)$$

**parallel** Conversely, elements sharing the same across-variable are said to be in *parallel* connection. N elements connected in parallel  have equivalent impedance Z and admittance Y:

$$Z(s) = 1 / \sum_{i=1}^N 1/Z_i(s) \quad \text{and} \quad Y(s) = \sum_{i=1}^N Y_i(s). \quad (10.9)$$

#### Example 10.01-1 input impedance of a simple circuit

For the circuit shown, find the input impedance.





## Lecture 10.02 Impedance with two-port elements

**transformers  
gyrators**

The two types of energy transducing elements, *transformers* and *gyrators*, “reflect” or “transmit” impedance through themselves, such that they are “felt” on the other side.

**transformer**

For a *transformer*, the elemental equations are

$$\mathcal{V}_2(t) = \mathcal{V}_1(t)/TF \quad \text{and} \quad \mathcal{F}_2(t) = -TF\mathcal{F}_1(t), \quad (10.10)$$

the Laplace transforms of which are

$$\mathcal{V}_2(s) = \mathcal{V}_1(s)/TF \quad \text{and} \quad \mathcal{F}_2(s) = -TF\mathcal{F}_1(s). \quad (10.11)$$

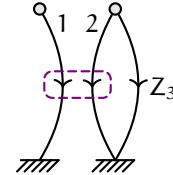
If, on the 2-side, the input impedance is  $Z_3$ , as in Figure 10.2, the equations of Equation 10.11 are subject to the continuity and compatibility equations

$$\mathcal{V}_2 = \mathcal{V}_3 \quad \text{and} \quad \mathcal{F}_2 = -\mathcal{F}_3. \quad (10.12)$$

Substituting these into Equation 10.11 and solving for  $\mathcal{V}_1$  and  $\mathcal{F}_1$ ,

$$\mathcal{V}_1 = TF\mathcal{V}_3 \quad \text{and} \quad \mathcal{F}_1 = \mathcal{F}_3/TF. \quad (10.13)$$

The elemental equation for element 3 is  $\mathcal{V}_3 = \mathcal{F}_3Z_3$ , which can be substituted into the through-variable equation to yield



**Figure 10.2:**

Working our way back from  $\mathcal{V}_3$  to  $\mathcal{V}_1$ , we apply the compatibility equation  $\mathcal{V}_2 = \mathcal{V}_3$  and the elemental equation  $\mathcal{V}_2 = \mathcal{V}_1/TF$ , as follows:

**effective input  
impedance**

Solving for the *effective input impedance*  $Z_1$ ,

$$Z_1 \equiv \frac{\mathcal{V}_1(s)}{\mathcal{F}_1(s)} \quad (10.14)$$

$$= TF^2 Z_3. \quad (10.15)$$

For a gyrator with gyrator modulus  $GY$ , in the configuration shown in Figure 10.3, a similar derivation yields the effective input impedance  $Z_1$ ,

$$Z_1 = GY^2 / Z_3. \quad (10.16)$$

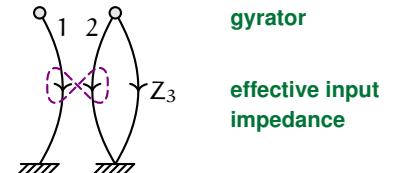
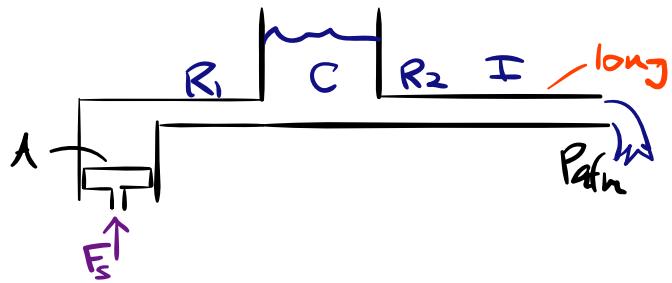


Figure 10.3:

Example 10.02-1 input impedance of fluid system with transducer

Draw a linear graph of the fluid system. What is the input impedance for an input force to the piston?



## Lecture 10.03 Transfer functions via impedance

Now the true power of impedance-based modeling is revealed: we can skip a time-domain model (e.g. state-space or io differential equation) and derive a transfer-function model, directly! Before we do, however, let's be sure to recall that a transfer-function model concerns itself with the *forced response* of a system, ignoring the free response. If we care to consider the free response, we can convert the transfer function model to an io differential equation and solve it.

There are two primary ways impedance-based modeling is used to derive transfer functions. The first and most general is described, here. The second is a shortcut most useful for relatively simple systems; it is described in [Section 10.05](#).

In what follows, it is important to recognize that, in the Laplace-domain, every elemental equation is just<sup>1</sup>

$$\mathcal{V} = \mathcal{F}\mathbf{Z}, \quad (10.17)$$

where the across-variable, through-variable, and impedance are all element-specific.

This algorithm is very similar to that for state-space models from linear graph models, presented in [Section 03.04](#). In the following, we consider a connected graph with  $B$  branches, of which  $S$  are sources (split between through-variable sources  $S_T$  and across  $S_A$ ). There are  $2B - S$  unknown across- and through-variables, so that's how many equations we need. We have  $B - S$  elemental equations and for the rest we will write continuity and compatibility equations.  $N$  is the number of nodes.

1. Derive  $2B - S$  independent Laplace-domain, algebraic equations from Laplace-domain elemental, continuity, and compatibility equations.

**normal tree**

- a) Draw a *normal tree*.

**elemental equation**

- b) Write a Laplace-domain *elemental equation* for each passive element.<sup>2</sup>

**continuity equation**

- c) Write a *continuity equation* for each passive branch by drawing a contour intersecting that and no other branch.<sup>3</sup>

**compatibility equation**

- d) Write a *compatibility equation* for each passive link by temporarily

---

<sup>1</sup>In electronics, this is sometimes called “generalized Ohm’s law.”

<sup>2</sup>There will be  $B - S$  elemental equations.

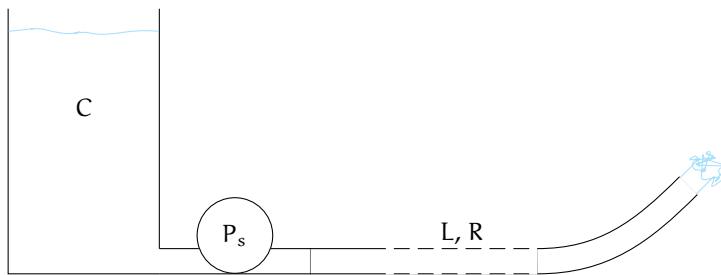
<sup>3</sup>There will be  $N - 1 - S_A$  independent continuity equations.

“including” it in the tree and finding the compatibility equation for the resulting loop.<sup>4</sup>

2. Solve the *algebraic* system of  $2B$  equations and  $2B$  unknowns for outputs in terms of inputs, only. Sometimes, solving for *all* unknowns via the usual methods is easier than trying to cherry-pick the desired outputs.
3. The solution for each output  $Y_i$  depends on zero or more inputs  $U_j$ . To solve for the transfer function  $Y_i/U_j$ , set  $U_k = 0$  for all  $k \neq j$ , then divide both sides of the equation by  $U_j$ .

#### Example 10.03-1 fire hose

For the schematic of a fire hose connected to a fire truck’s reservoir  $C$  via pump input  $P_s$ , use impedance methods to find the transfer function from  $P_s$  to the velocity of the spray. Assume the nozzle’s cross-sectional area is  $A$ .



<sup>4</sup>There will be  $B - N + 1 - S_T$  independent compatibility equations.



## Lecture 10.04 Norton and Thévenin theorems

The following remarkable theorem has been proven.<sup>5</sup>

### Theorem 10.04.1: generalized Thévenin's theorem

Given a linear network of across-variable sources, through-variable sources, and impedances, the behavior at the network's output nodes can be reproduced exactly by a single *across-variable source*  $\mathcal{V}_e$  in series with an impedance  $Z_e$ .

The equivalent linear network has two quantities to determine:  $\mathcal{V}_e$  and  $Z_e$ .

#### 10.04.0.1 Determining $Z_e$

The *equivalent impedance*  $Z_e$  of a network is the impedance between the output nodes with all inputs set to zero. Setting an across-variable source to zero means the across-variable on both its terminals are equal, which is equivalent to treating them as the same node. Setting a through-variable source to zero means the through-variable through it is zero, which is equivalent to treating its nodes as disconnected.

equivalent  
impedance  $Z_e$

#### 10.04.0.2 Determining $\mathcal{V}_e$

The *equivalent across-variable source*  $\mathcal{V}_e$  is the across-variable at the output nodes of the network when they are left open (disconnected from a load). Determining this value typically requires some analysis with the elemental, continuity, and compatibility equations (preferably via impedance methods).

equivalent  
across-variable  
source  $\mathcal{V}_e$

#### 10.04.1 Norton's theorem

Similarly, the following remarkable theorem has been proven.

---

<sup>5</sup>This lecture is intentionally strongly paralleled in our *Electronics* lecture on Norton's and Thévenin's theorems.

**Theorem 10.04.2: generalized Norton's theorem**

Given a linear network of across-variable sources, through-variable sources, and impedances, the behavior at the network's output nodes can be reproduced exactly by a single *through-variable source*  $\mathcal{F}_e$  in parallel with an impedance  $Z_e$ .

The equivalent network has two quantities to determine:  $\mathcal{F}_e$  and  $Z_e$ . The equivalent impedance  $Z_e$  is identical to that of Thévenin's theorem, which leaves the equivalent through-variable source  $\mathcal{F}_e$  to be determined.

**10.04.1.1 Determining  $\mathcal{F}_e$** 

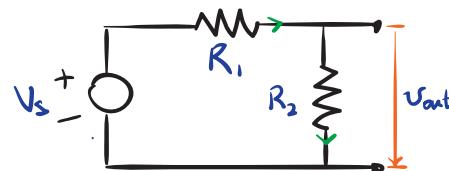
**equivalent through-variable source  $\mathcal{F}_e$**  The *equivalent through-variable source*  $\mathcal{F}_e$  is the through-variable through the output terminals of the network when they are shorted (collapsed to a single node). Determining this value typically requires some analysis with elemental, continuity, and compatibility equations (preferably via impedance methods).

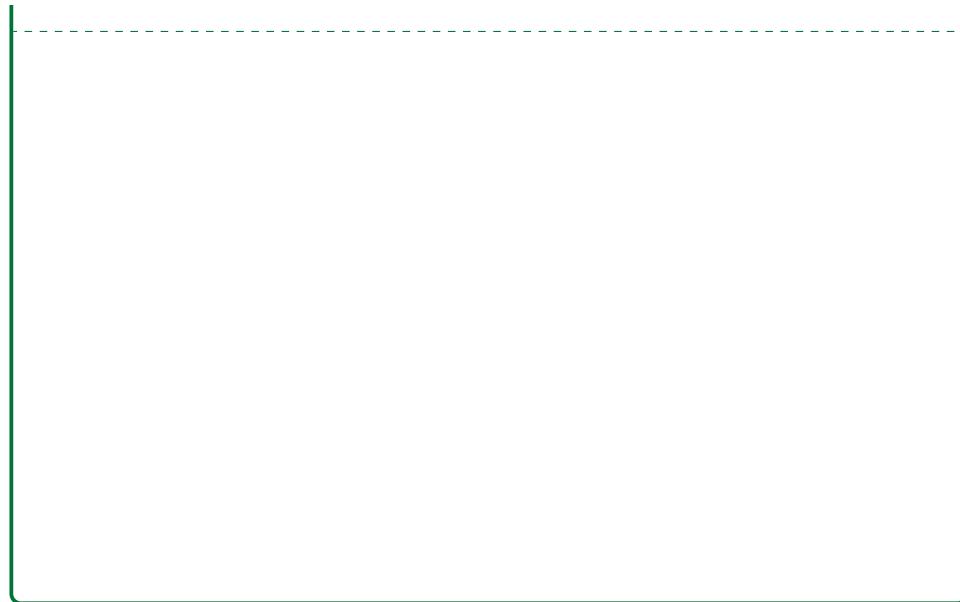
**10.04.2 Converting between Thévenin and Norton equivalents**

There is an equivalence between the two equivalent network models that allows one to convert from one to another with ease. The equivalent impedance  $Z_e$  is identical in each and provides the following equation for converting between the two representations:

**Equation 10.18 converting between Thévenin and Norton equivalents****Example 10.04-1 Thévenin and Norton equivalents**

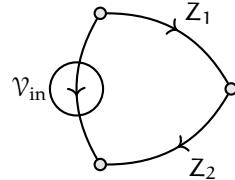
For the circuit shown, find a Thévenin and a Norton equivalent.





**across-variable  
divider****Lecture 10.05 The divider method**

In *Electronics*, we developed the useful voltage divider formula for quickly analyzing how voltage divides among series electronic impedances. This can be considered a special case of a more general *across-variable divider* equation for any elements described by an impedance. After developing the across-variable divider, we also introduce the through-variable divider, which divides an input through-variable among parallel elements.



**Figure 10.4:** the two-element across-variable divider.

### 10.05.1 Across-variable dividers

First, we develop the solution for the two-element across-variable divider shown in [Figure 10.4](#). We choose the across-variable across  $Z_2$  as the output. The analysis follows the impedance method of [Lecture 10.03](#), solving for  $V_2$ .

1. Derive four independent equations.
  - a) The normal tree is chosen to consist of  $V_{in}$  and  $Z_2$ .
  - b) The elemental equations are
  - c) The continuity equation is
  - d) The compatibility equation is
2. Solve for the output  $V_2$ . From the elemental equation for  $Z_2$ ,

A similar analysis can be conducted for  $n$  impedance elements.

