Advanced Control Systems

Lecture 1: State-Space Modelling

Aidan O. T. Hogg

EECS, Queen Mary University of London

a.hogg@qmul.ac.uk

Spring 2025

Examples of Systems: Growth of a Family of Rabbits

What is a system?

They are everywhere... for example...

The number of pairs of rabbits n months after a single pair begins breeding (and newly born bunnies are assumed to begin breeding when they are two months old) is given by the so-called Fibonacci numbers, which are recursively defined as

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}.$$

Examples of Systems: Fibonacci Numbers

Interestingly, there are many applications of Fibonacci numbers:

- The Fibonacci number F_{n+1} gives the number of ways for 2×1 dominoes to cover a $2 \times n$ checkerboard;
- The Fibonacci number F_{n+2} gives the number of ways of picking a set (including the empty set) from the numbers 1, 2, ... n, without picking two consecutive numbers;
- The probability of not getting two heads in a row in n tosses of a coin is $\frac{F_{n+2}}{2^n}$.
- Given a resistor network of 1Ω resistors, each incrementally connected in series or parallel to the preceding resistors, the net resistance is a rational number having the maximum possible denominator equal to F_{n+1}.

One mathematical model can represent multiple real-world phenomena

Examples of Systems: Model of an Infectious Disease

There are several models which describe the interactions between HIV and immunocytes in the human body. The most common model is

$$\dot{x} = \lambda - dx - \eta\beta xy, \quad \dot{y} = \eta\beta xy - ay - yI.$$

- x and y: Population of uninfected and infected CD4 T-helper cells
- I: Immune system action
- λ, d, β, η and a are positive parameters

Three operating conditions exist:

- Healthy patient.
- HIV infection without AIDS.
- AIDS-dominated state.

The first two operating conditions are **unstable**, which highlights why treating HIV patients is very challenging.

Examples of Systems: Scholastic Population (Graduation)

System Description:

$$
\begin{aligned}
x_1(k+1) &= (1-\alpha_1(k))x_1(k) + u(k),\\
x_2(k+1) &= (1-\alpha_2(k))x_2(k) + \alpha_1(k)x_1(k),\\
x_3(k+1) &= (1-\alpha_3(k))x_3(k) + \alpha_2(k)x_2(k),\\
y(k) &= \alpha_3(k)x_3(k).
\end{aligned}
$$

- $u(k)$ be the number of incoming first-year students at time k
- $y(k)$ be the number of graduated students at time k
- $x_i(k)$ be the number of students in the i-th year at time k
- $\alpha_i(k) \in [0, 1]$ be the rate of promotion in the i-th year at time k

Special Cases:

- **Ideal case:** $\alpha_i(k) = 1, \forall k \implies y(k) = u(k-3)$.
- **Extreme case:** $\alpha_i(k) = 0, \forall k$ and some $i \implies \lim_{k\to\infty} y(k) = 0$.

Examples of Systems: ABS System (Part 1)

Electronic Anti-lock Braking Systems (ABS) maximize friction while maintaining steerability. To model the braking system, the quarter-car model is used

$$J\dot{\omega} = rF_x - T_b, \quad m\dot{v} = -F_x$$

- ω: Angular speed of the wheel.
- v: Longitudinal speed of the vehicle.
- T_b: Braking torque.
- F_x: Longitudinal tire-road contact force.

Examples of Systems: ABS System (Part 2)

The dynamic behaviour is *hidden* in the expression of F_x, which depends on the variables v and ω, and can be approximated as follows

$$F_x = F_z \mu(\lambda, \beta_t, \theta_r)$$

- F_z is the vertical force at the tire-road contact point
- λ is the longitudinal slip, defined as $\lambda = \frac{v - \omega r}{\max\{\omega r, v\}}$
- β_t is the wheel side-slip angle
- θ_r is a set of parameters which characterize the shape of the static function $\mu(\lambda, \beta_t; \theta_r)$ and which depend upon the road conditions

Examples of Systems: Google PageRank Algorithm

Represent the web as a graph with nodes (web pages) and links (hyperlinks) to determine the importance of web pages for search engine results.

