Control Systems I
Lecture 2: Modeling and Linearization

Suggested Readings: Åström & Murray Ch. 2-3

Jacopo Tani

Institute for Dynamic Systems and Control
D-MAVT
ETH Zürich

September 28, 2018
## Tentative schedule

<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sept. 21</td>
<td>Introduction, Signals and Systems</td>
</tr>
<tr>
<td>2</td>
<td>Sept. 28</td>
<td><strong>Modeling and Linearization</strong></td>
</tr>
<tr>
<td>3</td>
<td>Oct. 5</td>
<td>Analysis 1: Time response, Stability</td>
</tr>
<tr>
<td>4</td>
<td>Oct. 12</td>
<td>Analysis 2: Diagonalization, Modal coordinates</td>
</tr>
<tr>
<td>5</td>
<td>Oct. 19</td>
<td>Transfer functions 1: Definition and properties</td>
</tr>
<tr>
<td>6</td>
<td>Oct. 26</td>
<td>Transfer functions 2: Poles and Zeros</td>
</tr>
<tr>
<td>7</td>
<td>Nov. 2</td>
<td>Analysis of feedback systems 1: internal stability, root locus</td>
</tr>
<tr>
<td>8</td>
<td>Nov. 9</td>
<td>Frequency response</td>
</tr>
<tr>
<td>9</td>
<td>Nov. 16</td>
<td>Analysis of feedback systems 2: the Nyquist condition</td>
</tr>
<tr>
<td>10</td>
<td>Nov. 23</td>
<td>Specifications for feedback systems</td>
</tr>
<tr>
<td>11</td>
<td>Nov. 30</td>
<td>Loop Shaping</td>
</tr>
<tr>
<td>12</td>
<td>Dec. 7</td>
<td>PID control</td>
</tr>
<tr>
<td>13</td>
<td>Dec. 14</td>
<td>State feedback and Luenberger observers</td>
</tr>
<tr>
<td>14</td>
<td>Dec. 21</td>
<td>On Robustness and Implementation challenges</td>
</tr>
</tbody>
</table>
Today’s learning objectives

After today’s lecture, you should be able to:

- Derive series, parallel and feedback system interconnections.
- Understand the difference between the main control architectures (feedforward, feedback).
- Can use the concept of “state” to model a system.
- Write the model of an LTI system with A, B, C, D matrices.
- Understands the notion of equilibrium points and can calculate them.
- The student is able to linearize a nonlinear system at an appropriately chosen equilibrium point to derive an approximate LTI state space representation.
Outline

1. Recap from previous lecture
2. Feedforward and feedback
3. State-space models
4. Modeling
A signal $u$ is a map from a set, e.g. $\mathbb{T}$, to another another set, e.g. $\mathbb{W}$. In short: $u : \mathbb{T} \to \mathbb{W}$.

A system is a mapping between input and output signals.

Control systems uses block diagrams to represent systems and their interconnections.

We classify the system in different ways:

- Static (memoryless) vs. Dynamic
- Causal vs. Non-causal
- Linear vs. Nonlinear
- Time-invariant vs. Time-varying
An input-output system $\Sigma$ is **memoryless** (or **static**) if for all $t \in \mathbb{T}$, $y(t)$ is a function of $u(t)$.

In other words, in a static system the output at the present time depends only on the value of input at the present time; not on the value of input in the past or the future time.
Causal vs. non-causal systems

- An input-output system $\Sigma$ is causal if, for any $t \in \mathbb{T}$, the output at time $t$ depends only on the values of the input on $(-\infty, t]$. It is strictly causal if the output at time $t$ depends only on values of the input $(-\infty, t)$.

- In other words: a system is causal if and only if the future input does not affect the present output.

- All practically realizable systems are causal. (It is impossible to implement a non causal system in “the real world”.)
Linear vs. nonlinear systems

- An input-output system is **linear** if it is **additive** and **homogeneous**.
  - Additivity: $\Sigma(u_1 + u_2) = \Sigma u_1 + \Sigma u_2$
  - Homogeneity: $\Sigma(ku) = k\Sigma u$.