**Equation 10.19 general across-variable divider**

### 10.05.2 Through-variable dividers

By a similar process, we can analyze a network that divides a through-variable into  $n$  *parallel* impedance elements.

**Equation 10.20 general through-variable divider**

### 10.05.3 Transfer functions using dividers

An excellent shortcut to deriving a transfer function is to use the across- and through-variable divider rules instead of solving the system of algebraic equations, as in [Section 10.03](#). An algorithm for this process is as follows.

1. Identify the element associated with an output variable  $Y_i$ . Call it the *output element*.
2. Identify the source associated with an input variable  $U_j$ . Set all other sources to zero.
3. Transform the network to be an across- or through-variable divider that includes the “bare” (uncombined) output element’s output variable.<sup>6</sup>
  - a) If necessary, form equivalent impedances of portions of the network, being sure to leave the output element’s output variable alone.
  - b) If necessary, transform the source *à la* Norton or Thévenin.

<sup>6</sup>In other words, if the across-variable of the output element is the output, do not combine it in series; if the through-variable is the output, do not combine it in parallel.

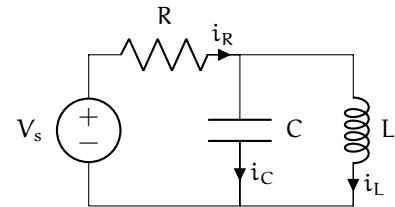
4. Apply the across- or through-variable divider equation.
5. If necessary, use the elemental equation of the output element to trade output across- and through-variables.
6. If necessary, use the source transformation equation of the input to trade input across- and through-variables.
7. Divide both sides by the input variable.

It turns out that, despite its many “if necessary” clauses, very often this “shortcut” is easier than the method of Lecture 10.03 for low-order systems if only a few transfer functions are of interest.

#### Example 10.05-1 a circuit transfer function using a divider

Given the circuit shown with voltage source  $V_s$  and output  $v_L$ ,

- a. what is the transfer function  $\frac{V_L}{V_s}$ ?
- b. Without transforming the source, find the transfer function  $\frac{I_L}{V_s}$ .
- c. Transforming the source, find  $\frac{I_L}{V_s}$ .

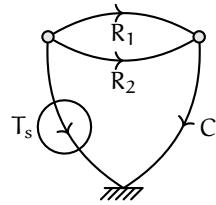




## Exercises

### Exercise 10.1

Use the linear graph below of a thermal system to (a) derive the transfer function  $T_{R_2}(s)/T_s(s)$ , where  $T_s$  is the input temperature and  $T_{R_2}$  is the temperature across the thermal resistor  $R_2$ . Use impedance methods. And (b) derive the input impedance the input  $T_s$  drives.

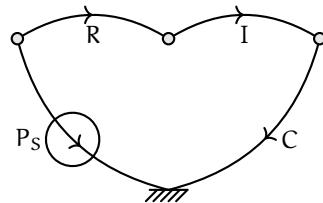


### Solution 10.1

a. blah

### Exercise 10.2

Use the linear graph below of a fluid system to (a) derive the transfer function  $P_C(s)/P_S(s)$ , where  $P_S$  is the input pressure and  $P_C$  is the pressure across the fluid capacitance  $C$ . Use impedance methods and a divider rule is highly recommended. (Simplify the transfer function.) And (b) derive the input impedance the input  $P_S$  drives. (Don't simplify the expression.)



### Solution 10.2

a. This is a straightforward across-variable divider:

$$\begin{aligned} \frac{P_C(s)}{P_S(s)} &= \frac{Z_C}{Z_I + Z_R + Z_C} \\ &= \frac{1}{ICs^2 + RCs + 1}. \end{aligned}$$

- b. The input impedance for the system consisting of three series elements is

$$Z_R + Z_I + Z_C = R + I_s + \frac{1}{C_s}.$$



# **Part V**

# **Frequency domain analysis**



---

Frequency domain, steady response

## Lecture 11.01 Fourier series

**frequency domain**  
**Fourier analysis**  
**frequency spectrum**

Fourier series are mathematical series that can represent a periodic signal as a sum of sinusoids at different amplitudes and frequencies. They are useful for solving for the response of a system to periodic inputs. However, they are probably most important *conceptually*: they are our gateway to thinking of signals in the *frequency domain*—that is, as functions of *frequency* (not time). To represent a function as a Fourier series is to *analyze* it as a sum of sinusoids at different frequencies<sup>1</sup>  $\omega_n$  and amplitudes  $a_n$ . Its *frequency spectrum* is the functional representation of amplitudes  $a_n$  versus frequency  $\omega_n$ .

Let's begin with the definition.

### Definition 11.01.1: Fourier series: trigonometric form

The *Fourier analysis* of a periodic function  $y(t)$  is, for  $n \in \mathbb{N}_0$ , period  $T$ , and angular frequency  $\omega_n = 2\pi n/T$ ,

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} y(t) \cos(\omega_n t) dt \quad (11.1)$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} y(t) \sin(\omega_n t) dt. \quad (11.2)$$

The *Fourier synthesis* of a periodic function  $y(t)$  with analysis components  $a_n$  and  $b_n$  corresponding to  $\omega_n$  is

$$y(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(\omega_n t) + b_n \sin(\omega_n t). \quad (11.3)$$

Let's consider the complex form of the Fourier series, which is analogous to Definition 11.01.1.

---

<sup>1</sup>It's important to note that the symbol  $\omega_n$ , in this context, is not the natural frequency, but a frequency indexed by integer  $n$ .

**Definition 11.01.2: Fourier series: complex form**

The *Fourier analysis* of a periodic function  $y(t)$  is, for  $n \in \mathbb{N}_0$ , period  $T$ , and angular frequency  $\omega_n = 2\pi n/T$ ,

$$c_{\pm n} = \frac{1}{T} \int_{-T/2}^{T/2} y(t) e^{-j\omega_n t} dt. \quad (11.4)$$

The *Fourier synthesis* of a periodic function  $y(t)$  with analysis components  $c_n$  corresponding to  $\omega_n$  is

$$y(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\omega_n t}. \quad (11.5)$$

We call the integer  $n$  a *harmonic* and the frequency associated with it,

$$\omega_n = 2\pi n/T, \quad (11.6)$$

the *harmonic frequency*. There is a special name for the first harmonic ( $n = 1$ ): the *fundamental frequency*. It is called this because all other frequency components are integer multiples of it.

It is also possible to convert between the two representations above.

**Definition 11.01.3: Fourier series: converting between forms**

The complex Fourier analysis of a periodic function  $y(t)$  is, for  $n \in \mathbb{N}_0$  and  $a_n$  and  $b_n$  as defined above,

$$c_{\pm n} = \frac{1}{2} (a_{|n|} \mp j b_{|n|}) \quad (11.7)$$

The sinusoidal Fourier analysis of a periodic function  $y(t)$  is, for  $n \in \mathbb{N}_0$  and  $c_n$  as defined above,

$$a_n = c_n + c_{-n} \text{ and} \quad (11.8)$$

$$b_n = j(c_n - c_{-n}). \quad (11.9)$$

The *harmonic amplitude* is

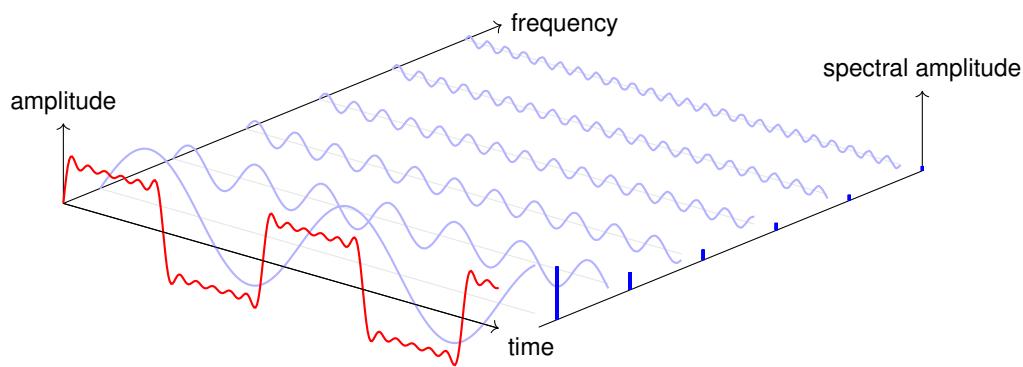
**harmonic  
amplitude**

$$C_n = \sqrt{a_n^2 + b_n^2} \quad (11.10)$$

$$= 2\sqrt{c_n c_{-n}}. \quad (11.11)$$

**line spectrum** A *line spectrum* is a graph of the harmonic amplitudes as a function of the harmonic frequencies.

The following illustration demonstrates how sinusoidal components sum to represent a square wave. A line spectrum is also shown.



Let us compute the associated spectral components in the following example.

#### Example 11.01-1 Fourier series analysis: line spectrum

Compute the first five harmonic amplitudes that represent the line spectrum for a square wave in the figure above.



## Lecture 11.02 Fourier transform

Let's consider a periodic function  $f$  with period  $T$ . Each period, the function has a triangular pulse of width  $\delta$  and height  $\delta/2$ .

First, we plot the function  $f$  in the time domain.

For  $\delta = 2$  and  $T \in [5, 15, 25]$ , the left-hand column of [Figure 11.1](#) shows two triangle pulses for each period  $T$ .

Consider the following argument. Just as a Fourier series is a frequency domain representation of a periodic signal, a Fourier transform is a frequency domain representation of an *aperiodic* signal (we will rigorously define it in a moment). The Fourier series components will have an analog, then, in the Fourier transform. Recall that they can be computed by integrating over a period of the signal. If we increase that period infinitely, the function is effectively aperiodic. The result (within a scaling factor) will be the Fourier transform analog of the Fourier series components.

Let us approach this understanding by actually computing the Fourier series components for increasing period  $T$ . As usual, we compute the Fourier series cosine and sine components  $a_n$  and  $b_n$  for component  $n$ .

Furthermore, we compute the *harmonic amplitude*

$$C_n = \sqrt{a_n^2 + b_n^2} \quad (11.12)$$

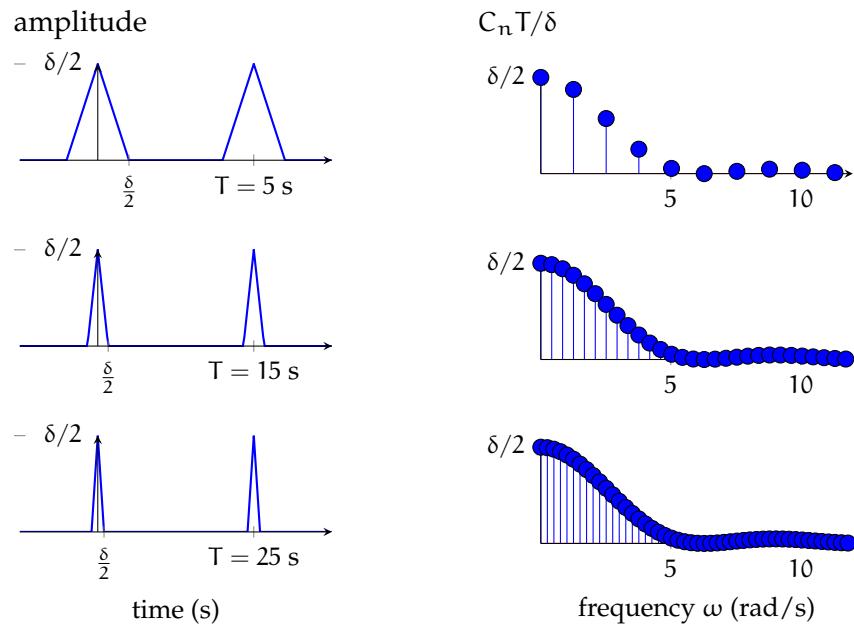
which we have also scale by a factor  $T/\delta$  in order to plot it with a convenient scale. The line spectra are shown in the right-hand column of [Figure 11.1](#). Note that with our chosen scaling, as  $T$  increases, the line spectra reveal a distinct waveform.

Let  $F$  be the continuous function of angular frequency  $\omega$

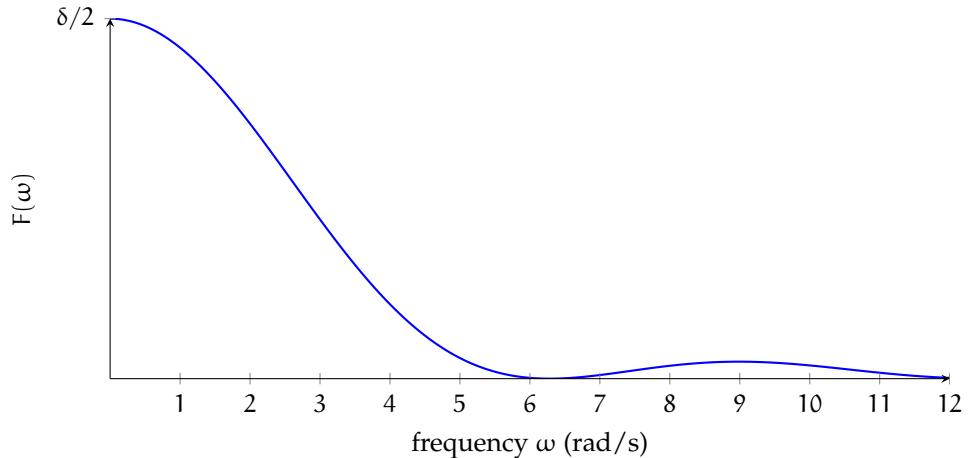
$$F(\omega) = \frac{\delta}{2} \cdot \frac{\sin^2(\omega\delta/4)}{(\omega\delta/4)^2}. \quad (11.13)$$

Consider the plot in [Figure 11.2](#) of  $F$ . It's obviously the function emerging in [Figure 11.1](#) from increasing the period of our pulse train.