Ranking Equation:

$$x_i(k+1) = (1-p)\sum_{j:j\to i}\frac{x_j(k)}{n_j} + p\sum_{j=1}^{N}\frac{x_j(k)}{N}.$$

Matrix Form:

$$x(k+1) = Ax(k).$$

Properties:

- Convergence to a steady-state vector $\bar{x}$.
- $\bar{x}$ is the normalized Google PageRank vector.

Classical Control Methods

Classical control methods: Frequency domain (1940s-1950s)

- Systems represented by transfer functions
- Performance and robustness specifications were either cast directly in or translated into the frequency domain
- Analysis techniques involving root locus plots, Bode plots, Nyquist plots, etc.
- Covered in **ECS601U/ECS788P Control Systems**

Modern State-space Methods

Modern state-space methods: Time domain (1960s-1970s)

- Systems represented in the time domain by a type of differential equation called a *state equation*
- Performance and robustness specifications also were specified in the time domain
- Analysis focuses on linear algebra techniques
- Covered in this module **ECS654U/ECS778P Advanced Control Systems**

Advantages of State-Space Modelling

Why use a state-space representation?

Some advantages:

- More suitable for multiple-input, multiple-output (MIMO) systems, like spacecraft, aircraft, automobiles, marine vessels, etc
- Classical control methods apply to linear (or linearised) systems only
- Computation: computers were better with numbers than with symbolic computation.
  In state-space methods, linear algebra techniques (like rank of a matrix, invariant subspaces, null-space, kernels, etc.) are found to describe various properties of control systems (like controllability, observability, feedback stabilizability, etc.)

State-Space Representation

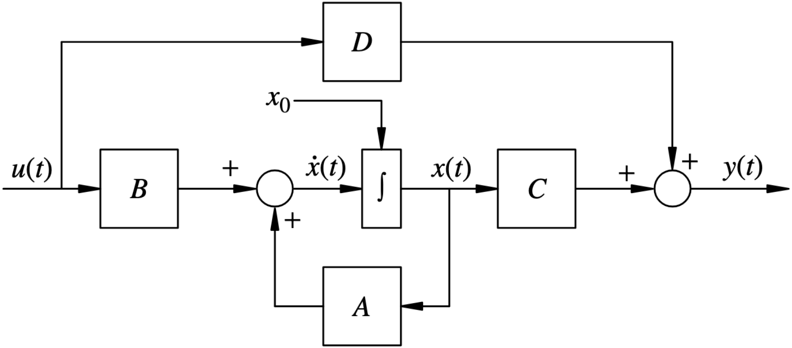
General Form:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

Components:

- $x(t)$: State vector ($n \times 1$).
- $u(t)$: Input vector ($p \times 1$).
- $y(t)$: Output vector ($q \times 1$).
- A, B, C, D: System matrices.

State-Equation Block Diagram



The main motivation for state-space modelling is to convert a coupled system of higher-order ordinary differential equations to a coupled set of first-order differential equations.

Linear Time-Invariant (LTI) Systems

In this module, we will focus on LTI systems,

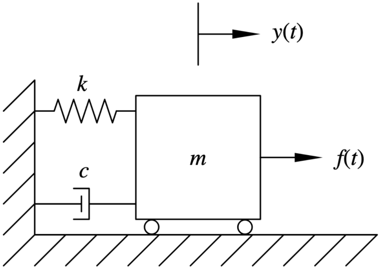
Key Properties:

- **Linearity:** Superposition and scaling principles apply.
- **Time-Invariance:** System dynamics do not change over time.

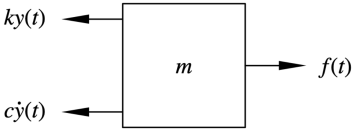
Examples of Applications:

- Control systems in robotics and aerospace.
- Signal processing systems.
- Electrical circuits and mechanical systems.

