01208583 ROBOTICS

Lecture 1 : Course Introduction

Dr.Varodom Toochinda
Dept. of Mechanical Engineering
Kasetsart University

Course outline

- Lecture 1 – Introduction to optimal control and reinforcement learning.
- Lecture 2 – Optimation basics: unconstrained, equality and inequality constrained problems. Regularization and line search methods.
- Lecture 3 – Deterministic optimal control. Pontryagin minimum princiuple. Linear quadratic Regulators (LQR). Controllability.
- Lecture 4 – Dynamic programming. Principle of optimality. Model Predictive Control (MPC)
- Lecture 5 – Differential dynamic programming and iterative LQR. Direct colocation methods. Trajectory stabilization.
- Lecture 6 – Attitude and rotation kinematics. Quaternion. Quadrotor control.
- Lecture 7 – Contact dynamics. Stochastic optimal control.

การให้คะแนน

- คะแนนรวม 100 คะแนน สำหรับครึ่งหลังของวิชานี้เท่านั้น
- การบ้าน : 40 %
- สอบ : 60 % ประกอบด้วย
  - สอบข้อเขียน : Closed-book. ให้เขียนโน๊ตใส่กระดาษ A4 เข้าได้จำนวน 2 แผ่น เขียนได้ทั้งหน้า-หลัง (30 %)
  - Take-home exam: ทำข้อสอบใน Jupyter notebook กำหนดส่งภายใน 24 ชั่วโมง (30 %)

GOOGLE CLASSROOM

https://classroom.google.com/c/ODI0MTQ2Mjc4OTYz?cjc=5q7tjvwl

Class code : 5q7tjvwl

REFERENCE COURSE

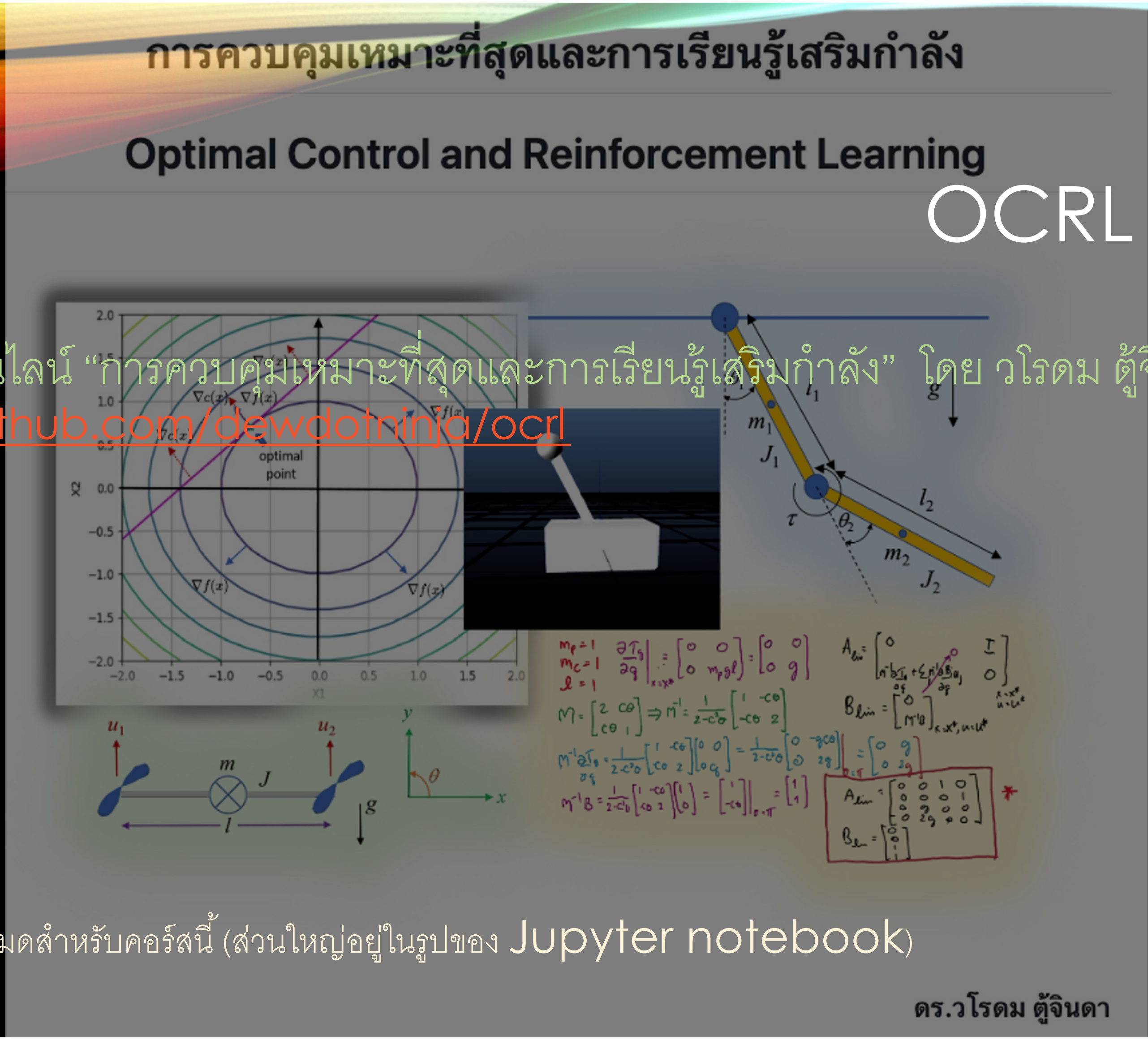
- CMU Optimal Control 16-745
  - Homepage : https://optimalcontrol.ri.cmu.edu/
  - Github : https://github.com/Optimal-Control-16-745
  - YouTube : https://www.youtube.com/playlist?list=PLZnJoM76RM6IAJfMXd1PgGNXn3dxhkVgI

หมายเหตุ : โปรแกรมตัวอย่างจากคอร์สนี้จะใช้ภาษา Julia ซึ่งผู้เรียนอาจไม่คุ้นเคย เรามี option แปลงเป็น Python ให้ อย่างไรก็ตาม แนะนำให้หาโอกาสเรียนรู้ Julia ซึ่งใช้การคอมไพล์ทำให้มีสมรรถนะดีกว่า

SUPPLEMENT

- MIT : Underactuated Robotics https://underactuated.csail.mit.edu/index.html
- MIT : Robotic Manipulation https://manipulation.csail.mit.edu/index.html
- Optional
  - UC Berkeley : CS287 Advanced Robotics (2019) https://youtube.com/playlist?list=PLwRJQ4m4UJjNBPJdt8WamRAt4XKc639wF&si=x5xk8z9Emg06Zan8
  - BYU Flow Lab : Optimization https://youtube.com/playlist?list=PLj6pNSgoumyfNUw0T_dOv5g6QzDf6tHmq&si=9IQZyr92Lcysy3qL