Now we are ready to define the Fourier transform and its inverse.



**Figure 11.1:** triangle pulse trains (left column) with longer periods, descending, and their corresponding line spectra (right column), scaled for convenient comparison.



**Figure 11.2:**  $F(\omega)$ , our mysterious Fourier series amplitude analog.

**Definition 11.02.1: Fourier transforms: trigonometric form**

Fourier transform (analysis):

$$A(\omega) = \int_{-\infty}^{\infty} y(t) \cos(\omega t) dt \quad (11.14)$$

$$B(\omega) = \int_{-\infty}^{\infty} y(t) \sin(\omega t) dt. \quad (11.15)$$

Inverse Fourier transform (synthesis):

$$y(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} A(\omega) \cos(\omega t) d\omega + \frac{1}{2\pi} \int_{-\infty}^{\infty} B(\omega) \sin(\omega t) d\omega. \quad (11.16)$$

**Definition 11.02.2: Fourier transforms: complex form**

Fourier transform  $\mathcal{F}$  (analysis):

$$\mathcal{F}(y(t)) = Y(\omega) = \int_{-\infty}^{\infty} y(t) e^{-j\omega t} dt. \quad (11.17)$$

Inverse Fourier transform  $\mathcal{F}^{-1}$  (synthesis):

$$\mathcal{F}^{-1}(Y(\omega)) = y(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Y(\omega) e^{j\omega t} d\omega. \quad (11.18)$$

So now we have defined the Fourier transform. There are many applications, including solving differential equations and *frequency domain* representations—called *spectra*—of *time domain* functions.

There is a striking similarity between the Fourier transform and the Laplace transform, with which you are already acquainted. In fact, the Fourier transform is a special case of a Laplace transform with Laplace transform variable  $s = j\omega$  instead of having some real component. Both transforms convert differential equations to algebraic equations, which can be solved and inversely transformed to find time-domain solutions. The Laplace transform is especially important to use when an input function to a differential equation is not absolutely integrable and the Fourier transform is undefined (for example a step or ramp function). However, the Laplace transform is also preferred for *initial value problems* due to its convenient way of handling them. The two transforms are equally useful for solving steady state problems. Although the Laplace transform has

many advantages, for spectral considerations, the Fourier transform is the only game in town.

A table of Fourier transforms and their properties can be found on the course website in Table C.3.

## Lecture 11.03 Frequency and impulse response

This lecture proceeds in three parts. First, the Fourier transform is used to derive the *frequency response function*. Second, this is used to derive the *frequency response*. Third, the frequency response for an impulse input is explored.

### 11.03.1 Frequency response functions

Consider a dynamic system described by the *input-output differential equation*—with variable  $y$  representing the *output*, dependent variable time  $t$ , variable  $u$  representing the *input*, constant coefficients  $a_i, b_j$ , order  $n$ , and  $m \leq n$  for  $n \in \mathbb{N}_0$ —as:

$$\begin{aligned} \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = \\ b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + \cdots + b_1 \frac{du}{dt} + b_0 u. \end{aligned} \quad (11.19)$$

#### Fourier transform

The *Fourier transform*  $\mathcal{F}$  of [Equation 11.19](#) yields something interesting (assuming zero initial conditions):

$$\begin{aligned} \mathcal{F}\left(\frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y\right) = \\ \mathcal{F}\left(b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + \cdots + b_1 \frac{du}{dt} + b_0 u\right) \Rightarrow \\ \mathcal{F}\left(\frac{d^n y}{dt^n}\right) + a_{n-1} \mathcal{F}\left(\frac{d^{n-1} y}{dt^{n-1}}\right) + \cdots + a_1 \mathcal{F}\left(\frac{dy}{dt}\right) + a_0 \mathcal{F}(y) = \\ b_m \mathcal{F}\left(\frac{d^m u}{dt^m}\right) + b_{m-1} \mathcal{F}\left(\frac{d^{m-1} u}{dt^{m-1}}\right) + \cdots + b_1 \mathcal{F}\left(\frac{du}{dt}\right) + b_0 \mathcal{F}(u) \Rightarrow \\ (j\omega)^n Y + a_{n-1} (j\omega)^{n-1} Y + \cdots + a_1 (j\omega) Y + a_0 Y = \\ b_m (j\omega)^m U + b_{m-1} (j\omega)^{m-1} U + \cdots + b_1 (j\omega) U + b_0 U. \end{aligned}$$

Solving for  $Y$ ,

#### forced response

The inverse Fourier transform  $\mathcal{F}^{-1}$  of  $Y$  is the *forced response*. However, this is not our primary concern; rather, we are interested to solve for the

frequency response function  $H(j\omega)$  as the ratio of the output transform  $Y$  to the input transform  $U$ , i.e.

$$H(j\omega) \equiv \frac{Y(\omega)}{U(\omega)} \quad (11.20a)$$

$$= \frac{b_m(j\omega)^m + b_{m-1}(j\omega)^{m-1} + \dots + b_1(j\omega) + b_0}{(j\omega)^n + a_{n-1}(j\omega)^{n-1} + \dots + a_1(j\omega) + a_0}. \quad (11.20b)$$

Note that a frequency response function can be converted to a transfer function via the substitution  $j\omega \mapsto s$  and, conversely, a transfer function can be converted to a frequency response function via the substitution  $s \mapsto j\omega$ , as in

It is often easiest to first derive a transfer function—using any of the methods described, previously—then convert this to a frequency response function.

### 11.03.2 Frequency response

From above, we can solve for the output response  $y$  from the frequency response function by taking the inverse Fourier transform:

$$y(t) = \mathcal{F}^{-1}Y(\omega). \quad (11.21)$$

From the definition of the frequency response function (11.20a),

$$y(t) = \mathcal{F}^{-1}(H(j\omega)U(\omega)). \quad (11.22)$$

The *convolution theorem* states that, for two functions of time  $h$  and  $u$ ,

**convolution theorem**

$$\mathcal{F}(h * u) = \mathcal{F}(h)\mathcal{F}(u) \quad (11.23a)$$

$$= H(j\omega)U(\omega), \quad (11.23b)$$

where the *convolution operator*  $*$  is defined by

**convolution operator \***

$$(h * u)(t) \equiv \int_{-\infty}^{\infty} h(\tau)u(t - \tau) d\tau. \quad (11.24)$$

Therefore,

**frequency response**

This is the *frequency response* in terms of all time-domain functions.

### 11.03.3 Impulse response

The frequency response result includes an interesting object:  $h(t)$ . What is the physical significance of  $h$ , other than its definition, as the inverse Fourier transform of  $H(j\omega)$ ?

Consider the singularity input  $u(t) = \delta(t)$ , an impulse. The frequency response is

$$y(t) = \int_{-\infty}^{\infty} h(\tau)\delta(t - \tau) d\tau. \quad (11.25)$$

**sifting property** The so-called *sifting property* of  $\delta$  yields

$$y(t) = h(t). \quad (11.26)$$

**impulse response** That is,  $h$  is the *impulse response*.

A very interesting aspect of this result is that

$$H(j\omega) = \mathcal{F}(h). \quad (11.27)$$

That is, the Fourier transform of the impulse response is the frequency response function. A way to estimate, via measurement, the frequency response function (and transfer function) of a system is to input an impulse, measure and fit the response, then Fourier transform it. Of course, putting in an actual impulse and fitting the response, perfectly are impossible; however, estimates using approximations remain useful.

It is worth noting that frequency response/transfer function estimation is a significant topic of study, and many techniques exist. Another method is described in [Section 11.04](#).

## Lecture 11.04 Sinusoidal input, frequency response

In this lecture, we explore the relationship—which turns out to be pretty chummy—between a system’s frequency response function  $H(j\omega)$  and its sinusoidal forced response.

Let’s build from the frequency response function  $H(j\omega)$  definition:

$$y(t) = \mathcal{F}^{-1}Y(\omega) \quad (11.28a)$$

$$= \mathcal{F}^{-1}(H(j\omega)U(\omega)). \quad (11.28b)$$

We take the input to be sinusoidal, with amplitude  $A \in \mathbb{R}$ , angular frequency  $\omega_0$ , and phase  $\psi$ :

$$u(t) = A \cos(\omega_0 t + \psi). \quad (11.29)$$

The Fourier transform of the input,  $U(\omega)$ , can be constructed via transform identities from [Table C.3](#). This takes a little finagling. Let

$$p(t) = Aq(t), \quad (11.30a)$$

$$q(t) = r(t - t_0), \text{ and} \quad (11.30b)$$

$$r(t) = \cos \omega_0 t, \text{ where} \quad (11.30c)$$

$$t_0 = -\psi/\omega_0. \quad (11.30d)$$

The corresponding Fourier transforms, from [Table C.3](#), are

$$P(\omega) = A Q(\omega), \quad (11.31a)$$

$$Q(\omega) = e^{-j\omega t_0} R(\omega), \text{ and} \quad (11.31b)$$

$$R(\omega) = \pi \delta(\omega - \omega_0) + \pi \delta(\omega + \omega_0). \quad (11.31c)$$

Putting these together,

And now we are ready to substitute into [Equation 11.28b](#); also applying the “linearity” property of the Fourier transform:

$$y(t) = A \pi (e^{j\psi} \mathcal{F}^{-1}(H(j\omega) \delta(\omega - \omega_0)) + e^{-j\psi} \mathcal{F}^{-1}(H(j\omega) \delta(\omega + \omega_0))). \quad (11.32)$$

The definition of the inverse Fourier transform gives

$$y(t) = \frac{A}{2} \left( e^{j\psi} \int_{-\infty}^{\infty} e^{j\omega t} H(j\omega) \delta(\omega - \omega_0) d\omega + e^{-j\psi} \int_{-\infty}^{\infty} e^{j\omega t} H(j\omega) \delta(\omega + \omega_0) d\omega \right). \quad (11.33)$$

Recognizing that  $\delta$  is an even distribution ( $\delta(t) = \delta(-t)$ ) and applying the sifting property of  $\delta$  allows us to evaluate each integral:

$$y(t) = \frac{A}{2} (e^{j\psi} e^{j\omega_0 t} H(j\omega_0) + e^{-j\psi} e^{-j\omega_0 t} H(-j\omega_0)). \quad (11.34)$$

Writing  $H$  in polar form,

The Fourier transform is conjugate symmetric—that is,  $F(-\omega) = F^*(\omega)$ —which allows us to further simplify:

$$y(t) = \frac{A|H(j\omega_0)|}{2} (e^{j(\omega_0 t + \psi)} e^{j\angle H(j\omega_0)} + e^{-j(\omega_0 t + \psi)} e^{-j\angle H(j\omega_0)}) \quad (11.36a)$$

$$= A|H(j\omega_0)| \frac{e^{j(\omega_0 t + \psi + \angle H(j\omega_0))} + e^{-j(\omega_0 t + \psi + \angle H(j\omega_0))}}{2}. \quad (11.36b)$$

Finally, Euler's formula yields something that deserves a box.

**Equation 11.37 sinusoidal response in terms of  $H(j\omega)$**

This is a remarkable result. For an input sinusoid, a linear system has a forced response that

- is also a sinusoid,
- is at the same frequency as the input,
- differs only in amplitude and phase,
- differs in amplitude by a factor of  $|H(j\omega)|$ , and
- differs in phase by a shift of  $\angle H(j\omega)$ .

Now we see one of the key facets of the frequency response function: it governs how a sinusoid transforms through a system. And just think how powerful it will be once we combine it with the powerful principle of superposition and the mighty Fourier series representation of a function—as a “superposition” of sinusoids!

## Lecture 11.05 Bode plots

This lecture also appears in *Control: an introduction*.

**Bode plots** Given Equation 11.37, we are often most-interested in the magnitude  $|H(j\omega)|$  and phase  $\angle H(j\omega)$  of the frequency response function. Each of these is a function of angular frequency  $\omega$ , so plotting  $|H(j\omega)|$  vs.  $\omega$  and  $\angle H(j\omega)$  vs.  $\omega$  is quite useful. *Bode plots* are such plots with axes scaled in a specific manner.

A Bode plot is a useful graphical representation of the frequency response of a system. Let  $|U(\omega)|$  and  $|Y(\omega)|$  be the complex amplitudes of the input and the output, respectively. Recall that the magnitude of the frequency response function  $|H(j\omega)|$  can be expressed as

**Equation 11.38 frequency response function as an amplitude ratio**

**decibels** This is a ratio of amplitudes, and so it is akin to amplitude ratios commonly expressed in *decibels* (dB). However, the magnitude ratio of Equation 11.38 is *not dimensionless*, and therefore cannot be expressed as decibel in the strict sense. Nevertheless, it is standard usage in system dynamics and control theory use the familiar formula to compute the logarithmic magnitude

**Equation 11.39 logarithmic magnitude of  $H(j\omega)$  in “dB”**

The phase is usually plotted in degrees, and the  $\omega$ -axis is logarithmic in both plots. The two plots are typically tiled vertically with the magnitude plot above the phase. We now work a simple example.

**Example 11.05-1 A simple Bode plot**

Let a system have transfer function  $H(s) = s$ , a single zero at the origin. Find the frequency response function and draw the Bode plot for the system.



It turns out that, when plotted on this logarithmic scale (recall that the  $\omega$ -axes are also plotted logarithmically), both the magnitude and phase are quite asymptotic to straight-lines for first- and second-order systems.

