Data-Driven Modeling and Control

Lecture 1

Dr. Shima Nazari

Class Syllabus

Meeting Times: MW 3:10 – 5:00 PM
Location: Olson 105

Instructor: Shima Nazari (snazari@ucdavis.edu)

Office hours: Monday 5:00-6:00 PM, 1001 Academic Surge Building

Recommended Readings:

1- “Data-Driven Science and Engineering Machine Learning, Dynamical Systems, and Control” by Steven Brunton and Nathan Kutz

2- “Robust Adaptive Control” by Petros Ioannou and Jing Sun

Course description:

This advanced level course will cover the theoretical and practical aspects of data-driven methods for dynamic system modeling and control. We will cover the general philosophy of data-driven approaches and introduce useful tools for each method. The intended data-driven modeling approaches encompass a collection of methods such as dynamic model decomposition, Koopman operator theory, system identification, and online parameter estimation. We will further address classical adaptive schemes such as model reference adaptive system and self-tuning regulators. Finally, we will briefly discuss machine learning methods for control, extreme seeking algorithms and other miscellaneous learning methods used in control systems development. Matlab/Simulink® will be used for numerical simulations.

Pre-requisite:

- Exposure to dynamic system modeling,
- state space models,
- linear control theory.
- Experience with MATLAB and Simulink.
- Familiarity with differential equations.

Grading:

Homework: 30%

Midterm Exam: 30%

Final Project: 40%

- Homework and project require use of Matlab and its toolboxes. Access to Matlab is required.
- Discussion on homework and collaboration on projects are encouraged.
- Every student needs to turn in his/her own homework to get the homework credit. Copied homework is a violation of Honor Code and it will not be accepted.
- Homework submission will be online and through Canvas.
- Late homework will be accepted for partial credit (20% off for each day), BEFORE the solutions are posted. No late homework will be accepted after the homework solutions are posted.
- The grading for the project is based on the project proposal (10%), presentation (40%), and final report (50%).
- There will be only one exam late midterm on 02/28/2024.
- **You can ask Homework/Project related questions in office hours.**

Class project:

The projects are to be carried out by self-organized teams (no more than three students on one team). The projects may focus on any concepts or techniques learned in this class. Comprehensive literature search or in-depth analysis and reproduction of the results of one technical paper is also acceptable. Extensive numerical simulations and design sensitivity analysis for examples found from books, journal papers are also good topics. Project proposals are due January 31st (Wednesday). Presentations will be scheduled towards the end of the quarter, and the presentations will be peer-graded.
All reports should be handed in by 5:00pm on March 13th, 2024 (last day of class).

Suggested format and content of the written project:

1. Introduction and problem statement (background of the problem, application motivation, new contribution if any)
2. Design model
3. Design approach
4. Design results
5. Conclusion
6. References
7. Attachment: Computer Simulation Code (if used)

Important dates:

01/31 (Wednesday): Project team formed and proposal due.
02/28 (Wednesday): Midterm Exam.
03/14 (Wednesday): Final report due.

Course Outline:

- Introductions (1 lecture)
- Singular Value Decomposition (1 lecture)
- Dynamic Model Decomposition (1 Lecture)
- Sparse Identification of Nonlinear Dynamics (1 lecture)
- Koopman Operator Theory (3 lectures)
- Balanced Model Reduction (1 lecture)
- System Identification (1 lecture)
- Online Parameters Estimation (2 lectures)
- Adaptive Control (2 lectures)
- Machine Learning for Control (1 Lectures)
- Adaptive Extremum-Seeking Methods (1 Lectures)

UCDAVIS CANVAS

Account
Dashboard
Courses
Calendar
Inbox 52
UCD Library
History
Commons
Help 10

MAE 298 008 WQ 2024 › Syllabus

Student View

Winter Quarter 2024

Home
Announcements
Assignments
Discussions
Grades
People
Pages
Files
Syllabus
Outcomes
Rubrics
Quizzes
Modules
BigBlueButton
Collaborations
Attendance
Chat
New Analytics
Media Gallery