OCRL BOOK



- หนังสือออนไลน์ “การควบคุมเหมาะที่สุดและการเรียนรู้เสริมกำลัง” โดย วโรดม ตู้จินดา
  https://github.com/dewdotninja/ocrl

แหล่งรวมเอกสารทั้งหมดสำหรับคอร์สนี้ (ส่วนใหญ่อยู่ในรูปของ Jupyter notebook)

ติดตั้งซอฟต์แวร์

- Python with Jupyter support
  - เนื่องจากแพ็กเกจที่ใช้ในคอร์สมีหลากหลายและบางตัวอาจลงยาก แนะนำให้ใช้ docker
    - ติดตั้ง docker desktop และรันโปรแกรม
    - สร้าง folder และคัดลอกไฟล์ Dockerfile และ compose.yml ใส่ใน folder นั้น
    - พิมพ์ docker compose up
  - Linux หรือ WSL บน windows : สร้าง environment ตามคำแนะนำ
  - Mac silicon : หากมีปัญหารัน docker ทดลองติดตั้ง miniconda สร้าง conda environment และติดตั้ง packages ที่ใช้
- Julia
  - ติดตั้งตามคำแนะนำจากเว็บ https://julialang.org/

LECTURE 1 MATERIALS

- ศึกษาพื้นฐานของโมเดลไม่เป็นเชิงเส้น : พลวัต จุดสมดุล เสถียรภาพ ระบบเวลาวิยุต
- การคำนวณอนุพันธ์อัตโนมัติ (automatic differentiation)
- การคำนวณปริพันธ์ (forward, backward, RK4)
- การทดสอบโปรแกรมใน container/environment ที่สร้างขึ้น

รายละเอียดอ่านได้จาก

merobo_hw1.ipynb

พลวัตในระบบเวลาต่อเนื่อง

สถานะ (state) $\in R^n$

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

อนุพันธ์เวลาของสถานะ

อินพุต (input) $\in R^m$

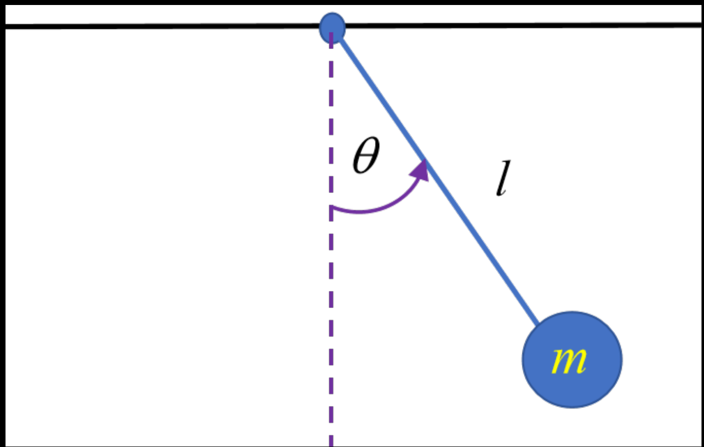
สำหรับระบบเชิงกล (เช่นหุ่นยนต์)

$$x = \begin{bmatrix} q \\ v \end{bmatrix}$$

"configuration"
"pose"
(อาจจะไม่ใช่เวกเตอร์)

ความเร็ว (velocity)
เป็นเวกเตอร์

ตัวอย่าง : ระบบลูกตุ้ม (pendulum)



$$ml^2\ddot{\theta} + mgl\sin(\theta) = \tau$$

สถานะ

$$x = \begin{bmatrix} q \\ v \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$$

อินพุต

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

$$\Rightarrow \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ -\frac{g}{l}\sin\theta + \frac{1}{ml^2}u \end{bmatrix}$$

ระบบควบคุมสัมพรรค (Control-affine systems)

$$\dot{x} = f_o(x) + B(x)u$$

drift

Jacobian input

ระบบลูกตุ้ม

$$f_o(x) = \begin{bmatrix} \dot{\theta} \\ -\frac{g}{l}\sin\theta \end{bmatrix}, \quad B(x) = \begin{bmatrix} 0 \\ -\frac{1}{ml^2} \end{bmatrix}$$

พลวัตตัวจัดการ (Manipulator dynamics)

สมการการเคลื่อนที่อีกรูปแบบหนึ่งที่เหมาะสมกับระบบเชิงกลเรียกว่า *พลวัตตัวจัดการ (manipulator dynamics)* เขียนได้เป็นดังนี้

$$M(q)\dot{v} + C(q,v) = B(q)u + F \tag{1.7}$$

นิยมเรียก $M(q)$ ว่าเมทริกซ์มวล (mass matrix) $C(q,v)$ ว่าค่าเอนเอียงพลวัต (dynamic bias) หรือคอริออลิส (coriolis) รวมพจน์แรงโน้มถ่วง (gravity) $B(q)$ คืออินพุตจาโคเบียน และ F รวมแรงจากภายนอกอื่นที่นอกเหนือจากอินพุต เช่นแรงเสียดทาน

เนื่องจาก (1.7) บรรยายพลวัตเท่านั้น หากต้องการบรรยายสถานะเต็มของระบบ ต้องการสมการเพิ่มเติมคือ

$$\dot{q} = G(q)v \tag{1.8}$$

เรียกว่า *จลนศาสตร์ความเร็ว (velocity kinematics)* โดย $G(q)$ คือ *จาโคเบียนจลนศาสตร์* ตัวอย่างเช่นกรณี $\dot{q}$ คือควอเทอร์เนียน 4 มิติ ขณะที่ v คือความเร็วเชิงมุมใน 3 มิติ จึงต้องผ่านตัวแปลง $G(q)$

ดังนั้นพลวัตในรูป (1,1) เขียนได้ดังนี้

$$\dot{x} = f(x,u) = \begin{bmatrix} G(q)v \\ M^{-1}(q)(B(x)u + F - C(q,v)) \end{bmatrix} \tag{1.9}$$

พลวัตตัวจัดการ (Manipulator dynamics)