Furthermore, higher-order system transfer functions can be re-written as the *product* of those of first-and second-order. For instance,

$$H(s) = \frac{-s + -}{s^3 + -s^2 + -s + -} \quad (11.40a)$$

$$= - \cdot (-s + 1) \cdot \frac{1}{-s + 1} \cdot \frac{1}{s^2 + -s + -} \quad (11.40b)$$

Recall (from, for instance, phasor representation) that for products of complex numbers, *phases*  $\phi_i$  *add* and *magnitudes*  $M_i$  *multiply*. For instance,

$$M_1 \angle \phi_1 \cdot \frac{1}{M_2 \angle \phi_2} \cdot \frac{1}{M_3 \angle \phi_3} = \frac{M_1}{M_2 M_3} \angle (\phi_1 - \phi_2 - \phi_3). \quad (11.41)$$

And if one takes the logarithm of the magnitudes, they add; for instance,

$$\log \frac{M_1}{M_2 M_3} = \log M_1 - \log M_2 - \log M_3. \quad (11.42)$$

There is only one more link in the chain: first- and second-order Bode plots depend on a handful of parameters that can be found *directly from transfer functions*. There is no need to compute  $|H(j\omega_0)|$  and  $\angle H(j\omega_0)$ !

In a manner similar to [Example 11.05-1](#), we construct Bode plots for several simple transfer functions in [11.05.1](#). Once we have these simple “building blocks,” we will be able to construct sketches of higher-order systems by graphical addition because logarithmic magnitudes and phases combine by summation, as shown in [11.05.2](#).

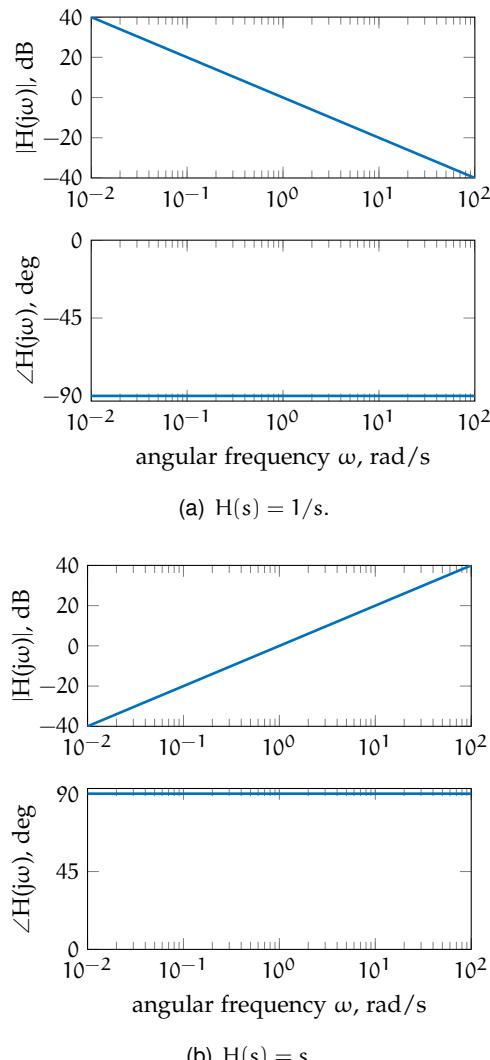
### 11.05.1 Bode plots for simple transfer functions

#### 11.05.1.1 Constant gain

For a transfer function that is simply a constant real gain  $H(s) = K$ , the frequency response function is trivially  $H(j\omega) = K$ . Its magnitude  $|H(j\omega)| = |K|$ . For positive gain  $K$ , the phase is  $\angle H(j\omega) = 0$ , and for negative  $K$ , the phase is  $\angle H(j\omega) = 180$  deg.

#### 11.05.1.2 Pole and zero at the origin

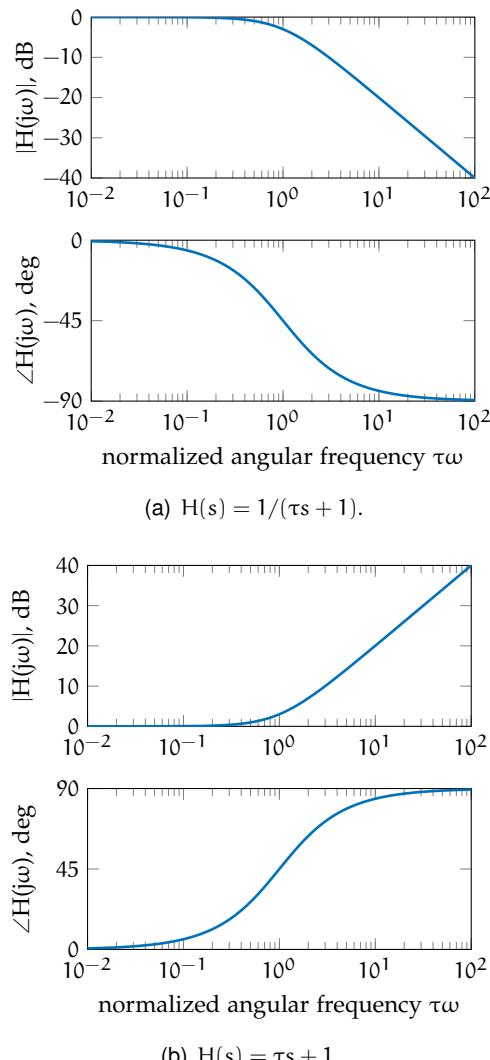
In [Example 11.05-1](#), we have already demonstrated how to derive from the transfer function  $H(s) = s$ , a zero at the origin, the frequency response function plotted in [Figure 11.3\(b\)](#). Similarly, for  $H(s) = 1/s$ , a pole at the origin, the frequency response function plotted in [Figure 11.3\(a\)](#).



**Figure 11.3:** Bode plots for (a) a pole at the origin and (b) a zero at the origin.

### 11.05.1.3 Real pole and real zero

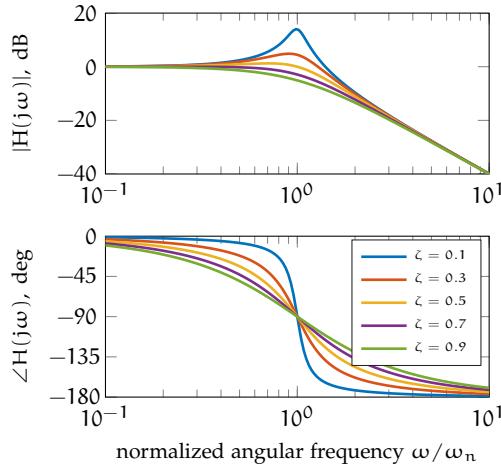
The derivations for real poles and zeros are not included, but the resulting Bode plots are shown in [Figure 11.4](#).



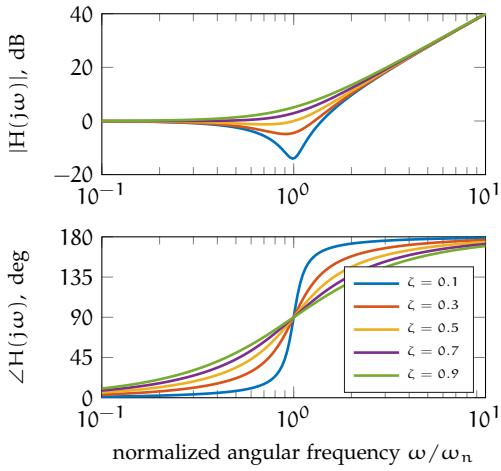
**Figure 11.4:** Bode plots for (a) a single real pole and (b) a single real zero.

#### 11.05.1.4 Complex conjugate pole pairs and zero pairs

The derivations for complex conjugate pole pairs and zero pairs are not included, but the resulting Bode plots are shown in [Figure 11.5](#).



$$(a) H(s) = \omega_n^2 / (s^2 + 2\zeta\omega_n s + \omega_n^2).$$



$$(b) H(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2) / \omega_n^2.$$

**Figure 11.5:** Bode plots for (a) a complex conjugate pole pair and (b) a complex conjugate zero pair.

### 11.05.2 Sketching Bode plots

We can use MATLAB's `bode` command to create Bode plots from LTI system models. However, we must understand how these plots relate to their transfer functions. In this section, we learn to sketch Bode plots in order to deepen our intuition of this relationship.

Let  $H(s) = \prod_i H_i(s)$ ; that is, let  $H(s)$  be the product of several factors  $H_i(s)$ . The magnitude and phase are

$$|H(s)| = \prod_i |H_i(s)| \quad \text{and} \quad \angle H(s) = \sum_i \angle H_i(s). \quad (11.43)$$

The Bode plot consists of plots of  $20 \log_{10}|H(s)|$  and  $\angle H(s)$  with  $s \mapsto j\omega$ . The magnitude and phase expressions, become

$$20 \log_{10}|H(j\omega)| = \sum_i 20 \log_{10}|H_i(j\omega)| \quad \text{and} \quad \angle H(j\omega) = \sum_i \angle H_i(j\omega). \quad (11.44)$$

This result means we can graphically sum both the magnitude and phase Bode plots of the individual factors of  $H(s)$ , as long as we are adding magnitudes in dB.

### Example 11.05-2 a transfer function under analysis

Given the transfer function

$$H(s) = \frac{200000(s+1)}{s^3 + 110s^2 + 11000s + 100000}$$

answer the following questions and imperatives.

- a Sketch a Bode plot on [Figure 11.6](#).
- b Confirm the accuracy of the sketch in Matlab, using the functions `bode` and `tf`.
- c If the input to a system with this transfer function is  $5 \sin(\omega t + \pi/7)$ , what is the output amplitude and phase for
  - i  $\omega = 1 \text{ rad/s}$ ,
  - ii  $\omega = 10 \text{ rad/s}$ , and
  - iii  $\omega = 1000 \text{ rad/s}$ ?

Use Matlab's function `evalfr` to perform the calculations.



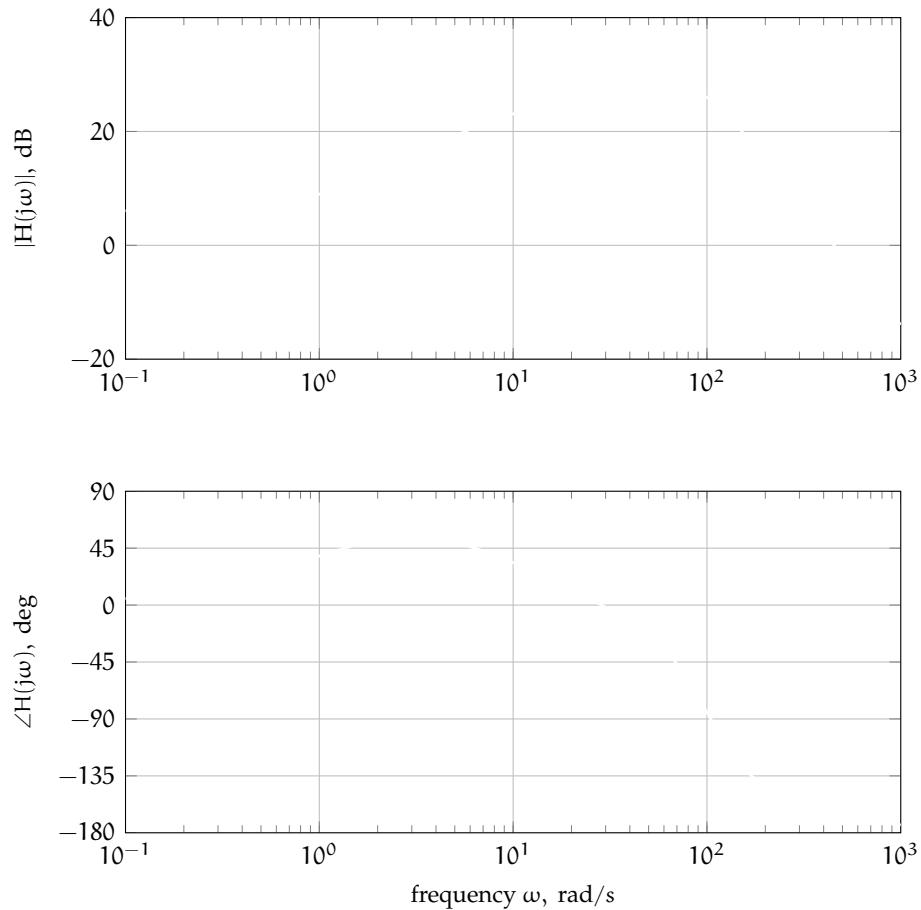


Figure 11.6: a Bode plot for Example 11.05-2.

```
sys = 2e5*...
tf(...,
[1,1],...
[1,110,11000,1e5]...
);
bode(sys);
```

Figure 11.7: a Matlab code listing for generating a Bode plot from Example 11.05-2.

```
in_amp = 5;
in_phase = pi/7; % rad
omega_a = [1,10,1e3]; % rad/s

for i = 1:length(omega_a)
    H_eval = evalfr(sys,j*omega_a(i));
    H_mag = abs(H_eval);
    H_phase = angle(H_eval);
    out_amp = 5*H_mag;
    out_phase = in_phase + H_phase;
    sprintf([
        'For input angular freq %0.2g,\n',...
        ' input amplitude %0.2g,\n',...
        ' input phase %0.2g,\n',...
        ' H magnitude %0.2g, and\n',...
        ' H phase %0.2g,\n',...
        ' the output amplitude is %0.2g and\n',...
        ' the output phase is %0.2g.\n',...
    ],...
        omega_a(i),...
        in_amp,...  
in_phase,...  
H_mag,...  
H_phase,...  
out_amp,...  
out_phase...
    )
end
```

**Figure 11.8:** a Matlab code listing for evaluating the frequency response function  $H(j\omega)$  from Example 11.05-2.

