ASEN 5044, Fall 2018

Statistical Estimation for Dynamical Systems

Lecture 4: Time-domain Solutions for LTI Systems: Matrix Exponential and Properties

Prof. Nisar Ahmed (Nisar.Ahmed@Colorado.edu)

Thurs 9/6/2018

University of Colorado Boulder

Announcements

- **HW 1 Posted: Due Thurs 9/13 at 11 am (before start of next lecture)**
- **Submit to Canvas –**
  - **All submissions must be legible!!! – zero credit otherwise**
  - **All submissions must have your name on them!!! – zero credit otherwise**

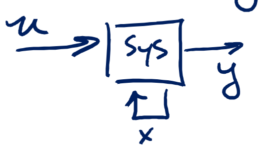
- Advanced Questions:
  - required for PhD students
  - optional/extra credit for everyone else

Overview

- Last time: State Space (SS) Models
  - motivation, examples
  - →**(A,B,C,D) matrices for linear time invariant (LTI) systems**

$$\dot{x}(t) = Ax(t) + Bu(t), \quad A \in \mathbb{R}^{n \times n}, \quad B^{n \times m}, \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m$$

$$y(t) = Cx(t) + Du(t), \quad C \in \mathbb{R}^{p \times n}, \quad D \in \mathbb{R}^{p \times m}, \quad y \in \mathbb{R}^p$$



How to actually solve? what $x(t) = ?$

[as fxn of time, given some $x(t=0)$]

- **Today: Matrix exponential as solution to LTI matrix vector IVP**

  READ: Chapter 1.3-1.4 in Simon book

Matrix-Vector Initial Value Problems (IVPs)

- **Given SS model for an LTI system (i.e. given its [A,B,C,D] parameters), how do we solve for x(t)?** vector sol'n!
- Suppose x(0) given, u(t) =0 (no external forcing) and we ignore output y(t)
- Left with a matrix-vector ODE, i.e. a system of linear ODEs with initial conditions

$$\dot{x}(t) = A x(t), \quad x(0) = \begin{bmatrix} x_1(0) \\ \vdots \\ x_n(0) \end{bmatrix} \longleftrightarrow \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} x_1(t) + \dots + a_{1n} x_n(t) \\ \vdots \\ a_{n1} x_1(t) + \dots + a_{nn} x_n(t) \end{bmatrix}$$

→ So what is sol'n for $x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix}$?

→ Generally speaking: We need a State transition matrix (STM)

$$x(t) = \Phi(t, 0) \cdot x(0) \quad \text{(or)} \quad x(t) = \Phi(t, t_0) \cdot x(t_0)$$

where the STM $= \Phi(t, t_0) \in \mathbb{R}^{n \times n}$ such that

$$\frac{d}{dt}[x(t)] = \frac{d}{dt}[\Phi(t, t_0)\, x(t_0)] = A x(t) \text{ w/ IC } \Phi(t_0, t_0) = I$$

Matrix-Vector Initial Value Problems (IVPs)

- If we plug the STM into the original matrix-vector ODE, we get:

$$\dot{x}(t) = \frac{d}{dt}\left[\Phi(t,t_0)\, x(t_0)\right] = \dot{\Phi}(t,t_0)\, x(t_0) + \Phi(t,t_0)\cdot \overbrace{\frac{d}{dt}\cancel{\left[x(t_0)\right]}}^{0} \quad (*)$$

But also have: $\dot{x}(t) = A\,x(t) = A\left[\Phi(t,t_0)\cdot x(t_0)\right] \quad (**)$

$\longrightarrow$ Equate RHS of $(*)$ & $(**)$ to each other:

$$\dot{\Phi}(t,t_0)\, x(t_0) = A\, \Phi(t,t_0)\, x(t_0)$$

So we need to solve this matrix ODE to find $x(t)$! $\Longrightarrow$

$$\boxed{\underbrace{\dot{\Phi}(t,t_0)}_{[n\times n]} = \underbrace{A}_{[n\times n]}\, \underbrace{\Phi(t,t_0)}_{[n\times n]}, \quad \text{w/ IC } \Phi(t_0,t_0) = \underbrace{I}_{[n\times n]}}$$

how?

