01208583 Robotics

LECTURE 3 : OPTIMAL CONTROL

Dr. Varodom Toochinda

Dept. of Mechanical Engineering

Kasetsart University

หัวข้อ

- การควบคุมเหมาะที่สุดแบบเชิงกำหนด
  - โดเมนเวลาต่อเนื่อง
  - โดเมนเวลาดีสครีต
- หลักการค่าต่ำสุดของพอนเทรียกิน (Pontryagin's minimum principle)
- ตัวควบคุมกำลังสองเชิงเส้น (Linear Quadratic Regulator : LQR)
- วิธีการยิงโดยอ้อม (Indirect shooting)
- การโปรแกรมกำลังสอง (Quadric Program : QP)
- การหาคำตอบโดยสมการริกคาติ (Riccati equation)
- ตัวควบคุม LQR แบบแนวนอนอนันต์ (infinite horizon)
- คุณสมบัติควบคุมได้ (controllability)

3.1 การควบคุมเหมาะที่สุดแบบเชิงกำหนด (deterministic optimal control)

3.1.1 รูปแบบปัญหาในโดเมนเวลาต่อเนื่อง

ตัวอย่างของโจทย์ปัญหาการควบคุมเหมาะที่สุดแบบเชิงกำหนดเป็นดังนี้

$$\min_{x(t),u(t)} J(x(t),u(t)) = \int_{t_0}^{t_f} l(x(t),u(t))\,dt + l_F(x(t_f))$$

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

$$U_{min} \leq u(t) \leq U_{max}\ \ \text{หรือ}\ \ u(t) \in \mathscr{U},$$

$$c(x(t)) \geq 0 \tag{3.1}$$

บรรทัดแรกคือฟังก์ชันมูลค่าซึ่งเป็นตัวชี้วัดสมรรถนะ ประกอบด้วยพจน์แรกที่อยู่ในปริพันธ์เรียกว่ามูลค่าขั้น (stage cost) และพจน์สุดท้ายคือ มูลค่าปลาย (terminal cost)

3.1.2 รูปแบบปัญหาในโดเมนเวลาดีสครีต

ตัวอย่างของปัญหาการควบคุมเหมาะที่สุดในโดเมนเวลาวิยุตเขียนได้ดังนี้

$$\min_{x_{1:N}, u_{1:N-1}} J(x_{1:N}, u_{1:N-1}) = \sum_{k=1}^{N-1} l(x_k, u_k) + l_F(x_N)$$

$$s.t. \;\; x_{k+1} = f(x_k, u_k),$$

$$U_{min} \leq u_k \leq U_{max} \;\; \text{หรือ} \;\; u_k \in \mathscr{U} \;\; \forall k,$$

$$c(x_k) \geq 0 \;\; \forall k \tag{3.2}$$

โดยคำอธิบายสำหรับพจน์ทั้งหมดสอดคล้องกับกรณีเวลาต่อเนื่อง (3.1) เพียงแต่ค่าตัวแปรอยู่ในรูปของตัวอย่าง (samples) จำนวน N ค่าสำหรับ x_k และ $N-1$ ค่าสำหรับ u_k พลวัตที่เป็นเงื่อนไขสมการและการจำกัดค่าในเงื่อนไขอสมการอยู่ในรูปดีสครีตทั้งหมด

ข้อสังเกตสำหรับ (3.2)

ข้อสังเกตสำหรับ (3.2)

- ปัญหานี้เรียกว่า *มิติจำกัด (finite dimensional)*
- ค่าตัวอย่าง x_k, u_k มักถูกเรียกว่าจุดเงื่อน
- การแปลงปัญหาจากเวลาต่อเนื่องเป็นดีสครีตทำได้โดยการประมาณค่าปริพันธ์วิธีต่างๆ เช่น ผลต่างข้างหน้า/ย้อนหลัง การแปลงเชิงเส้นคู่ หรือวิธีรุงเงอคุททา (Runge-Kutta)
- การแปลงปัญหาจากเวลาดีสครีตเป็นต่อเนื่องทำได้โดยการประมาณค่าในช่วง (interpolation)

3.1.3 หลักการค่าต่ำสุดของพอนเทรียกิน

สรุปจาก (3.6)-(3.9) ได้ความสัมพันธ์ดังนี้

$$x_{k+1} = \nabla_{\lambda} H(x_k, u_k, \lambda_{k+1}) \tag{3.10}$$

$$\lambda_k = \nabla_{x} H(x_k, u_k, \lambda_{k+1}) \tag{3.11}$$

$$u_k = \underset{u}{argmin}\ H(x_k, u, \lambda_{k+1})$$

$$s.t\ \ u \in \mathscr{U} \tag{3.12}$$

$$\lambda_N = \frac{\partial l_F(x_N)}{\partial x_N} \tag{3.13}$$

รายละเอียดศึกษาได้จาก chapter3.ipynb

โดยวิธีการหาลิมิต สามารถบรรยายในระบบเวลาต่อเนื่องได้เป็นดังนี้

$$\dot{x} = \nabla_{\lambda} H(x, u, \lambda) \tag{3.14}$$

$$-\dot{\lambda} = \nabla_{x} H(x, u, \lambda) \tag{3.15}$$

$$u = \underset{\tilde{u}}{argmin}\ H(x, \tilde{u}, \lambda)$$

$$s.t\ \ \tilde{u} \in \mathscr{U} \tag{3.16}$$

$$\lambda(t_f) = \frac{\partial l_F}{\partial x} \tag{3.17}$$

ข้อสังเกต

- วิธีการนี้มีรูปแบบของปัญหาค่าขอบ 2 จุด คือหาค่า x ในทิศทางข้างหน้าโดยหาปริพันธ์ของสมการอนุพันธ์ และหาค่า λ ในทิศทางย้อนหลังโดยการลดค่าเกรเดียนต์ ซึ่งมีอัลกอริทึมหลายประเภทที่ใช้แนวทางนี้ ตัวอย่างหนึ่งคือการแพร่กระจายย้อนหลัง (back propagation) .ในการเรียนรู้เชิงลึก
- วิธีการนี้มีชื่อเรียกว่า *วิธีโดยอ้อม (indirect method)* หรือ *วิธีการยิงโดยอ้อม (indirect shooting)*
- ในระบบเวลาต่อเนื่อง $\lambda(t)$ มีชื่อเรียกว่าแนววิถีสถานะร่วม (costate)
- วิธีการนี้มีข้อด้อยในการคำนวณ โดยใช้เวลามากและประสบปัญหาเชิงเลขในแบบเดียวกับที่พบในโครงข่ายประสาทเทียม คือการเพิ่มค่าอย่างมากหรือหายไปของเกรเดียนต์ นอกจากนั้นยังไม่สามารถจัดการกับเงื่อนไขบังคับอสมการได้โดยตรง ดังนั้นความนิยมเริ่มลดลงสวนทางกับการพัฒนาด้านคอมพิวเตอร์ เพราะปัจจุบันการคำนวณแบบอื่นเช่นวิธีนิวตันให้ผลดีกว่า

3.2 ตัวควบคุมกำลังสองเชิงเส้น

ตัวควบคุมกำลังสองเชิงเส้นหรือ LQR เรียกได้ว่าเป็นกรณีพิเศษของการควบคุมเหมาะที่สุดแบบเชิงกำหนดที่มีความสำคัญและมีประโยชน์อย่างยิ่ง เนื่องจากสามารถหาคำตอบได้ในรูปปิด (สำหรับปัญหาขั้นพื้นฐานที่ไม่มีเงื่อนไขบังคับอสมการ) รูปแบบของปัญหาเขียนได้ดังนี้

$$\min_{x_{1:N},u_{1:N-1}} J = \sum_{k=1}^{N-1} \frac{1}{2}x_k^T Q_k x_k + \frac{1}{2}u_k^T R_k u_k + \frac{1}{2}x_N^T Q_N x_N$$

$$s.t.\ \ x_{k+1} = A_k x_k + B_k u_k,\ \ Q_k \succcurlyeq 0,\ \ R_k \succ 0 \tag{3.18}$$