MAE 298 008 WQ 2024

Jump to Today | Edit

Data Driven Modeling and Control

Department of Mechanical and Aerospace Engineering

University of California Davis

Winter 2024

Class Information:

Class Time: 3:10 – 5:00 pm MW

Class Location: Olson 105

Instructor:

Professor Shima Nazari

Office: 1001 Academic Surge Building

Email: snazari@ucdavis.edu

Office hours: Monday 5:00-6:00 pm in 1001 Academic Surge

Textbooks:

- "Data-Driven Science and Engineering Machine Learning, Dynamical Systems, and Control" by Steven

Course Status

Unpublish | Published

Import Existing Content
Import from Commons
Choose Home Page
View Course Stream
Course Setup Checklist
New Announcement
New Analytics
View Course Notifications

January 2024

31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Course assignments are not weighted.

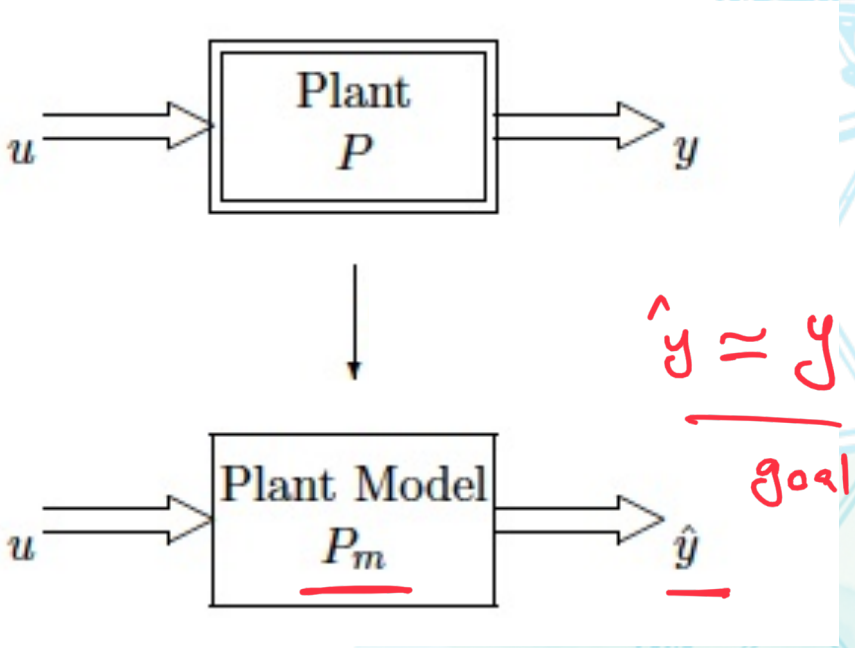
System Modeling

Goal:

- predict the system output for the given inpu

Why?

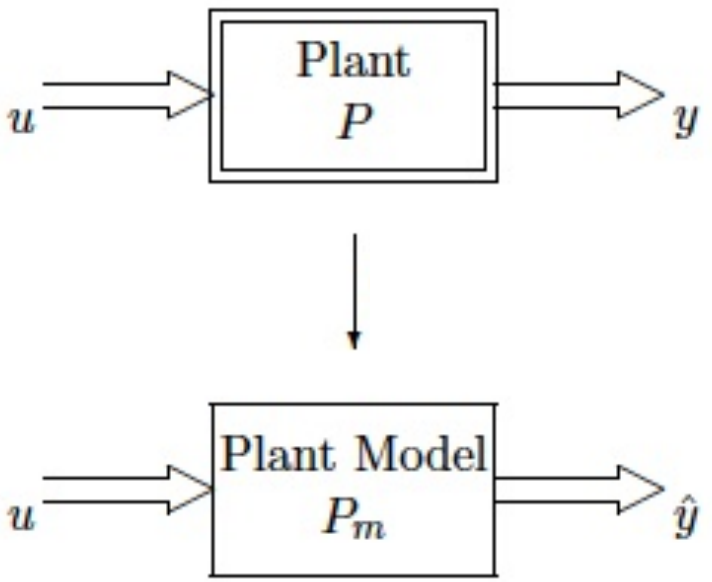
- Better design
- Predict future
- Analysis and understanding of systems
- Control and estimation
- ⋮