$\longrightarrow$ Consider $n=1$ (simplest): scalar $A = a$

$\longrightarrow \dot{\Phi}(t,t_0) = a\,\Phi(t,t_0)$, $\Phi(t_0,t_0) = 1$

$\longrightarrow$ clearly: $\Phi(t,t_0) = e^{a(t-t_0)}$ $\longrightarrow$ STM is exp fxn in a & $t-t_0$

what about for $n \geq 1$

The STM for LTI Systems: the Matrix Exponential

- Remarkably, the STM for <u>any</u> square <u>LTI</u> matrix A is given by the **matrix exponential**

$$\Phi(t,t_0) = e^{A(t-t_0)} \in \mathbb{R}^{n\times n} \iff \Phi(t,0) = e^{At}$$

where matrix exponential is <u>defined</u> as the infinite series:

$$e^{A(t-t_0)} \triangleq I + A(t-t_0) + A^2\frac{(t-t_0)^2}{2!} + A^3\frac{(t-t_0)^3}{3!} + \dots + A^r\frac{(t-t_0)^r}{r!} + \dots$$

$$= \sum_{i=0}^{\infty} A^i \frac{(t-t_0)^i}{i!} \quad (\text{where } A^i = A\cdot A\cdot A\cdots A = \text{product of } A \text{ } i \text{ times})$$

this provably converges for <u>any</u> $A \in \mathbb{R}^{n\times n}$

→ <u>can easily verify that</u>:

$$\frac{d}{dt}\left[e^{A(t-t_0)}\right] = A + A^2(t-t_0) + A^3\frac{(t-t_0)^2}{2!} + \dots$$

$$= A\left[I + A(t-t_0) + A^2\frac{(t-t_0)^2}{2!} + \dots\right]$$

$$\stackrel{\checkmark}{=} A\,e^{A(t-t_0)}$$

Properties of the Matrix Exponential

The matrix exponential function of matrix M is generally defined as:

$$\underset{[n\times n]}{e^{M}} \triangleq \sum_{i=0}^{\infty} \frac{M^i}{i!} = I + M + \frac{M^2}{2!} + \frac{M^3}{3!} + \cdots$$

for $M \in \mathbb{R}^{n\times n}$

This maps an arbitrary (n x n) matrix M to another (n x n) matrix.

Matrix exponential has following useful properties:

- Always invertible, even if M itself is singular

i.e. $(e^{M})^{-1}$ always exists: $(e^{M})^{-1} = e^{-M}$ s.t. $(e^{M})(e^{-M}) = (e^{-M})(e^{M}) = I$

- Product of two matrix exponentials commutes <u>iff</u> input matrices commute

(if & only if)

$\Phi \in \mathbb{R}^{n\times n}$

$\Psi \in \mathbb{R}^{n\times n}$

$\longrightarrow e^{\Phi} \cdot e^{\Psi} = e^{\Phi+\Psi} = e^{\Psi+\Phi} \underset{\text{if \& only if}}{\Longleftrightarrow} \Phi\Psi = \Psi\Phi$

Computing the Matrix Exponential/STM

The matrix exponential is the STM for LTI state space models:

$$\dot{x} = A\,x, \quad x(t_0) = x_0 \quad \longrightarrow \quad x(t) = \Phi(t, t_0)\cdot x(t_0)$$

$$\text{where } \Phi(t, t_0) = e^{A(t-t_0)} = \sum_{i=0}^{\infty} A^i \frac{(t-t_0)^i}{i!}$$

this means $\sum_{i=0}^{\infty} \rightarrow \sum_{i=0}^{n-1}$!

The STM is extremely useful for doing computer simulations of LTI systems --

but how to actually compute an infinite series of matrix powers?

- Brute force: truncated series, or lucky properties of matrix
- Eigenvalue decomposition
- Laplace transforms
- Cayley-Hamilton theorem :
- **Matlab: "expm"** command

for any $A \in \mathbb{R}^{n\times n}$: $|A - \lambda I| = \lambda^n + c_{n-1}\lambda^{n-1} + c_{n-2}\lambda^{n-2} + \dots + c_1\lambda^1 + c_0 = 0$

= characteristic equation for A

C-H Theorem says:

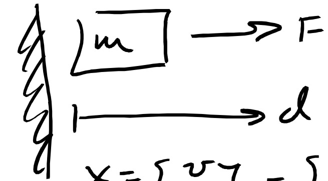
$$A^n + c_{n-1}A^{n-1} + c_{n-2}A^{n-2} + \dots + c_1 A^1 + c_0 \cdot I = 0$$

(ie A satisfies its own char. eq!)

$$\Rightarrow A^n = -\left(c_{n-1}A^{n-1} + c_{n-2}A^{n-2} + \dots + c_1 A^1 + c_0 I\right)$$

Example: STM Computation for 1D Mass System

- Recall: state space model for displacement d(t) vs. force F(t)



$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \longrightarrow$ So what $\Phi(t, t_0) = e^{A(t-t_0)} = ?$

$x = \begin{bmatrix} v \\ d \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$\dot{x} = \begin{bmatrix} \dot{v} \\ \dot{d} \end{bmatrix} = Ax \quad (F=0)$

$\longrightarrow$ suppose $t - t_0 \stackrel{\Delta}{=} \Delta t$ for some constant t_0

$\longrightarrow \Phi(t, t_0) = e^{A\Delta t} = \sum_{i=0}^{\infty} A^i \frac{(\Delta t)^i}{i!} \longrightarrow$

we know that:

$i=0: \quad A^0 = I$

$i=1: \quad A^1 = A$

$\longrightarrow$ for $i=2$: $A^2 : A \cdot A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

for $i=3$: $A^3 = A \cdot A \cdot A = A \cdot A^2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

$\Longrightarrow A^i = 0$ for all $i \geq 2$!