โดย Q_k, R_k คือเมทริกซ์ให้ค่าน้ำหนักกับสถานะและเอาต์พุตควบคุมตามลำดับ ซึ่งมีคุณสมบัติทำให้ปัญหาเป็นแบบคอนเวกซ์ Q_k เป็นแบบกึ่งบวกแน่นอนเพราะเรายอมให้ค่าน้ำหนักของสถานะบางตัวเป็นศูนย์ได้ แต่ R_k ต้องเป็นแบบบวกแน่นอนเสมอ เพราะหากไม่มีการให้ค่าน้ำหนักกับเอาต์พุตตัวควบคุม ในการคำนวณคำตอบที่มีค่าต่ำสุด อัลกอริทึมจะโกงในการหาคำตอบโดยการใช้ u_k สูงมากเข้าสู่อนันต์ กลายเป็นปัญหาที่สร้างขึ้นอย่างไม่เหมาะสม (ill-posed) แม้ในทางปฏิบัติอาจใช้การจำกัดค่า u โดยเงื่อนไขอสมการ แต่เป็นวิธีการที่นอกเหนือจากปัญหา LQR พื้นฐานที่กล่าวถึงในหัวข้อนี้

3.2.1 การหาคำตอบโดยวิธีการยิงโดยอ้อม

ในการแก้ปัญหา LQR โดยวิธีการยิงโดยอ้อมที่อธิบายในหัวข้อ 3.1.3 เราใช้ความสัมพันธ์ดังนี้

$$x_{k+1} = \nabla_\lambda H(x_k, u_k, \lambda_{k+1}) = Ax_k + Bu_k \tag{3.19}$$

$$\lambda_k = \nabla_x H(k_k, u_k, \lambda_{k+1}) = Qx_k + A^T\lambda_{k+1} \tag{3.20}$$

$$\lambda_N = Q_N x_N \tag{3.21}$$

$$u_k = \nabla_u H(x_k, u_k, \lambda_{k+1}) = 0 \Rightarrow -R^{-1}B^T\lambda_{k+1} \tag{3.22}$$

เค้าโครงของอัลกอริทึมการยิงโดยอ้อมเป็นดังนี้

เค้าโครงของอัลกอริทึมการยิงโดยอ้อมเป็นดังนี้

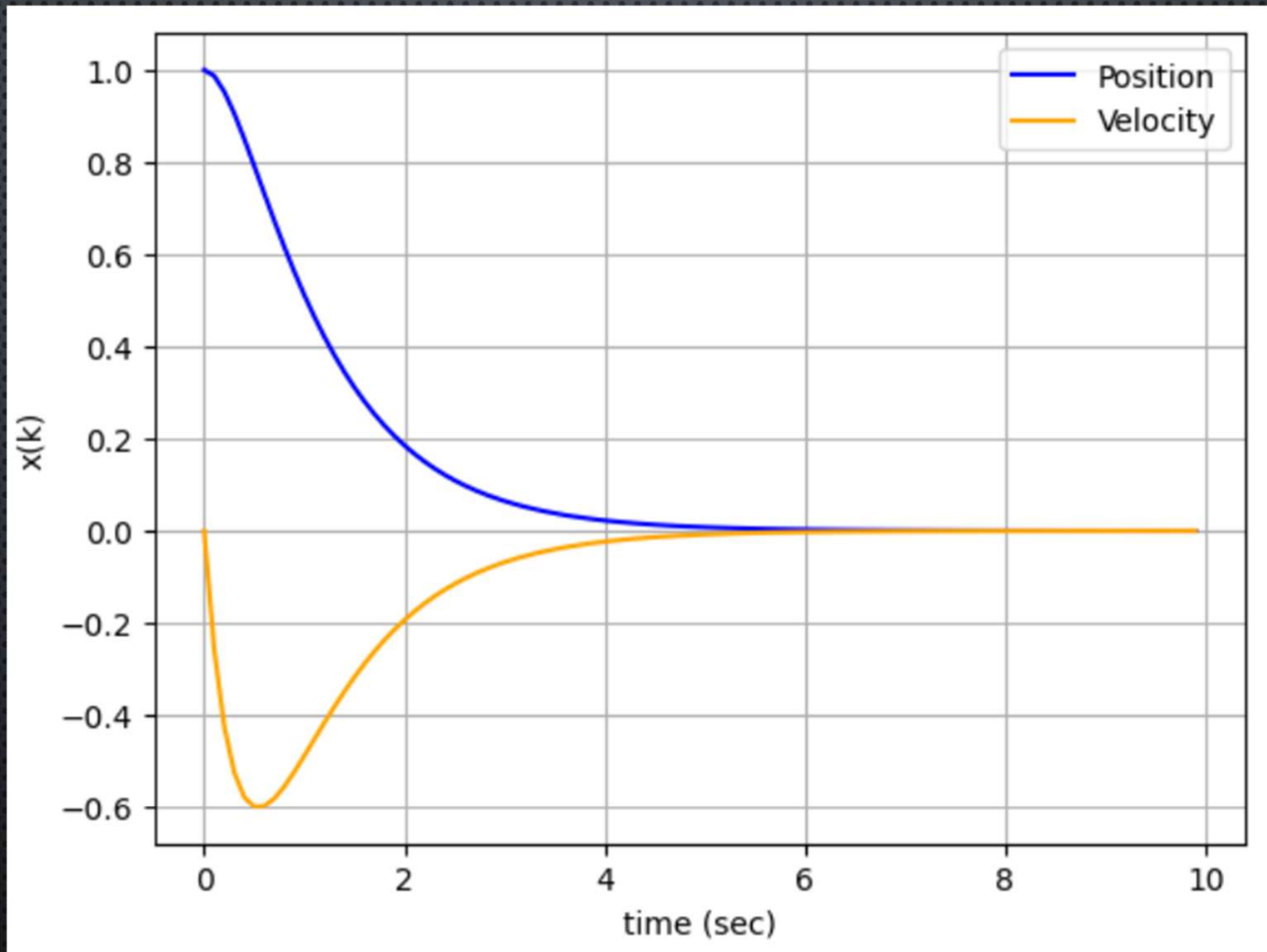
1. คาดเดาค่าเริ่มต้น $u_{1:N-1}$ โดยอาจเริ่มจากค่าศูนย์ทั้งหมด
2. จำลองโดยการแผ่ออก (rollout) ในทิศทางข้างหน้า เพื่อหาค่า $x_{1:N}$
3. คำนวณย้อนกลับเพื่อหาค่า $\lambda, \Delta u$
4. แผ่ออกค่า Δu โดยมีการค้นหาตามเส้น
5. ไปที่ขั้นตอน 3. จนกว่าจะลู่เข้า

ตัวอย่าง 3.1 ต้องการใช้ตัวควบคุม LQR กับระบบปริพันธ์คู่ (double integrator)