System Modeling

White-box modeling: To develop models from physical principles and system parameters,

- Conservation of mass
- Conservation of momentum
- Conservation of energy
- Electromagnetism
- …



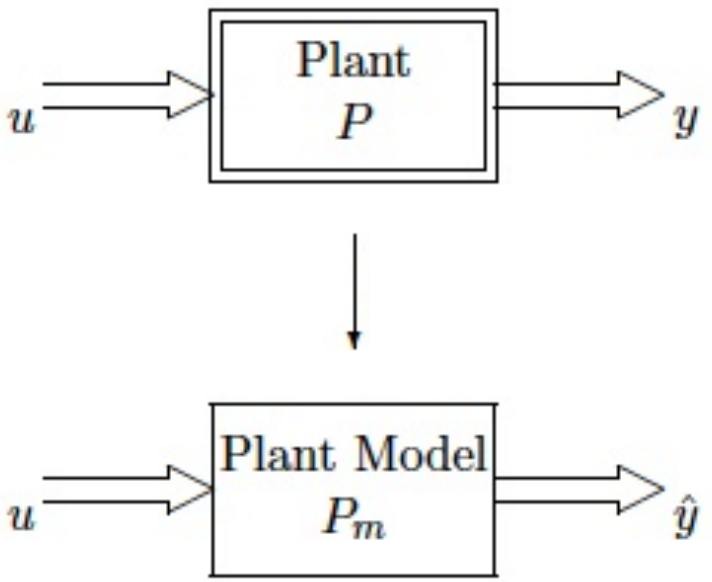
What can go wrong?

- Unknown and complex physics
- Difficult to find system parameters
- **Examples:** predicting climate change, understanding cognition from neural recordings, predicting and suppressing the spread of disease

System Modeling

Why data-driven models:

- Data available from sensors simulations
- Increased storage and computational power of the underlying hardware and communication infrastructure



Data-Driven Control

- Uses data to find control law or improve the control law
- A collection of methods:
  - Reinforcement learning
  - Iterative learning control
  - Extreme seeking methods
  - Using adaptive laws
  - …

Dynamical systems: continuous

General **nonlinear** state-space system:

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

$\dot{x} = f(x, u, t, \beta)$

$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t, \boldsymbol{\beta})$$

$y = g(x, u, t, \beta)$

$\longrightarrow$ short notation

$\boldsymbol{x}$: state of the system (a vector) $x \in R^n$

$\boldsymbol{y}$: vector of outputs $y = R^p$

$\boldsymbol{f}$: vector field of functions

$\boldsymbol{u}$: vector of inputs $u \in R^m$

t: time

$\boldsymbol{\beta}$: set of parameters (hyper parameters)

— time variant system since f and g depend on t

Dynamical systems: continuous

Nonlinear **time-invariant** state-space system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\beta})$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\beta})$$

* x, y, u change with time.

no explicit dependency of f and g on time.

Nonlinear time-invariant **autonomous** state-space system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\beta})$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\beta})$$

$\dot{x} = f(x)$

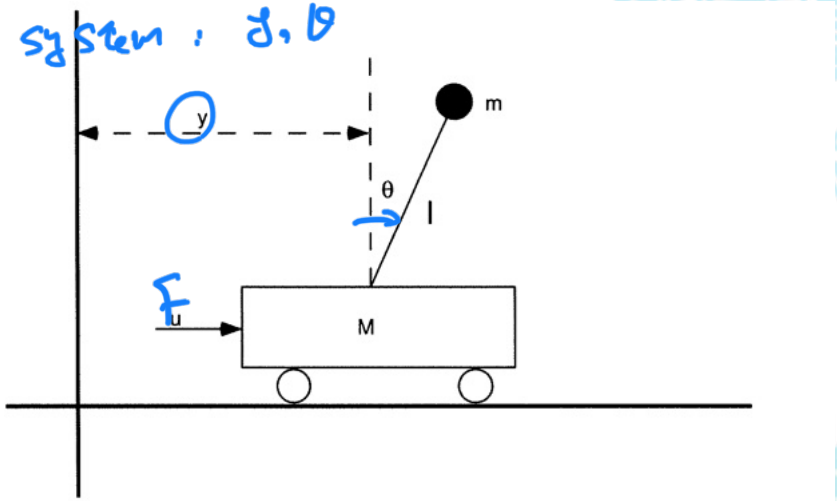
$y = g(x)$

* system with no input

Dynamical systems: example

Inverted pendulum on a Cart:

2 degree of freedom system : y, θ



state : $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix}$

system input : $u = F$

EoMs:

$$\begin{cases} (M+m)\ddot{y} + (ml\cos\theta)\ddot{\theta} - ml\dot{\theta}^2\sin\theta = F \\ ml\ddot{\theta} + ml\cos\theta\,\ddot{y} - mgl\sin\theta = 0 \end{cases}$$

time invariant

Nonlinear system

what is F : you need to decouple the EoMs to find F.

Linearization

DEF of equilibrium.

- Local linearization about an equilibrium point $(x^{eq}, u^{eq}), f(x^{eq}, u^{eq}) = 0$

$\dot{x} = f(x, u)$

$\dot{y} = g(x, u)$

Taylor

$$\begin{cases} \delta x = x - x^{eq} \\ \delta y = y - y^{eq} \\ \delta u = u - u^{eq} \end{cases}$$

Taylor series expansion:

$$\delta \dot{x} = \underbrace{\frac{\partial f}{\partial x}(x^{eq}, u^{eq})}_{A} \delta x + \underbrace{\frac{\partial f}{\partial u}(x^{eq}, u^{eq})}_{B} \delta u + H.O.T$$

$$\delta \dot{x} = A \delta x + B \delta u$$

$$\dot{x} = A x + B u$$

x: deviation of system state from equilibrium state.

- Local linearization about a trajectory

$\dot{x} = A(t) x + B(t) u \rightarrow$ time varying linear system.

* please review the revelant linearization concepts.

Dynamical systems: Discrete

General **nonlinear** state-space system:

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k, k, \boldsymbol{\beta}) \\ \boldsymbol{y}_k = \boldsymbol{G}(\boldsymbol{x}_k, \boldsymbol{u}_k, k, \boldsymbol{\beta}) \end{cases}$$

where $\boldsymbol{x}_k = \boldsymbol{x}(k\Delta t)$

Δt: sample time

* The discrete models should be used in digital control and when working with sampled data.

Nonlinear **time-invariant** state-space system:

$$\left.\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\beta}) \\ \boldsymbol{y}_k &= \boldsymbol{G}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\beta}) \end{aligned}\right\}$$

F & G not dependent K

Autonomous: $x_{k+1} = F(x_k)$

$y_k = G(x_k)$

Converting continuous systems to discrete systems

Continuous system: $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t, \boldsymbol{\beta})$

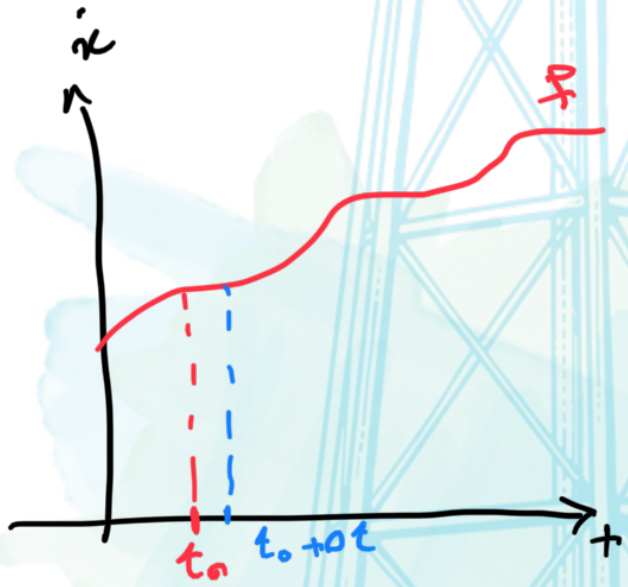
How to find F using f ?

