NA 565 - Fall 2023

State Space Models & Linear Systems

September 6, 2023

UNIVERSITY OF MICHIGAN

- The equation type of

$$\frac{d}{dt}x = \dot{x} = f(x)$$

  that involves derivatives of the dependent variable is called an Ordinary Differential Equation (ODE).

- here x depends on time t, i.e., $x(t)$;

- In mathematics, physics, and engineering, ODEs are extremely useful for modeling both physical and non-physical processes.

State Space Models

- The state of a system is a collection of variables that summarize the past of a system for the purpose of predicting the future.
- The state variables are gathered in a vector $x \in \mathbb{R}^n$ called the *state vector*.
- The control variables are represented in a vector $u \in \mathbb{R}^p$;
- measured signal by the vector $y \in \mathbb{R}^q$

A system can be represented by the differential equation

$$\frac{d}{dt}x = f(x,u)$$

$$y = h(x,u)$$

$$f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n \quad \text{and} \quad h : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^q$$

We call a model of this form a state space model.

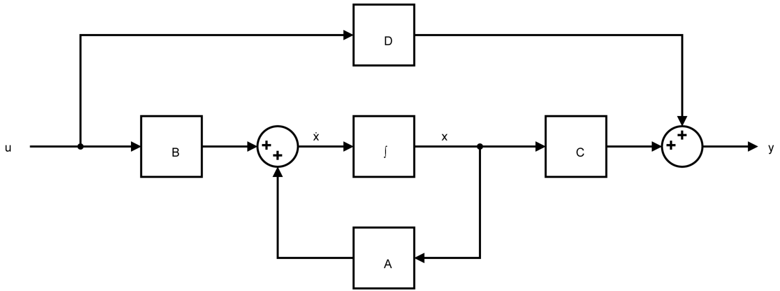
A *linear* and time-invariant system, or LTI, can be represented by the differential equation

$$\frac{d}{dt}x = Ax + Bu$$
$$y = Cx + Du$$

A (dynamics or system matrix), B (control matrix), C (sensor matrix), and D (direct sum) are constant matrices.

Often, $D = 0$; indicating that the control signal u does not influence the output directly.

Block Diagram Representation



Linear vs. Nonlinear

Remark (Existence and uniqueness theorem)

Picard–Lindelöf theorem.

- When the power series of the function $f(x) = \exp(x)$ is applied to a real $n \times n$ matrix A, the result is called *matrix exponentiation*:

$$\exp(A) = e^A = I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \frac{1}{4!}A^4 + \cdots .$$

- This series converges for all real $n \times n$ matrices.

Theorem

The convolution equation gives the solution to the linear differential equation

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau.$$

Theorem

The convolution equation gives the solution to the linear differential equation

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau.$$

Proof.

Using the Leibniz integral rule, we have

$$\frac{d}{dt}x = Ae^{At}x(0) + \int_0^t Ae^{A(t-\tau)}Bu(\tau)d\tau + Bu(t) = Ax + Bu.$$

$\square$

Assuming a constant timestep T and a zero-order hold on the input $u(t)$ during T, we have

$$\begin{aligned} x_{t_{k+1}} :=\ & x\left((k+1)T\right) = \\ &= e^{AT}x(kT) + \left(\int_0^T e^{A\nu}d\nu\right)Bu(kT) \\ &=: e^{AT}x_{t_k} + \left(\int_0^T e^{A\nu}d\nu\right)Bu_{t_k} \\ &=: A_d x_{t_k} + B_d u_{t_k}. \end{aligned}$$

$$\boxed{A_d = e^{AT}} \quad \textbf{and} \quad \boxed{B_d = \left(\int_0^T e^{A\nu}d\nu\right)B}.$$

Common notations are (all equivalent and acceptable):

$$
\begin{aligned}
x_{t_{k+1}} &= A_d x_{t_k} + B_d u_{t_k} \\
x_{k+1} &= A_d x_k + B_d u_k \\
x_{t+1} &= A_d x_t + B_d u_t
\end{aligned}
$$

$$t_k, k, t = 0, 1, 2, \ldots$$

The simplest discretization of an ODE is known as the Euler method.

- Euler method: At timestep k, use a forward-difference approximation $\dot{x} \approx \frac{x_{k+1}-x_k}{T}$,

$$x_{k+1} = (I + TA)x_k + TBu_k.$$

- Works for nonlinear ODEs as well,

$$x_{k+1} = x_k + Tf(x_k, u_k).$$

Notation $h = T$ as timestep (or step size) is also very common ($x_{k+1} = x_k + hf(x_k, u_k)$).

We can linearize nonlinear models around an operating point to obtain a linear model. Note that this model is only locally valid.