$\longrightarrow$ So $\Phi(t, t_0) = e^{A\Delta t} = \frac{A^0 (\Delta t)^0}{0!} + \frac{A^1 (\Delta t)^1}{1!} = \boxed{I + A\Delta t = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} = \Phi(t, t_0)}$

So:

$x(t) = x(t_0 + \Delta t)$

$= \Phi(t, t_0) \cdot x(t_0)$

$= \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix}\begin{bmatrix} v(t_0) \\ d(t_0) \end{bmatrix}$

$= \begin{bmatrix} v(t_0) \\ v(t_0)\Delta t + d(t_0) \end{bmatrix}$

General Solution to Forced LTI Matrix-Vector IVPs

- Recall: General LTI state space model with inputs given by

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x(t_0), \quad u(t) \neq 0 \text{ for } t \geq 0$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times n}$ & $B \in \mathbb{R}^{n \times m}$

- If non-zero u(t) for some initial condition x(0), then

$$\circledast \quad \boxed{x(t) = \underbrace{e^{A(t-t_0)} \cdot x(t_0)}_{\substack{\text{unforced response} \\ \text{(ie "free response") due IC's}}} + \underbrace{\int_{t_0}^{t} \underset{[n \times n]}{e^{A(t-\tau)}} \cdot \underset{[n \times m]}{B} \underset{[m \times 1]}{u(\tau)} \, d\tau}_{\substack{\text{forced response} \\ \text{[convolution integral}}}}$$

⊛ lsim fxn in matlab

Choice and Transformation of State Representations

- The [A,B,C,D] matrices are not unique for ~~that~~ given set of linear ODEs
- Infinitely many possible [A,B,C,D] -- governed by choice of state x
- These choices are all related by invertible similarity transformations
- Example for 1D mass system again: $x = \begin{bmatrix} v \\ d \end{bmatrix}$, $A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$, $\dot{x} = Ax$

→ Suppose we want to change basis to get new state vector!

$\tilde{x} = \begin{bmatrix} d \\ v \end{bmatrix} \longrightarrow$ clearly, $\underset{[\text{new state}]}{\tilde{x}} = Tx$, where $T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

[invertible similarity transform]

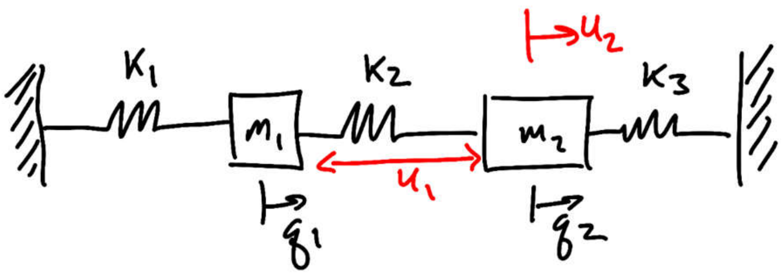
→ So $\dot{\tilde{x}} = \frac{d}{dt}(Tx) = T\dot{x} + \cancel{\frac{d}{dt}\{T\}}^{\,0} x$

$= T(Ax) \Longrightarrow \dot{\tilde{x}} = TAx$

But since $\tilde{x} = Tx \rightarrow x = T^{-1}\tilde{x} \Longrightarrow \dot{\tilde{x}} = \underbrace{TA(T^{-1})}_{= \tilde{A}}\tilde{x} \rightarrow \boxed{\dot{\tilde{x}} = \tilde{A}\tilde{x}}$

Linear vs. Nonlinear System Models

- Linear dynamics/ODEs = good approx. for many physical laws, but not all!
- Example: 2 mass / 3 spring system
- Physical springs and actuators always have nonlinear behavior – but sometimes we can ignore these for *a priori*/first principles models in control/estimation



$$x = [q_1(t), \dot{q}_1(t), q_2(t), \dot{q}_2(t)]^T$$

$$u = [u_1(t), u_2(t)]^T$$

$$y = [q_1(t), q_2(t)]^T$$

u_1 = relative actuator;
u_2 = absolute actuator

- For $k_1 = k_2 = k_3 = 1$ N/m and $m_1 = m_2 = 1$ kg, use simple physics to get LTI SS model

How would this all change/look if we accounted for nonlinearities?

$$\dot{x} = Ax(t) + Bu(t) \qquad A = \begin{bmatrix} 0 & 1.0 & 0 & 0 \\ -2.0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \\ 1.0 & 0 & -2.0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

$$y(t) = Cx(t) + Du(t) \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$