- In other words, $\Sigma$ is linear if, for all input signals $u_a$, $u_b$, and scalars $\alpha, \beta \in \mathbb{R}$,
  \[
  \Sigma(\alpha u_a + \beta u_b) = \alpha(\Sigma u_a) + \beta(\Sigma u_b) = \alpha y_a + \beta y_b.
  \]

- The key idea is **superposition** property:
Time-invariant vs. time-variant systems

- An input-output system $\Sigma$ is **time-invariant** if $\Sigma \sigma_\tau u = \sigma_\tau \Sigma u = \sigma_\tau y$, $\forall \tau \in \mathbb{T}$, where $\sigma_\tau$ is a shift operator such that $\sigma_\tau u(t) = u(t - \tau)$.

- In other words, the system manipulates the input in a way that does not depend on when the system is used.
Which systems will we treat in this course, and why?

- In this course, we will consider only **LTI SISO** systems:
  - Linear,
  - **Time Invariant,**
  - **Single Input, Single Output,**
  - Causal.

- **This is a very restrictive class of systems**; in fact, most systems are NOT LTI. On the other hand, many systems are approximated very well by LTI models. This is a key idea that we will explore today.

- As long as we are mindful of the errors induced by the LTI approximation, the methods discussed in the class are very powerful.

- Indeed, most control systems in operation are designed according to the principles that will be covered in the course.
Interconnections of systems: series

- Serial interconnection:

\[ \Sigma = \Sigma_2 \Sigma_1 \]
Interconnections of systems: parallel

- Parallel interconnection:

\[ \Sigma = \Sigma_1 + \Sigma_2 \]
Interconnections of systems: (negative) feedback

- (Negative) Feedback interconnection:

\[ \Sigma = (I + \Sigma_1 \Sigma_2)^{-1} \Sigma_1 \]
Outline

1. Recap from previous lecture
2. Feedforward and feedback
3. State-space models
4. Modeling
Control objectives

- **Stabilization**: make sure the system does not “blow up” (i.e., it stays within “normal” operating conditions).

- **Regulation**: maintain a desired operating point in spite of disturbances.

- **Tracking**: follow a reference trajectory that changes over time, as closely as possible.

- **Robustness**: the control system satisfies the above objectives even if the system is “slightly” different than we expected.
The ideal controller, or, how to make a brick wall fly?
The ideal controller, or, how to make a brick wall fly?
The ideal controller, or, how to make a brick wall fly?

Example from Prof. Sandipan Mishra's, "Feedforward and Iterative Learning Control", RPI, 2012.
Basic control architectures: Feed-Forward

- Basic idea: given $r$, attempt to compute what should be the control input $u$ that would make $y = r$.

- Essentially, $F$ should be an “inverse” of $P$.

- Relies on good knowledge of $P$ — sensitive to modeling errors.

- Cannot alter the dynamics of $P$, i.e., cannot make an unstable system stable.
Basic control architectures: Feedback

- Basic idea: given the error $r - y$, compute $u$ such that the error is “small.”
- Intuition: the bigger $C$, the smaller the error $e$ will be, regardless of $P$ (under some assumption, e.g., closed-loop stability).
- Does not require a precise knowledge of $P$ — robust to modeling errors.
- Can stabilize unstable systems. But can also make stable systems unstable (!).
- Needs an “error” to develop in order to figure out the appropriate control.
Basic control architectures: two degrees of freedom

- Basic idea: given $r$, compute a guess for the control input $u$ that would make $y = r$; correct this guess based on the measurement of the error $e$, so that the error is minimized.

- Combines the main advantages of feed-forward and feedback architectures. Ensures stability, robustness, but can also speed up tracking/regulation.
Outline

1 Recap from previous lecture
2 Feedforward and feedback
3 State-space models
4 Modeling
Causality revisited

- An input-output system \( \Sigma \) is causal if, for any \( t_1 \in \mathbb{T} \), the output at time \( t_1 \) depends only on the values of the input on \(( -\infty, t_1 ]\).
State of a system

- We know that, if a system is causal, in order to compute its output at a given time $t_0$, we need to know “only” the input signal over $(-\infty, t_0]$.

- This is a lot of information. Can we summarize it with something more manageable?

**Definition (state)**

The state $x(t_0)$ of a causal system at time $t_0$ is the information needed, together with the input $u$ between times $t_0$ and $t_1$, to uniquely predict the output at time $t_1$, for all $t_1 \geq t_0$.