สำหรับกรณีระบบลูกตุ้มในตัวอย่าง 1.1

$$M(q) = ml^2, \quad C(q,v) = mgl\sin(\theta), \quad B = I, G = I \tag{1.10}$$

จุดที่น่าสนใจสำหรับการบรรยายพลวัตตัวจัดการคือ ระบบเชิงกลของวัตถุคงรูป (rigid body) เกือบทั้งหมด เช่นหุ่นยนต์สามารถเขียนอยู่ในรูปนี้ได้ เนื่องจากเป็นเสมือนวิธีหนึ่งในการเขียน *สมการออยเลอร์-ลากรานจ์ (Euler-Lagrange equation)*

$$L = \frac{1}{2}v^T M(q)v - U(q) \tag{1.11}$$

โดยทางด้านขวาของ (1.11) พจน์แรกคือพลังงานจลน์ (kinetic energy) และพจน์ที่สองคือพลังงานศักย์ (potential energy) สังเกตว่าจากหลักการทางฟิสิกส์ พลังงานจลน์ใน (1.11) ต้องเป็นบวกเสมอ ดังนั้นเมทริกซ์มวล M จะต้องเป็นแบบสมมาตรและบวกแน่นอนเสมอ ทำให้สามารถหาค่าผกผันของ M ใน (1.9) ได้

ระบบเชิงเส้น (linear systems)

สมการพลวัตเรียกว่าเป็น *ระบบเชิงเส้น (linear systems)* หากเขียนได้อยู่ในรูป

$$\dot{x} = A(t)x + B(t)u \tag{1.12}$$

ในกรณีทั่วไปเมทริกซ์ $A(t), B(t)$ ขึ้นกับเวลา เรียกว่าระบบแปรตามเวลา (time varying) แต่หากเมทริกซ์ $A(t) = A,\ B(t) = B$ เรียกว่าระบบไม่แปรตามเวลา (time invariant) (นิยมเรียกตัวย่อ LTI)

ระบบเชิงเส้นมีความสำคัญอย่างยิ่งยวดในสาขาระบบควบคุม เนื่องจากเราสามารถประมาณค่าระบบไม่เป็นเชิงเส้นเป็นระบบเชิงเส้นเฉพาะที่ (local) และระบบควบคุมมักทำงานได้ดี การประมาณค่ารอบจุดทำงาน x^*, u^* ทำได้ดังนี้

$$\dot{x} = f(x,u) \quad \Rightarrow \quad A = \left.\frac{\partial f}{\partial x}\right|_{x^*,u^*},\ B = \left.\frac{\partial f}{\partial u}\right|_{x^*,u^*} \tag{1.13}$$

ในโปรแกรม เหมาะสมที่จะใช้วิธีการหาอนุพันธ์อัตโนมัติ (autodiff) อ่านเพิ่มเติมในภาคผนวก C

จุดสมดุล (equilibria)

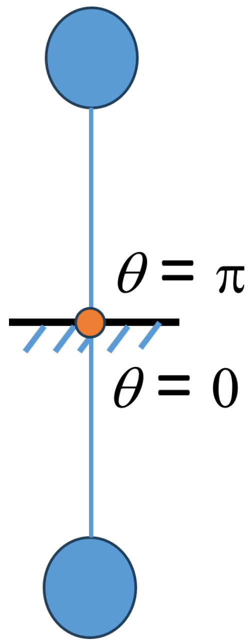
นิยามของสมดุล (equilibrium) คือจุดที่ระบบคงอยู่ในสถานะพัก (remains at rest) ในทางคณิตศาสตร์

$$\dot{x} = f(x, u) = 0 \tag{1.14}$$

พิจารณาจาก (1.14) ในทางพีชคณิตสมดุลคือค่ารากของสมการพลวัต

ในระบบลูกตุ้ม เมื่อกำหนดเงื่อนไขสมดุล

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ -\frac{g}{l}\sin(\theta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{1.15}$$

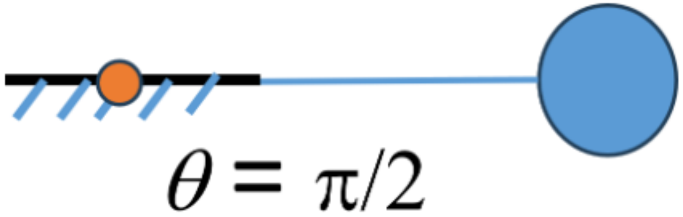


จะได้ว่า

$$\begin{aligned} \dot{\theta} &= 0 \\ \theta &= 0, \pi, 2\pi, \ldots \end{aligned} \tag{1.16}$$

คือลูกตุ้มอยู่ในตำแหน่งตั้งฉากกับพื้น ในทิศทางตกสู่พื้นหรือชี้ขึ้น ดังแสดงในรูปที่ 1.2

หากเราต้องการจะย้ายตำแหน่งสมดุลย์ของลูกตุ้มจะทำได้หรือไม่? สมมุติว่าต้องการตำหน่งสมดุลเป็นทิศทางขนานกับพื้นดังในรูปที่ 1.3



การควบคุมลูกตุ้มอย่างง่าย

รูปที่ 1.3 การย้ายจุดสมดุลของระบบลูกตุ้มโดยตัวควบคุม

จะเห็นว่ากรณีนี้เราต้องการตัวควบคุม u ที่จะต้านแรงโน้มถ่วง จากสมการพลวัตลูกตุ้ม จะได้ว่า

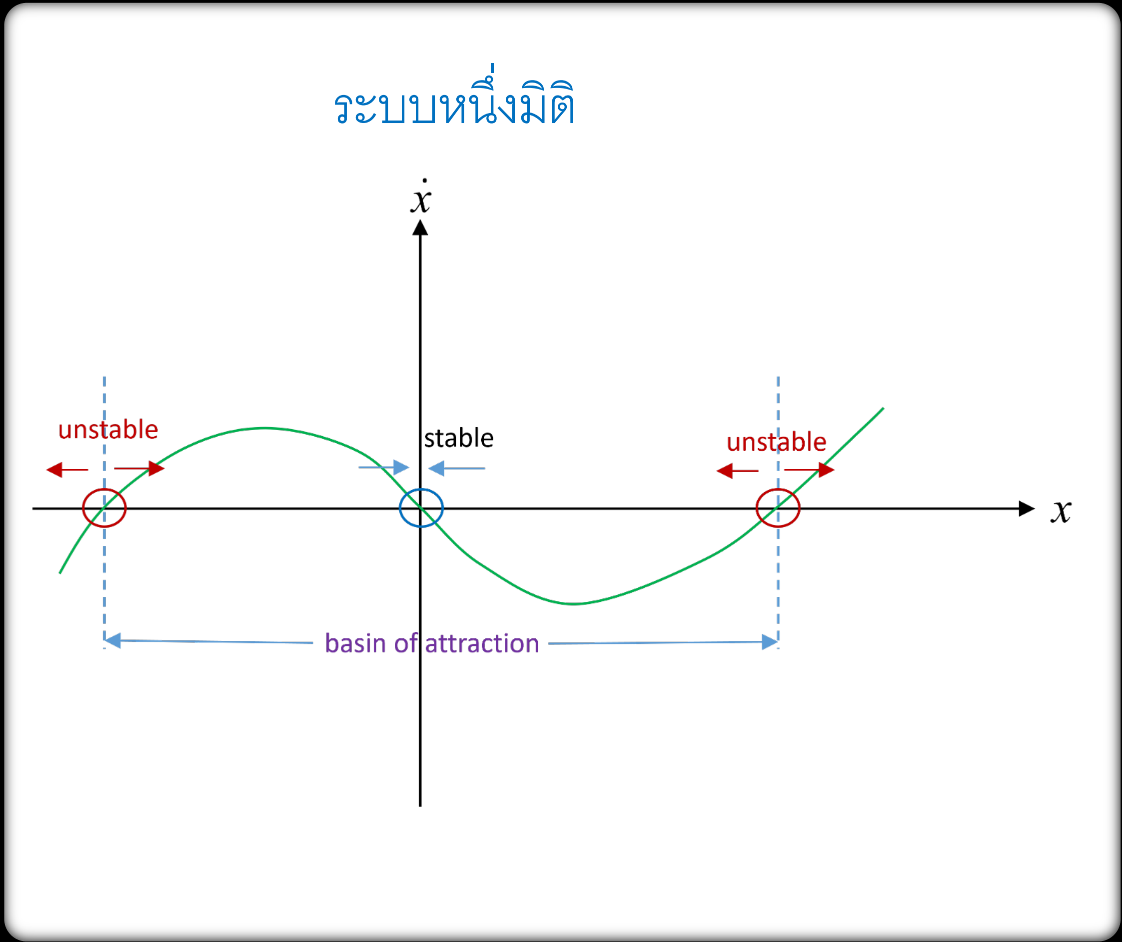
$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ -\frac{g}{l}\sin(\pi/2) + \frac{1}{ml^2}u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{1.15}$$