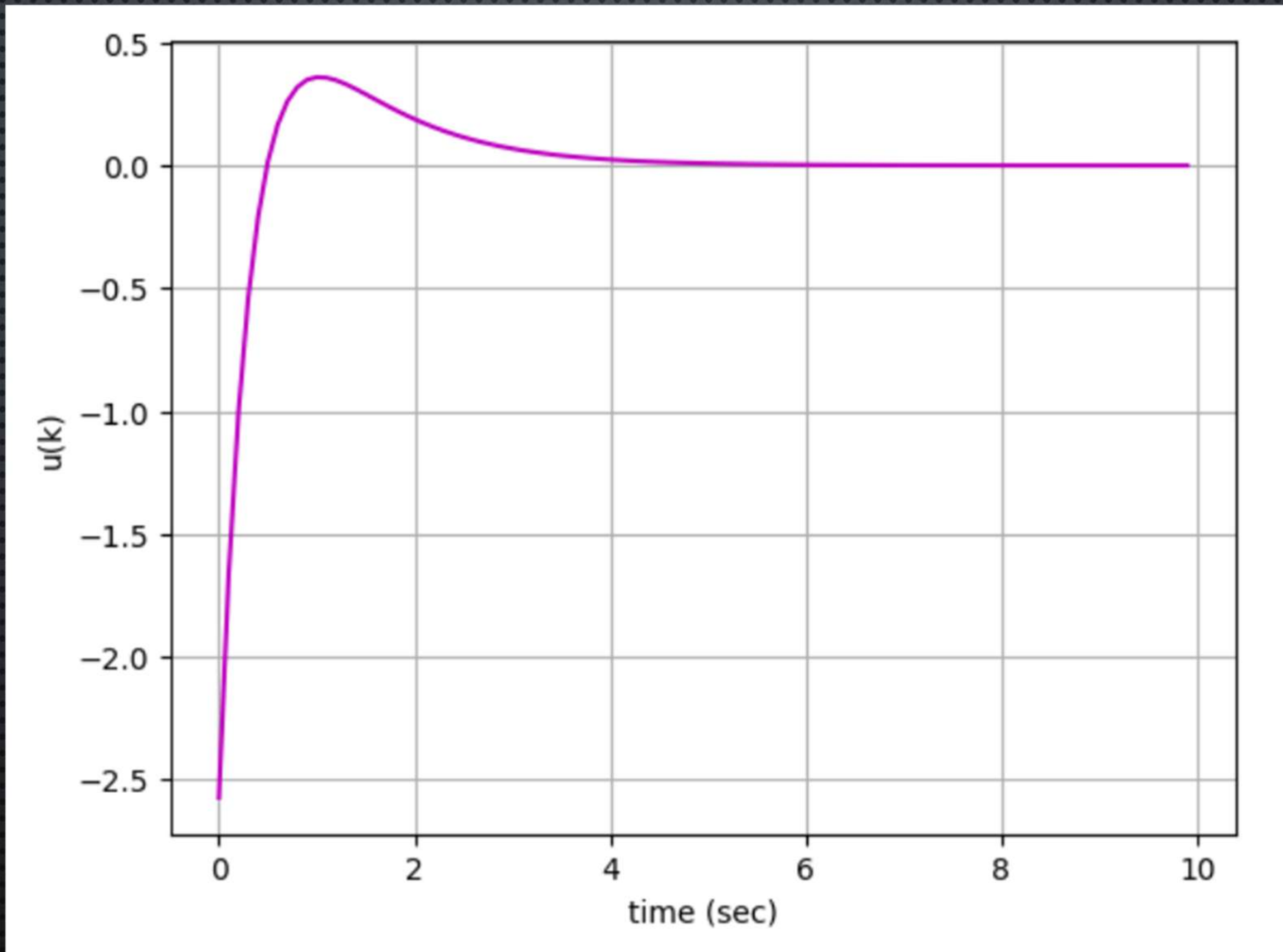
$$\dot{x} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \qquad (3.23)$$

เราอาจมองระบบนี้เป็นเสมือนก้อนอิฐบล็อกที่เลื่อนไปบนพื้นน้ำแข็งโดยไม่มีแรงเสียดทาน สถานะ $[q, \dot{q}]^T$ คือค่าตำแหน่งและความเร็วของก้อนอิฐ และ u คือแรงที่กระทำ เราสามารถแปลงสมการปริภูมิสถานะนี้เป็นระบบดีสครีตได้เมื่อกำหนดคาบเวลาการสุ่ม h

$$x_{k+1} = Ax_k + Bu_k = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_k \\ \dot{q}_k \end{bmatrix} + \begin{bmatrix} \frac{1}{2}h^2 \\ h \end{bmatrix} u_k \quad (3.24)$$



รูปที่ 3.1 แนววิถีของสถานะ x จากตัวควบคุม LQR ในตัวอย่าง 3.1 (วิธีการยิงโดยอ้อม)



รูปที่ 3.2 แนววิถีของเอาต์พุต u จากตัวควบคุม LQR ในตัวอย่าง 3.1 (วิธีการยิงโดยอ้อม)

3.2.2 การหาคำตอบโดยการโปรแกรมกำลังสอง

จากปัญหา LQR (3.18) ต้องการจัดรูปใหม่เพื่อใช้วิธี QP กำหนดสถานะเริ่มต้น x_1 ซึ่งเป็นค่าที่เราเลือก มิใช่ตัวแปรการตัดสินใจ ดังนั้นนิยามเวกเตอร์ของตัวแปรการตัดสินใจดังนี้

$$z = \begin{bmatrix} u_1 \\ x_2 \\ u_2 \\ \vdots \\ x_{N-1} \\ u_{N-1} \\ x_N \end{bmatrix} \tag{3.25}$$

นิยามเมทริกซ์น้ำหนัก

$$H = \begin{bmatrix} R_1 & 0 & \cdots & \cdots & 0 \\ 0 & Q_2 & 0 & \cdots & 0 \\ 0 & 0 & R_2 & 0 & 0 \\ 0 & \cdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & Q_N \end{bmatrix} \tag{3.26}$$

จะเห็นว่าเราสามารถแทนฟังก์ชันมูลค่าใน (3.18) ในรูปที่กระชับขึ้นดังนี้

$$J = \frac{1}{2} z^T H z \tag{3.27}$$

ใช้การจัดรูปลักษณะเดียวกันนี้กับเงื่อนไขบังคับที่เป็นพลวัตเชิงเส้นใน (3.18)

$$\begin{bmatrix} B_1 & -I & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & A_2 & B_2 & -I & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \cdots & 0 \\ 0 & \cdots & \cdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & A_{N-1} & B_{N-1} & -I \end{bmatrix} \begin{bmatrix} u_1 \\ x_2 \\ u_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} -A_1 x_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.28}$$

นิยามเมทริกซ์ทางด้านซ้ายของ (3.28) คือ C และเวกเตอร์ด้านขวาคือ d ดังนั้นจะได้เงื่อนไขบังคับสมการเป็น