Example: Mass-Spring-Damper System (Part 1)



Translational mechanical system



Free-body diagram

Example: Mass-Spring-Damper System (Part 2)

Using Newton's second law, the dynamic force balance for the free-body diagram yields the following second-order ordinary differential equation

$$m\ddot{y}(t) + c\dot{y}(t) + ky(t) = f(t)$$

Define the state variables as displacement and velocity

$$\begin{aligned} x_1(t) &= y(t) \\ x_2(t) &= \dot{y}(t) = \dot{x}_1(t) \end{aligned}$$

Therefore

$$\begin{aligned} \dot{y}(t) &= x_2(t) \\ \ddot{y}(t) &= \dot{x}_2(t) \end{aligned}$$

Example: Mass-Spring-Damper System (Part 3)

Substituting these two state definitions into the original system equation gives

$$m\dot{x}_2(t) + cx_2(t) + kx_1(t) = f(t)$$

The original single second-order differential equation can be written as a coupled system of two first-order differential equations

$$\begin{aligned}\dot{x}_1(t) &= x_2(t)\\ \dot{x}_2(t) &= -\frac{c}{m}x_2(t) - \frac{k}{m}x_1(t) + \frac{1}{m}f(t)\end{aligned}$$

Example: Mass-Spring-Damper System (Part 4)

Recall state-space equations

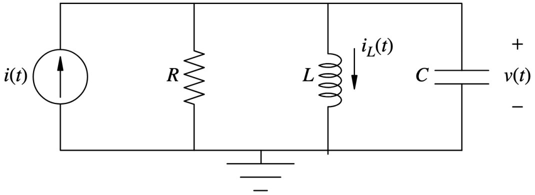
$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

State-space representation of the system

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

Example: Parallel RLC Circuit (Part 1)



Parallel electrical circuit

The input to the system is the current produced by the independent current source $u(t) = i(t)$, and the output is the capacitor voltage $y(t) = v(t)$.

It is often convenient to associate state variables with the energy storage elements in the network, namely, the capacitors and inductors.

Example: Parallel RLC Circuit (Part 2)

Specifically, capacitor voltages and inductor current and, therefore, we choose state variables

$$x_1(t) = i_L(t)$$
$$x_2(t) = v(t)$$

Using the inductor's voltage-current relationship given by

$$x_2(t) = v(t) = L\frac{di_L(t)}{dt} = L\dot{x}_1(t)$$

While applying Kirchhoff's current law produces (recall $i_c(t) = C\frac{v(t)}{dt}$)

$$\frac{1}{R}x_2(t) + x_1(t) + C\dot{x}_2(t) = u(t)$$

Example: Parallel RLC Circuit (Part 3)

These relationships can be rearranged so as to isolate state-variable time derivatives as follows

$$\dot{x}_1(t) = \frac{1}{L} x_2(t)$$
$$\dot{x}_2(t) = -\frac{1}{C} x_1(t) - \frac{1}{RC} x_2(t) + \frac{1}{C} u(t)$$

This pair of coupled first-order differential equations, along with the output definition $y(t) = x_2(t)$, yields the following state-space description for this electrical circuit

$$\dot{x}(t) = \begin{bmatrix} 0 & \frac{1}{L} \\ -\frac{1}{C} & -\frac{1}{RC} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{C} \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) + [0] u(t)$$

Example: Parallel RLC Circuit (Part 4)

Buy inspection, the coefficient matrices A, B, C, and D are found to be

$$A = \begin{bmatrix} 0 & \frac{1}{L} \\ -\frac{1}{C} & -\frac{1}{RC} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{C} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = [0]$$

Note that D $=$ 0 in this example because there is no direct coupling between the current source and the capacitor voltage.

Python Library for State-Space Analysis

Install the Python Control Systems library using

```
pip install control
```

Documentation:

https://web.math.princeton.edu/~cwrowley/python-control

Example:

```
import control as ct
from matplotlib import pyplot as plt
A = [[-1, -2], [3, -4]]
B = [[5], [7]]
C = [[6, 8]]
D = [[9]]
sys = ct.ss(A, B, C, D)
ct.step_response(sys).plot()
plt.show()
```