## Lecture 11.06 Periodic input, frequency response

Let a system  $H$  have a periodic input  $u$  represented by a Fourier series. For reals  $a_0$ ,  $\omega_1$  (fundamental frequency),  $A_n$ , and  $\phi_n$ , let

$$u(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} A_n \sin(n\omega_1 t + \phi_n). \quad (11.45)$$

The  $n$ th harmonic is

which, from Equation 11.37 yields forced response

Applying the principle of superposition, the forced response of the system to periodic input  $u$  is

$$y(t) = \frac{a_0}{2} H(j0) + \sum_{n=1}^{\infty} A_n |H(jn\omega_1)| \sin(n\omega_1 t + \phi_n + \angle H(jn\omega_1)). \quad (11.46)$$

Similarly, for inputs expressed as a complex Fourier series with components

$$u_n(t) = c_n e^{jn\omega_1 t}, \quad (11.47)$$

each of which has output

$$y_n(t) = c_n H(jn\omega_1) e^{jn\omega_1 t}, \quad (11.48)$$

the principle of superposition yields

$$y(t) = \sum_{n=-\infty}^{\infty} c_n H(jn\omega_1) e^{jn\omega_1 t}. \quad (11.49)$$

Equations 11.46 and 11.49 tell us that, for a periodic input, we obtain a periodic output with each harmonic  $\omega_n$  amplitude scaled by  $|H(j\omega_n)|$  and phase offset by  $\angle H(j\omega_n)$ . As a result, the response will usually undergo significant *distortion*, called *phase distortion*. The system  $H$  can be considered to *filter* the input by amplifying and suppressing different harmonics. This

**phase distortion  
filter**

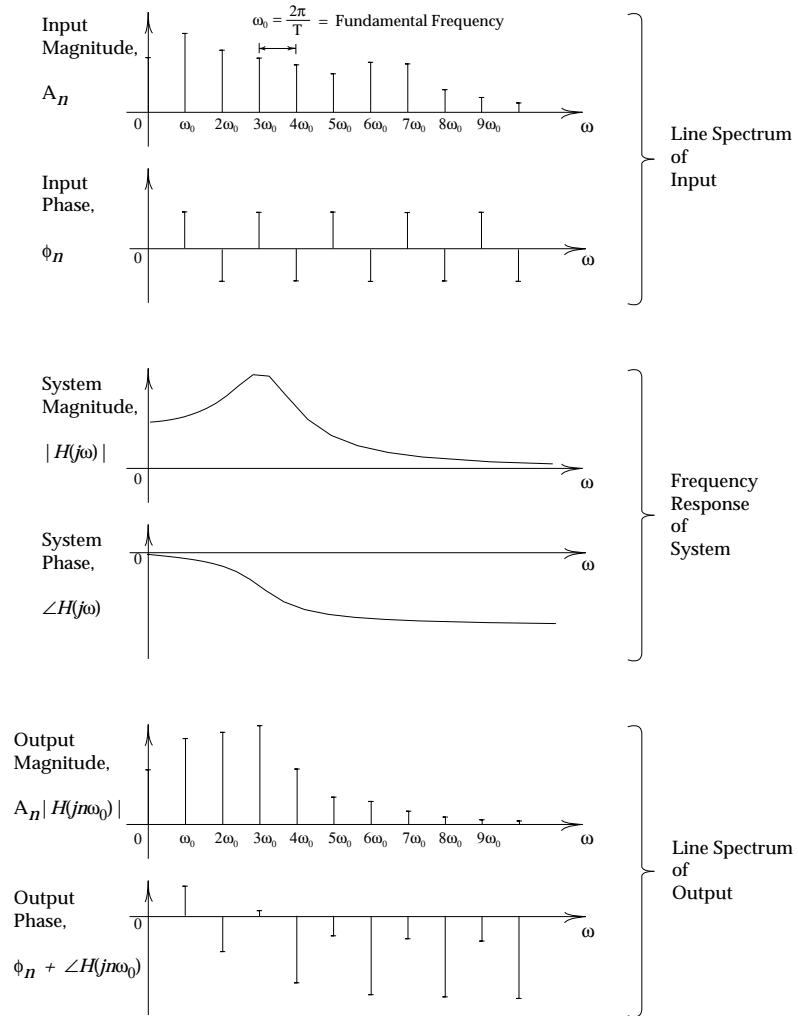


Figure 11.9: response  $y$  of a system  $H$  to periodic input  $u$ .

is why systems not intended to be used as such are still sometimes called “filters.” This way of thinking about systems is very useful to the study of vibrations, acoustics, measurement, and electronics.

All this can be visualized via a Bode plot, which is a significant aspect of its analytic power. An example of such a visualization is illustrated in Figure 11.9.

## Lecture 11.07 Design problem: rainwater catchment and irrigation system

Design a home rainwater catchment system and sprinkler distribution system. Most places, a surprising amount of water falls on a house's roof throughout a year. Capturing it for irrigation can save water costs and reduce the environmental impact of watering lawns, plants, and gardens.

Design a home rainwater catchment and irrigation system. The design constraints are as follows.

1. It should be designed for Olympia, Washington rainfall, as described in [Table 11.1](#).
2. For a house, large tanks are unsightly. Instead, use a series of connected barrels.

After discussions with the customer, the following design requirements for the system are identified.

1. It should be capable of distributing one inch of water per unit area June through September, even during drought conditions, during which there is half the average rainfall in the months March through September (see [Table 11.1](#)).
2. The roof area for collection is 400 square ft.
3. The lawn area for distribution is 600 square ft.
4. It should be low-maintenance.
5. The distribution system should be capable of being "blown out" during winter months *or* it must be designed to handle sudden dips from 33 down to 22 deg F for up to two days.<sup>2</sup>
6. When tanks are full, it should be able to gracefully dump excess water. If possible, designing it to refresh itself by dumping old water for new water is desired.
7. It should be able to handle a heavy rain of 1 inch per hour via an overflow mechanism, but be able to handle a moderate rain of 0.2 inches per hour without requiring overflow (unless the tanks are full).
8. It should be designed to be fed from typical house rain gutter downspouts.
9. Distribution should be automated.

<sup>2</sup>A potential way to mitigate freezing is keeping the water in motion. Care must be taken not to create inadvertent ice skating rinks.

**Table 11.1:** mean monthly rainfall data and corresponding “drought conditions” for Olympia, Washington, USA (NOAA, 2017).

month	mean precip. (in)	drought precip. (in)
January	8.51	8.51
February	5.82	5.82
March	4.85	2.43
April	3.11	1.55
May	1.84	0.92
June	1.42	0.71
July	0.67	0.34
August	1.31	0.65
September	2.36	1.18
October	4.66	4.66
November	7.66	7.66
December	8.52	8.52

10. Energy efficiency is desired. If possible, using tanks’ potential energy for distribution is desired. In this case, unconventional distribution networks are allowable (e.g. “drip” systems without conventional sprinkler heads that require high pressure). However, the distribution hardware should not be custom-designed.
11. Commercially available parts are desired. Minimize the number of custom parts (zero is best).

The focus of the design problem is the sizing of the pipes, barrels, and mechanisms based on a dynamic system analysis.<sup>3</sup>

It is highly recommended that you use the following Fourier Series fit to the Olympia drought rainfall data, presented as trigonometric series coefficient vectors  $a$  and  $b$  for easy definition in Matlab.<sup>4</sup>

```
w = 0.5236; % fundamental frequency
a0 = 3.579; % dc offset
a(1) = 4.144;
b(1) = 0.6244;
a(2) = 1.332;
b(2) = 0.07578;
```

<sup>3</sup>A design that is not informed by a thoroughly presented system model will receive no credit.

<sup>4</sup>The fit is an 8-term Fourier series fit performed via Matlab’s `fit` function.

```
a(3) = -0.07667;
b(3) = 0.03167;
a(4) = -0.2469;
b(4) = 0.0004836;
a(5) = -0.09448;
b(5) = 0.01735;
a(6) = 0.07417;
b(6) = -2.131e-06;
a(7) = -0.06748;
b(7) = -0.0124;
a(8) = -0.1235;
b(8) = -0.0002381;
```

A system model response to this input can be used to determine the system parameters, such as the number of barrels required. Do not forget to include the effect of distribution, which can be modeled as a *negative* source. Although we have the tools to perform the analysis analytically, it is highly recommended that a Matlab simulation is developed using `ss` to define the system and `lsim` to simulate the response. A frequency response analysis using `bode` may also prove useful. It may be possible to simply iteratively tweak design parameters until the simulation meets the requirements.

A thorough report is required. It is highly recommended that LaTeX is used. Thorough analysis, results, and design is required. All sizing and specific parts are required. Either an analytic or a numerical (simulation) demonstration of the design's fulfillment of the requirements is required.

## Lecture 11.08 Nonlinear fluid system example

Consider a fluid system with an input volumetric flowrate  $Q_s$  into a capacitance  $C$  that is drained by only a single pipe of *nonlinear* resistance  $R$  and  $L$ , as shown in the linear graph of [Figure 11.10](#). The nonlinearity of  $R$  is a good way to model an overflow. In this lecture, we will derive a *nonlinear state-space model* for the system—specifically, a state equation—and solve it, numerically using Matlab.

**nonlinear  
state-space model**

### 11.08.1 Normal tree, order, and variables

[Figure 11.10](#) already shows the normal tree. There are two independent energy storage elements, making it a second-order ( $n = 2$ ) system. We define the state vector to be

$$\mathbf{x} = [P_C \quad Q_L]^\top. \quad (11.50)$$

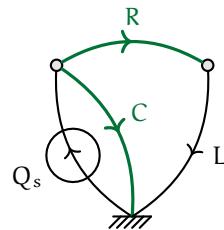
The input vector is defined as  $\mathbf{u} = [Q_s]$ .

### 11.08.2 Elemental, continuity, and compatibility equations

Before turning to our familiar elemental equations, we'll consider the nonlinear resistor.

#### 11.08.2.1 Nonlinear elemental equation

Suppose we are trying to model an overflow with the pipe R-L to ground. An overflow would have no flow until the fluid capacitor fills to a certain height, then it would transition to flowing quite rapidly. This process seems to be inherently nonlinear because we cannot write an element that



**Figure 11.10:** a linear graph and normal tree (green) for a nonlinear fluid system.

depends linearly on the height of the fluid in the capacitor (even if height was one of our state variables, which it is not).

The volume in the tank can be found by integrating in flow ( $Q_s$ ) minus out flow ( $Q_R$ ), but this is not accessible within a simulation, since it must be integrated, so it's not an ideal variable for our model. However, the pressure  $P_C$ —a state variable—is proportional to the fluid height in the capacitor, which we'll call  $h$ :

$$P_C = \rho gh, \quad (11.51)$$

where  $\rho$  is the density of the fluid and  $g$  is the gravitational acceleration. Since the height of the capacitor is presumably known, we can use  $P_C$  to be our fluid height metric.

When the height  $h$  reaches a certain level, probably near the capacitor's max, which we'll denote  $h_m$ , we want our overflow pipe R-L to start flowing. Since  $P_C$  is our height metric, we want to define a resistance as a function of it,  $R(P_C)$ .

Now we must determine the form of  $R(P_C)$ . Clearly, when  $h \sim P_C$  is small, we want as little as possible flow through R-L, so  $R(P_C)$  should be large. If  $R$  was infinitely large, divisions by zero would likely arise in a simulation, so we choose to set our low-pressure  $R$  to some finite value:

$$R(P_C)|_{P_C \rightarrow 0} = R_0. \quad (11.52)$$

Conversely, when  $h \sim P_C$  is large (near max), we want maximum flow through R-L, so  $R(P_C)$  should be some finite value, say, that of the pipe:

$$R(P_C)|_{P_C \rightarrow \infty} = R_\infty. \quad (11.53)$$

Clearly, this model requires  $R_\infty \ll R_0$ .

The transition from  $R_0$  to  $R_\infty$  should be smooth in order to minimize numerical solver difficulties. Furthermore, a smooth transition is consistent with, say, a float opening a valve at the bottom of the capacitor,<sup>5</sup> since the valve would transition continuously from closed to open. Many functions could be used to model this transition, especially if piecewise functions are considered. However, the tanh function has the merit of enabling us to easily define a single non-piecewise function for the entire domain. Let  $\bar{P}_C$

---

<sup>5</sup>Note that this model might be said to assume the overflow pipe is attached to the bottom of the capacitor since the pressure driving fluid through this pipe is supposed to be  $P_C$ . However, no matter the overflow valve's inlet height, if its outlet is at the height of the bottom of the capacitor, this model is still valid.

be the transition pressure and  $\Delta P_C$  be the transition width. A convenient nonlinear resistor, then, is

$$R(P_C) = R_\infty + \frac{R_0 - R_\infty}{2} \left( 1 - \tanh \frac{5(P_C - \bar{P}_C)}{\Delta P_C} \right). \quad (11.54)$$

Note that this function only approximately satisfies  $R(P_C)|_{P_C \rightarrow 0} = R_0$ , but the small deviation from this constraint is worth it for the convenience it provides. Another noteworthy aspect of Equation 11.54 is the factor of 5, which arises from the tanh function's natural transition width, which we alter via  $\Delta P_C$ .