Discrete system: $\boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k, k, \boldsymbol{\beta})$

Accurate conversion: $\boxed{\boldsymbol{F}(\boldsymbol{x}(t_0)) = \boldsymbol{x}(t_0) + \int_{t_0}^{t_0+t} \boldsymbol{f}(\boldsymbol{x}(\tau))d\tau}$

$$x(t_0 + \Delta t) = x(t_0) + \int_{t_0}^{t_0+\Delta t} f(x(\tau))\,d\tau$$

$$\underbrace{x(t_0 + \Delta t)}_{x_{k+1}} = F(x(t_0)) = F(x_k)$$



* dropped dependency on u, t, β in notation.

Converting continuous systems to discrete systems

Approximations:

Forward Euler discretization

$$\dot{\boldsymbol{x}}(t) = \frac{\boldsymbol{x}(t+\Delta t) - \boldsymbol{x}(t)}{\Delta t}$$

let $t = k\Delta t$

$$\dot{x}(t) = \frac{x_{k+1} - x_k}{\Delta t} \quad \text{I}$$

$$\begin{cases} x(t) \rightarrow x_k \\ y(t) \rightarrow y_k \\ u(t) \rightarrow u_k \end{cases} \quad \text{II}$$

use I + II to convert to a discrete equation.

Backward Euler discretization

$$\boxed{\dot{\boldsymbol{x}}(t) = \frac{\boldsymbol{x}(t) - \boldsymbol{x}(t-\Delta t)}{\Delta t}}$$

→ what is used in real time simulations & controllers.

Norms

A norm generalizes the concept of length or magnitude of a vector.

$$\|.\|: V \to R^+$$

A norm is a function of a single vector and produces a scalar result and satisfies:

1. $\|x\| = 0 \iff x = 0$
2. $\|\alpha x\| = |\alpha| \, \|x\| \quad \forall \alpha$ scalar
3. $\|x + y\| \leq \|x\| + \|y\|$ → triangle inequality
4. $\|\langle x, y \rangle\| \leq \|x\| \, \|y\|$ → cauchy schwartz inequality

inner product of x & y

Example Norms

1. $\|x\|_2 = \left(\sum_{i=1}^{n} |x_i^2| \right)^{1/2} = \left(\langle x, x \rangle \right)^{1/2} = (x^T x)^{1/2}$

   Ecludian norm

2. $\|x\|_p = \left(\sum_{i=1}^{n} |x_i^p| \right)^{1/p} \rightarrow$ lp norm

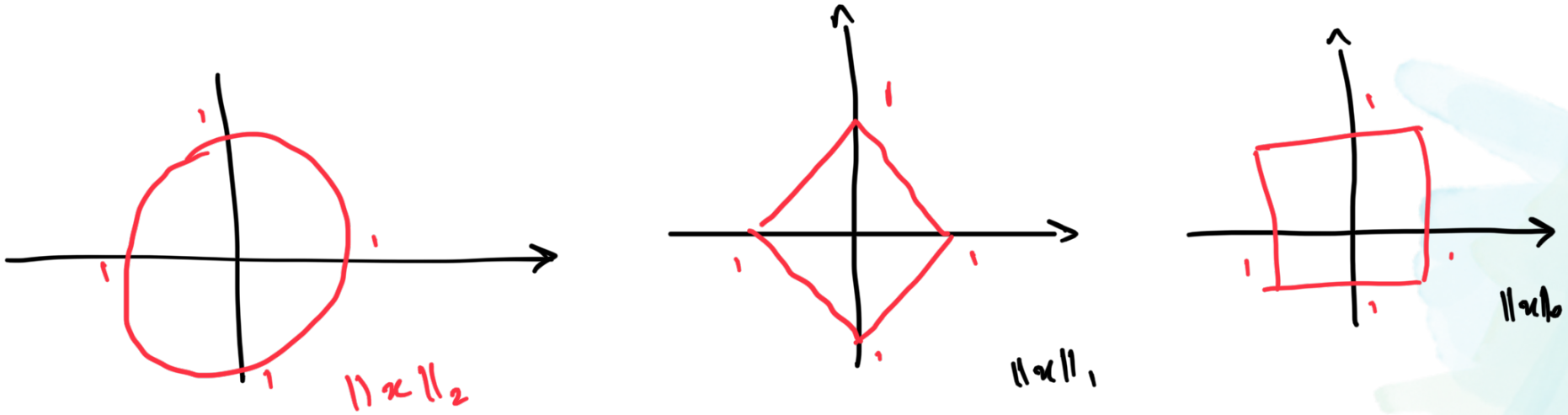
3. $\|x\|_1 = \sum_{i=1}^{n} |x_i| \rightarrow$ l_1 Norm