คำนวณหาตัวควบคุมได้ดังนี้

$$\begin{aligned} \frac{1}{ml^2}u &= \frac{g}{l}\sin(\pi/2) \\ u &= mgl \end{aligned} \tag{1.16}$$

ในกรณีทั่วไปคือปัญหาการหาค่าราก (root finding problem) สำหรับตัวควบคุม u

เสถียรภาพของจุดสมดุล



$$\frac{\partial f}{\partial x} < 0 \Rightarrow \text{เสถียร}, \quad \frac{\partial f}{\partial x} > 0 \Rightarrow \text{ไม่เสถียร}$$

เสถียรภาพของจุดสมดุล

หลักการนี้ยังสามารถขยายไปสู่ระบบที่มิติสูงขึ้น ในระบบมิติเท่ากับ n $\frac{\partial f}{\partial x}$ เป็นเมทริกซ์จาโคเบียน $n \times n$ ดังนั้นจึงต้องใช้วิธีการแยกค่าลักษณะเฉพาะ (eigen-decompostion) เพื่อแยกเป็นระบบมิติเดี่ยวจำนวน n ระบบ จะได้ว่า

$$\begin{aligned} Re\left[eigvals\left(\frac{\partial f}{\partial x}\right)\right] &< 0 \Rightarrow \text{เสถียร} \\ Re\left[eigvals\left(\frac{\partial f}{\partial x}\right)\right] &> 0 \Rightarrow \text{ไม่เสถียร} \end{aligned} \tag{1.18}$$

โดย $Re[\ ]$ คือส่วนค่าจริงของจำนวนเชิงซ้อน และ $eigvals(\ )$ คือการคำนวณค่าลักษณะเฉพาะ

เสถียรภาพของระบบลูกตุ้ม

เมื่อใช้หลักการนี้ตัดสินเสถียรภาพของจุดสมดุลในระบบลูกตุ้ม

$$f(x) = \begin{bmatrix} \dot{\theta} \\ -\frac{g}{l}\sin\theta \end{bmatrix} \;\Rightarrow\; \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos\theta & 0 \end{bmatrix}$$

ที่จุดสมดุล $\theta = \pi$

$$\left.\frac{\partial f}{\partial x}\right|_{\theta=\pi} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{bmatrix} \;\Rightarrow\; eigvals\left(\frac{\partial f}{\partial x}\right) = \pm\sqrt{\frac{g}{l}}$$

จะเห็นว่าค่าลักษณะเฉพาะตัวหนึ่งเป็นจำนวนจริงค่าบวก ดังนั้นสรุปได้ว่าจุดสมดุลในตำแหน่งชี้ขึ้นไม่เสถียร ที่จุดสมดุล $\theta = 0$

$$\left.\frac{\partial f}{\partial x}\right|_{\theta=0} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{bmatrix} \;\Rightarrow\; eigvals\left(\frac{\partial f}{\partial x}\right) = \pm i\sqrt{\frac{g}{l}}$$

เป็นกรณีที่ค่าลักษณะเฉพาะมีค่าจริงเป็นศูนย์ และค่าจินตภาพเป็นสังยุค ทำให้ผลตอบสนองเป็น *การแกว่งแบบไร้การหน่วง (undamped oscillation)*

พลวัตในระบบเวลาวิยุต (discrete-time)

พลวัตในระบบเวลาวิยุตโดยทั่วไปมักอยู่ใน *รูปชัดแจ้ง (explicit form)*

$$x_{k+1} = f_d(x_k, u_k) \tag{1.19}$$

โดยในรูปนี้ สถานะในการสุ่มเวลาครั้งต่อไป x_{k+1} บรรยายได้เป็นฟังก์ชันของสถานะก่อนหน้า x_k และอินพุต u_k