$$Cz = d \tag{3.29}$$

โดยใช้การจัดรูปใหม่นี้ จะเขียนปัญหา LQR ได้เป็นโจทย์ QP

$$\min_z \frac{1}{2} z^T H z$$

$$s.t. \;\; Cz - d = 0 \tag{3.30}$$

ซึ่งมีลากรานเจียน

$$L(z,\lambda) = \frac{1}{2} z^T H z + \lambda^T (Cz - d) \tag{3.31}$$

และเงื่อนไข KKT

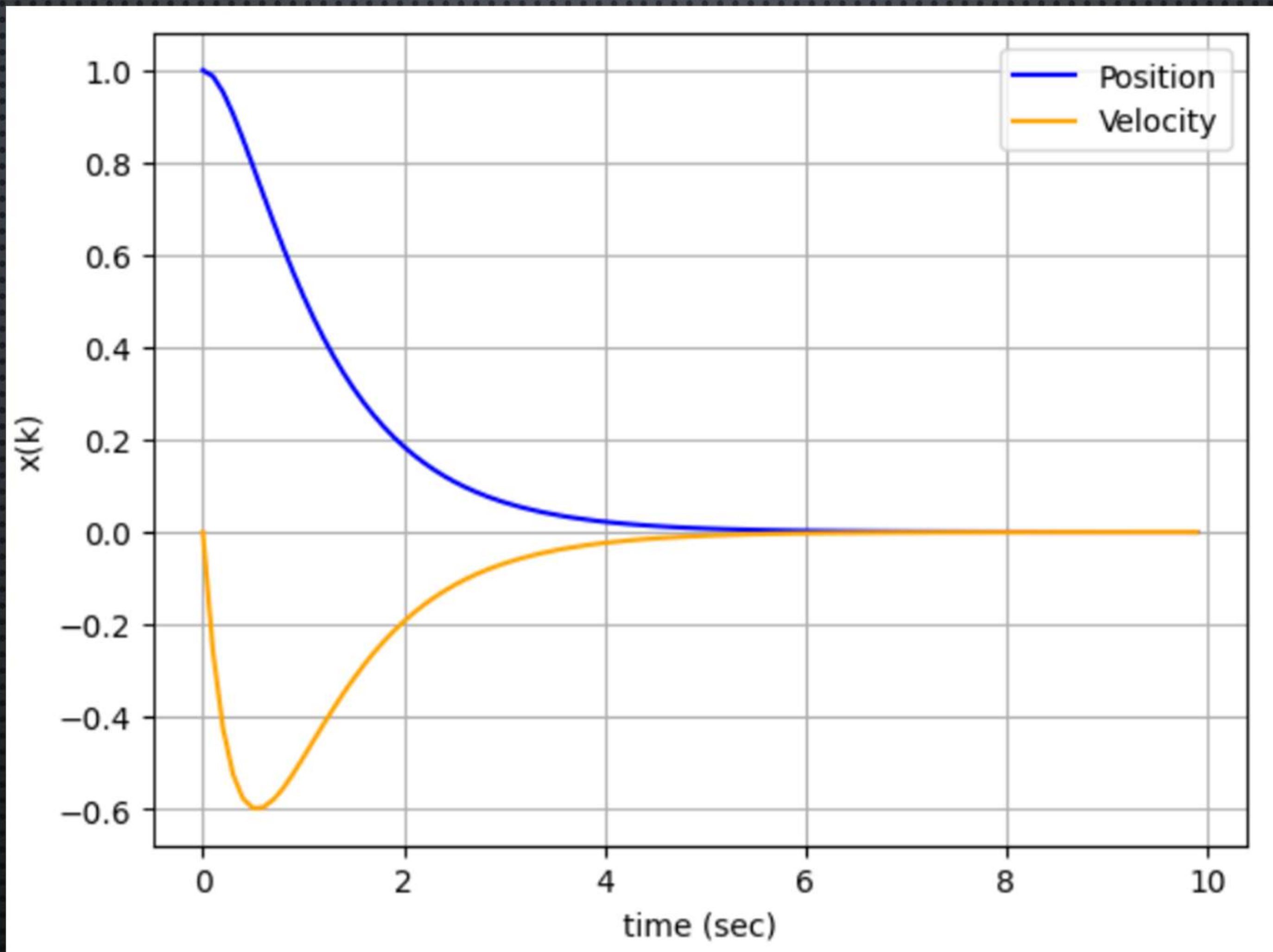
$$\nabla_z L = Hz + C^T \lambda = 0 \tag{3.32}$$

$$\nabla_\lambda L = Cz - d = 0 \tag{3.33}$$

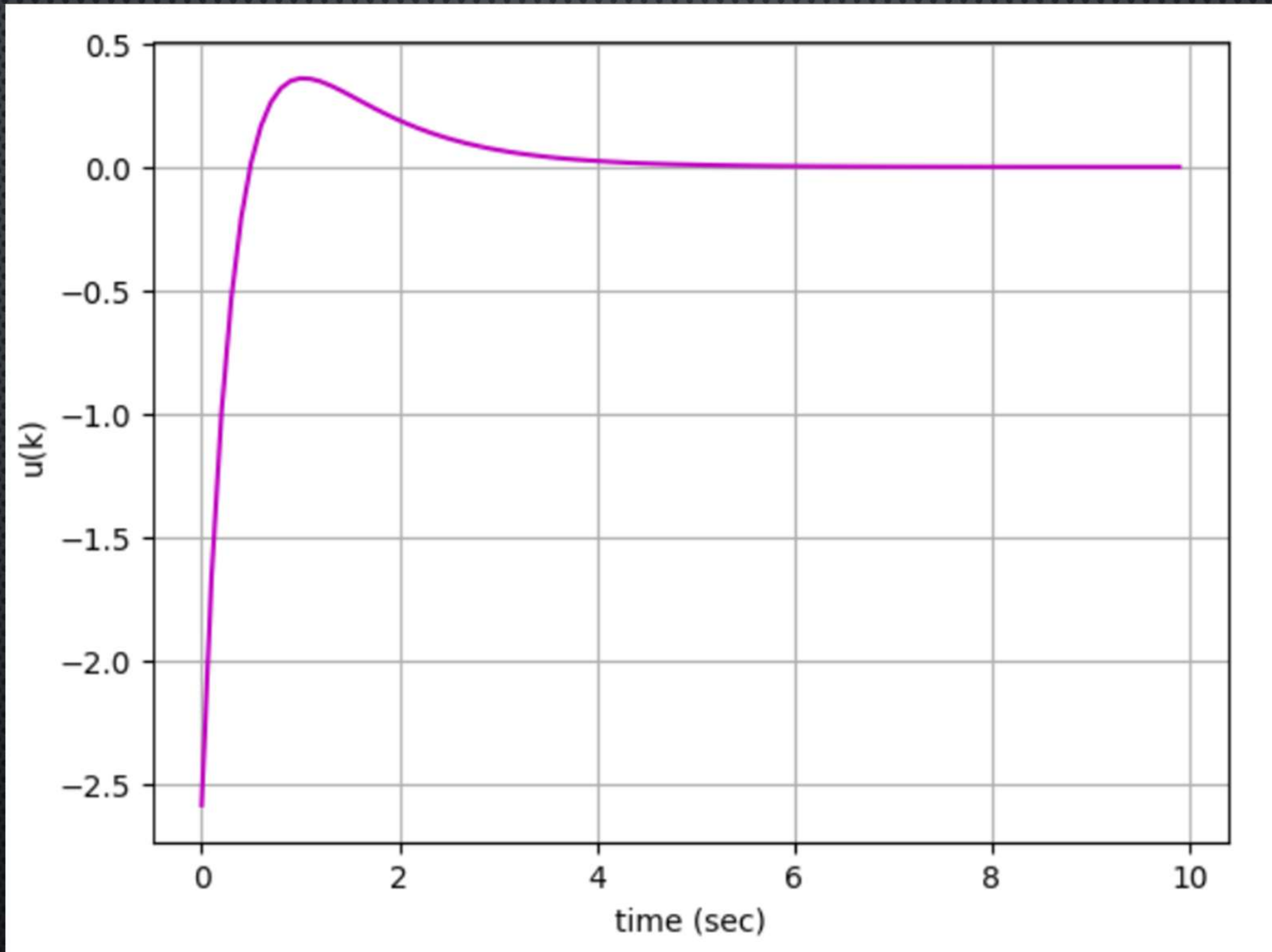
หรือเมื่อเขียนในรูปเมทริกซ์ได้เป็นดังนี้

$$\begin{bmatrix} H & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} z \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ d \end{bmatrix} \tag{3.34}$$