- Usually, the state of a system is a vector in some Euclidean space $\mathbb{R}^n$. 
The state of the system at a given time $t_0$ summarizes the whole history of the past inputs over $(-\infty, t_0)$, for the purpose of predicting the output at future times.
The state of the system at a given time $t_0$ summarizes the whole history of the past inputs over $(-\infty, t_0)$, for the purpose of predicting the output at future times.
The state of the system at a given time $t_0$ summarizes the whole history of the past inputs over $(-\infty, t_0)$, for the purpose of predicting the output at future times.
Suppose that the input is $u = u_+ - u_- \text{ (i.e., net flow of water into the tank).}$

Is there a way to “summarize” the effect of all past inputs until time $t$?
Dimension of a system

- The choice of a state for a system is not unique (in fact, there are infinite choices, or realizations).

- However, there are some choices of state which are preferable to others; in particular, we can look at “minimal” realizations.

**Definition (Dimension of a system)**

The dimension of a causal system is the minimal number of variables sufficient to describe the system’s state (i.e., the dimension of the smallest state vector).

- We will deal mostly with finite-dimensional systems, i.e., systems which can be described with a finite number of variables.
It can be shown that any finite-dimensional LTI system can be described by a state-space model of the form:

\[
\frac{d}{dt} x(t) = Ax(t) + Bu(t);
\]
\[
y(t) = Cx(t) + Du(t);
\]

- The state \( x \) represents the **memory** of the system.
- The state \( x \) is an “internal” variable that cannot be accessed directly, but only “controlled” though the input \( u \) and “observed” through the output \( y \).
- The **order** of the system is the dimension of the state \( x \).
- A system is **memoryless** if the “dimension” of the state is zero (i.e., there is no need to keep a state, or memory)
- A system is **strictly causal** if the “feedthrough” term \( D \) is zero.
Outline

1. Recap from previous lecture
2. Feedforward and feedback
3. State-space models
4. Modeling
Note that: “All models are wrong”, but some are useful. (depending on the application) – George E. P. Box.

- **Notation:**
  - Flow: \( u = u_+ - u_- \),
  - Water level: \( h \),
  - Tank cross-section area: \( T_{cs} \),

- **Dynamic model:**
  \[
  \frac{d h}{dt} = \frac{1}{T_{cs}} u
  \]

- **State-space model, with \( y = x = h \):**
  \[
  \begin{bmatrix}
  A & B \\
  C & D
  \end{bmatrix}
  =
  \begin{bmatrix}
  0 & \frac{1}{T_{cs}} \\
  1 & 0
  \end{bmatrix}
  \]
Temperature Control (1st order system)

- **Notation:**

- **Dynamic model:**
  \[ m \frac{d T}{dt} = c (T_e - T) + u \]

- **State-space model, with**
  \[ y = x = T - T_e: \]
  \[
  \begin{bmatrix}
  A & B \\
  C & D
  \end{bmatrix} = \begin{bmatrix}
  -\frac{c}{m} & \frac{1}{m} \\
  1 & 0
  \end{bmatrix}
  \]
Nonlinear systems

- Unfortunately, most real-world systems are nonlinear.

- Finite-dimensional, time-invariant, causal nonlinear control systems, with input $u$ and output $y$ can typically be modeled using a set of differential equations as follows:

  \begin{align*}
  \dot{x}(t) &:= \frac{d}{dt} x(t) = f(x(t), u(t)), \\
  y(t) &= g(x(t), u(t));
  \end{align*}

- Can we construct an LTI approximation of this system, so that we can apply the techniques designed for LTI systems?

- Remarkably, control systems designed for the LTI approximation will work very well for the nonlinear system.
Pendulum (2nd-order nonlinear system)

- [https://www.youtube.com/watch?v=oWiuSp6qAPk](https://www.youtube.com/watch?v=oWiuSp6qAPk)
- **Notation:**
  - Angular position: $\theta$
  - Torque input: $u$
  - Pendulum length: $L$
  - Pendulum mass: $m$
  - Viscous damping coefficient: $c$

- **Dynamic model:**

\[
ml^2\ddot{\theta} = -c\dot{\theta} - mgL\sin\theta + u.
\]

Define $x_1 := \theta$, and $x_2 := \dot{\theta}$.

Then one can write

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= -\frac{c}{ml^2}x_2 - \frac{g}{L}\sin(x_1) + u.
\end{align*}
\]
Equilibrium point

- The pendulum is a nonlinear system. How can we construct an approximate model which is linear?

- The key idea is that of constructing an approximation that is valid “near” some special operating condition, e.g., equilibrium points.

**Definition (Equilibrium point)**

A system described by an ODE $\dot{x}(t) = f(x(t), u(t))$ has an equilibrium point $(x_e, u_e)$ if $f(x_e, u_e) = 0$.