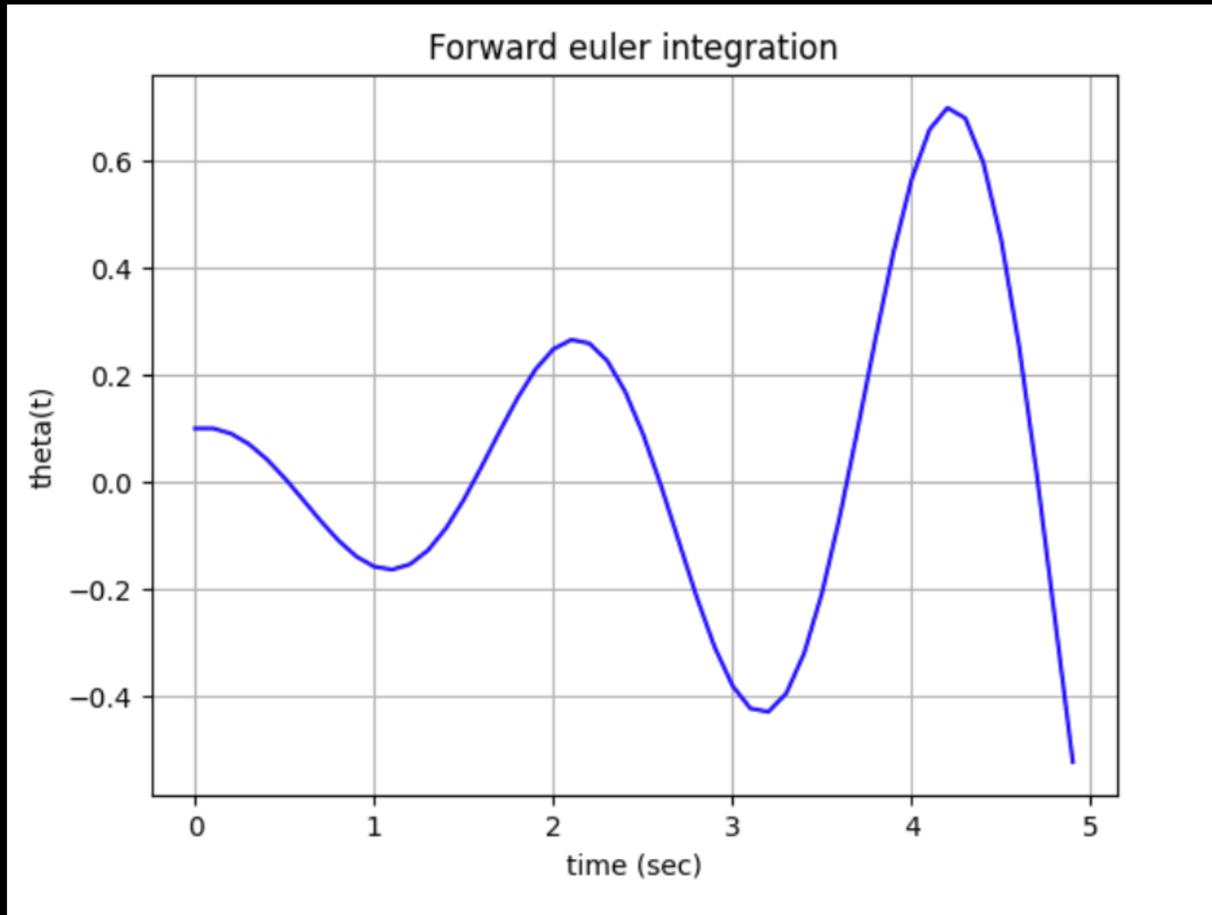
การหาปริพันธ์โดยวิธีออยเลอร์ข้างหน้า (forward Euler integration)

การแปลงเป็นระบบวิยุตทำได้หลายวิธี วิธีง่ายที่สุดเรียกว่า *การหาปริพันธ์ออยเลอร์ข้างหน้า (forward Euler integration)*

$$x_{k+1} = f_d(x_k, u_k) = x_k + hf(x_k, u_k) \tag{1.20}$$

โดย h คือค่าขั้นเวลา เราอาจมองว่าวิธีการออยเลอร์ข้างหน้าคือการประมาณค่าโดยความชันคงที่

จำลองการแกว่งลูกตุ้ม
ที่จุด $\theta = 0$
โดยวิธีออยเลอร์ข้างหน้า



เสถียรภาพของระบบเวลาวิยุต

พิจารณาพลวัตในระบบเวลาวิยุต (1.19) สมมุติว่าสถานะเริ่มต้นคือ x_0 การคำนวณสถานะในขั้นเวลา N ก็คือการส่งแบบทำซ้ำ (iterative map)

$$x_N = f_d(f_d(f_d(\dots f_d(x_0)))) \tag{1.21}$$

ประมาณค่าเชิงเส้นและใช้กฎลูกโซ่ จะได้ว่า

$$\frac{\partial x_N}{\partial x_0} = \left.\frac{\partial f_d}{\partial x}\right|_{x_0} \left.\frac{\partial f_d}{\partial x}\right|_{x_0} \dots \left.\frac{\partial f_d}{\partial x}\right|_{x_0} = A_d^N \tag{1.22}$$

กำหนด $x_0 = 0$ เป็นจุดสมดุล (โดยไม่เสียความเป็นทั่วไป เพราะสามารถใช้การแปลงพิกัด) ดังนั้นหากระบบเสถียรจะได้ว่า

$$\begin{aligned} &\lim_{k\to\infty} A_d^k x_0 = 0 \forall x_0 \\ \Rightarrow &\lim_{k\to\infty} A_d^k = 0 \\ \Rightarrow &|eigvals(A_d)| < 1 \end{aligned} \tag{1.23}$$

กล่าวคือ จุดสมดุลเสถียร ค่าลักษณะเฉพาะของ A_d จะต้องอยู่ใน *วงกลมหนึ่งหน่วย (unit circle)* ซึ่งเป็นหลักการที่สำคัญในการศึกษาระบบเวลาวิยุต

เสถียรภาพของระบบลูกตุ้มในเวลาวิยุต

พิจารณาระบบลูกตุ้มที่แปลงเป็นระบบเวลาวิยุตโดยวิธีออยเลอร์ข้างหน้า (1.20) จะได้ว่า