สังเกตว่า (3.34) เป็นสมการเมทริกซ์เชิงเส้นที่สามารถแก้สมการเพื่อหาคำตอบได้ในทันที แตกต่างจากวิธีการยิงโดยอ้อมที่ต้องมีการวนซ้ำจำนวนมาก ดังนั้นจึงเป็นวิธีการที่ดีกว่า



รูปที่ 3.3 แนววิถีของสถานะ x จากตัวควบคุม LQR ในตัวอย่าง 3.2 (วิธี QP)



รูปที่ 3.4 แนววิถีของเอาต์พุตควบคุม u จากตัวควบคุม LQR ในตัวอย่าง 3.2 (วิธี QP)

3.2.3 การหาคำตอบโดยสมการริกคาติ

จากวิธีการยิงโดยอ้อมและวิธี QP จะพบว่าได้แนววิถีคำตอบในรูปของวงเปิด ทำให้ไม่สามารถประยุกต์ใช้เป็นตัวควบคุมป้อนกลับได้โดยตรง ดังนั้นในหัวข้อนี้จะลองเจาะลึกลงในสมการ (3.34) ที่มีลักษณะเป็นมากเลขศูนย์ และมีแบบรูปเฉพาะที่สามารถอนุพัทธ์เป็นระบบสมการที่มีประโยชน์อย่างยิ่งสำหรับการควบคุมเหมาะที่สุด

เพื่อความง่ายในการศึกษา พิจารณา (3.34) สำหรับขั้นเวลาจำนวนน้อยเช่น 4 ขั้น และเมทริกซ์ R, Q, Q_N, A, B เป็นค่าคงที่

$$\left[\begin{array}{cccccc|ccc} R & 0 & 0 & 0 & 0 & 0 & B^T & 0 & 0 \\ 0 & Q & 0 & 0 & 0 & 0 & -I & A^T & 0 \\ 0 & 0 & R & 0 & 0 & 0 & 0 & B^T & 0 \\ 0 & 0 & 0 & Q & 0 & 0 & 0 & -I & A^T \\ 0 & 0 & 0 & 0 & R & 0 & 0 & 0 & B^T \\ 0 & 0 & 0 & 0 & 0 & Q_N & 0 & 0 & -I \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ B & -I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A & B & -I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A & B & -I & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} u_1 \\ x_2 \\ u_2 \\ x_3 \\ u_3 \\ x_4 \\ -- \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -- \\ -Ax_1 \\ 0 \\ 0 \end{bmatrix} \quad (3.35)$$

สังเกตตัวชี้ของ λ ที่เริ่มจาก 2 (ดูได้จากสมการ (3.3)) โดยโครงสร้างของระบบสมการเมทริกซ์นี้ เราสามารถใช้วิธีแทนค่าย้อนหลังเพื่อหาคำตอบ ซึ่งสอดคล้องกับวิธีโดยอ้อมในหัวข้อ 3.2.1 เริ่มจากแถวล่างสุดของเมทริกซ์บล็อก H (แถวที่ 6 ของ (3.35)) เขียนเป็นสมการได้ดังนี้

$$Q_N x_4 - \lambda_4 = 0 \Rightarrow \lambda_4 = Q_N x_4 \tag{3.36}$$

แทนค่า λ_4 ลงในแถวที่ 5 ของ (3.35)

$$\begin{aligned} Ru_3 + B^T\lambda_4 = Ru_3 + B^T Q_N x_4 = 0 \\ \Rightarrow Ru_3 + B^T Q_N(Ax_3 + Bu_3) = 0 \\ \Rightarrow u_3 = -(R + B^T Q_N B)^{-1} B^T Q_N A x_3 \end{aligned} \tag{3.37}$$

จะเห็นว่า (3.37) อยู่ในรูปของการป้อนกลับสถานะ ซึ่งเป็นวัตถุประสงค์หนึ่งของการอนุพัทธ์นี้ในการจัดรูปแนววิถีวงเปิดเดิมเป็นระบบป้อนกลับ นิยาม

$$K_3 \triangleq (R + B^T Q_N B)^{-1} B^T Q_N A \tag{3.38}$$

เพื่อเขียน (3.37) ให้กระชับขึ้นเป็น

$$u_3 = -K_3 x_3 \tag{3.39}$$

ต่อมาพิจารณาแถวที่ 4 ของ (3.35)

$$Qx_3 - \lambda_3 + A^T \lambda_4 = 0 \tag{3.40}$$

แทนค่า λ_4 จาก (3.36)

$$Qx_3 - \lambda_3 + A^T Q_N x_4 = 0 \tag{3.41}$$

แทนค่าพลวัต $x_4 = Ax_3 + Bu_3$

$$Qx_3 - \lambda_3 + A^T Q_N (Ax_3 + Bu_3) = 0 \tag{3.42}$$

แทนค่า u_3 จาก (3.39) และจัดรูปสมการเพื่อหาคำตอบ λ_3

$$\lambda_3 = (Q + A^T Q_N (A - BK_3))x_3 \tag{3.43}$$

นิยาม

$$P_3 \triangleq (Q + A^T Q_N (A - BK_3)) \tag{3.44}$$

โดยวิธีการนี้เราจะได้อัลกอริทึมวนซ้ำย้อนหลังสำหรับ K และ P ที่มีเค้าโครงดังนี้

$$P_N = Q_N$$

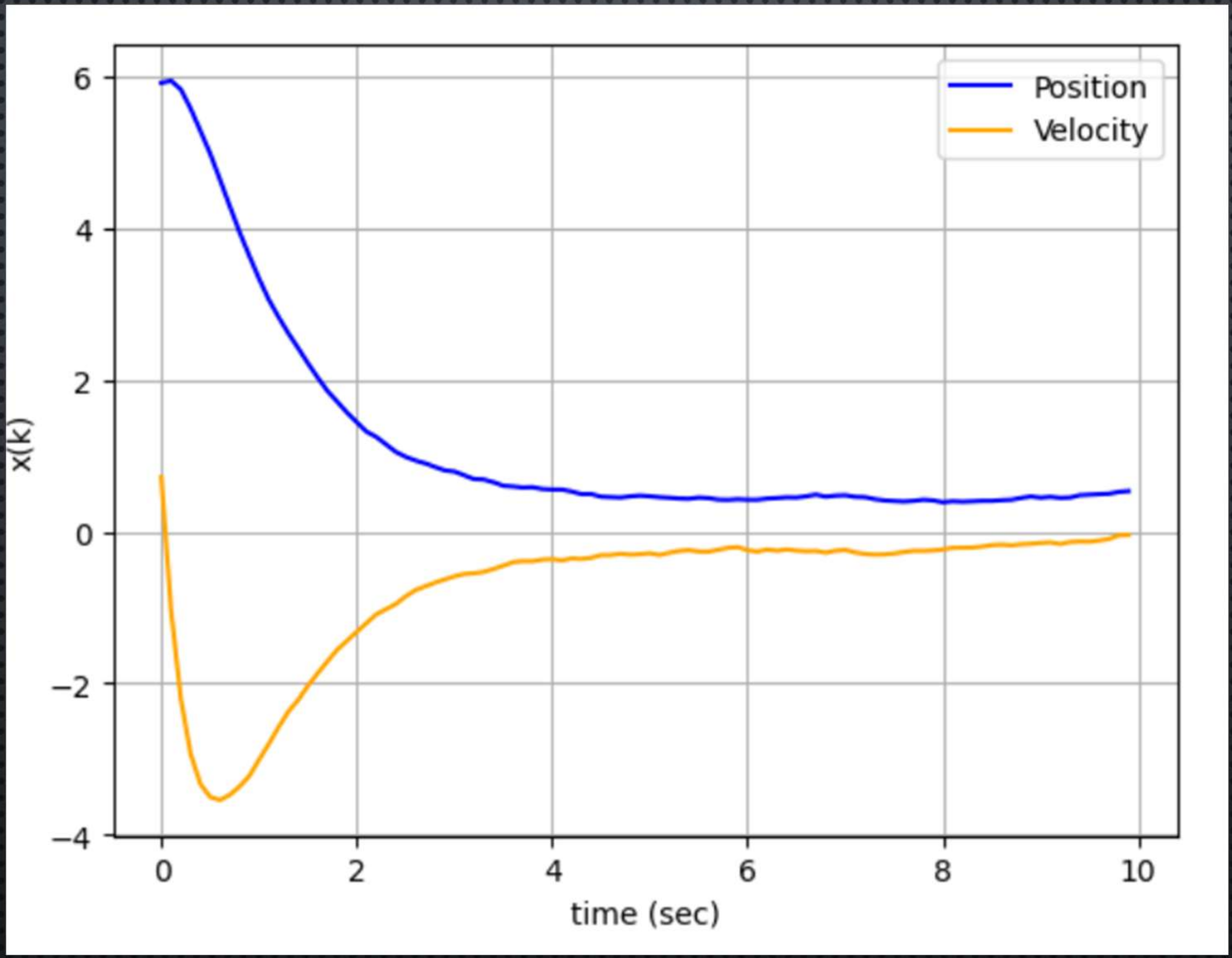
วนซ้ำย้อนหลัง k จาก $N-1$ ถึง 1:

$$K_k = (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A$$

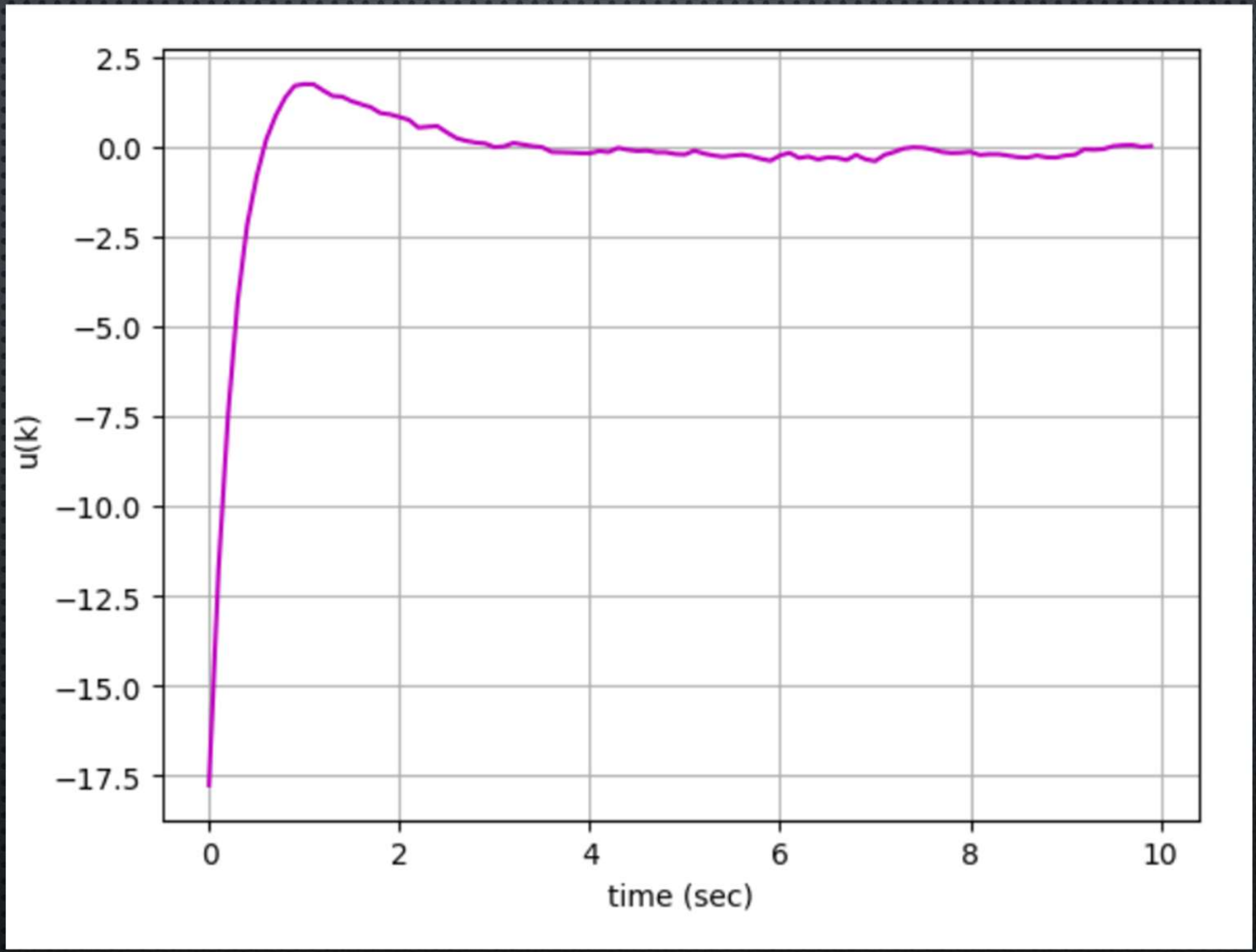
$$P_k = Q + A^T P_{k+1}(A - BK_k)$$

เรียกว่า *สมการริกคาติ (Riccati equation)* หรือการวนซ้ำริกคาติ สรุปได้ว่าเราสามารถแก้ปัญหา QP โดยการคำนวณสมการริกคาติย้อนหลัง ตามด้วยการแผ่ไปข้างหน้าเพื่อคำนวณ $x_{1:N}$ และ $u_{1:N-1}$

วิธีการนี้ให้ผลเช่นเดียวกับ 2 วิธีก่อนหน้านี้ (ดูรูปที่ 3.5, 3.6 ใน chapter3.ipynb)



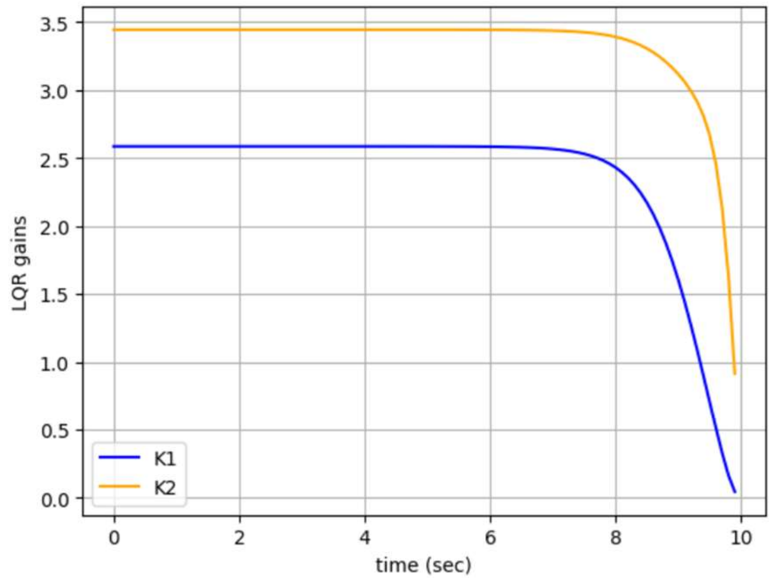
รูปที่ 3.7 แนววิถีของสถานะ x จากตัวควบคุม LQR เมื่อค่าเริ่มต้นเป็นค่าสุ่มและเพิ่มการรบกวนกับพลวัต