4. $\|x\|_\infty = \max_i |x_i| \rightarrow$ l_∞ Norm

Unit Balls

DEF: Unit ball of a norm: $\{x : \|x\| = 1\}$

Some unit balls:



Norm of a function

Given a vector space $(c[t_0, t_1], R)$ ← f is define

1. $\|f\|_2 = \left(\int_{t_0}^{t_1} |f(t)|^2 \, dt \right)^{1/2}$
2. $\|f\|_p = \left(\int_{t_0}^{t_1} |f(t)|^p \, dt \right)^{1/2}$
3. $\|f\|_\infty = \max_{t \in [t_0, t_1]} |f(t)|$

Pseudoinverse

Consider an under-determined system: $m \overset{n}{\boxed{\quad A \quad}} \; \boxed{x}_{n} = \boxed{b}_{m}$

$$\boxed{Ax = b,} \qquad A \in \mathbb{R}^{m\times n},\ x \in \mathbb{R}^{n},\ b \in \mathbb{R}^{m},\quad m < n$$

We seek to minimize norm solution from the infinitely many solutions:

$$* \begin{cases} \min \frac{1}{2} x^T x \\ s.t. \quad Ax - b = 0 \end{cases}$$

constrained minimization problem.

solution:

form the Hamiltonian:

$$H = \frac{1}{2} x^T x + \lambda^T (Ax - b) \quad (1) \rightarrow$$

minimize over all values of x & λ

↳ lagrange multiplier

Pseudoinverse

set two derivatives to zero :

$$\frac{\partial^T H}{\partial x} = 0 \rightarrow x + A^T \lambda = 0 \quad (2)$$

$$\frac{\partial^T H}{\partial \lambda} = 0 \rightarrow Ax - b = 0 \quad (3) \rightarrow$$ simply recovers the constraint.

Rearrang 2 & pre-multiply by A : $Ax = -AA^T\lambda$

$$\rightarrow \lambda = (-AA^T)^{-1} b \quad (4)$$

substitute (4) into (2) $x = -A^T\lambda = \overset{A^+}{\boxed{A^T(AA^T)^{-1}}} b$: solution

Pseudoinverse

Consider an Over-determined system:

$$\boxed{A}\,\boxed{x} = \boxed{b}$$

tall A

$$Ax = b, \qquad A \in \mathbb{R}^{m\times n},\ x \in \mathbb{R}^n,\ b \in \mathbb{R}^m,\ \underline{m} > \underline{n}$$

No solution exists for $Ax = b$

Form error $\underline{e = b - Ax}$ and minimize $\underline{\frac{1}{2}e^T e}$ → second Norm of error vector!

$$\frac{1}{2}e^T e = \frac{1}{2}(b - Ax)^T (b - Ax) = \frac{1}{2}(b^T b - x^T A^T b - b^T A x + x^T A^T A x)$$

let $\frac{\partial}{x}\left(\frac{1}{2}e^T e\right) = 0 \rightarrow \frac{1}{2}(-2A^T b + 2A^T A x) = 0 \rightarrow A^T b = A^T A x$

$$x = \underbrace{\boxed{(A^T A)^{-1} A^T}}_{A^+} b$$

Pseudoinverse

- Right pseudoinverse: $A^\dagger = A^T(AA^T)^{-1}$ (Moore-Penrose pseudoinverse)

$$AA^\dagger = I_m$$ → identity matrix

- Left pseudoinverse: $A^\dagger = (A^TA)^{-1}A^T$

$$\boxed{A^\dagger A = I_n}$$