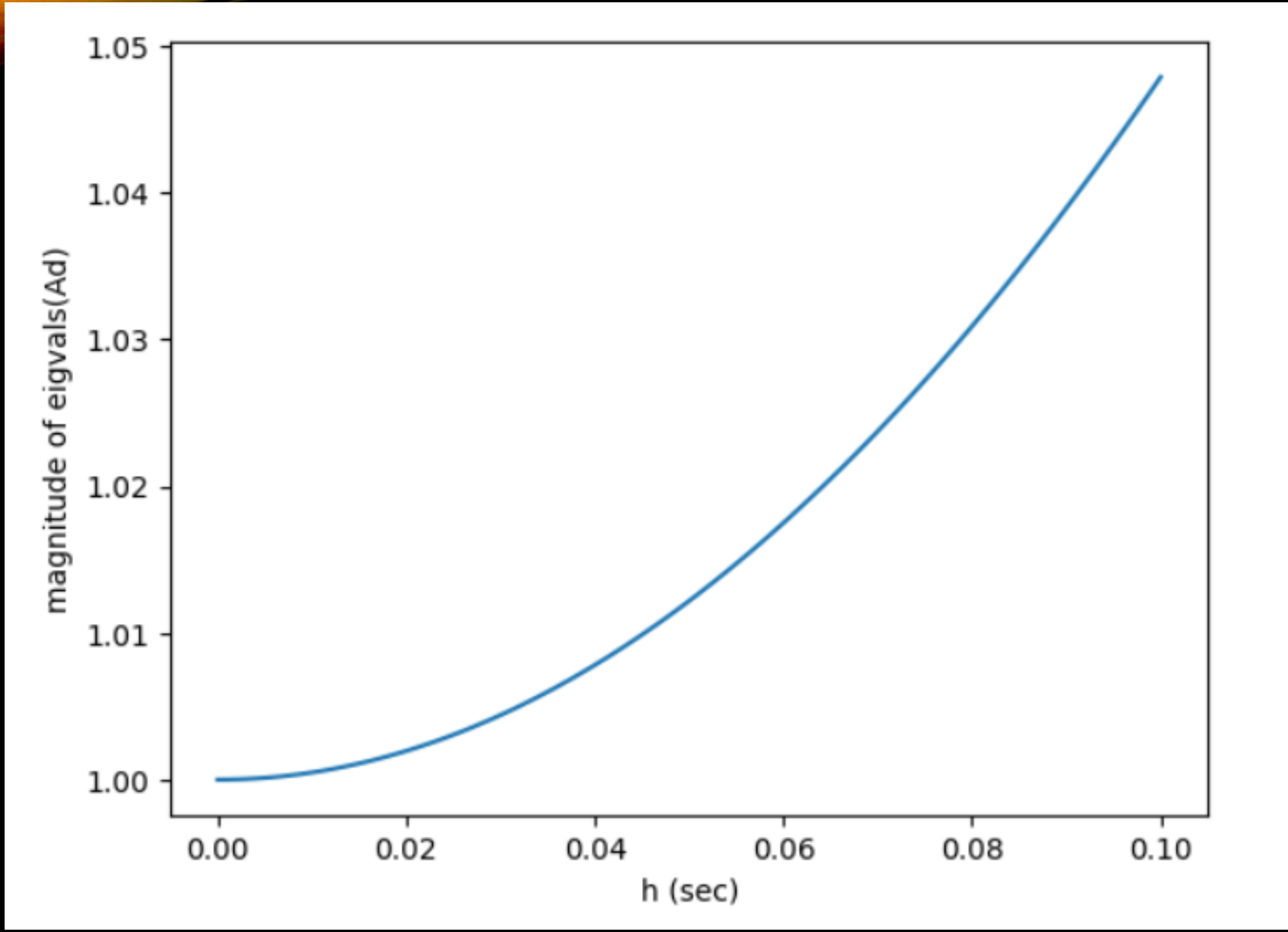
$$A_d = \frac{\partial f_d}{\partial x} = I + hA = I = h \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}cos(\theta) & 0 \end{bmatrix} \tag{1.24}$$

เมื่อคำนวณค่าลักษณะเฉพาะ จะได้ว่า

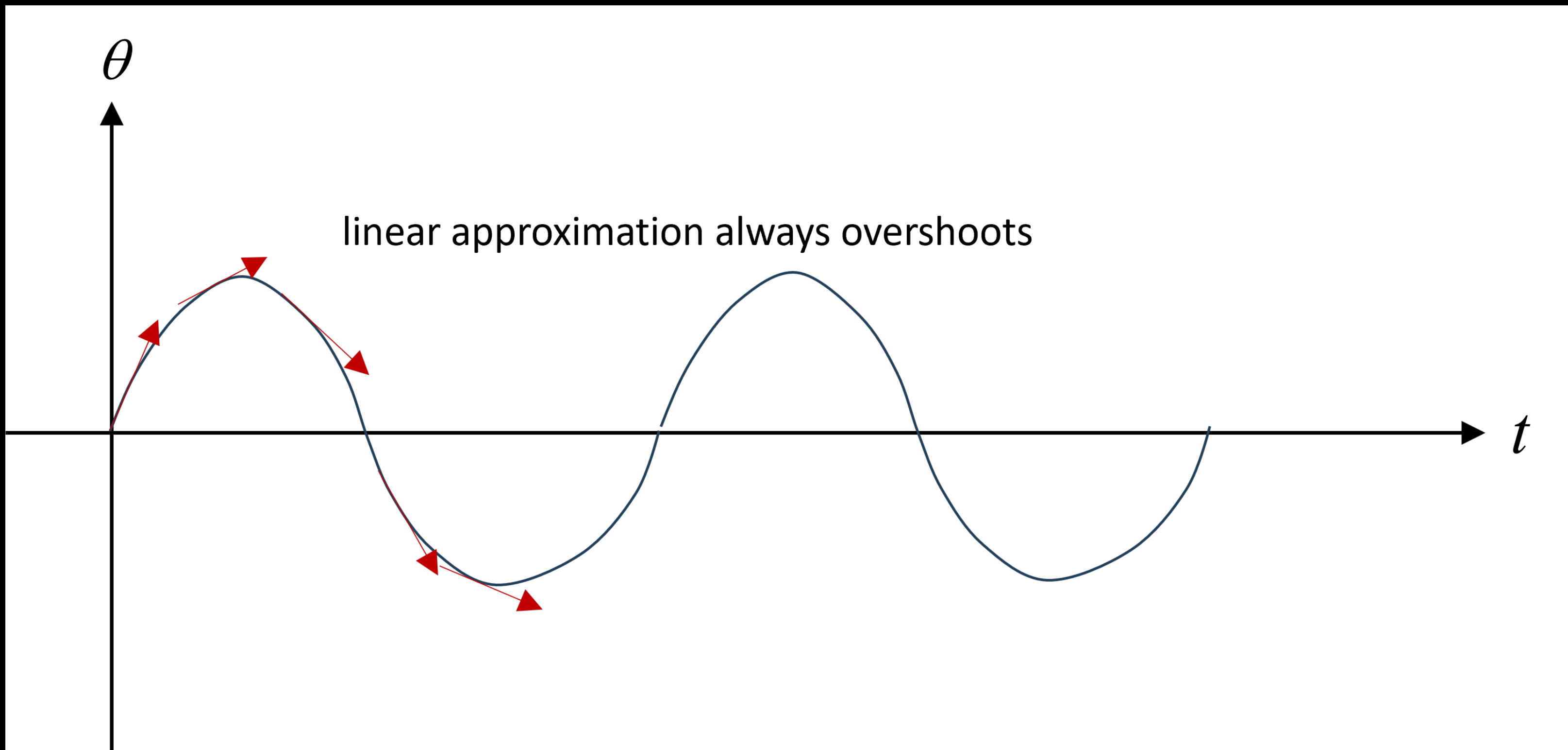
$$eigvals(A_d|_{\theta=0}) \approx 1 \pm \epsilon i$$

โดยค่า ϵ เป็นค่าน้อยขึ้นอยู่กับค่าขั้นเวลา h เช่น $\epsilon = 0.3$ เมื่อ $h = 0.1$ วินาที แต่ไม่ว่าค่าจะน้อยเท่าไรจะส่งผลทำให้ค่าลักษณะเฉพาะของ A_d ออกนอกวงกลมหนึ่งหน่วย ทำให้เป็นระบบที่ไม่เสถียร และการแกว่งจะมีขนาดใหญ่ขึ้นดังแสดงในผลการจำลอง