#### 11.08.2.2 Other elemental equations and the continuity and compatibility equations

The other elemental equations have been previously encountered and are listed in the table, below. Furthermore, continuity and compatibility equations can be found in the usual way—by drawing contours and temporarily creating loops by including links in the normal tree. We proceed by drawing a table of all elements and writing an elemental equation for each element, a continuity equation for each branch of the normal tree, and a compatibility equation for each link.

el.	elemental eq.	el.	cont/comp. eq.
C	$\frac{dP_C}{dt} = \frac{1}{C} Q_C$	C	$Q_C = Q_s - Q_L$
L	$\frac{dQ_L}{dt} = \frac{1}{L} P_L$	L	$P_L = P_C - P_R$
R	$P_R = Q_R R(P_C)$	R	$Q_R = Q_L$

#### 11.08.3 State equation

The system of equations composed of the elemental, continuity, and compatibility equations can be reduced to the state equation. This equation *nonlinear*, so it cannot be written in the linear form with A and B matrices. However, it can still be written as a system of first-order ordinary differen-

tial equations, as follows:

$$\begin{aligned}\frac{dx}{dt} &= f(x, u) \\ &= \left[ \frac{(Q_s - Q_L)/C}{(P_C - Q_L R(P_C))/L} \right].\end{aligned}\quad (11.55)$$

Although it appears simple, this nonlinear differential equation likely has no known analytic solution. Two other options are available:

1. linearize the model about an operating point and solve the linearized equation or
2. numerically solve the nonlinear equation.

Both methods are widely useful, but let's assume we require the model to be accurate over a wide range of capacitor fullness. Therefore, we choose to investigate via numerical solution.

#### 11.08.4 Simulation

Broadly, the numerical investigation will be conducted via Matlab's `ode23t` solver.<sup>6</sup> Of course, as with any numerical solution, specific values of the parameters must be selected. We begin with declaring the fluid to be water, endowing it with a density, and specify the gravitational acceleration  $g$ . Furthermore, an "anonymous" function `P_fun` is defined, accepting the height  $h$  of the fluid in the capacitor and returning the corresponding pressure. Other parameters specified include the fluid capacitance  $C$  and the overflow pipe inertance  $L$ .

```
global C L % global to be used in state equation
C = 1e3; % ... fluid capacitance
L = 1e-3; % ... fluid inertance
rho = 997; % kg/m^3 ... density of water
g = 9.81; % m/s^2 ... gravitational constant
P_fun = @(h) rho*g*h; % pressure as a function of height
```

Next, we define the maximum height  $h_{max}$  of fluid in the capacitor, the transition height  $h_t$ , and the distance  $dh$  over which the resistor will transition from high to low impedance.

---

<sup>6</sup>Source Matlab files can be found on `ricopic.one/dynamic_systems_ii/source` as `ricopic.one/dynamic_systems_ii/source/nonlinear_tank_example.m` and `ricopic.one/dynamic_systems_ii/source/nonlinear_fluid_state.m`.

```

h_max = 1; % m ... maximum height of fluid
h_t = .88; % m ... transition height
dh = .05; % m ... height difference for transition

```

Corresponding pressures, which we prefer for computation, can be computed with `P_fun`.

```

P_t = P_fun(h_t); % N/m^2 ... transition pressure
dP = P_fun(dh); % N/m^2 ... pressure dif for transition

```

### 11.08.5 Nonlinear resistance

Now, let's define the variable resistance function `R_fun` ( $R(P_C)$ ). We define the anonymous function via the two "limiting" resistances  $R_0$  ( $R_0$ ) and  $R_{\text{inf}}$  ( $R_{\infty}$ ).

```

R_inf = 1e-1; % N/m^2 / m^3/s ... resistance with full cap
R_0 = 1e2; % ... resistance with empty capacitor
R_fun = @(P) (R_0-R_inf)/2*(1-tanh(5/dP*(P-P_t)))+R_inf;

```

Let's take a moment to plot this function. See [Figure 11.11](#) for the results. This is a reasonable approximation of a valve that allows no flow until the capacitor fluid height reaches a threshold, then allows a significant amount of flow.

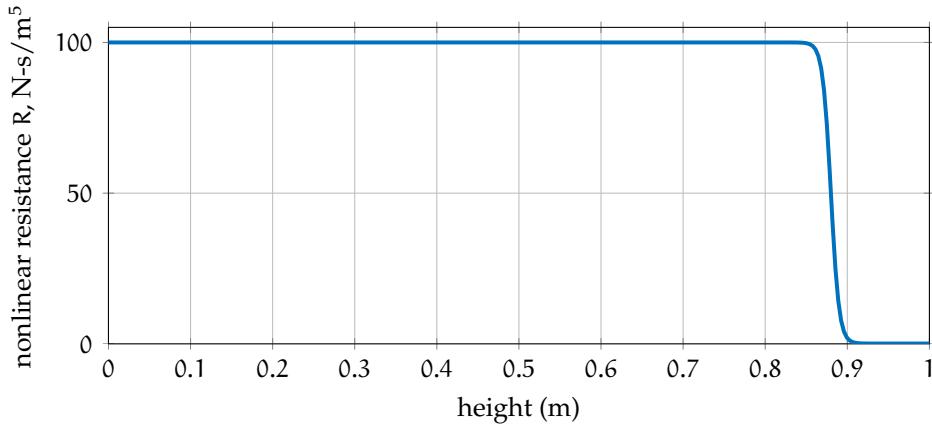
```

h_half = linspace(0,h_t-dh,50);
h_a = [... % heights to plot
    h_half(1:end-1), ...
    linspace(h_half(end),h_max,50) ...
];
P_a = P_fun(h_a); % N/m^2 ... pressures to plot

```

#### 11.08.5.1 Numerical solution

The numerical ODE solver we'll use (`ode23t`) requires we define the first-order system of differential equations from [Equation 11.55](#). This is done by writing a function file `nonlinear_fluid_state.m` that the function return the time derivative of the state vector `x_a` ( $\dot{x}$ ) at a given time.

**Figure 11.11:** nonlinear resistance versus tank water level.

```

function dx = nonlinear_fluid_state(t,x,u_fun,R_fun)
global C L
% x(1) is P_C
% x(2) is Q_L
% R_fun is the nonlinear resistance

% call input function at this time step
Q_s = u_fun(t);

% compute nonlinear resistance at this time step
R = R_fun(x(1));

dx = zeros(2,1);      % a column vector
dx(1) = (Q_s - x(2))/C;      % d P_C/dt
dx(2) = (x(1) - x(2)*R)/L; % d Q_L/dt
end

```

We also pass it the nonlinear resistance function `R_fun` and the input function `Q_s_fun`. Let's model an offset sinusoidal input flowrate, defined as an anonymous function as follows.

```

Q_s_fun = @(t) 1e4*(1+sin(2*pi/1e2*t));

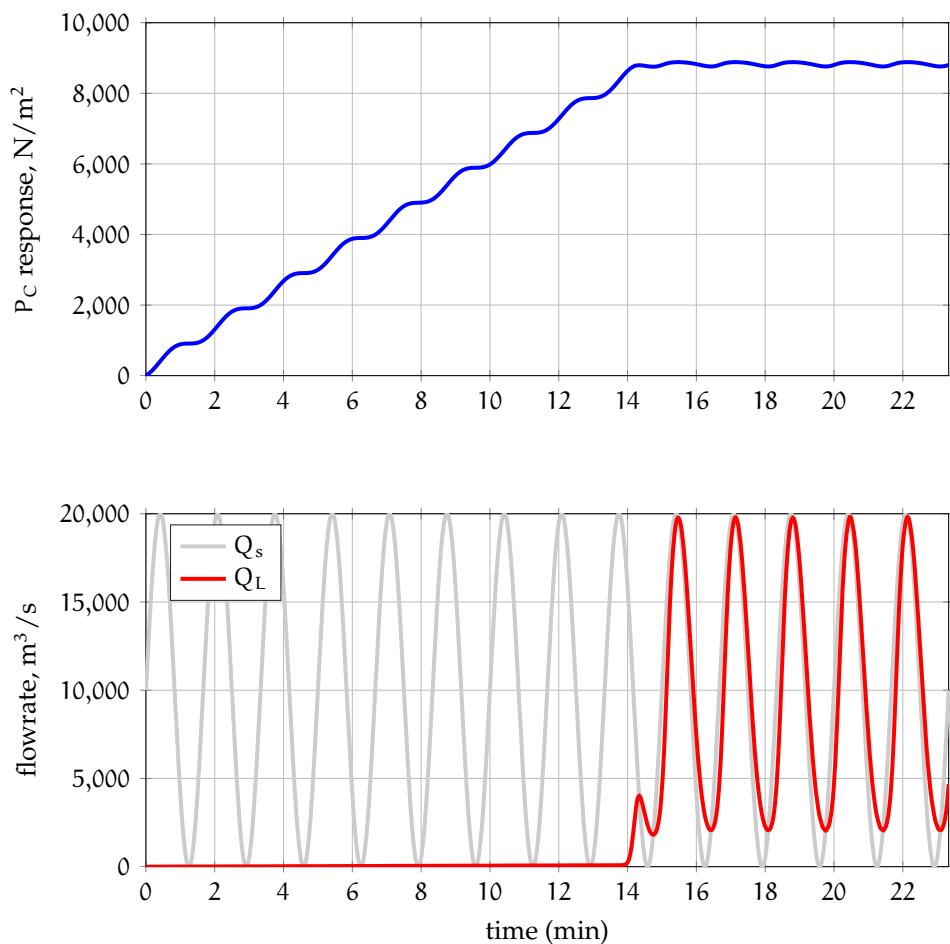
```

We're ready to simulate! The time array and zero initial conditions are specified, then simulation commences. There are several Matlab ODE

solver routines with the same or similar syntax. Many ODEs can be solved with the `ode45` function. However, this problem is what is called “stiff,” which runs much better on the solver `ode23t`.

```
t_a = linspace(0,1.4e3,1e3);
x_0 = [0;0];
x_sol_struc = ode23t(... 
    @(t,x) nonlinear_fluid_state(t,x,Q_s_fun,R_fun), ...
    t_a, ...
    x_0...
);
x_sol = deval(x_sol_struc,t_a);
```

We plot the results in Figure 11.12. So the overflow is relatively inactive while the capacitor fills, until  $P_C$  achieves the pressure associated with a near-full capacitor. Then the flowrate suddenly increases rapidly due to the sudden drop in  $R(P_C)$ . Since the input is oscillating, the overflow pipe loses flowrate, then gains it again when the input flowrate increases enough to increase the capacitor pressure.



**Figure 11.12:** state response to input  $Q_s$ .

## Exercises

### Exercise 11.1

Explain, in your own words (supplementary drawings are ok), what the *frequency domain* is, how we derive models in it, and why it is useful.

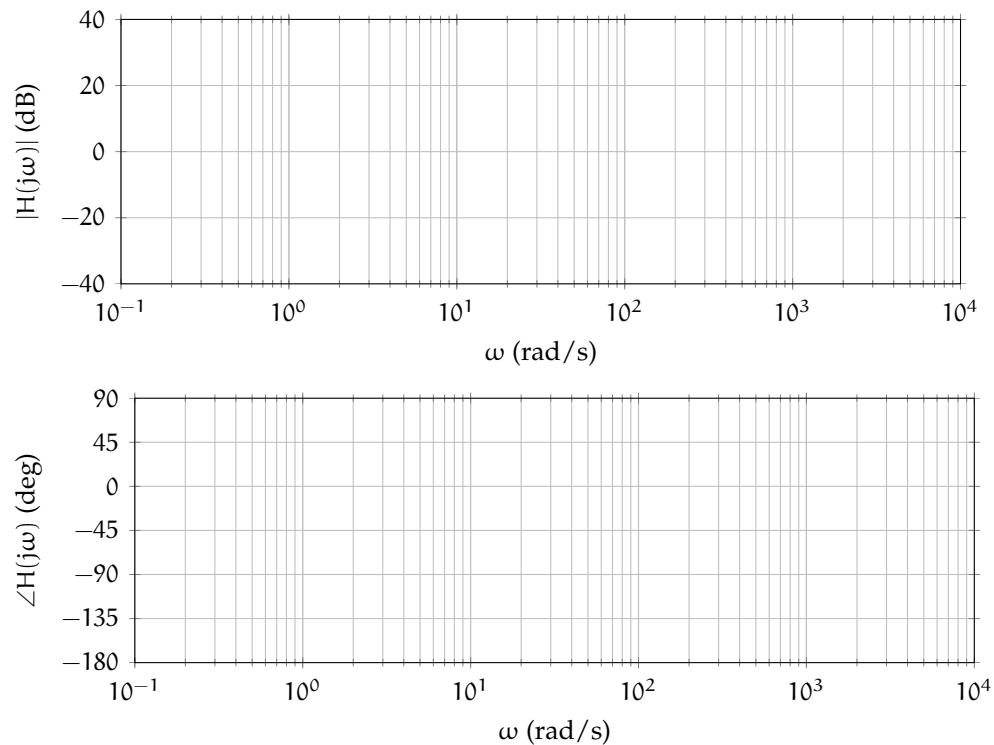
### Exercise 11.2

Let a transfer function  $H$  be

$$\frac{10(s + 100)}{s^2 + 2s + 100}. \quad (11.56)$$

Use  $H$  to respond to the following questions and imperatives.

- a. Write  $H$  as a product of standard-form transfer functions.
- b. Find the frequency response function  $H(j\omega)$  *without* simplifying.
- c. Use the axes below to sketch the Bode plot of  $H$ .