- Clearly, if $x(0) = x_e$, and $u(t) = u_e$ for all $t \geq 0$, then

  $$x(t) = x(0) = x_e, \quad \text{for all } t \geq 0.$$
Equilibria for a pendulum

What would be an equilibrium for the pendulum? We need to solve for

\[ \dot{x}_1 = x_2 = 0, \]
\[ \dot{x}_2 = -\frac{c}{mL^2}x_2 - \frac{g}{L} \sin(x_1) + u = 0. \]

There are two solutions:

- \( x_e = (0, 0), \ u_e = 0, \) and

- \( x_e = (\pi, 0), \ u_e = 0. \)

The first equilibrium is stable, the second unstable; we will make these notions more precise.
(Jacobian) Linearization procedure

- Given an equilibrium point \((x_e, u_e)\), with output \(y_e = g(x_e, u_e)\), a linearized model is obtained by setting

\[
\begin{align*}
x & \leftarrow x_e + \delta x, \\
u & \leftarrow u_e + \delta u, \\
y & \leftarrow y_e + \delta y,
\end{align*}
\]

and then neglecting all terms of second (or higher) order in \(\delta x\), \(\delta u\), and \(\delta y\) in the (nonlinear) dynamics model.

- Note that since \(x_e\) is a constant, \(\dot{x} = \delta \dot{x}\).

- The resulting model would be a good approximation of the original nonlinear model when “near” the equilibrium point, i.e., for \(|\delta x| << 1\), \(|\delta u| << 1\), and \(|\delta y| << 1\).
Linearization about the stable equilibrium

- Susbstituting as in the previous slide, with $x_e = (0, 0)$ and $u_e = 0$,

$$\begin{align*}
\delta \dot{x}_1 &= 0 + \delta x_2, \\
\delta \dot{x}_2 &= -\frac{c}{mL^2} (0 + \delta x_2) - \frac{g}{L} (0 + \delta x_1) + \frac{1}{mL^2} (0 + \delta u), \\
\delta y &= \delta x_1.
\end{align*}$$

- The above can be rewritten in the state-space form as

$$\begin{align*}
\delta \dot{x} &= A \delta x + B \delta u, \\
\delta y &= C \delta x + D \delta u,
\end{align*}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ -g/L & -c/(mL^2) \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1/(mL^2) \end{bmatrix},$$

$$C = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad D = 0.$$
Linearization about the unstable equilibrium

- Substituting $x_e = (\pi, 0)$ and $u_e = 0$,

\[
\begin{align*}
\delta \dot{x}_1 &= \pi + \delta x_2, \\
\delta \dot{x}_2 &= -\frac{c}{ml^2}(0 + \delta x_2) - \frac{g}{L}(0 - \delta x_1) + \frac{1}{ml^2}(0 + \delta u), \\
\delta y &= \delta x_1.
\end{align*}
\]

- The above can be rewritten in the state-space form as

\[
\begin{align*}
\delta \dot{x} &= A\delta x + B\delta u, \\
\delta y &= C\delta x + D\delta u,
\end{align*}
\]

where

\[
\begin{align*}
A &= \begin{bmatrix} 0 & 1 \\ +g/L & -c/(ml^2) \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1/(ml^2) \end{bmatrix}, \\
C &= \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0.
\end{align*}
\]
Some remarks on infinite-dimensional systems

Even though we will not address infinite-dimensional systems in detail, some examples are very useful:

- **Time-delay systems**: Consider the very simple time delay $S_T$, defined as a continuous-time system such that its input and outputs are related by

$$y(t) = u(t - T).$$

In order to predict the output at times after $t$, the knowledge of the input for all times in $(t - T, t]$ is necessary.

- **PDE-driven systems**: Many systems in engineering, arising, e.g., in structural control and flow control applications, can only be described exactly using a continuum of state variables (stress, displacement, pressure, temperature, etc.). These are infinite-dimensional systems.

In order to deal with infinite-dimensional systems, approximate discrete models are often used to reduce the dimension of the state.
Today’s learning objectives

After today’s lecture, you should be able to:

- Derive series, parallel and feedback system interconnections.
- Understand the difference between the main control architectures (feedforward, feedback).
- Can use the concept of “state” to model a system.
- Write the model of an LTI system with A, B, C, D matrices.
- Understands the notion of equilibrium points and can calculate them.
- The student is able to linearize a nonlinear system at an appropriately chosen equilibrium point to derive an approximate LTI state space representation.