ค่าลักษณะเฉพาะเทียบกับขั้นเวลาของระบบลูกตุ้มโดยวิธีออยเลอร์ข้างหน้า



การพุ่งเกินจากการประมาณค่าโดยเส้นตรง



วิธีรุงเงอ-คุททาอันดับ 4 (4^{th} order Runke-Kutta method หรือ RK4)

RK4 ประมาณค่า $x(t)$ โดยพหุนามกำลังสาม (cubic polynomials) แทนที่จะใช้เส้นตรง ทำให้มีความแม่นยำสูงกว่าวิธีการออยเลอร์ข้างหน้า

สำหรับพลวัต

$$x_{k+1} = f(x_k)$$

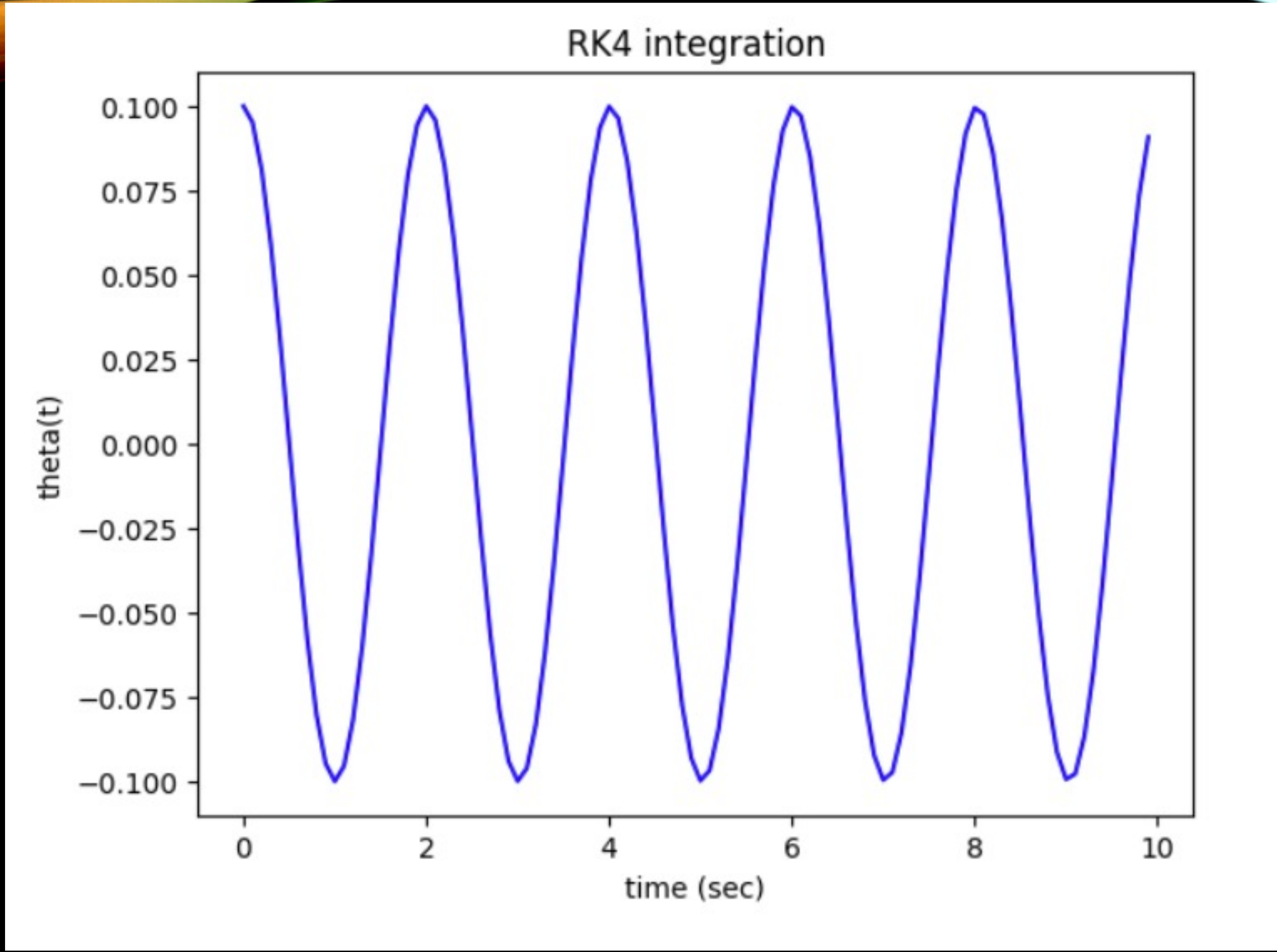
เขียนโครงสร้างโค้ดได้ดังนี้

Pseudo-code :

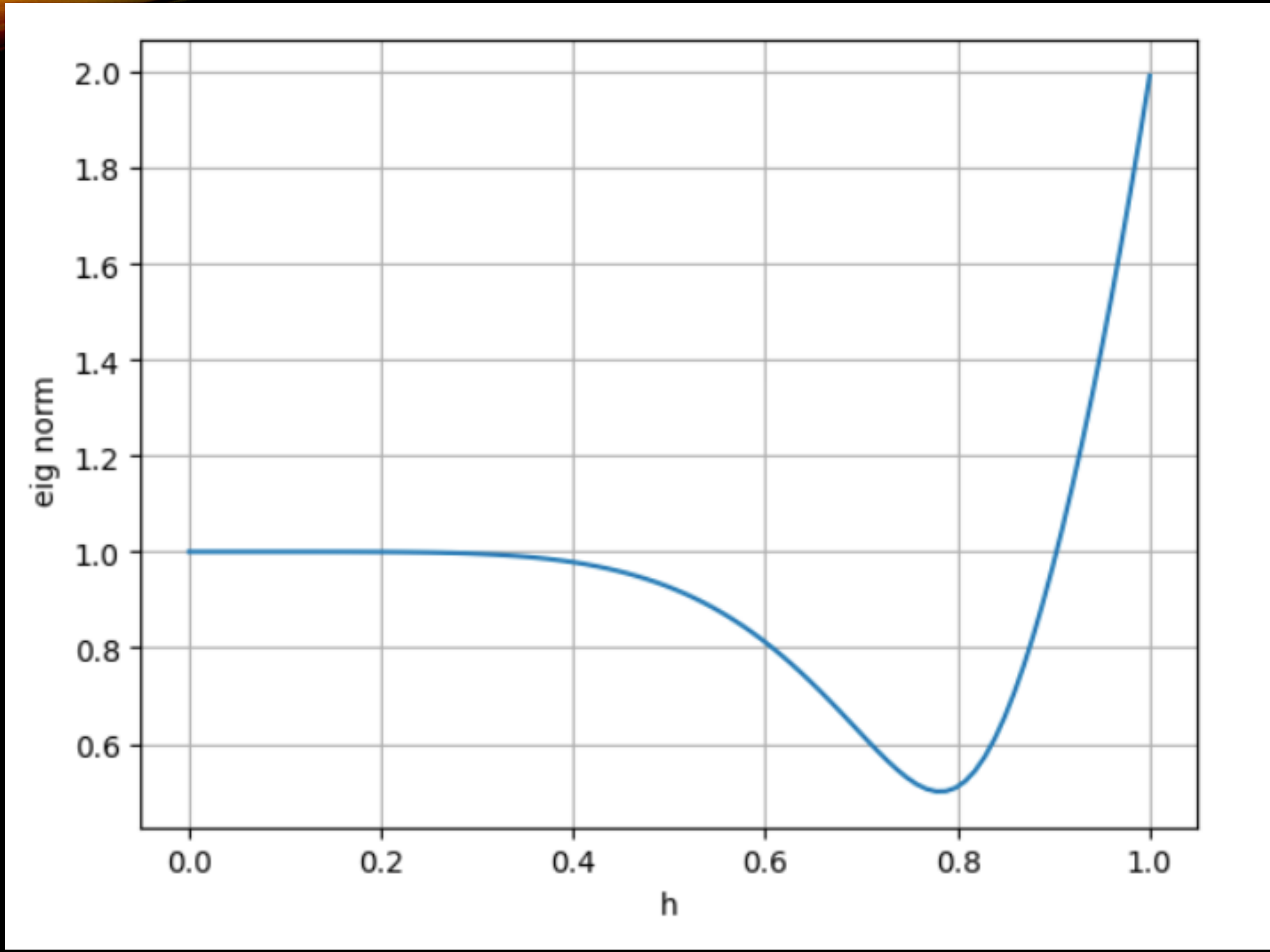
```
k1 = f(xk)
k2 = f(xk+(h/2)*k1)
k3 = f(xk+(h/2)*k2)
k3 = f(xk+(h/2)*k3)
xk_new = xk + (h/6)*(k1 + 2*k2 + 2*k3 + k4)
```