รูปที่ 3.8 แนววิถีของ u จากตัวควบคุม LQR เมื่อค่าเริ่มต้นเป็นค่าสุ่มและเพิ่มการรบกวนกับพลวัต

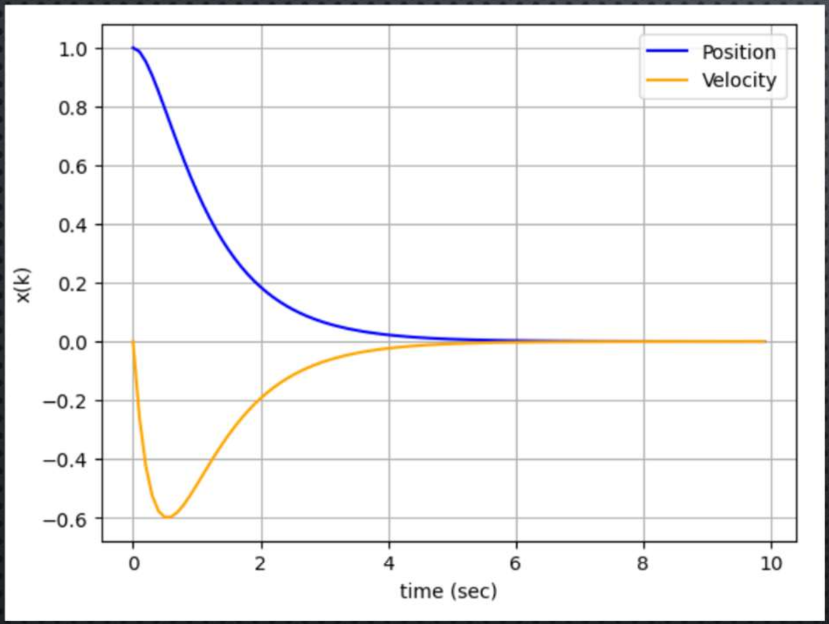
3.2.4 ตัวควบคุม LQR แบบแนวนอนอนันต์

ความหมายของตัวควบคุม LQR แบบแนวนอนอนันต์ (infinite horizon) คือเมื่อเราขยายแกนเวลาออกไปไม่มีที่สิ้นสุด ซึ่งสำหรับปัญหา LQR ที่ไม่แปรตามเวลา ค่าของ P_k และ K_k ที่คำนวณโดยการวนซ้ำริกคาติจะลู่เข้าสู่ค่าคงที่ P_∞, K_∞ ซึ่งสามารถแสดงตัวอย่างได้จากค่าที่เก็บไว้ในตัวอย่าง 3.3 รูปที่ 3.9 แสดงค่าของตัวควบคุมป้อนกลับสถานะ K_k เทียบกับเวลา โดยค่าเริ่มต้นจะอยู่ที่เวลาสิ้นสุด แต่เมื่อคำนวณวนซ้ำย้อนหลังไปช่วงเวลาหนึ่งจะเริ่มเห็นการลู่เข้าสู่ค่าคงที่

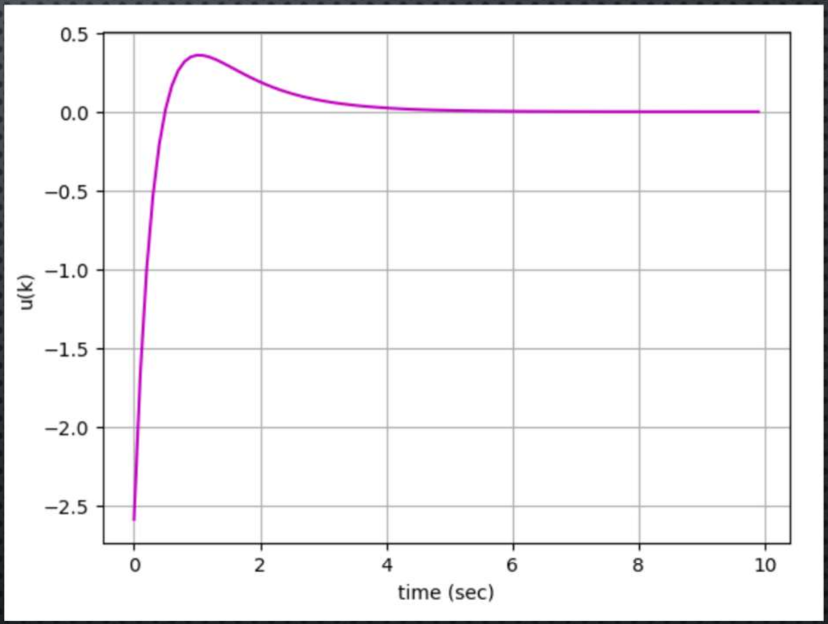


รูปที่ 3.9 การลู่เข้าสู่ค่าคงที่ของ K_k จากการวนซ้ำริกคาติย้อนหลัง

เมื่อทดลองใช้ตัวควบคุม Kinf ในการแผ่พลวัตข้างหน้าและพล็อตแนววิถีในรูปที่ 3.10, 3.11 เมื่อเปรียบเทียบกับผลในรูปที่ 3.5, 3.6 พบว่าแทบไม่มีความแตกต่างกัน



รูปที่ 3.10 แนววิถีของสถานะ x จากตัวควบคุมที่สังเคราะห์โดย dlqr()



รูปที่ 3.11 แนววิถีของเอาต์พุตควบคุม u จากตัวควบคุมที่สังเคราะห์โดย dlqr()

3.3 คุณสมบัติควบคุมได้ของระบบพลวัต

อธิบายโดยภาพรวม การที่เราจะควบคุมพลวัตใด ๆ ได้นั้น เอาต์พุตตัวควบคุมต้องสามารถเข้าถึงทุกสถานะของพลวัตนั้นได้ไม่ว่าจะช้าเร็วเพียงใด คุณสมบัตินี้มิได้ขึ้นกับจำนวนของตัวกระตุ้น (actuator) ตัวอย่างเช่นในระบบหุ่นยนต์ที่มีตัวกระตุ้นน้อยกว่าข้อต่อเรียกว่าถูกกระตุ้นไม่เต็มที่ (underactuated) แต่มิได้หมายความว่าจะควบคุมไม่ได้ เพราะอาจเป็นเพียงเอาต์พุตควบคุมจะไม่สามารถเข้าถึงสถานะบางตัวได้ในทันทีเท่านั้น

ความควบคุมได้จะชัดเจนขึ้นในระบบเชิงเส้นที่บรรยายในรูปเมทริกซ์ $x_{k+1} = Ax_k + Bu_k$ เมื่อใช้การป้อนกลับสถานะ $u_k = -Kx_k$ เช่นเมื่อ K เป็นคำตอบของ LQR ค่าลักษณะเฉพาะของเมทริกซ์วงปิด $A - BK$ จะต้องมีขนาดน้อยกว่าหนึ่งทุกตัวระบบป้อนกลับจึงเสถียร แม้ในกรณีพลวัตระบบวงเปิดอาจไม่เสถียร ดังนั้นจะเห็นว่าหากเราไม่สามารถย้ายค่าลักษณะเฉพาะตัวที่ไม่เสถียรให้อยู่ในวงกลมรัศมีหนึ่งหน่วยได้โดยอินพุตควบคุม ระบบป้อนกลับไม่มีทางจะเสถียรได้ กล่าวคือระบบนี้ไม่สามารถควบคุมได้ จึงไม่สามารถใช้วิธี LQR

หมายเหตุ : คุณสมบัติที่อ่อนกว่าเรียกว่า ทำให้เสถียรได้ (stabilizable) หมายถึงระบบวงเปิดที่ยอมให้มีค่าลักษณะเฉพาะบางตัวที่เคลื่อนย้ายไม่ได้โดยอินพุตควบคุม แต่ค่าลักษณะเฉพาะตัวนั้นจะต้องเสถียร เนื้อหาในหัวข้อนี้จะกล่าวถึงเฉพาะความสามารถควบคุมได้เท่านั้น

ความควบคุมได้จะบรรยายได้ชัดเจนขึ้นสำหรับกรณีระบบเชิงเส้นไม่แปรตามเวลา (linear time-invariant เรียกย่อว่า LTI) เพราะเมทริกซ์ A, B ในพลวัตเป็นค่าคงที่ สำหรับสถานะเริ่มต้น x_0 ใดๆ พลวัตของ x_N คำนวณได้จากการแทนค่าพลวัตจากสถานะปลายย้อนไปสู่ต้นดังนี้

$$
\begin{aligned}
x_N &= Ax_{N-1} + Bu_{N-1} \\
&= A(Ax_{N-2} + Bu_{N-2}) + Bu_{N-1} \\
&= \qquad \vdots \\
&= A^N x_0 + A^{N-1}Bu_0 + A^{N-2}Bu_1 + \cdots + Bu_{N-1}
\end{aligned} \tag{3.45}
$$

สามารถเขียนเป็นรูปสมการเมทริกซ์

$$
x_N = \begin{bmatrix} B & AB & A^2B & \cdots & A^{N-1}B \end{bmatrix} \begin{bmatrix} u_{N-1} \\ u_{N-2} \\ u_{N-3} \\ \vdots \\ u_0 \end{bmatrix} + A^N x_0 \tag{3.46}
$$

นิยามเมทริกซ์แรกทางด้านขวาของ (3.46)

$$\phi \triangleq \begin{bmatrix} B & AB & A^2B & \cdots & A^{N-1}B \end{bmatrix} \tag{3.47}$$

ในระบบเชิงเส้นเราสามารถหาคำตอบสำหรับกรณี $x_N = 0$ โดยไม่เสียความเป็นทั่วไป ต้องการหาคำตอบสำหรับ $u_{0:N-1}$ ใน (3.47) แต่เนื่องจากเมทริกซ์ ϕ เป็นแบบกว้าง ดังนั้นต้องใช้วิธีกำลังสองน้อยสุดเพื่อหาคำตอบดังนี้

$$\begin{bmatrix} u_{N-1} \\ u_{N-2} \\ u_{N-3} \\ \vdots \\ u_0 \end{bmatrix} = [\phi^T(\phi\phi^T)^{-1}](x_N - A^N x_0) \tag{3.48}$$

โดยเมทริกซ์ $\phi^T(\phi\phi^T)^{-1}$ เรียกว่าตัวผกผันเทียม (pseudo-inverse)

จากพื้นฐานพีชคณิตเชิงเส้น การที่เมทริกซ์ $\phi\phi^T$ ขนาด $n \times n$ จะมีตัวผกผันได้นั้นจะต้องไม่เป็นเอกฐาน คือมีค่าลำดับชั้นเท่ากับ n คือมิติของสถานะ ซึ่งเมื่อแปลงเป็นเงื่อนไขของ ϕ จะได้ว่า

$$rank(\phi) = n, \;\; n = dim(x) \tag{3.49}$$

สังเกตว่าเมทริกซ์ ϕ เป็นแบบกว้าง มีจำนวนแถว n แต่มีคอลัมน์จำนวนมากเท่ากับ N คือจำนวนขั้นเวลา แต่เมื่ออาศัยหลักการของทฤษฎีเคย์เลย์-แฮมิลตัน (Cayley-Hamilton theorem) ที่กล่าวว่า สำหรับเมทริกซ์ A ขนาด $n \times n$ ค่ายกกำลังของเมทริกซ์ A^N สามารถเขียนในรูปการรวมเชิงเส้น (linear combination) ของค่ายกกำลัง $0, 1, \ldots, n-1$ ของ A

$$A^N = \sum_{k=0}^{n-1} \alpha_k A^k \tag{3.50}$$

ดังนั้นใน (3.47) การเพิ่มคอลัมน์ให้กับ ϕ หลังจาก $A^{n-1}B$ ไม่สามารถเพิ่มค่าลำดับชั้นมากขึ้น เราจึงนิยามเมทริกซ์ความสามารถควบคุมได้ (controllability matrix) ขนาด $n \times n$ ได้ดังนี้

$$\Phi \triangleq \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{3.51}$$

3.4 สรุปท้ายบท

เนื้อหาในบทนี้เป็นการเริ่มต้นศึกษาการควบคุมเหมาะที่สุด โดยยกตัวอย่างโจทย์ปัญหา LQR พื้นฐานที่ยังไม่มีการจำกัดค่าเอาต์พุตตัวควบคุมและสถานะ เรานำเสนอ 3 วิธีในการหาคำตอบ เริ่มจากวิธีการยิงโดยอ้อม อาศัยหลักการค่าต่ำสุดของพอนเทรียกิน ซึ่งเป็นวิธีที่ใช้ในยุคต้นของการควบคุมเหมาะที่สุด โดยมีการคำนวณในทิศทางข้างหน้าและย้อนหลังคล้ายกับการเรียนรู้เชิงลึกในปัจจุบัน ดังนั้นจะประสบปัญหาเช่นเดียวกันกับการมีค่าสูงและจางหายของเกรเดียนต์ในโครงข่ายประสาทเทียม ใช้เวลานานในการประมวลผล ทำให้ปัจจุบันได้รับความนิยมน้อย

วิธีการที่สองคือการจัดรูปเป็นปัญหา QP สร้างเมทริกซ์ขนาดใหญ่ที่เป็นแบบมากเลขศูนย์ และหาคำตอบสมการเชิงเส้นได้ในทันทีซึ่งจะเร็วกว่าวิธีแรก ในการเขียนโค้ดจะต้องไม่หาคำตอบจากเมทริกซ์ที่สร้างเป็นแบบหนาแน่น แต่อาศัยแพ็กเกจเสริมเช่น scipy เพื่อใช้ประโยชน์จากโครงสร้างมากเลขศูนย์

คำตอบที่ได้จากวิธีการที่ 1 และ 2 จะอยู่ในรูปแนววิถีของ x และ u ที่เป็นระบบวงเปิด แตกต่างจากวิธีวนซ้ำริกคาติ ที่จะได้ตัวควบคุมป้อนกลับสถานะ $u_k = Kx_k$ สามารถใช้กับค่าเริ่มต้นสถานะใดๆ รวมถึงขจัดการรบกวนในระบบ นอกจากนั้นวิธีนี้ยังมีความซับซ้อนในการคำนวณแปรผันเชิงเส้นกับจำนวนขั้นเวลา เมื่อเปรียบเทียบกับวิธี QP ที่ความซับซ้อนเป็นกำลังสามของจำนวนขั้นเวลา

ในส่วนท้ายของบทกล่าวถึงตัวควบคุมแบบแนวนอนอนันต์ ซึ่งสามารถใช้แพ็กเกจเสริมเช่น Python Control library ในการคำนวณตัวควบคุมป้อนกลับสถานะ และสุดท้ายคือคุณสมบัติการควบคุมได้ของพลวัตที่ควรต้องตรวจสอบก่อนแก้ปัญหา LQR

บรรณานุกรม

1. Z. Manchester et.al. 16-745 Optimal Control & Reinforcement Learning, Course materials, Lecture 7, 8, 9. Carnegie Mellon University. 2024,2025.

2. R. Tedrake. Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation .Course Notes for MIT 6.832. 2023.

3. D.P. Bertsekas. Reinforcement Learning and Optimal Control. MIT Press. 2019.