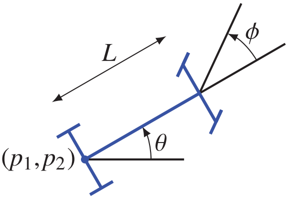
Linearization of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ around point x_0 is

$$
\begin{aligned}
f(x) \approx f(x_0) + \left.\frac{\partial f}{\partial x}\right|_{x=x_0} (x - x_0) &= f(x_0) + \underbrace{J(x_0)}_{\text{Jacobian}} (x - x_0) \\
&= (f(x_0) - J(x_0)x_0) + J(x_0)x \\
&=: a + Fx,
\end{aligned}
$$

which is an affine function.

$$J(x) := \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m\times n}$$

Simple model of a car



$$
\begin{aligned}
\frac{dp_1}{dt} &= s(t)\cos\theta(t) \\
\frac{dp_2}{dt} &= s(t)\sin\theta(t) \\
\frac{d\theta}{dt} &= \frac{s(t)}{L}\tan\phi(t)
\end{aligned}
$$

- $s(t)$ is speed of vehicle, $\phi(t)$ is steering angle
- $p(t)$ is position, $\theta(t)$ is orientation

Discretized model

- discretized model (for small time interval h):

$$
\begin{aligned}
p_1(t+h) &\approx p_1(t) + hs(t)\cos(\theta(t)) \\
p_2(t+h) &\approx p_2(t) + hs(t)\sin(\theta(t)) \\
\theta(t+h) &\approx \theta(t) + h\frac{s(t)}{L}\tan(\phi(t))
\end{aligned}
$$

- define input vector $u_k = (s(kh), \phi(kh))$
- define state vector $x_k = (p_1(kh), p_2(kh), \theta(kh))$
- discretized model is $x_{k+1} = f(x_k, u_k)$ with

$$
f(x_k, u_k) = \begin{bmatrix} (x_k)_1 + h(u_k)_1\cos((x_k)_3) \\ (x_k)_2 + h(u_k)_1\sin((x_k)_3) \\ (x_k)_3 + h(u_k)_1\tan((u_k)_2)/L \end{bmatrix}
$$

Control problem

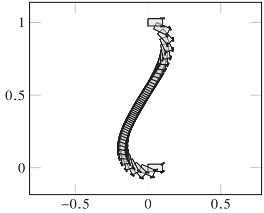
- move car from given initial to desired final position and orientation
- using a small and slowly varying input sequence
- this is a constrained nonlinear least squares problem:

$$
\begin{array}{ll}
\text{minimize} & \sum_{k=1}^{N} \|u_k\|^2 + \gamma \sum_{k=1}^{N-1} \|u_{k+1} - u_k\|^2 \\
\text{subject to} & x_2 = f(0, u_1) \\
& x_{k+1} = f(x_k, u_k), \quad k = 2, \ldots, N-1 \\
& x_{\text{final}} = f(x_N, u_N)
\end{array}
$$

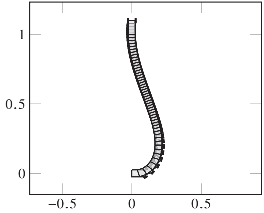
- variables are $u_1, \ldots, u_N$, $x_2, \ldots, x_N$

Four solution trajectories

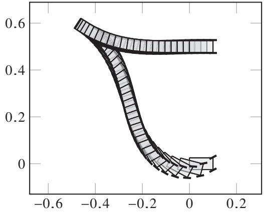
$x_{\text{final}} = (0, 1, 0)$



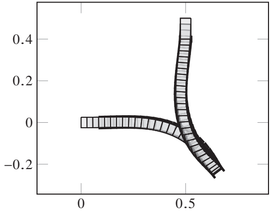
$x_{\text{final}} = (0, 1, \pi/2)$



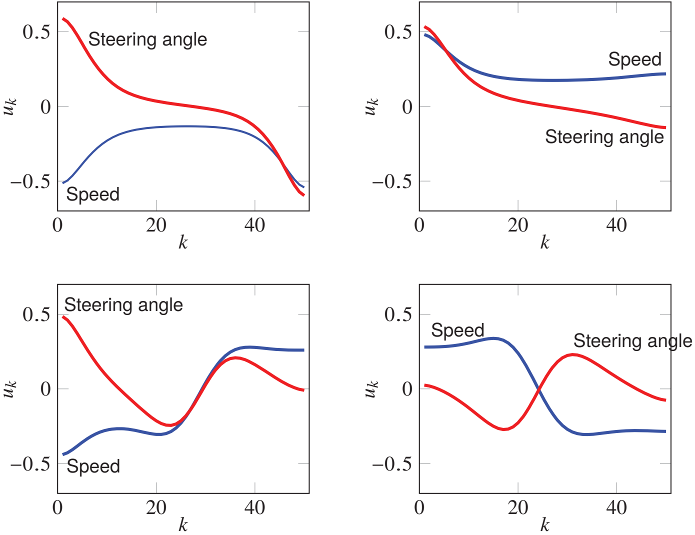
$x_{\text{final}} = (0, 0.5, 0)$

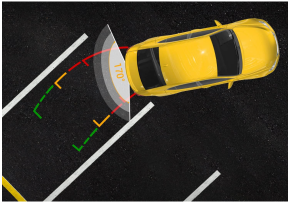


$x_{\text{final}} = (0.5, 0.5, -\pi/2)$



Inputs for four trajectories





- HW0 on Canvas; Submission via Gradescope.
- Ask your questions on Piazza; Due on Sept. 15.
- You'll
  - study qualitative behavior of linear (or locally linearized) dynamical systems using eigenvalues of the system matrix.
  - learn how to *simulate* linear and nonlinear dynamical systems such as a simple car model.
  - play with built-in ode solvers (integrators; RK4) and your own first-order forward Euler solver.

Next Time

- Constrained Optimization