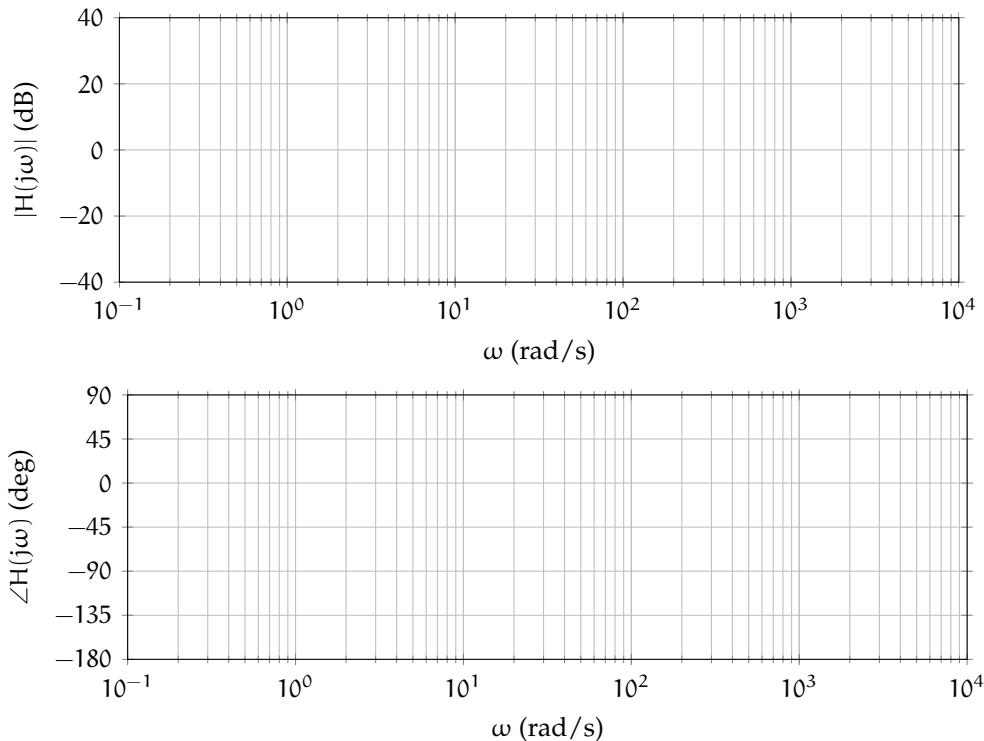
**Exercise 11.3**

Let a transfer function  $H$  be

$$H(s) = \frac{1000(s + 10)}{(s + 100)(s + 1000)}.$$

Use  $H$  to respond to the following questions and imperatives.

- a. Write  $H$  as a product of standard-form transfer functions.
- b. Find the frequency response function  $H(j\omega)$  *without* simplifying.
- c. Use the axes below to sketch the Bode plot of  $H$ .



# **Part VI**

# **Appendices**



A

---

## Mathematics reference

## Lecture A.01 Quadratic forms

The solution to equations of the form  $ax^2 + bx + c = 0$  is

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad (\text{A.1})$$

### A.01.1 Completing the square

This is accomplished by re-writing the quadratic formula in the form of the left-hand-side (LHS) of this equality, which describes factorization

$$x^2 + 2xh + h^2 = (x + h)^2. \quad (\text{A.2})$$

## Lecture A.02 Trigonometry

### A.02.1 Triangle identities

With reference to the below figure, the *law of sines* is

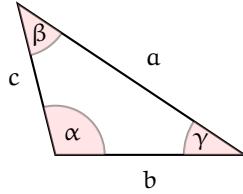
$$\frac{\sin \alpha}{a} = \frac{\sin \beta}{b} = \frac{\sin \gamma}{c} \quad (\text{A.3})$$

and the *law of cosines* is

$$c^2 = a^2 + b^2 - 2ab \cos \gamma \quad (\text{A.4a})$$

$$b^2 = a^2 + c^2 - 2ac \cos \beta \quad (\text{A.4b})$$

$$a^2 = c^2 + b^2 - 2cb \cos \alpha \quad (\text{A.4c})$$



### A.02.2 Reciprocal identities

$$\csc u = \frac{1}{\sin u} \quad (\text{A.5a})$$

$$\sec u = \frac{1}{\cos u} \quad (\text{A.5b})$$

$$\cot u = \frac{1}{\tan u} \quad (\text{A.5c})$$

### A.02.3 Pythagorean identities

$$1 = \sin^2 u + \cos^2 u \quad (\text{A.6a})$$

$$\sec^2 u = 1 + \tan^2 u \quad (\text{A.6b})$$

$$\csc^2 u = 1 + \cot^2 u \quad (\text{A.6c})$$

#### A.02.4 Co-function identities

$$\sin\left(\frac{\pi}{2} - u\right) = \cos u \quad (\text{A.7a})$$

$$\cos\left(\frac{\pi}{2} - u\right) = \sin u \quad (\text{A.7b})$$

$$\tan\left(\frac{\pi}{2} - u\right) = \cot u \quad (\text{A.7c})$$

$$\csc\left(\frac{\pi}{2} - u\right) = \sec u \quad (\text{A.7d})$$

$$\sec\left(\frac{\pi}{2} - u\right) = \csc u \quad (\text{A.7e})$$

$$\cot\left(\frac{\pi}{2} - u\right) = \tan u \quad (\text{A.7f})$$

#### A.02.5 Even-odd identities

$$\sin(-u) = -\sin u \quad (\text{A.8a})$$

$$\cos(-u) = \cos u \quad (\text{A.8b})$$

$$\tan(-u) = -\tan u \quad (\text{A.8c})$$

#### A.02.6 Sum-difference formulas (AM or lock-in)

$$\sin(u \pm v) = \sin u \cos v \pm \cos u \sin v \quad (\text{A.9a})$$

$$\cos(u \pm v) = \cos u \cos v \mp \sin u \sin v \quad (\text{A.9b})$$

$$\tan(u \pm v) = \frac{\tan u \pm \tan v}{1 \mp \tan u \tan v} \quad (\text{A.9c})$$

#### A.02.7 Double angle formulas

$$\sin(2u) = 2 \sin u \cos u \quad (\text{A.10a})$$

$$\cos(2u) = \cos^2 u - \sin^2 u \quad (\text{A.10b})$$

$$= 2 \cos^2 u - 1 \quad (\text{A.10c})$$

$$= 1 - 2 \sin^2 u \quad (\text{A.10d})$$

$$\tan(2u) = \frac{2 \tan u}{1 - \tan^2 u} \quad (\text{A.10e})$$

### A.02.8 Power-reducing or half-angle formulas

$$\sin^2 u = \frac{1 - \cos(2u)}{2} \quad (\text{A.11a})$$

$$\cos^2 u = \frac{1 + \cos(2u)}{2} \quad (\text{A.11b})$$

$$\tan^2 u = \frac{1 - \cos(2u)}{1 + \cos(2u)} \quad (\text{A.11c})$$

### A.02.9 Sum-to-product formulas

$$\sin u + \sin v = 2 \sin \frac{u+v}{2} \cos \frac{u-v}{2} \quad (\text{A.12a})$$

$$\sin u - \sin v = 2 \cos \frac{u+v}{2} \sin \frac{u-v}{2} \quad (\text{A.12b})$$

$$\cos u + \cos v = 2 \cos \frac{u+v}{2} \cos \frac{u-v}{2} \quad (\text{A.12c})$$

$$\cos u - \cos v = -2 \sin \frac{u+v}{2} \sin \frac{u-v}{2} \quad (\text{A.12d})$$

### A.02.10 Product-to-sum formulas

$$\sin u \sin v = \frac{1}{2} [\cos(u-v) - \cos(u+v)] \quad (\text{A.13a})$$

$$\cos u \cos v = \frac{1}{2} [\cos(u-v) + \cos(u+v)] \quad (\text{A.13b})$$

$$\sin u \cos v = \frac{1}{2} [\sin(u+v) + \sin(u-v)] \quad (\text{A.13c})$$

$$\cos u \sin v = \frac{1}{2} [\sin(u+v) - \sin(u-v)] \quad (\text{A.13d})$$

### A.02.11 Two-to-one formulas

$$A \sin u + B \cos u = C \sin(u + \phi) \quad (\text{A.14a})$$

$$= C \cos(u + \psi) \text{ where} \quad (\text{A.14b})$$

$$C = \sqrt{A^2 + B^2} \quad (\text{A.14c})$$

$$\phi = \arctan \frac{B}{A} \quad (\text{A.14d})$$

$$\psi = -\arctan \frac{A}{B} \quad (\text{A.14e})$$

## Lecture A.03 Matrix inverses

This is a guide to inverting  $1 \times 1$ ,  $2 \times 2$ , and  $n \times n$  matrices.

Let  $A$  be the  $1 \times 1$  matrix

$$A = [a].$$

The inverse is simply the reciprocal:

$$A^{-1} = [1/a].$$

Let  $B$  be the  $2 \times 2$  matrix

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

It can be shown that the inverse follows a simple pattern:

$$\begin{aligned} B^{-1} &= \frac{1}{\det B} \begin{bmatrix} b_{22} & -b_{12} \\ -b_{21} & b_{11} \end{bmatrix} \\ &= \frac{1}{b_{11}b_{22} - b_{12}b_{21}} \begin{bmatrix} b_{22} & -b_{12} \\ -b_{21} & b_{11} \end{bmatrix}. \end{aligned}$$

Let  $C$  be an  $n \times n$  matrix. It can be shown that its inverse is

$$C^{-1} = \frac{1}{\det C} \text{adj } C,$$

**adjoint** where  $\text{adj}$  is the *adjoint* of  $C$ .

## Lecture A.04 Laplace transforms

The definition of the one-side Laplace and inverse Laplace transforms follow.

### Definition A.04.1: Laplace transforms (one-sided)

Laplace transform  $\mathcal{L}$ :

$$\mathcal{L}(y(t)) = Y(s) = \int_0^{\infty} y(t)e^{-st} dt. \quad (\text{A.15})$$

Inverse Laplace transform  $\mathcal{L}^{-1}$ :

$$\mathcal{L}^{-1}(Y(s)) = y(t) = \frac{1}{2\pi j} \int_{\sigma-j\infty}^{\sigma+j\infty} Y(s)e^{st} ds. \quad (\text{A.16})$$

See Table C.2 for a list of properties and common transforms.



B

---

## Advanced topics

## B.01 Systems with repeated eigenvalues

This topic is fully treated by Brogan (1991, p 250), but not by Rowell and Wormley (1997). Every  $n \times n$  matrix has  $n$  eigenvalues, and for each distinct eigenvalue  $\lambda_i$ , a linear independent eigenvector  $m_i$  exists. For every eigenvalue  $\lambda_i$  repeated  $\mu_i$  times (termed *algebraic multiplicity* of  $\lambda_i$ ), any number  $q_i$  (termed *geometric multiplicity* or *degeneracy* of  $\lambda_i$ ) up to and including  $\mu_i$  of independent eigenvectors may exist:  $1 \leq q_i \leq \mu_i$ .  $q_i$  is equal to the dimension of the null space of  $A - I\lambda_i$ ,

$$q_i = n - \text{rank}(A - \lambda_i I). \quad (\text{B.1})$$

This gives rise to the three cases that follow.

**full degeneracy** When  $q_i = \mu_i$ , the eigenvalue problem has  $q_i = \mu_i$  independent solutions for  $m_i$ . So, even though there were not  $n$  distinct eigenvalues,  $n$  distinct eigenvectors still exist and we can diagonalize or decouple the system as before.

**simple degeneracy** When  $q_i = 1$ , the eigenvalue problem has  $q_i = 1$  independent solutions for  $m_i$ . We would still like to construct a basis set of  $n$  independent vectors, but they can no longer be eigenvectors, and we will no longer be able to fully diagonalize or decouple the system. There are multiple ways of doing this (e.g. Gram-Schmidt), but the typical and most nearly diagonal way is to construct  $\mu_i - q_i$  *generalized eigenvectors* (here also called  $m_i$ ), which will be included in the modal matrix  $M$  along with the eigenvectors. The generalized eigenvectors are found by solving the usual eigenvalue/vector problem for the first eigenvector  $m_i^1$  corresponding to  $\lambda_i$ , then solving it again with the following equations to find the generalized eigenvectors

$$\begin{aligned} (A - \lambda_i) m_i^2 &= m_i^1 \\ (A - \lambda_i) m_i^3 &= m_i^2 \\ &\vdots \end{aligned}$$

This forms the modal matrix  $M$ . The block-diagonal *Jordan form* matrix  $J$ , analogous to the diagonal  $\Lambda$  is

$$J = M^{-1} A M, \quad (\text{B.2})$$

which gives the most-decoupled state transition matrix

$$\Phi(t) = M e^{Jt} M^{-1}. \quad (\text{B.3})$$

**general degeneracy** If  $1 < q_i < \mu_i$ , the preceding method applies, but it may be ambiguous as to which eigenvector the generalized eigenvectors correspond (or how many for each). This can be approached by trial and error or a systematic method presented by Brogan (1991, p 255).



C

---

## Summaries

## C.01 Summary of system representations

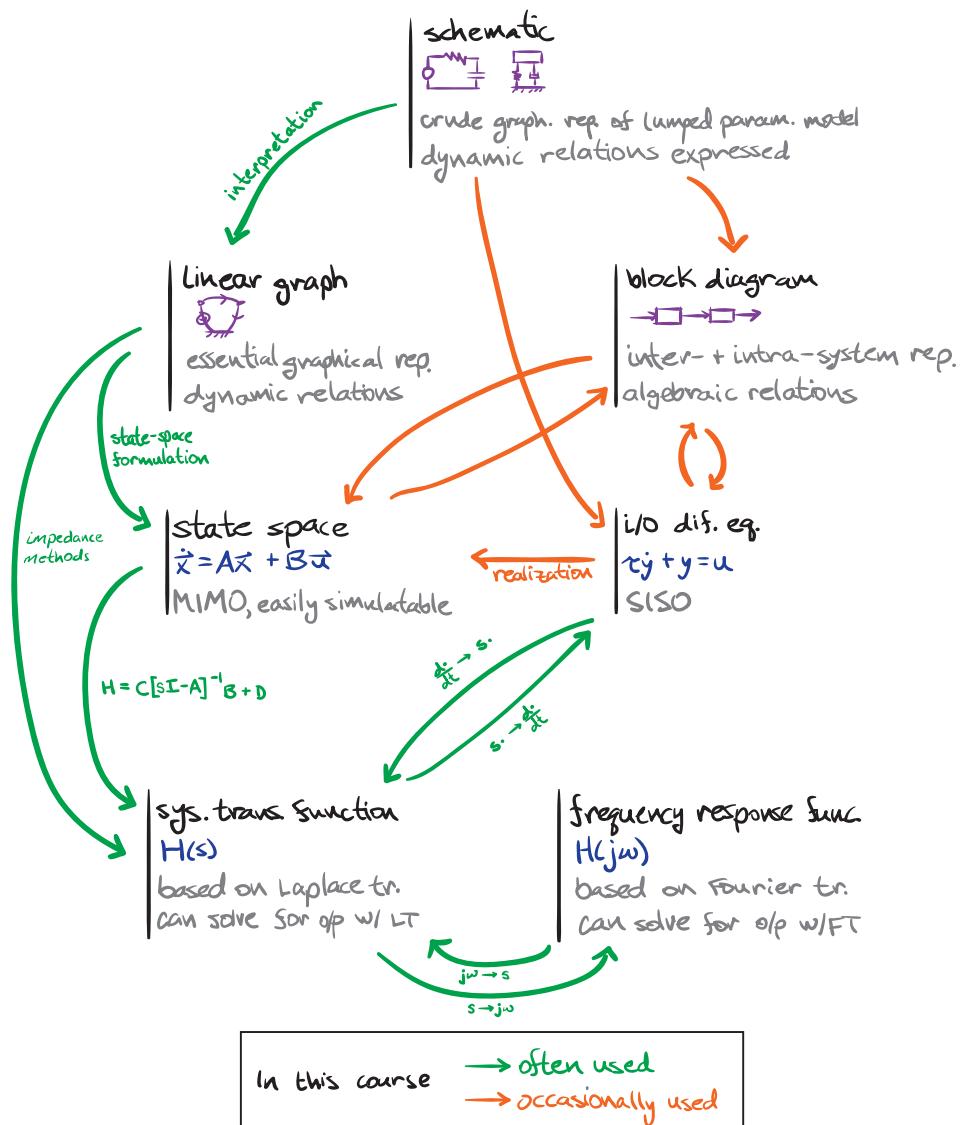


Figure C.1: relations among system representations.

## C.02 Summary of one-port elements

The one table to rule them all, **Table C.1**.

**Table C.1:** parameters, elemental equations, and impedances of one-port elements for generalized, mechanical, electrical, fluid, and thermal systems.

	generalized	mechanical translation	mechanical rotation	electrical	fluid	thermal
variables	across	$\mathcal{V}$	velocity $v$	angular vel. $\Omega$	voltage $v$	pressure $P$
	through	$\mathcal{F}$	force $f$	torque $T$	current $i$	heat fr. $q$
A-type	capacitor	capacitor	mass	mom. inertia	capacitor	capacitor
	capacitance	$C$	$m$	$J$	$C$	$C$
	elem. eq.	$\frac{d\mathcal{V}_C}{dt} = \frac{1}{C}\mathcal{F}_C$	$\frac{dv_m}{dt} = \frac{1}{m}f_m$	$\frac{d\Omega_J}{dt} = \frac{1}{J}T_J$	$\frac{dv_C}{dt} = \frac{1}{C}i_C$	$\frac{dP_C}{dt} = \frac{1}{C}Q_C$
	impedance	$\frac{1}{Cs}$	$\frac{1}{ms}$	$\frac{1}{Js}$	$\frac{1}{Cs}$	$\frac{1}{Cs}$
T-type	inductor	inductor	spring	rot. spring	inductor	inertance
	inductance	$L$	$1/k$	$1/k$	$L$	$I$
	elem. eq.	$\frac{d\mathcal{F}_L}{dt} = \frac{1}{L}\mathcal{V}_L$	$\frac{df_k}{dt} = kv_k$	$\frac{dT_k}{dt} = k\Omega_k$	$\frac{di_L}{dt} = \frac{1}{L}v_L$	$\frac{dQ_I}{dt} = \frac{1}{I}P_I$
	impedance	$Ls$	$s/k$	$s/k$	$Ls$	$Is$
D-type	resistor	resistor	damper	rot. damper	resistor	resistor
	resistance	$R$	$1/B$	$1/B$	$R$	$R$
	elem. eq.	$\mathcal{V}_R = \mathcal{F}_R R$	$v_B = f_B/B$	$\Omega_B = T_B/B$	$v_R = i_R R$	$P_R = Q_R R$
	impedance	$R$	$1/B$	$1/B$	$R$	$R$

## C.03 Laplace transforms

**Table C.2** is a table with functions of time  $f(t)$  on the left and corresponding Laplace transforms  $L(s)$  on the right. Where applicable,  $s = \sigma + j\omega$  is the Laplace transform variable,  $T$  is the time-domain period,  $\omega_0 2\pi/T$  is the corresponding angular frequency,  $j = \sqrt{-1}$ ,  $a \in \mathbb{R}^+$ , and  $b, t_0 \in \mathbb{R}$  are constants.

**Table C.2:** Laplace transform identities.

function of time $t$	function of Laplace $s$
$a_1 f_1(t) + a_2 f_2(t)$	$a_1 F_1(s) + a_2 F_2(s)$
$f(t - t_0)$	$F(s)e^{-t_0 s}$
$f'(t)$	$sF(s) - f(0)$
$\frac{d^n f(t)}{dt^n}$	$s^n F(s) + s^{(n-1)} f(0) + s^{(n-2)} f'(0) + \dots + f^{(n-1)}(0)$
$\int_0^t f(\tau) d\tau$	$\frac{1}{s} F(s)$
$tf(t)$	$-F'(s)$
$f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau$	$F_1(s)F_2(s)$
$\delta(t)$	1
$u_s(t)$	$1/s$
$u_r(t)$	$1/s^2$
$t^{n-1}/(n-1)!$	$1/s^n$
$e^{-at}$	$\frac{1}{s+a}$
$te^{-at}$	$\frac{1}{(s+a)^2}$

$\frac{1}{(n-1)!} t^{n-1} e^{-at}$	$\frac{1}{(s+a)^n}$
$\frac{1}{a-b} (e^{at} - e^{bt})$	$\frac{1}{(s-a)(s-b)} \quad (a \neq b)$
$\frac{1}{a-b} (ae^{at} - be^{bt})$	$\frac{s}{(s-a)(s-b)} \quad (a \neq b)$
$\sin \omega t$	$\frac{\omega}{s^2 + \omega^2}$
$\cos \omega t$	$\frac{s}{s^2 + \omega^2}$
$e^{at} \sin \omega t$	$\frac{\omega}{(s-a)^2 + \omega^2}$
$e^{at} \cos \omega t$	$\frac{s-a}{(s-a)^2 + \omega^2}$

## C.04 Fourier transforms

Table C.3 is a table with functions of time  $f(t)$  on the left and corresponding Fourier transforms  $F(\omega)$  on the right. Where applicable,  $T$  is the time-domain period,  $\omega_0 2\pi/T$  is the corresponding angular frequency,  $j = \sqrt{-1}$ ,  $a \in \mathbb{R}^+$ , and  $b, t_0 \in \mathbb{R}$  are constants. Furthermore,  $f_e$  and  $f_o$  are even and odd functions of time, respectively, and it can be shown that any function  $f$  can be written as the sum  $f(t) = f_e(t) + f_o(t)$ . (Hsu, 1967)

**Table C.3:** Fourier transform identities.

function of time $t$	function of frequency $\omega$
$a_1 f_1(t) + a_2 f_2(t)$	$a_1 F_1(\omega) + a_2 F_2(\omega)$
$f(at)$	$\frac{1}{ a } F(\omega/a)$
$f(-t)$	$F(-\omega)$
$f(t - t_0)$	$F(\omega)e^{-j\omega t_0}$
$f(t) \cos \omega_0 t$	$\frac{1}{2} F(\omega - \omega_0) + \frac{1}{2} F(\omega + \omega_0)$
$f(t) \sin \omega_0 t$	$\frac{1}{j2} F(\omega - \omega_0) - \frac{1}{j2} F(\omega + \omega_0)$
$f_e(t)$	$\text{Re } F(\omega)$
$f_o(t)$	$j \text{Im } F(\omega)$
$F(t)$	$2\pi f(-\omega)$
$f'(t)$	$j\omega F(\omega)$
$\frac{d^n f(t)}{dt^n}$	$(j\omega)^n F(\omega)$
$\int_{-\infty}^t f(\tau) d\tau$	$\frac{1}{j\omega} F(\omega) + \pi F(0)\delta(\omega)$
$-jtf(t)$	$F'(\omega)$

$(-jt)^n f(t)$	$\frac{d^n F(\omega)}{d\omega^n}$
$f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau$	$F_1(\omega) F_2(\omega)$
$f_1(t) f_2(t)$	$\frac{1}{2\pi} F_1(\omega) * F_2(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(\alpha) F_2(\omega - \alpha) d\alpha$
$e^{-at} u_s(t)$	$\frac{1}{j\omega + a}$
$e^{-a t }$	$\frac{2a}{a^2 + \omega^2}$
$e^{-at^2}$	$\sqrt{\pi/a} e^{-\omega^2/(4a)}$
1 for $ t  < a/2$ , else 0	$\frac{a \sin(a\omega/2)}{a\omega/2}$
$te^{-at} u_s(t)$	$\frac{1}{(a + j\omega)^2}$
$\frac{t^{n-1}}{(n-1)!} e^{-at})^n u_s(t)$	$\frac{1}{(a + j\omega)^n}$
$\frac{1}{a^2 + t^2}$	$\frac{\pi}{a} e^{-a \omega }$
$\delta(t)$	1
$\delta(t - t_0)$	$e^{-j\omega t_0}$
$u_s(t)$	$\pi\delta(\omega) + \frac{1}{j\omega}$
$u_s(t - t_0)$	$\pi\delta(\omega) + \frac{1}{j\omega} e^{-j\omega t_0}$
1	$2\pi\delta(\omega)$
t	$2\pi j\delta'(\omega)$
$t^n$	$2\pi j^n \frac{d^n \delta(\omega)}{d\omega^n}$
$e^{j\omega_0 t}$	$2\pi\delta(\omega - \omega_0)$
$\cos \omega_0 t$	$\pi\delta(\omega - \omega_0) + \pi\delta(\omega + \omega_0)$

$\sin \omega_0 t$	$-j\pi\delta(\omega - \omega_0) + j\pi\delta(\omega + \omega_0)$
$u_s(t) \cos \omega_0 t$	$\frac{j\omega}{\omega_0^2 - \omega^2} + \frac{\pi}{2}\delta(\omega - \omega_0) + \frac{\pi}{2}\delta(\omega + \omega_0)$
$u_s(t) \sin \omega_0 t$	$\frac{\omega_0}{\omega_0^2 - \omega^2} + \frac{\pi}{2j}\delta(\omega - \omega_0) - \frac{\pi}{2j}\delta(\omega + \omega_0)$
$t u_s(t)$	$j\pi\delta'(\omega) - 1/\omega^2$
$1/t$	$\pi j - 2\pi j u_s(\omega)$
$1/t^n$	$\frac{(-j\omega)^{n-1}}{(n-1)!} (\pi j - 2\pi j u_s(\omega))$
$\operatorname{sgn} t$	$\frac{2}{j\omega}$
$\sum_{n=-\infty}^{\infty} \delta(t - nT)$	$\omega_0 \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0)$

## Bibliography

- A. Agarwal and J. Lang. *Foundations of Analog and Digital Electronic Circuits*. The Morgan Kaufmann Series in Computer Architecture and Design. Elsevier Science, 2005. ISBN 9780080506814. URL <https://books.google.com/books?id=1GgP7FDEv3AC>.
- Tennessee Valley Authority and Tomia. Hydroelectric dam—wikipedia, the free encyclopedia, 2018. URL [https://commons.wikimedia.org/wiki/File:Hydroelectric\\_dam.svg](https://commons.wikimedia.org/wiki/File:Hydroelectric_dam.svg). [Online; accessed 13-February-2018].
- Stefán Baldursson. Bldc motor modelling and control – a matlab®/simulink®implementation. Master’s thesis, Chalmers University, 2005.
- Richard C. Booton and Simon Ramo. The development of systems engineering. *IEEE Transactions on Aerospace and Electronic Systems*, AES-20: 306–9, July 1984.
- William L Brogan. *Modern Control Theory*. Prentice Hall, third edition, 1991.
- Cameron N. Devine and Rico A.R. Picone. Statum. <https://github.com/CameronDevine/Statum>, 2018.
- P Horowitz and W Hill. *The Art of Electronics*. Cambridge University Press, 2015. ISBN 9780521809269. URL <https://books.google.com/books?id=LAIWPwAACAAJ>.
- H.P. Hsu. *Fourier Analysis*. Simon & Schuster, 1967. ISBN 9780671270377. URL <https://books.google.com/books?id=aqYPAgAACAAJ>.

- N.S. Nise. *Control Systems Engineering, 7th Edition.* Wiley, 2015.  
ISBN 9781118800829. URL <https://books.google.com/books?id=BwTYBgAAQBAJ>.
- NOAA. Mothly average precipitation 1951–2008 olympia regional airport—noaa station, August 2017. URL <http://www.co.thurston.wa.us/monitoring/noaa/noaa-home.html>.
- Derek Rowell and David N. Wormley. *System Dynamics: An Introduction.* Prentice Hall, 1997.