หมายเหตุ : xk_new ในโค้ดด้านบน หมายถึง x_{k+1}

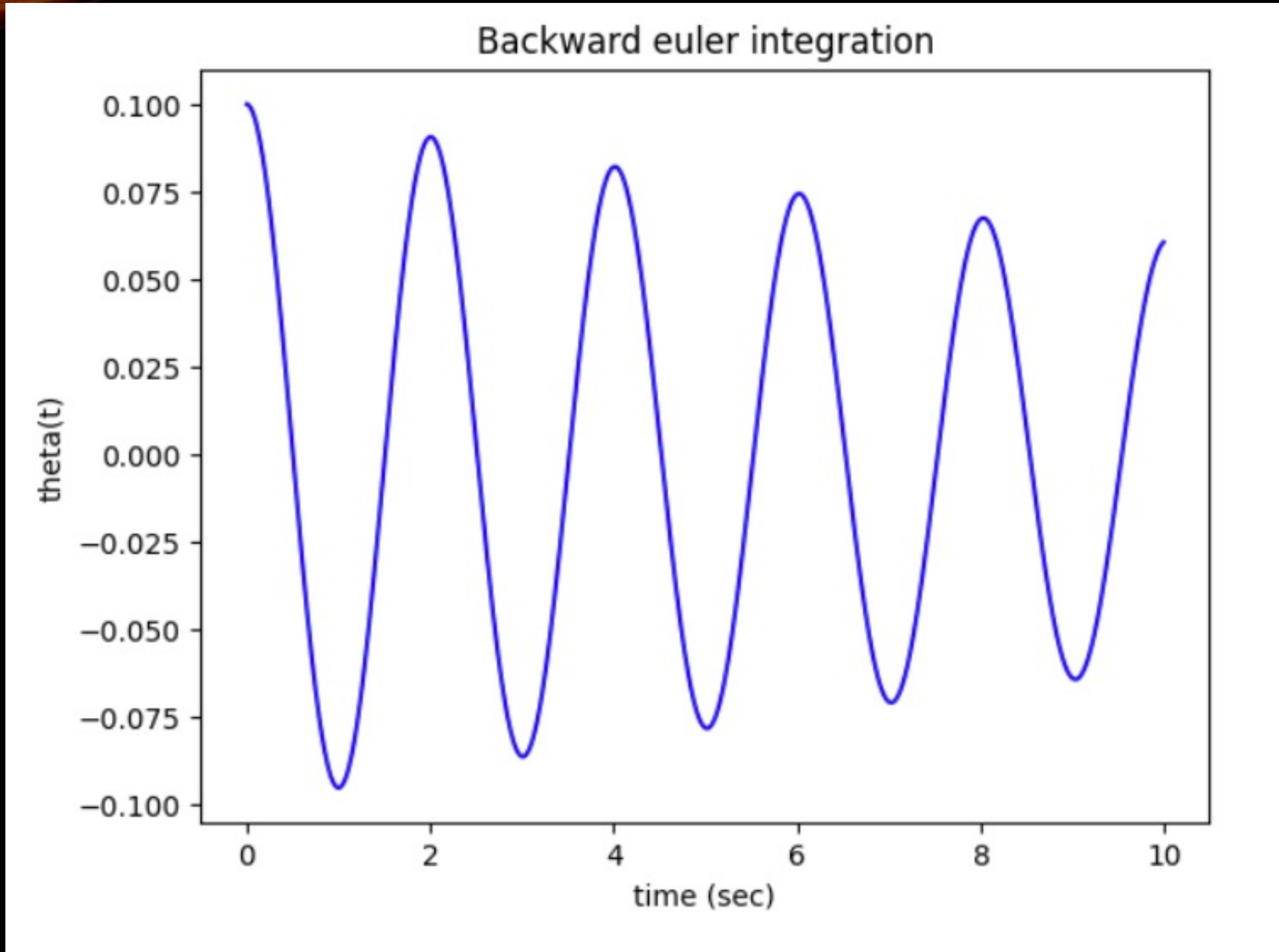
จำลองการแกว่งลูกตุ้ม
ที่จุด $\theta = 0$
โดยวิธี RK4



ค่าลักษณะเฉพาะเทียบกับขั้นเวลาของระบบลูกตุ้มโดยวิธี RK4



จำลองการแกว่งลูกตุ้ม
ที่จุด $\theta = 0$
โดยวิธีออยเลอร์ย้อนหลัง
(backward Euler)



REFERENCES

1. Z. Manchester et.al. [16-745 Optimal Control & Reinforcement Learning, Course materials https://optimalcontrol.ri.cmu.edu/#learning-resources , Carnegie Mellon University. 2025.
2. Wikipedia. Runge-Kutta methods https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods