รายการอ้างอิง

1. Craig, J. J. **Introduction to robotics: Mechanics and control** 2nd ed. Reading, MA: Addison-Wesley Publishing Co., 1989.
2. Craig, J. J., Hsu, P., and Sastry, S. S. Adaptive control of mechanical manipulators. **IEEE Int. Conf. on Robotic and Automation**, San Francisco, CA, 1986, quoted in 19, 23.
3. Dubowsky, S., and Desforges, D. T. The application of model reference adaptive control to robotic manipulators. **ASME J. of Dyn. Syst., Meas., and Contr.** 101 (1979): 193-220.
4. Fu, K. S., Gonzales, R. C., and Lee, C. S. G. **Robotics: Control, sensing, vision, and intelligence.** New York: McGrow-Hill, 1987.
5. Goldenberg, A. A., Apkarian, J. A., and Smith, H. W. An approach to adaptive control of robot manipulators using the Computed Torque technique. **ASME J. of Dyn. Syst., Meas., and Contr.** 111 (March 1989): 1-8.
6. Gu, K., and Tongue, B. H. A new strategy for adaptive motion control of robots. **ASME J. of Dyn. Syst., Meas., and Contr.** 112 (September 1990): 410-416.
7. Ioannou, P. A., and Datta, A. Robust adaptive control: A unified approach. **Proceedings of the IEEE** 79 (December 1991): 1736-1768.
8. Johansson, R. Adaptive control of robot manipulator motion. **IEEE Trans. on Robotics and Automation** 6 (August 1990): 483-490.
9. Koivo, A. J., and Guo, T.-H. Adaptive linear controller for robotic manipulators. **IEEE Trans. on Automatic Control** 28 (1983): 162-171.
10. Kuo, C. Y., and Wang, S-P. T. Nonlinear robust industrial robot control. **ASME J. of Dyn. Syst., Meas., and Contr.** 111 (March 1989): 24-30.
11. Leal, R. L., and De wit, C. C. Passivity based adaptive control for mechanical manipulators using LS-type estimation. **IEEE Trans. on Automatic Control** 35 (1990): 1363-1365.
12. Lee, C. S. G., and Chung, M. J. An adaptive control strategy for mechanical manipulator. **IEEE Trans. on Automatic Control** 29 (1984): 837-840.
13. Liu, M.-H., and Lin, W. Multivariable self-tuning control with decoupling for robotic manipulators.

IEE **Proc.-D: Control Theory and Application** 135 (January 1988): 43-48.

14. Narendra, K. S., and Annaswamy, A. M. **Stable adaptive system.** Englewood Cliffs, NJ: Prentice Hall, 1989.
15. Sage, A. P., and White III, C. C. **Optimum system control.** Englewood Cliffs, NJ: Prentice Hall, 1977.
16. Schilling, R. J. **Fundamentals of robotics: Analysis and control.** Englewood Cliffs, NJ: Prentice Hall, 1990.
17. Slotine, J.-J. E., and Li, W. Adaptive manipulator control : A case study. **IEEE Trans. on Automation Control** 33 (November 1988): 995-1003.
18. Slotine, J.-J. E., and Li, W. **Applied nonlinear control.** Englewood Cliffs, NJ: Prentice Hall, 1991.
19. Spong, M. W., Ortega, R. On adaptive inverse dynamics control of rigid robots. **IEEE Trans. on Automatic Control** 35 (January 1990): 92-95.
20. Spong, M. W., Ortega, R., and Kelly, R. Comment on "Adaptive manipulatorcontrol: A case study". **IEEE Trans. on Automatic Control** 35 (June 1990): 761-762.
21. Stoten, D. P. **Model reference adaptive control of manipulators.** Somerset, England: Research Studies Press, 1990.
22. Vukobratovic, M., and Stokic, D. **Control of manipulation.** Springer-Verlag, 1982, quoted in 10.
23. Walker, M. W. Adaptive control of manipulators containing closed kinematic loops. **IEEE Trans. on Robotics and Automation** 6 (February 1990): 10-19.
24. Young, K.-K. D. Controller design for a manipulator using theory of variable structure systems. **IEEE Trans. on syst., Man, Cybern.** SMC-8 (February 1978): 101-109.
25. Yeung, K. S., and Chen, Y. P. A new controller design for manipulators using the theory of variable structure systems **IEEE Trans. on Automatic Control** 33 (February 1988): 200-206.

ภาคผนวก

ภาคผนวก ก

ประเภทของแขนหุ่นยนต์

เราสามารถจัดแบ่งประเภทของแขนหุ่นยนต์ได้หลายวิธี ขึ้นอยู่ว่าเลือกใช้กฎเกณฑ์ในการจัดแบ่งอย่างไร ในภาคผนวก ก นี้ เราจะจัดแบ่งประเภทของแขนหุ่นยนต์ตามลักษณะทางเรขาคณิตของเส้นขอบของบริเวณการทำงาน (work-envelope geometrics) ซึ่งประกอบด้วยตำแหน่งทั้งหมดในพื้นที่สามมิติที่แขนหุ่นยนต์สามารถเข้าถึง หรือกล่าวในอีกลักษณะหนึ่งก็คือ จัดแบ่งตามลักษณะการทำงานของดีกรีออฟฟรีดอมส่วนหลัก (major degrees of freedom, major DOF) แต่ก่อนที่จะพิจารณาประเภทต่าง ๆ ของแขนหุ่นยนต์ เราจะทำความเข้าใจเกี่ยวกับรายละเอียดของดีกรีออฟฟรีดอมของแขนหุ่นยนต์กันสักเล็กน้อย

<u>การพิจารณาดีกรีออฟฟรีดอมของแขนหุ่นยนต์</u>

ในขณะทำงานข้อต่อทั้งหมดของแขนหุ่นยนต์จะช่วยกันทำหน้าที่อยู่สองประการ นั่นคือ การทำให้ตำแหน่งของจุดข้อมือ (wrist point) เข้าถึงตำแหน่งใด ๆ ในบริเวณการทำงาน (work space) ซึ่งเป็นพื้นที่สามมิติ และการปรับหมุนทิศทางของเครื่องมือหรือส่วนมือของแขนหุ่นยนต์ เพื่อให้อยู่ในมุมที่เหมาะสมต่อกิจกรรมของงาน จากลักษณะดังกล่าวทำให้เราแบ่งดีกรีออฟฟรีดอมออกเป็นส่วนหลัก ๆ ได้ 2 ส่วน คือ

1. <u>Major DOF</u>

   เป็น DOF 3 ส่วนแรกที่ใช้ในการจัดวางตำแหน่งของจุดข้อมือ ให้ตรงกับตำแหน่งที่ต้องการ เนื่องจากตำแหน่งใด ๆ ในขณะทำงานมักจะอ้างอิงในลักษณะของตำแหน่งในพื้นที่สามมิติ ดังนั้นเราต้องการข้อต่อเพียง 3 ข้อต่อเท่านั้น เพื่อใช้สำหรับการวางตำแหน่งของจุดข้อมือ

2. <u>Minor DOF</u>

   เป็น DOF 3 ส่วนหลังที่ใช้ในการหมุนเครื่องมือให้อยู่ในมุมที่สอดคล้องกับวัตถุเป้าหมาย เนื่องจากการหมุนที่สมบูรณ์แบบในทุกทิศทางจะประกอบด้วยการหมุนใน 3 แกน เช่น X-Y-Z fixed angles หรือ Y-X Euler angles (Craig, 1989) เป็นต้น ดังนั้นเราต้องการข้อต่ออีก 3 ข้อต่อ เพื่อใช้สำหรับการหมุนเครื่องมือให้อยู่ในมุมที่ต้องการ

<u>การจัดประเภทของแขนหุ่นยนต์</u>

แขนหุ่นยนต์ส่วนใหญ่ที่ได้รับความนิยมจะมีลักษณะการเชื่อมโยงของลิงค์เป็นแบบอนุกรมวงเปิด และมีจำนวนของ DOF เท่ากับจำนวนข้อต่อ ลักษณะของแขนหุ่นยนต์เหล่านี้มักจะมีการแยกส่วนที่จุดข้อมือ (wrist-partitioned) กล่าวคือ มีการแบ่งข้อต่อจำนวน n-3 ข้อต่อสุดท้ายไว้สำหรับหมุนเครื่องมือ ข้อต่อในกลุ่มนี้จะมีแกนตัดผ่านกันที่จุดข้อมือ (wrist point) เพียงจุดเดียว ในส่วนของ 3 ข้อต่อแรกจะใช้วางตำแหน่งของจุดข้อมือ อย่างไรก็ตามเราอาจจะเคยเห็นแขนหุ่นยนต์บางรุ่นมีข้อต่อที่ใช้ในการวางตำแหน่งมากกว่า 3 ข้อต่อ เราเรียกแขนหุ่นยนต์ประเภทหลังนี้ว่า แขนหุ่นยนต์แบบรีดันแดนท์ (redundant robot) ข้อต่อที่เพิ่มขึ้นนี้จะมีประโยชน์ในการหลบหลีกสิ่งกีดขวางในขณะทำงาน ด้วยเหตุนี้การจัดแบ่งประเภทของแขนหุ่นยนต์ตามลักษณะทางเรขาคณิตของเส้นขอบของบริเวณการทำงาน จะพิจารณาจากบริเวณการทำงานซึ่งเกิดจากการเคลื่อนที่ของ 3 ข้อต่อแรกเท่านั้น

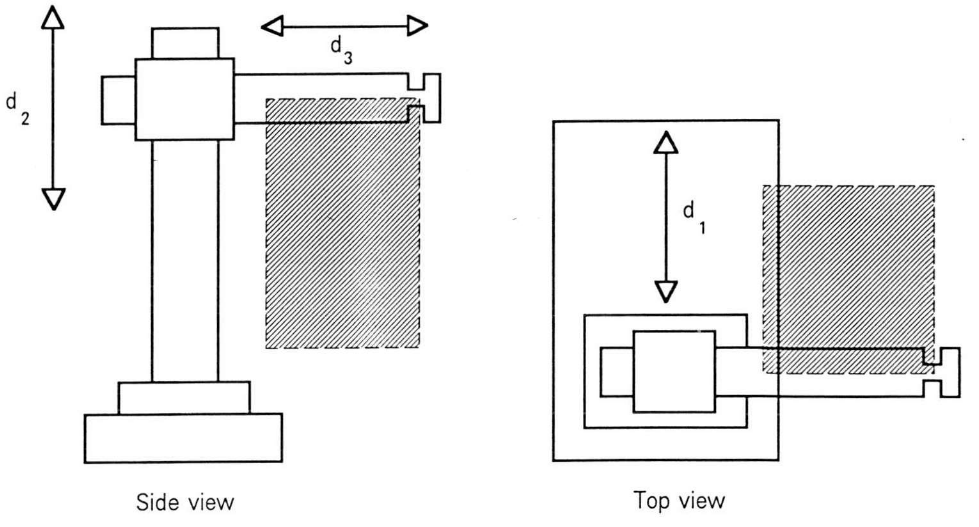
เนื่องจากแขนหุ่นยนต์ส่วนใหญ่ที่ใช้อยู่ทั่วไปในกระบวนการผลิต จะประกอบด้วยลิงค์ที่ออกแบบให้มีรูปร่างเป็นแท่งตรง ข้อต่อที่พบเห็นการใช้งานบ่อย ๆ ก็มีเพียง 2 ชนิด ดังที่แสดงในตารางที่ 2 ดังนั้นรูปแบบที่ต่างกันของแขนหุ่นยนต์จึงมีไม่สูงมากนัก ในปี ค.ศ. 1990 Schilling จัดแบ่งประเภทแขนหุ่นยนต์ที่ได้รับความนิยมตามลักษณะทางเรขาคณิตของเส้นขอบของบริเวณการทำงาน ได้เป็น 5 ประเภทดังนี้

1. แขนหุ่นยนต์แบบแกนตั้งฉาก (Cartesian robot)
2. แขนหุ่นยนต์แบบข้อปล้อง (Articulated robot)
3. แขนหุ่นยนต์แบบทรงกระบอก (Cylindrical robot)
4. แขนหุ่นยนต์แบบทรงกลม (Spherical robot)
5. แขนหุ่นยนต์แบบสการา (SCARA robot)

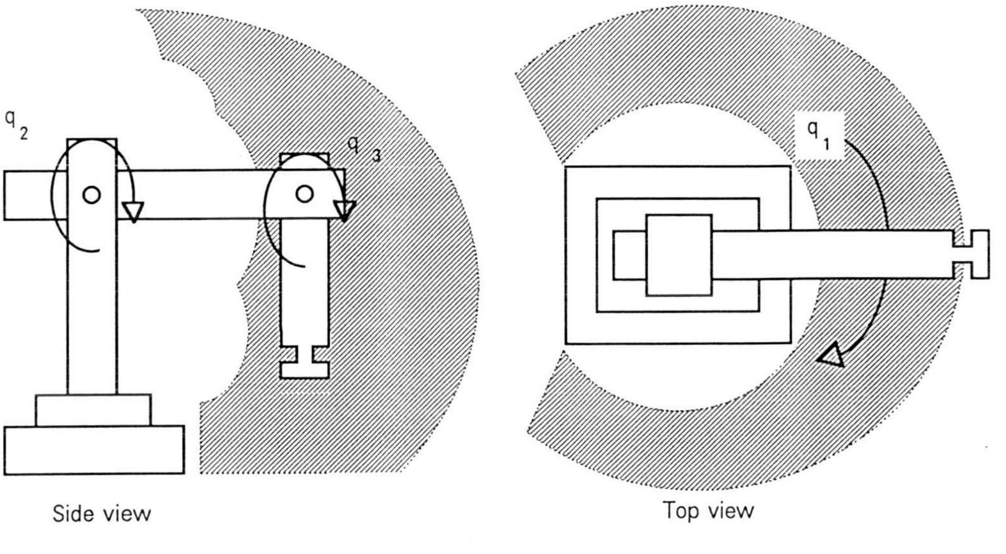
ประเภท	เครื่องหมาย	สัญลักษณ์	คำอธิบาย
ข้อต่อแบบหมุน (revolute joint)	R		หมุนรอบแกน
ข้อต่อแบบพริสแมทิค (prismatic joint)	P		เลื่อนไปตามแกน

ตารางที่ 2 แสดงประเภทต่าง ๆ ของข้อต่อที่ใช้งานบ่อย ๆ ในแขนหุ่นยนต์

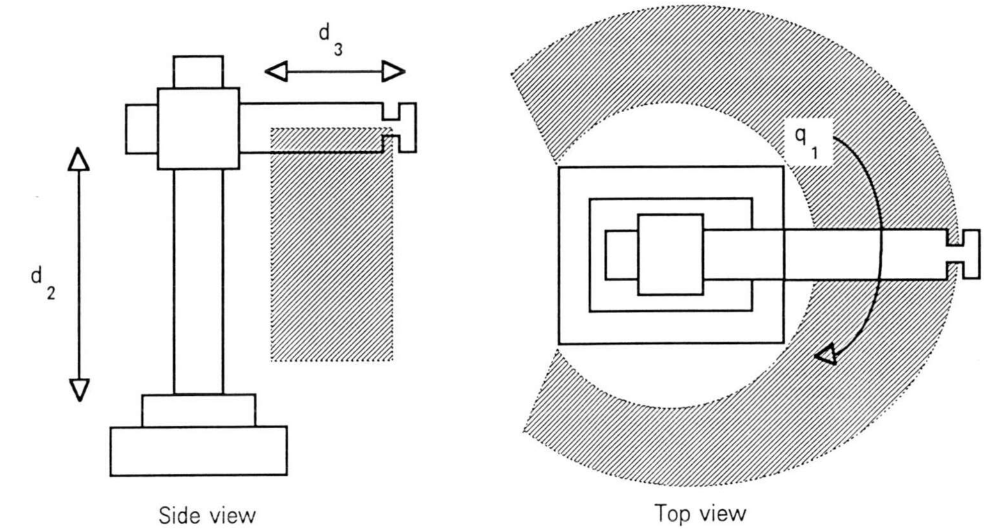
ลักษณะรูปร่าง และขอบเขตของบริเวณการทำงานของแขนหุ่นยนต์แต่ละประเภทจะแตกต่างกันไป ดังแสดงไว้อย่างละเอียดในภาพที่ ก



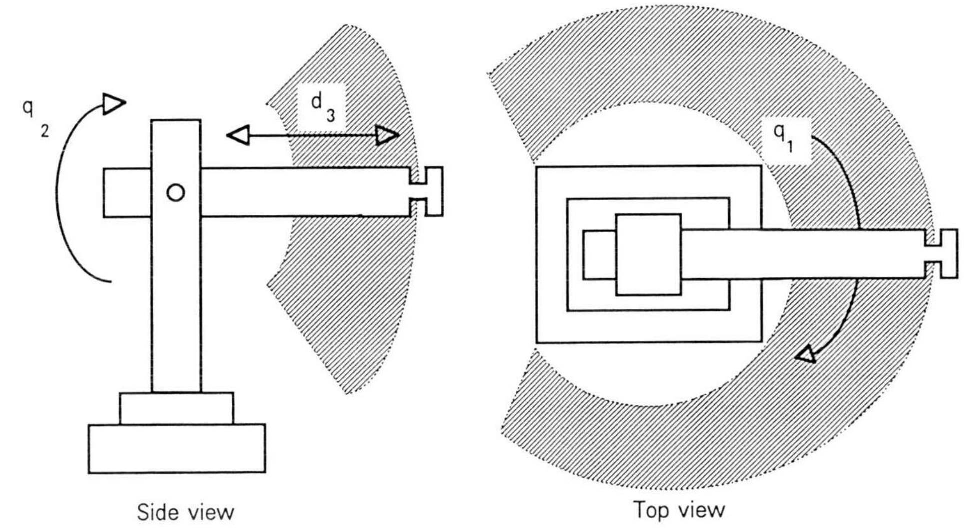
ภาพที่ ก.1 โครงสร้างและบริเวณการทำงานของแขนหุ่นยนต์แบบแกนตั้งฉาก



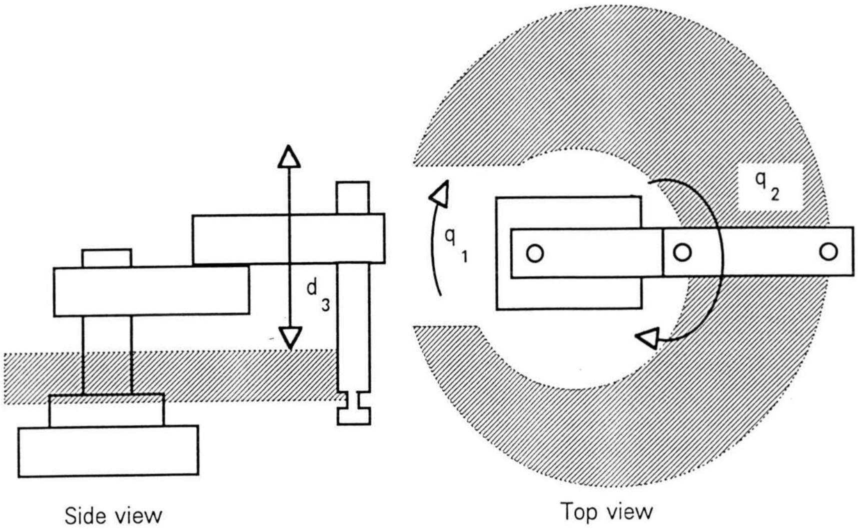
ภาพที่ ก.2 โครงสร้างและบริเวณการทำงานของแขนหุ่นยนต์แบบข้อปล้อง



ภาพที่ ก.3 โครงสร้างและบริเวณการทำงานของแขนหุ่นยนต์แบบทรงกระบอก



ภาพที่ ก.4 โครงสร้างและบริเวณการทำงานของแขนหุ่นยนต์แบบทรงกลม



ภาพที่ ก.5 โครงสร้างและบริเวณการทำงานของแขนหุ่นยนต์แบบสการา

ภาคผนวก ข

การใช้วิธีโปรแกรมพลวัตในตัวควบคุมชนิดปรับตัวเองโดยตรงที่ใช้คุณสมบัติแพสซีฟ

ทฤษฎีและแนวคิดของวิธีโปรแกรมพลวัต

1. รูปแบบของปัญหา

พิจารณาสมการทั่ว ๆ ไปของระบบ ดังที่แสดงในสมการ (ข.1)

$$\dot{x} = f[x(t),u(t),t] \tag{ข.1}$$

จุดมุ่งหมาย คือ การคำนวณหาสัญญาณควบคุม u ที่ทำให้ค่าของดรรชนีสมรรถนะ ซึ่งแสดงอยู่ในสมการ (ข.2) มีขนาดน้อยที่สุด

$$J = \theta[x(t_f),t_f] + \int_{t_0}^{t_f} \phi[x(t),u(t),t]dt \tag{ข.2}$$

2. การคำนวณหาสัญญาณควบคุม u ที่เหมาะที่สุด

กำหนดให้ฟังก์ชันหลักของแฮมิลตัน (Hamilton's principal function) เป็นดังสมการที่ (ข.3)

$$V_H[x(t_0),t_0] = \min_{U_t}\{J\} = \min_{U_t}\{\theta[x(t_f),t_f] + \int_{t_0}^{t_f} \phi[x(s),u(s),s]ds\} \tag{ข.3}$$

เมื่อ $V_H[x(t_f),t_f] = \theta[x(t_f),t_f]$

$U_t = \{u(s),t_0 \le s \le t_f\}$

โดยการใช้วิธีโปรแกรมพลวัต (Sage และ White, 1977) เราจะได้สมการแฮมิลตัน-จาโคบิ (Hamilton-Jacobi equation) ดังแสดงอยู่ในสมการ (ข.4)

$$-(\partial V_H[x(t),t]/\partial t) = \min_{u} \phi[x(t),u(t),t] + [\partial V_H[x(t),t]/\partial x]^T f[x(t),u(t),t] \tag{ข.4}$$

สำหรับทุก ๆ $t \in [t_0,t_f]$

จากฟังก์ชันแฮมิลโตเนียน (Hamiltonian function) ซึ่งแสดงอยู่ในสมการที่ (ข.5) เมื่อแทนลงในสมการแฮมิลตัน-จาโคบิ จะได้ผลตามที่แสดงอยู่ในสมการ (ข.6)

$$H[x(t),u(t),\partial V_H/\partial x] = \phi[x(t),u(t),t] + [\partial V_H[x(t),t]/\partial x]^T f[x(t),u(t),t] \tag{ข.5}$$

$$-(\partial V_H[x(t),t]/\partial t) \quad = \quad \min_u \{H[x(t),u(t),\partial V_H/\partial x]\} \qquad (ข.6)$$

จากสมการ (ข.6) จะเห็นว่า ค่าของสัญญาณควบคุมที่เหมาะที่สุด (u^*) จะต้องทำให้ฟังก์ชันแฮมิลโตเนียนมีค่าน้อยที่สุด และทำให้สมการ (ข.7) เป็นจริง

$$-(\partial V_H[x(t),t]/\partial t) = \phi[x(t),u^*(t),t]+[\partial V_H[x(t),t]/\partial x]^T f[x(t),u^*(t),t] \qquad (ข.7)$$

การประยุกต์ใช้งานกับแขนหุ่นยนต์

1. แขนหุ่นยนต์กรณีไม่คิดแรงเสียดทานฝืดคัช

1.1 แบบจำลองทางคณิตศาสตร์ของแขนหุ่นยนต์

$$\begin{bmatrix} \ddot{\tilde{q}} \\ \dot{\tilde{q}} \\ \dot{\tilde{A}} \end{bmatrix} = \begin{bmatrix} -H^{-1}C-K_{pp} & -H^{-1}CK_{pp} & H^{-1}Y \\ I_{n\times n} & 0 & 0 \\ -\Gamma Y^T & -\Gamma Y^T K_{pp} & 0 \end{bmatrix} \begin{bmatrix} \dot{\tilde{q}} \\ \tilde{q} \\ \tilde{A} \end{bmatrix} + \begin{bmatrix} H^{-1} \\ 0 \\ 0 \end{bmatrix} u$$

$$\dot{x} = \begin{bmatrix} -H^{-1}C-K_{pp} & -H^{-1}CK_{pp} & H^{-1}Y \\ I_{n\times n} & 0 & 0 \\ -\Gamma Y^T & -\Gamma Y^T K_{pp} & 0 \end{bmatrix} x + \begin{bmatrix} H^{-1} \\ 0 \\ 0 \end{bmatrix} u \qquad (ข.8)$$

1.2 ดรรชนีสมรรถนะ

$$J \; = \; (1/2)\tilde{A}^T(\infty)\Gamma_1^{-1}\tilde{A}(\infty)+(1/2)\int_0^\infty [x^TQx+u^TRu+2x^TSu]\,dt \qquad (ข.9)$$

เมื่อ $Q = \begin{bmatrix} Q_1 & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}$

$S = \begin{bmatrix} S_1 \\ 0_{n\times n} \end{bmatrix}$

และ Q_1 เป็นเมตริกซ์ที่แสดงค่าถ่วงน้ำหนักของค่าความผิดพลาดตำแหน่ง และความเร็ว ซึ่งมีมิติ $2n\times 2n$

R เป็นเมตริกซ์ที่แสดงค่าถ่วงน้ำหนักของขนาดสัญญาณควบคุม u และเป็นเมตริกซ์ที่มากกว่าศูนย์ ซึ่งมีมิติ $n\times n$

S_1 เป็นเมตริกซ์ที่แสดงค่าถ่วงน้ำหนัก สำหรับพจน์ที่ประกอบด้วยความผิดพลาดของตำแหน่งและความเร็ว กับขนาดของสัญญาณควบคุม u ซึ่งมีมิติ $2n \times n$

Γ_1^{-1} เป็นเมตริกซ์ที่แสดงค่าถ่วงน้ำหนักของค่าความผิดพลาด ที่เกิดจากการประมาณค่าพารามิเตอร์ของระบบ ที่เวลาอนันต์

ค่าที่อยู่ในวงเล็บของพจน์สุดท้ายทางขวามือของสมการ (ข.9) สามารถจัดรูปใหม่ ดังแสดงในสมการ (ข.10)

$$x^T Q x + u^T R u + 2x^T S u = x^T (Q - S^T R^{-1} S) x + (u + R^{-1} S x)^T R (u + R^{-1} S x) \qquad \text{(ข.10)}$$

เนื่องจากค่าในวงเล็บของสมการ (ข.9) ที่แสดงอยู่ในสมการ (ข.10) ต้องเป็นค่าที่มากกว่าศูนย์ ดังนั้นเราจะได้ข้อกำหนดในการเลือกค่าถ่วงน้ำหนักทั้งหมดว่า $(Q - S^T R^{-1} S) > 0$ และเมตริกซ์ R ต้องเป็นเมตริกซ์ที่มากกว่าศูนย์

1.3 <u>ฟังก์ชันหลักของแฮมิลตัน</u>

ถ้า $V_H[x(t),t]$ เป็นสเกลาร์ฟังก์ชันที่สอดคล้องกับสมการแฮมิลตัน-จาโคบิ เราจะได้ว่าค่าที่น้อยที่สุดของดรรชนีสมรรถนะเป็นไปตามสมการ (ข.11)

$$\begin{aligned} V_H[x(0),0] &= \min_{U_t}\{(1/2)\tilde{A}^T(\infty)\Gamma_1^{-1}\tilde{A}(\infty) \\ &\qquad + (1/2)\int_0^\infty [x^T Q x + u^T R u + 2x^T S u]\, dt\} \\ &= \min_{U_t}\{J\} \end{aligned} \qquad \text{(ข.11)}$$

เมื่อ
$$\begin{aligned} V_H[x(\infty),\infty] &= (1/2)\tilde{A}^T(\infty)\Gamma^{-1}\tilde{A}(\infty) \\ U_t &= \{u(s), 0 \le s \le \infty\} \\ x &= x(s) \\ u &= u(s) \end{aligned}$$

1.4 <u>ฟังก์ชันแฮมิลโตเนียน</u>

จากสมการ (ข.11) สมการฟังก์ชันแฮมิลโตเนียนจะเป็นไปตามสมการที่ (ข.12)

$$H[x(t),u(t),\partial V_H/\partial x] = (1/2)x^T Q x + (1/2)u^T R u + x^T S u + (\partial V_H/\partial x)^T \dot{x}$$

(ข.12)

เนื่องจากเราต้องใช้ฟังก์ชันแฮมิลโตเนียน เพื่อคำนวณหาสัญญาณควบคุมที่เหมาะที่สุด ดังนั้นเราต้องหาค่าของ $\dot{x}$ และ $(\partial V_H/\partial x)^T$ เพื่อแทนลงในสมการ (ข.12) ในส่วนของ $\dot{x}$ นั้นเราสามารถหาได้จากสมการ (ข.8) แต่สำหรับส่วน$(\partial V_H/\partial x)^T$ เราต้องคำนวณเพิ่มเติม ดังที่จะแสดงต่อไป

กำหนดให้ฟังก์ชันหลักของแฮมิลตันเป็นดังสมการที่ (ข.13)

$$V_H[x(t),t] = (1/2)x^T \begin{bmatrix} T_{11} & 0 & 0 \\ T_{12} & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} H(q) & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & \Gamma_1^{-1} \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} & O \\ 0 & I & O \\ 0 & 0 & I \end{bmatrix} x$$

(ข.13)

จะได้

$$\begin{aligned} dV_H[x(t),t]/dt & \\ &= (1/2)x^T T_0^T \begin{bmatrix} \dot{H}(q) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_0 x + x^T T_0^T \begin{bmatrix} H(q) & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & \Gamma_1^{-1} \end{bmatrix} T_0 \dot{x} \\ &= \partial V_H[x(t),t]/\partial t + (\partial V_H[x(t),t]/\partial x)^T \dot{x} \\ &= (1/2)x^T T_0^T \begin{bmatrix} \partial H(q)/\partial t & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_0 x \\ &\quad + (1/2)x^T T_0^T \begin{bmatrix} [\partial H(q)/\partial x]^T & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_0 x \dot{x} \\ &\quad + x^T T_0^T \begin{bmatrix} H(q) & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & \Gamma_1^{-1} \end{bmatrix} T_0 \dot{x} \end{aligned}$$

ดังนั้น

$$\begin{aligned} (1/2)x^T T_0^T \begin{bmatrix} \dot{H}(q) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_0 x &= (1/2)x^T T_0^T \begin{bmatrix} \partial H(q)/\partial t & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_0 x \\ &\quad + (1/2)x^T T_0^T \begin{bmatrix} [\partial H(q)/\partial x]^T & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_0 x \dot{x} \end{aligned}$$

(ข.14)

หรือ

$$x^TT_0^T\begin{bmatrix}[\partial H(q)/\partial x]^T & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}T_0x\dot{x} = x^TT_0^T\begin{bmatrix}\dot{H}(q)-\partial H(q)/\partial t & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}T_0x$$

(ข.15)

แทนค่าต่าง ๆ ลงในสมการ (ข.12) จะได้

$$\begin{aligned}
H[x(t),u(t),\partial V_H/\partial x] &= (1/2)x^TQx+(1/2)u^TRu+x^TSu+(\partial V/\partial x)^T\dot{x}\\
&= (1/2)x^TQx+(1/2)u^TRu+x^TSu\\
&\quad + (1/2)x^TT_0^T\begin{bmatrix}[\partial H(q)/\partial x]^T & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}T_0x\dot{x}\\
&\quad + x^TT_0^T\begin{bmatrix}H(q) & 0 & 0\\ 0 & P & 0\\ 0 & 0 & \Gamma_1^{-1}\end{bmatrix}T_0\dot{x}\\
&= (1/2)x^TQx+(1/2)u^TRu+x^TSu\\
&\quad + (1/2)x^TT_0^T\begin{bmatrix}\dot{H}(q)-\partial H(q)/\partial t & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}T_0x\\
&\quad + x^TT_0^T\begin{bmatrix}H(q) & 0 & 0\\ 0 & P & 0\\ 0 & 0 & \Gamma_1^{-1}\end{bmatrix}T_0\dot{x}
\end{aligned}\qquad (ข.16)$$

จากนั้นจึงแทนค่า $\dot{x}$ จากสมการ (ข.8) ลงในสมการที่ (ข.16) แล้วทำการดิฟเฟอเรนชิเอทบางส่วน (partial differentiate) เทียบกับ u และบังคับให้เท่ากับศูนย์จะได้

$$\begin{aligned}
\partial H[x(t),u(t),\partial V_H/\partial x]/\partial u &= 0\\
&= u^TR+x^TS+x^T\begin{bmatrix}T_{11}H(q)T_{11}H(q)^{-1}\\ T_{12}H(q)T_{11}H(q)^{-1}\\ 0\end{bmatrix}
\end{aligned}$$

$\therefore$ จะได้

$$u^T R = -x^T S - x^T \begin{bmatrix} T_{11} H(q) T_{11} H(q)^{-1} \\ T_{12} H(q) T_{11} H(q)^{-1} \\ 0 \end{bmatrix}$$

$$u = u^*$$
$$= -R^{-1} H^{-1}(q) T_{11} H(q) B^T T_0 x - R^{-1} S^T x \quad (ข.17)$$

เมื่อ $B^T = \begin{bmatrix} I & 0 & 0 \end{bmatrix}$

จากสมการ (ข.17) เราจะได้สัญญาณควบคุม u^* ที่ทำให้ดรรชนีสมรรถนะมีค่าน้อยที่สุด ซึ่งเป็นเป้าหมายของการคำนวณ

1.5 <u>การตรวจสอบสมการแฮมิลตัน-จาโคบิ</u>

เมื่อแทนค่าสัญญาณควบคุม u^* ที่คำนวณได้ลงในสมการ (ข.7) จะได้

$$H[x(t), u^*(t), \partial V_H / \partial x] + (\partial V_H [x(t), t] / \partial t) = 0$$
$$= (1/2) x^T Q x$$
$$+ (1/2) x^T (S + T_0{}^T B H T_{11} H^{-1}) R^{-1} (S^T + H^{-1} T_{11} H B^T T_0) x$$
$$+ x^T S R^{-1} (-H^{-1} T_{11} H B^T T_0 x - S^T x)$$
$$+ (1/2) x^T T_0^T \begin{bmatrix} \partial H(q)/\partial t & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_0 x$$
$$+ (1/2) x^T T_0^T \begin{bmatrix} \dot{H}(q) - \partial H(q)/\partial t & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_0 x$$
$$+ x^T T_0^T \begin{bmatrix} H(q) & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & \Gamma_1^{-1} \end{bmatrix} T_0 \dot{x}$$
$$= (1/2) x^T Q x$$
$$+ (1/2) x^T (S + T_0{}^T B H T_{11} H^{-1}) R^{-1} (S^T + H^{-1} T_{11} H B^T T_0) x$$
$$+ x^T S R^{-1} (-H^{-1} T_{11} H B^T T_0 x - S^T x)$$
$$+ (1/2) x^T T_0^T \begin{bmatrix} \dot{H} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_0 x$$

$$+x^T\begin{bmatrix} -T_{11}HT_{11}H^{-1}C & -T_{11}HT_{11}H^{-1}CT_{11}^{-1}T_{12} & T_{11}HT_{11}H^{-1}Y \\ -T_{12}HT_{11}H^{-1}C+P & -T_{12}HT_{11}H^{-1}CT_{11}^{-1}T_{12} & T_{12}HT_{11}H^{-1}Y \\ -\Gamma_1^{-1}\Gamma Y^T & -\Gamma_1^{-1}\Gamma Y^T K_{pp} & 0 \end{bmatrix}x$$

$$+x^T\begin{bmatrix} -T_{11}HT_{11}K_{pp}+T_{11}HT_{12} & 0 & 0 \\ -T_{12}HT_{11}K_{pp}+T_{12}HT_{12} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}x$$

$$+x^T\begin{bmatrix} 0 & -T_{11}HT_{11}H^{-1}CK_{pp}+T_{11}HT_{11}H^{-1}CT_{11}^{-1}T_{12} & 0 \\ 0 & -T_{12}HT_{11}H^{-1}CK_{pp}+T_{12}HT_{11}H^{-1}CT_{11}^{-1}T_{12} & 0 \\ 0 & 0 & 0 \end{bmatrix}x$$

$$+x^T\begin{bmatrix} T_{11}HT_{11}H^{-1} \\ T_{12}HT_{11}H^{-1} \\ 0 \end{bmatrix}R^{-1}(S^T+H^{-1}T_{11}HB^TT_0)x$$

กำหนดให้ $K_{pp} = T_{11}^{-1}T_{12}$

$\Gamma = \Gamma_1 T_{11} T_{11}$

$T_{11} = t_{11}I_{n\times n}$

$\therefore$ จะได้

$$H[x(t),u^*(t),\partial V_H/\partial x]+(\partial V_H[x(t),t]/\partial t) = 0$$

$$= (1/2)x^TQx+(1/2)x^T(S+T_0{}^TBt_{11})R^{-1}(S^T+t_{11}B^TT_0)x$$

$$+x^TSR^{-1}(-t_{11}B^TT_0x-S^Tx)$$

$$+(1/2)x^TT_0^T\begin{bmatrix} \dot{H} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}T_0x$$

$$+x^T\begin{bmatrix} -t_{11}t_{11}C & -t_{11}CT_{12} & t_{11}t_{11}Y \\ -T_{12}t_{11}C+P & -T_{12}CT_{12} & T_{12}t_{11}Y \\ -t_{11}t_{11}Y^T & -t_{11}Y^TT_{12} & 0 \end{bmatrix}x$$

$$-x^T\begin{bmatrix} t_{11}t_{11} \\ T_{12}t_{11} \\ 0 \end{bmatrix}R^{-1}(S^T+t_{11}B^TT_0)x$$

$$= (1/2)x^TQx-(1/2)x^T(S+T_0{}^TBt_{11})R^{-1}(S^T+t_{11}B^TT_0)x$$

$$+(1/2)x^TT_0^T\begin{bmatrix} \dot{H} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}T_0x$$

$$+x^T T_0^T \begin{bmatrix} -C & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_0 x$$

$$+ x^T \begin{bmatrix} 0 & 0 & 0 \\ P & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x$$

$$= (1/2)x^T Qx - (1/2)x^T (S + T_0{}^T Bt_{11}) R^{-1} (S^T + t_{11} B^T T_0) x$$

$$+(1/2)\ x^T \begin{bmatrix} 0 & P & 0 \\ P & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x$$

$$= 0 \qquad (ข.18)$$

จากสมการ (ข.18) ได้แสดงให้เห็นว่า สัญญาณควบคุมที่คำนวณได้ จะเป็นค่าที่เหมาะที่สุด ก็ต่อเมื่อสมการ (ข.19) เป็นจริง

$$\begin{bmatrix} 0 & P & 0 \\ P & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + Q - (S^T + t_{11} B^T T_0)^T R^{-1} (S^T + t_{11} B^T T_0) = 0 \qquad (ข.19)$$

2. <u>แขนหุ่นยนต์กรณีคิดรวมแรงเสียดทานฝืชคัช</u>

2.1 <u>แบบจำลองทางคณิตศาสตร์ของแขนหุ่นยนต์</u>

$$\begin{bmatrix} \ddot{\tilde{q}} \\ \dot{\tilde{q}} \\ \dot{\tilde{A}} \\ \dot{\tilde{K}}_f \end{bmatrix} = \begin{bmatrix} -H^{-1}C - K_{pp} & -H^{-1}CK_{pp} & H^{-1}Y & H^{-1}Y_f \\ I_{n\times n} & 0 & 0 & 0 \\ -\Gamma Y^T & -\Gamma Y^T K_{pp} & 0 & 0 \\ -\Gamma_f Y_f{}^T & -\Gamma_f Y_f{}^T K_{pp} & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\tilde{q}} \\ \tilde{q} \\ \tilde{A} \\ \tilde{K}_f \end{bmatrix} + \begin{bmatrix} H^{-1} \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

$$\dot{x} = \begin{bmatrix} -H^{-1}C - K_{pp} & -H^{-1}CK_{pp} & H^{-1}Y & H^{-1}Y_f \\ I_{n\times n} & 0 & 0 & 0 \\ -\Gamma Y^T & -\Gamma Y^T K_{pp} & 0 & 0 \\ -\Gamma_f Y_f{}^T & -\Gamma_f Y_f{}^T K_{pp} & 0 & 0 \end{bmatrix} x + \begin{bmatrix} H^{-1} \\ 0 \\ 0 \\ 0 \end{bmatrix} u \qquad (ข.20)$$

2.2 <u>ดรรชนีสมรรถนะ</u>

$$J = (1/2)\tilde{A}^T(\infty)\Gamma_1{}^{-1}\tilde{A}(\infty) + (1/2)\tilde{K}_f{}^T(\infty)\Gamma_2{}^{-1}\tilde{K}_f(\infty)$$

$$+ (1/2)\int_0^\infty [x^T Qx + u^T Ru + 2x^T Su]\, dt \qquad (ข.21)$$

เมื่อ $Q = \begin{bmatrix} Q_1 & 0_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} \end{bmatrix}$

$S = \begin{bmatrix} S_1 \\ 0_{n\times n} \\ 0_{n\times n} \end{bmatrix}$

และ Γ_1^{-1} เป็นเมตริกซ์ที่แสดงค่าถ่วงน้ำหนักของค่าความผิดพลาด ที่เกิดจากการประมาณค่าพารามิเตอร์ A ที่เวลาอนันต์

Γ_2^{-1} เป็นเมตริกซ์ที่แสดงค่าถ่วงน้ำหนักของค่าความผิดพลาด ที่เกิดจากการประมาณค่าพารามิเตอร์ K_f ที่เวลาอนันต์

2.3 ฟังก์ชันหลักของแฮมิลตัน

$$\begin{aligned} V_H[x(0),0] &= \min_{U_t}\{(1/2)\tilde{A}^T(\infty)\Gamma_1^{-1}\tilde{A}(\infty)+(1/2)\tilde{K}_f^{\ T}(\infty)\Gamma_2^{-1}\tilde{K}_f(\infty) \\ &\quad +(1/2)\int_0^\infty[x^TQx+u^TRu+2x^TSu]\,dt\} \\ &= \min_{U_t}\{J\} \end{aligned} \qquad (ข.22)$$

2.4 ฟังก์ชันแฮมิลโตเนียน

$$H[x(t),u(t),\partial V_H/\partial x]=(1/2)x^TQx+(1/2)u^TRu+x^TSu+(\partial V_H/\partial x)^T\dot{x} \qquad (ข.23)$$

จากฟังก์ชันหลักของแฮมิลตันที่แสดงในสมการ (ข.24) เราจะแทนค่า $(\partial V_H/\partial x)^T$ และ $\dot{x}$ ลงในสมการ (ข.23)

$$V_H[x(t),t]=(1/2)x^TT_0^{\ T}\begin{bmatrix} H(q) & 0 & 0 & 0 \\ 0 & P & 0 & 0 \\ 0 & 0 & \Gamma_1^{-1} & 0 \\ 0 & 0 & 0 & \Gamma_2^{-1} \end{bmatrix}T_0x \qquad (ข.24)$$

เมื่อ $T_0 = \begin{bmatrix} T_{11} & T_{12} & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$

ดังนั้นจะได้

$$
\begin{aligned}
H[x(t),u(t),\partial V_H/\partial x] &= (1/2)x^TQx + (1/2)u^TRu + x^TSu + (\partial V/\partial x)^T\dot{x} \\
&= (1/2)x^TQx + (1/2)u^TRu + x^TSu \\
&\quad + (1/2)x^TT_0^T\begin{bmatrix} [\partial H(q)/\partial x]^T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}T_0x\dot{x} \\
&\quad + x^TT_0^T\begin{bmatrix} H(q) & 0 & 0 & 0 \\ 0 & P & 0 & 0 \\ 0 & 0 & \Gamma_1^{-1} & 0 \\ 0 & 0 & 0 & \Gamma_2^{-1} \end{bmatrix}T_0\dot{x} \\
&= (1/2)x^TQx + (1/2)u^TRu + x^TSu \\
&\quad + (1/2)x^TT_0^T\begin{bmatrix} \dot{H}(q) - \partial H(q)/\partial t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}T_0x \\
&\quad + x^TT_0^T\begin{bmatrix} H(q) & 0 & 0 & 0 \\ 0 & P & 0 & 0 \\ 0 & 0 & \Gamma_1^{-1} & 0 \\ 0 & 0 & 0 & \Gamma_2^{-1} \end{bmatrix}T_0\dot{x}
\end{aligned}
\tag{ข.25}
$$

จากนั้นจึงแทนค่า $\dot{x}$ จากสมการ (ข.20) ลงในสมการที่ (ข.25) แล้วทำการดิฟเฟอเรนชิเอทบางส่วน (partial differentiate) เทียบกับ u และบังคับให้เท่ากับศูนย์จะได้

$$
\begin{aligned}
\partial H[x(t),u(t),\partial V_H/\partial x]/\partial u &= 0 \\
&= u^TR + x^TS + x^T\begin{bmatrix} T_{11}H(q)T_{11}H(q)^{-1} \\ T_{12}H(q)T_{11}H(q)^{-1} \\ 0 \\ 0 \end{bmatrix}
\end{aligned}
$$

$\therefore$ จะได้

$$
\begin{aligned}
u^TR &= -x^TS - x^T\begin{bmatrix} T_{11}H(q)T_{11}H(q)^{-1} \\ T_{12}H(q)T_{11}H(q)^{-1} \\ 0 \\ 0 \end{bmatrix} \\
u &= u^* \\
&= -R^{-1}H^{-1}(q)T_{11}H(q)B^TT_0x - R^{-1}S^Tx
\end{aligned}
\tag{ข.26}
$$

เมื่อ $B^T = \begin{bmatrix} I & 0 & 0 & 0 \end{bmatrix}$

2.5 <u>การตรวจสอบสมการแฮมิลตัน-จาโคบิ</u>

เมื่อแทนค่าสัญญาณควบคุม u^* ที่คำนวณได้ลงในสมการ (ข.7) จะได้

$$
\begin{aligned}
H[x(t),u^*(t),\partial V_H/\partial x] + (\partial V_H[x(t),t]/\partial t) = 0 \\
&= (1/2)x^T Q x \\
&\quad + (1/2)x^T (S + T_0{}^T BHT_{11}H^{-1})R^{-1}(S^T + H^{-1}T_{11}HB^T T_0)x \\
&\quad + x^T SR^{-1}(-H^{-1}T_{11}HB^T T_0 x - S^T x) \\
&\quad + (1/2)x^T T_0^T \begin{bmatrix} \partial H(q)/\partial t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} T_0 x \\
&\quad + (1/2)x^T T_0^T \begin{bmatrix} \dot{H}(q) - \partial H(q)/\partial t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} T_0 x \\
&\quad + x^T T_0^T \begin{bmatrix} H(q) & 0 & 0 & 0 \\ 0 & P & 0 & 0 \\ 0 & 0 & \Gamma_1^{-1} & 0 \\ 0 & 0 & 0 & \Gamma_2^{-1} \end{bmatrix} T_0 \dot{x}
\end{aligned}
$$

$$
\begin{aligned}
= \; & (1/2)x^T Q x \\
& + (1/2)x^T (S + T_0{}^T BHT_{11}H^{-1})R^{-1}(S^T + H^{-1}T_{11}HB^T T_0)x \\
& + x^T SR^{-1}(-H^{-1}T_{11}HB^T T_0 x - S^T x) \\
& + (1/2)x^T T_0^T \begin{bmatrix} \dot{H}(q) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} T_0 x
\end{aligned}
$$

$$+x^T\begin{bmatrix} -T_{11}HT_{11}H^{-1}C & -T_{11}HT_{11}H^{-1}CT_{11}^{-1}T_{12} & T_{11}HT_{11}H^{-1}Y & T_{11}HT_{11}H^{-1}Y_f \\ -T_{12}HT_{11}H^{-1}C+P & -T_{12}HT_{11}H^{-1}CT_{11}^{-1}T_{12} & T_{12}HT_{11}H^{-1}Y & T_{12}HT_{11}H^{-1}Y_f \\ -\Gamma_1^{-1}\Gamma Y^T & -\Gamma_1^{-1}\Gamma Y^T K_{pp} & 0 & 0 \\ -\Gamma_2^{-1}\Gamma_f {Y_f}^T & -\Gamma_2^{-1}\Gamma_f {Y_f}^T K_{pp} & 0 & 0 \end{bmatrix}x$$

$$+x^T\begin{bmatrix} -T_{11}HT_{11}K_{pp}+T_{11}HT_{12} & 0 & 0 & 0 \\ -T_{12}HT_{11}K_{pp}+T_{12}HT_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}x$$

$$+x^T\begin{bmatrix} 0 & -T_{11}HT_{11}H^{-1}CK_{pp}+T_{11}HT_{11}H^{-1}CT_{11}^{-1}T_{12} & 0 & 0 \\ 0 & -T_{12}HT_{11}H^{-1}CK_{pp}+T_{12}HT_{11}H^{-1}CT_{11}^{-1}T_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}x$$

$$+x^T\begin{bmatrix} T_{11}HT_{11}H^{-1} \\ T_{12}HT_{11}H^{-1} \\ 0 \\ 0 \end{bmatrix}R^{-1}(S^T+H^{-1}T_{11}HB^TT_0)x$$

กำหนดให้ $K_{pp} = T_{11}^{-1}T_{12}$

$\Gamma = \Gamma_1 T_{11} T_{11}$

$\Gamma_f = \Gamma_2 T_{11} T_{11}$

$T_{11} = t_{11} I_{n\times n}$

$\therefore$ จะได้

$$H[x(t),u^*(t),\partial V_H/\partial x]+(\partial V_H[x(t),t]/\partial t) = 0$$

$$= (1/2)x^TQx+(1/2)x^T(S+T_0{}^TBt_{11})R^{-1}(S^T+t_{11}B^TT_0)x$$

$$+x^TSR^{-1}(-t_{11}B^TT_0x-S^Tx)$$

$$+ (1/2)x^TT_0^T\begin{bmatrix} \dot{H} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}T_0x$$

$$+x^T\begin{bmatrix} -t_{11}t_{11}C & -t_{11}CT_{12} & t_{11}t_{11}Y & t_{11}t_{11}Y_f \\ -T_{12}t_{11}C+P & -T_{12}CT_{12} & T_{12}t_{11}Y & T_{12}t_{11}Y_f \\ -t_{11}t_{11}Y^T & -t_{11}Y^TT_{12} & 0 & 0 \\ -t_{11}t_{11}{Y_f}^T & -t_{11}{Y_f}^TT_{12} & 0 & 0 \end{bmatrix}x$$

$$- x^T \begin{bmatrix} t_{11}t_{11} \\ T_{12}t_{11} \\ 0 \\ 0 \end{bmatrix} R^{-1}(S^T + t_{11}B^T T_0)x$$

$$= (1/2)x^T Qx - (1/2)x^T(S + T_0{}^T Bt_{11})R^{-1}(S^T + t_{11}B^T T_0)x$$

$$+ (1/2)x^T T_0^T \begin{bmatrix} \dot{H} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} T_0 x$$

$$+ x^T T_0^T \begin{bmatrix} -C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} T_0 x$$

$$+ x^T \begin{bmatrix} 0 & 0 & 0 & 0 \\ P & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x$$

$$= (1/2)x^T Qx - (1/2)x^T(S + T_0{}^T Bt_{11})R^{-1}(S^T + t_{11}B^T T_0)x$$

$$+ (1/2)x^T \begin{bmatrix} 0 & 0 & 0 & 0 \\ P & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x$$

$$= 0 \qquad \text{(ข.27)}$$

จากสมการ (ข.27) ได้แสดงให้เห็นว่า สัญญาณควบคุมที่คำนวณได้ จะเป็นค่าที่เหมาะที่สุด ก็ต่อเมื่อสมการ (ข.28) เป็นจริง

$$\begin{bmatrix} 0 & P & 0 & 0 \\ P & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + Q - (S^T + t_{11}B^T T_0)^T R^{-1}(S^T + t_{11}B^T T_0) = 0 \qquad \text{(ข.28)}$$

ภาคผนวก ค

แฟ้มคำสั่งที่ใช้ในการจำลองการทำงาน

ระบบเป็นแขนหุ่นยนต์แบบข้อปล้อง

```
CONTINUOUS SYSTEM robot
"simple Articulated robot
STATE x1 x2 x3 x4 x5 x6
DER dx1 dx2 dx3 dx4 dx5 dx6
INPUT t1 t2 t3
OUTPUT p1 p2 p3 v1 v2 v3 a1 a2 a3 a4 a5 a6 a7 a8 h1 h2 h3 h4
OUTPUT kf1 kf2 kf3
time t
"abbreviate notation
s1=sin(x1)
s2=sin(x2)
c1=cos(x1)
c2=cos(x2)
s12=sin(x1+x2)
c12=cos(x1+x2)
"constant parameters
a1=m2*l1*lc2
a2=(j1+(m1*lc1*lc1)+(m2*l1*l1)-j1p)
a3=(j2+(m2*lc2*lc2)-j2p)
a4=(j1+(m1*lc1*lc1)+(m2*l1*l1)+j2+(m2*lc2*lc2))
a5=(j2+(m2*lc2*lc2))
a6=j1p+j2p+j3
a7=(m1*lc1*g)+(m2*l1*g)
```

```
a8=m2*lc2*g
"inertia matrix elements
h11=a4+2*a1*c2
h12=a5+a1*c2
h21=h12
h22=a5
h33=a2*s1*s1+a3*s12*s12+a6+2*a1*s1*s12
delta=(h11*h22)-(h12*h21)
"viscous-friction force
kf1=75
kf2=10
kf3=1
fv1=kf1*x4
fv2=kf2*x5
fv3=kf3*x6
"temporary variable
temp11=(k1*t1)-fv1+(2*a1*s2*x5*x4)+(a1*s2*x5*x5)
temp12=((0.5*a2*sin(2*x1))+(0.5*a3*sin(2*(x1+x2))))*x6*x6
temp13=(a1*sin((2*x1)+x2))*x6*x6
temp14=(a7*s1)+(a8*s12)
temp21=(k2*t2)-fv2-(a1*s2*x4*x4)
temp22=0.5*a3*sin(2*(x1+x2))*x6*x6
temp23=(a1*s1*c12*x6*x6)+(a8*s12)
temp31=(k3*t3)-fv3-(a3*sin(2*(x1+x2))*x6*x5)
temp32=-(2*a1*s1*c12*x6*x5)
temp33=-(((a2*sin(2*x1))+(a3*sin(2*(x1+x2))))*x6*x4)
temp34=-(2*a1*sin((2*x1)+x2)*x6*x4)
```

```
temp1=temp11+temp12+temp13+temp14
temp2=temp21+temp22+temp23
temp3=temp31+temp32+temp33+temp34
"state equaions
dx1=x4
dx2=x5
dx3=x6
dx4=(1/delta)*((h22*temp1)-(h12*temp2))
dx5=(1/delta)*((h11*temp2)-(h21*temp1))
dx6=temp3/h33
"convert radians to degrees
pp1=((180*7)/22)*x1
pp2=((180*7)/22)*x2
pp3=((180*7)/22)*x3
vv1=((180*7)/22)*x4
vv2=((180*7)/22)*x5
vv3=((180*7)/22)*x6
aa1=((180*7)/22)*dx4
aa2=((180*7)/22)*dx5
aa3=((180*7)/22)*dx6
"output
p1=x1
p2=x2
p3=x3
v1=x4
v2=x5
v3=x6
```

```
h1=h11
h2=h12
h3=h22
h4=h33
"nominal values
l1:1
l2:1
c1:0.5
c2:0.5
1:10
2:10
1:0.5
1p:0.01
j2:0.5
j2p:0.01
j3:1
k1:40
k2:20
k3:40
g:9.81
x1:0
x2:0
x3:0
x4:0
x5:0
x6:0
END
```

```
CONTINUOUS SYSTEM contr
"adaptive controller for articulated robot
INPUT p1 p2 p3 v1 v2 v3 pd1 pd2 pd3 vd1 vd2 vd3 ad1 ad2 ad3
OUTPUT t1 t2 t3 perr1 perr2 perr3 verr1 verr2 verr3
OUTPUT u1 u2 u3 aa1 aa2 aa3 aa4 aa5 aa6 aa7 aa8 kff1 kff2 kff3
STATE a1 a2 a3 a4 a5 a6 a7 a8 kf1 kf2 kf3
DER da1 da2 da3 da4 da5 da6 da7 da8 dkf1 dkf2 dkf3
"abbreviate notation
c1=cos(p1)
c2=cos(p2)
s1=sin(p1)
s2=sin(p2)
c12=cos(p1+p2)
s12=sin(p1+p2)
s212=sin((2*p1)+p2)
"errors
perr1=p1-((22*pd1)/(7*180))
perr2=p2-((22*pd2)/(7*180))
perr3=p3-((22*pd3)/(7*180))
verr1=v1-((22*vd1)/(7*180))
verr2=v2-((22*vd2)/(7*180))
verr3=v3-((22*vd3)/(7*180))
perr1d=((180*7)/22)*perr1
perr2d=((180*7)/22)*perr2
perr3d=((180*7)/22)*perr3
verr1d=((180*7)/22)*verr1
```

```
verr2d=((180*7)/22)*verr2
verr3d=((180*7)/22)*verr3
err1=(verr1)+(t12*perr1/t11)
err2=(verr2)+(t12*perr2/t11)
err3=(verr3)+(t12*perr3/t11)
"controller variables
dqr1=((22*vd1)/(7*180))-((t12*perr1)/t11)
dqr2=((22*vd2)/(7*180))-((t12*perr2)/t11)
dqr3=((22*vd3)/(7*180))-((t12*perr3)/t11)
ddqr1=((22*ad1)/(7*180))-((t12*verr1)/t11)
ddqr2=((22*ad2)/(7*180))-((t12*verr2)/t11)
ddqr3=((22*ad3)/(7*180))-((t12*verr3)/t11)
y11=2*c2*ddqr1+c2*ddqr2-s2*v2*(dqr1+dqr2)-s2*v1*dqr2-s212*v3*dqr3
y12=-(s1*c1*v3*dqr3)
y13=-(s12*c12*v3*dqr3)
y14=ddqr1
y15=ddqr2
y17=-s1
y18=-s12
y21=c2*ddqr1+s2*v1*dqr1-s1*c12*v3*dqr3
y23=-(s12*c12*v3*dqr3)
y25=ddqr1+ddqr2
y28=-s12
temp1=s212*v1*dqr3+s1*c12*v2*dqr3
y31=2*s1*s12*ddqr3+s212*v3*dqr1+s1*c12*v3*dqr2+temp1
y32=s1*s1*ddqr3+s1*c1*v3*dqr1+s1*c1*v1*dqr3
temp2=s12*c12*v1*dqr3+s12*c12*v2*dqr3
```

```
y33=s12*s12*ddqr3+s12*c12*v3*dqr1+s12*c12*v3*dqr2+temp2

y36=ddqr3

"adaptive equations

da1=-(r1*t11*t11*(y11*err1+y21*err2+y31*err3))

da2=-(r2*t11*t11*(y12*err1+y32*err3))

da3=-(r3*t11*t11*(y13*err1+y23*err2+y33*err3))

da4=-(r4*t11*t11*(y14*err1))

da5=-(r5*t11*t11*(y15*err1+y25*err2))

da6=-(r6*t11*t11*(y36*err3))

da7=-(r7*t11*t11*(y17*err1))

da8=-(r8*t11*t11*(y18*err1+y28*err2))

dkf1=-(r9*t11*t11*v1*err1)

dkf2=-(r10*t11*t11*v2*err2)

dkf3=-(r11*t11*t11*v3*err3)

"output

u1=(kv1*verr1)+(kp1*perr1)

u2=(kv2*verr2)+(kp2*perr2)

u3=(kv3*verr3)+(kp3*perr3)

t1=(y11*a1+y12*a2+y13*a3+y14*a4+y15*a5+y17*a7+y18*a8+kf1*v1-u1)/k1

t2=(y21*a1+y23*a3+y25*a5+y28*a8+kf2*v2-u2)/k2

t3=(y31*a1+y32*a2+y33*a3+y36*a6+kf3*v3-u3)/k3

aa1=a1

aa2=a2

aa3=a3

aa4=a4

aa5=a5

aa6=a6
```

```
aa7=a7
aa8=a8
kff1=kf1
kff2=kf2
kff3=kf3
"nominal values
r1:0.2
r2:0.5
r3:0.2
r4:0.5
r5:0.2
r6:0.2
r7:10
r8:5
r9:5
r10:1
r11:0.2
t11:4
t12:5
kv1:1600
kv2:1600
kv3:1600
kp1:2000
kp2:2000
kp3:2000
k1:40
k2:20
```

```
k3:40
END

CONTINUOUS SYSTEM path
"Path generation for articulated robot
OUTPUT pd1 pd2 pd3 vd1 vd2 vd3 ad1 ad2 ad3
TIME t
pd1=30*(1-COS(2*22*t/7))
pd2=30*(1-COS(2*22*t/7))
pd3=45*(1-COS(2*22*t/7))
vd1=(30*2*22/7)*SIN(2*22*t/7)
vd2=(30*2*22/7)*SIN(2*22*t/7)
vd3=(45*2*22/7)*SIN(2*22*t/7)
ad1=(30*(2*22/7)*(2*22/7))*COS(2*22*t/7)
ad2=(30*(2*22/7)*(2*22/7))*COS(2*22*t/7)
ad3=(45*(2*22/7)*(2*22/7))*COS(2*22*t/7)
END

CONTINUOUS SYSTEM cost
"performance index for articulated robot
INPUT perr1 perr2 perr3 verr1 verr2 verr3 u1 u2 u3 h1 h2 h3 h4
INPUT a1 a2 a3 a4 a5 a6 a7 a8 aa1 aa2 aa3 aa4 aa5 aa6 aa7 aa8
INPUT kf1 kf2 kf3 kff1 kff2 kff3
STATE x
DER dx
TIME t
"temporary variable
```

```
tem1=q11*SQR(verr1)
tem2=q22*SQR(verr2)
tem3=q33*SQR(verr3)
tem4=q44*SQR(perr1)
tem5=q55*SQR(perr2)
tem6=q66*SQR(perr3)
tem7=r11*SQR(u1)
tem8=r22*SQR(u2)
tem9=r33*SQR(u3)
error1=(t11*verr1)+(t12*perr1)
error2=(t11*verr2)+(t12*perr2)
error3=(t11*verr3)+(t12*perr3)
dif1=rr1*SQR(aa1-a1)
dif2=rr2*SQR(aa2-a2)
dif3=rr3*SQR(aa3-a3)
dif4=rr4*SQR(aa4-a4)
dif5=rr5*SQR(aa5-a5)
dif6=rr6*SQR(aa6-a6)
dif7=rr7*SQR(aa7-a7)
dif8=rr8*SQR(aa8-a8)
dif9=rr9*SQR(kff1-kf1)
dif10=rr10*SQR(kff2-kf2)
dif11=rr11*SQR(kff3-kf3)
dif12=h1*SQR(error1)
dif13=h3*SQR(error2)
dif14=h4*SQR(error3)
dif15=2*h2*error1*error2
```

```
dif16=p11*SQR(perr1)

dif17=p22*SQR(perr2)

dif18=p33*SQR(perr3)

temp1=tem1+tem2+tem3+tem4+tem5+tem6+tem7+tem8+tem9

temp2=(dif1+dif2+dif3+dif4+dif5+dif6+dif7+dif8+dif9+dif10+dif11)

temp3=(dif12+dif13+dif14+dif15+dif16+dif17+dif18)

"sum and integrate of cost functions

dx=temp1

sum=temp2+temp3

"nominal values

q11:25600

q22:25600

q33:25600

q44:40000

q55:40000

q66:40000

r11:0.01

r22:0.01

r33:0.01

rr1:5

rr2:2

rr3:5

rr4:2

rr5:5

rr6:5

rr7:0.1

rr8:0.2
```

```
rr9:0.2

rr10:1

rr11:5

p11:32000

p22:32000

p33:32000

t11:4

t12:5

END


CONNECTING SYSTEM conn

"connecting system for articulated robot and adaptive contr

t1[robot]=t1[contr]

t2[robot]=t2[contr]

t3[robot]=t3[contr]

u1[cost]=u1[contr]

u2[cost]=u2[contr]

u3[cost]=u3[contr]

p1[contr]=p1[robot]

p2[contr]=p2[robot]

p3[contr]=p3[robot]

v1[contr]=v1[robot]

v2[contr]=v2[robot]

v3[contr]=v3[robot]

pd1[contr]=pd1[path]

pd2[contr]=pd2[path]

pd3[contr]=pd3[path]
```

```
vd1[contr]=vd1[path]

vd2[contr]=vd2[path]

vd3[contr]=vd3[path]

ad1[contr]=ad1[path]

ad2[contr]=ad2[path]

ad3[contr]=ad3[path]

perr1[cost]=perr1[contr]

perr2[cost]=perr2[contr]

perr3[cost]=perr3[contr]

verr1[cost]=verr1[contr]

verr2[cost]=verr2[contr]

verr3[cost]=verr3[contr]

a1[cost]=a1[robot]

a2[cost]=a2[robot]

a3[cost]=a3[robot]

a4[cost]=a4[robot]

a5[cost]=a5[robot]

a6[cost]=a6[robot]

a7[cost]=a7[robot]

a8[cost]=a8[robot]

kf1[cost]=kf1[robot]

kf2[cost]=kf2[robot]

kf3[cost]=kf3[robot]

aa1[cost]=aa1[contr]

aa2[cost]=aa2[contr]

aa3[cost]=aa3[contr]

aa4[cost]=aa4[contr]
```

```
aa5[cost]=aa5[contr]
aa6[cost]=aa6[contr]
aa7[cost]=aa7[contr]
aa8[cost]=aa8[contr]
kff1[cost]=kff1[contr]
kff2[cost]=kff2[contr]
kff3[cost]=kff3[contr]
h1[cost]=h1[robot]
h2[cost]=h2[robot]
h3[cost]=h3[robot]
h4[cost]=h4[robot]
END
```

ระบบเป็นแขนหุ่นยนต์แบบสการา

```
CONTINUOUS SYSTEM robot
"simple SCARA robot
STATE x1 x2 x3 x4 x5 x6
DER dx1 dx2 dx3 dx4 dx5 dx6
INPUT t1 t2 t3
OUTPUT p1 p2 p3 v1 v2 v3 a1 a2 a3 a4 h1 h2 h3 h4
OUTPUT kf1 kf2 kf3
TIME t
"abbreviate notation
s1=sin(x1)
s2=sin(x2)
c1=cos(x1)
c2=cos(x2)
```

```
"constant parameters
a1=((m1/3)+m2+m3)*l1*l1
a2=((2*m3)+m2)*l1*l2
a3=((m2/3)+m3)*l2*l2
a4=m3*g
"inertia matrix elements
h11=a1+a2*c2+a3
h12=-(a2*c2/2)-a3
h21=h12
h22=a3
h33=a4/g
delta=(h11*h22)-(h12*h21)
"viscous-friction force
kf1=75
kf2=10
kf3=1
fv1=kf1*x4
fv2=kf2*x5
fv3=kf3*x6
"temporary variable
temp1=(k1*t1)-fv1+(a2*s2*x5*x4)-(0.5*a2*s2*x5*x5)
temp2=(k2*t2)-fv2-(0.5*a2*s2*x4*x4)
temp3=(k3*t3)-fv3+a4
"state equaions
dx1=x4
dx2=x5
dx3=x6
```

```
dx4=(1/delta)*((h22*temp1)-(h12*temp2))

dx5=(1/delta)*((h11*temp2)-(h21*temp1))

dx6=temp3/h33

"convert radians to degrees

pp1=((180*7)/22)*x1

pp2=((180*7)/22)*x2

pp3=x3

vv1=((180*7)/22)*x4

vv2=((180*7)/22)*x5

vv3=x6

aa1=((180*7)/22)*dx4

aa2=((180*7)/22)*dx5

aa3=dx6

"output

p1=x1

p2=x2

p3=x3

v1=x4

v2=x5

v3=x6

h1=h11

h2=h12

h3=h22

h4=h33

"nominal values

l1:1

l2:1
```

```
m1:10
m2:10
m3:5
k1:40
k2:20
k3:40
g:9.81
x1:0
x2:0
x3:0
x4:0
x5:0
x6:0
END

CONTINUOUS SYSTEM contr
"adaptive controller for SCARA robot
INPUT p1 p2 p3 v1 v2 v3 pd1 pd2 pd3 vd1 vd2 vd3 ad1 ad2 ad3
OUTPUT t1 t2 t3 perr1 perr2 perr3 verr1 verr2 verr3
OUTPUT u1 u2 u3 aa1 aa2 aa3 aa4 kff1 kff2 kff3
STATE a1 a2 a3 a4 kf1 kf2 kf3
DER da1 da2 da3 da4 dkf1 dkf2 dkf3
"abbreviate notation
c1=cos(p1)
c2=cos(p2)
s1=sin(p1)
s2=sin(p2)
```

```
"errors
perr1=p1-((22*pd1)/(7*180))
perr2=p2-((22*pd2)/(7*180))
perr3=p3-pd3
verr1=v1-((22*vd1)/(7*180))
verr2=v2-((22*vd2)/(7*180))
verr3=v3-vd3
perr1d=((180*7)/22)*perr1
perr2d=((180*7)/22)*perr2
perr3m=perr3
verr1d=((180*7)/22)*verr1
verr2d=((180*7)/22)*verr2
verr3m=verr3
err1=(verr1)+(t12*perr1/t11)
err2=(verr2)+(t12*perr2/t11)
err3=(verr3)+(t12*perr3/t11)
"controller variables
dqr1=((22*vd1)/(7*180))-((t12*perr1)/t11)
dqr2=((22*vd2)/(7*180))-((t12*perr2)/t11)
dqr3=vd3-((t12*perr3)/t11)
ddqr1=((22*ad1)/(7*180))-((t12*verr1)/t11)
ddqr2=((22*ad2)/(7*180))-((t12*verr2)/t11)
ddqr3=ad3-((t12*verr3)/t11)
Y11=ddqr1
y12=c2*ddqr1-0.5*(c2*ddqr2+s2*v2*dqr1+s2*v1*dqr2-s2*v2*dqr2)
y13=ddqr1-ddqr2
y22=0.5*(s2*v1*dqr1-c2*ddqr1)
```

```
y23=ddqr2-ddqr1

y34=(ddqr3/g)-1

"adaptive equations

da1=-(r1*t11*t11*(y11*err1))

da2=-(r2*t11*t11*(y12*err1+y22*err2))

da3=-(r3*t11*t11*(y13*err1+y23*err2))

da4=-(r4*t11*t11*(y34*err3))

dkf1=-(r5*t11*t11*v1*err1)

dkf2=-(r6*t11*t11*v2*err2)

dkf3=-(r7*t11*t11*v3*err3)

"output

u1=(kv1*verr1)+(kp1*perr1)

u2=(kv2*verr2)+(kp2*perr2)

u3=(kv3*verr3)+(kp3*perr3)

t1=(y11*a1+y12*a2+y13*a3+kf1*v1-u1)/k1

t2=(y22*a2+y23*a3+kf2*v2-u2)/k2

t3=(y34*a4+kf3*v3-u3)/k3

aa1=a1

aa2=a2

aa3=a3

aa4=a4

kff1=kf1

kff2=kf2

kff3=kf3

"nominal values

r1:0.5

r2:0.5
```

```
r3:0.2
r4:5
r5:5
r6:1
r7:0.2
t11:4
t12:5
kv1:1600
kv2:1600
kv3:1600
kp1:2000
kp2:2000
kp3:2000
k1:40
k2:20
k3:40
g:9.81
END

CONTINUOUS SYSTEM path
"Path generation for SCARA robot
OUTPUT pd1 pd2 pd3 vd1 vd2 vd3 ad1 ad2 ad3
TIME t
pd1=30*(1-COS(2*22*t/7))
pd2=30*(1-COS(2*22*t/7))
pd3=1*(1-COS(2*22*t/7))
vd1=(30*2*22/7)*SIN(2*22*t/7)
```

```
vd2=(30*2*22/7)*SIN(2*22*t/7)
vd3=(1*2*22/7)*SIN(2*22*t/7)
ad1=(30*(2*22/7)*(2*22/7))*COS(2*22*t/7)
ad2=(30*(2*22/7)*(2*22/7))*COS(2*22*t/7)
ad3=(1*(2*22/7)*(2*22/7))*COS(2*22*t/7)
END

CONTINUOUS SYSTEM cost
"performance index for SCARA robot
INPUT perr1 perr2 perr3 verr1 verr2 verr3 u1 u2 u3 h1 h2 h3 h4
INPUT a1 a2 a3 a4  aa1 aa2 aa3 aa4
INPUT kf1 kf2 kf3 kff1 kff2 kff3
STATE x
DER dx
TIME t
"temporary variable
tem1=q11*SQR(verr1)
tem2=q22*SQR(verr2)
tem3=q33*SQR(verr3)
tem4=q44*SQR(perr1)
tem5=q55*SQR(perr2)
tem6=q66*SQR(perr3)
tem7=r11*SQR(u1)
tem8=r22*SQR(u2)
tem9=r33*SQR(u3)
error1=(t11*verr1)+(t12*perr1)
error2=(t11*verr2)+(t12*perr2)
```

```
error3=(t11*verr3)+(t12*perr3)
dif1=rr1*SQR(aa1-a1)
dif2=rr2*SQR(aa2-a2)
dif3=rr3*SQR(aa3-a3)
dif4=rr4*SQR(aa4-a4)
dif5=rr5*SQR(kff1-kf1)
dif6=rr6*SQR(kff2-kf2)
dif7=rr7*SQR(kff3-kf3)
dif8=h1*SQR(error1)
dif9=h3*SQR(error2)
dif10=h4*SQR(error3)
dif11=2*h2*error1*error2
dif12=p11*SQR(perr1)
dif13=p22*SQR(perr2)
dif14=p33*SQR(perr3)
temp1=tem1+tem2+tem3+tem4+tem5+tem6+tem7+tem8+tem9
temp2=(dif1+dif2+dif3+dif4+dif5+dif6+dif7)
temp3=(dif8+dif9+dif10+dif11+dif12+dif13+dif14)
"sum and integrate of cost functions
dx=temp1
sum=temp2+temp3
"nominal values
q11:25600
q22:25600
q33:25600
q44:40000
q55:40000
```

```
q66:40000
r11:0.01
r22:0.01
r33:0.01
rr1:2
rr2:2
rr3:5
rr4:0.2
rr5:0.2
rr6:1
rr7:5
p11:32000
p22:32000
p33:32000
t11:4
t12:5
END

CONNECTING SYSTEM conn
"connecting system for SCARA robot and adaptive contr
t1[robot]=t1[contr]
t2[robot]=t2[contr]
t3[robot]=t3[contr]
u1[cost]=u1[contr]
u2[cost]=u2[contr]
u3[cost]=u3[contr]
p1[contr]=p1[robot]
```

```
p2[contr]=p2[robot]

p3[contr]=p3[robot]

v1[contr]=v1[robot]

v2[contr]=v2[robot]

v3[contr]=v3[robot]

pd1[contr]=pd1[path]

pd2[contr]=pd2[path]

pd3[contr]=pd3[path]

vd1[contr]=vd1[path]

vd2[contr]=vd2[path]

vd3[contr]=vd3[path]

ad1[contr]=ad1[path]

ad2[contr]=ad2[path]

ad3[contr]=ad3[path]

perr1[cost]=perr1[contr]

perr2[cost]=perr2[contr]

perr3[cost]=perr3[contr]

verr1[cost]=verr1[contr]

verr2[cost]=verr2[contr]

verr3[cost]=verr3[contr]

a1[cost]=a1[robot]

a2[cost]=a2[robot]

a3[cost]=a3[robot]

a4[cost]=a4[robot]

kf1[cost]=kf1[robot]

kf2[cost]=kf2[robot]

kf3[cost]=kf3[robot]
```

```
aa1[cost]=aa1[contr]
aa2[cost]=aa2[contr]
aa3[cost]=aa3[contr]
aa4[cost]=aa4[contr]
kff1[cost]=kff1[contr]
kff2[cost]=kff2[contr]
kff3[cost]=kff3[contr]
h1[cost]=h1[robot]
h2[cost]=h2[robot]
h3[cost]=h3[robot]
h4[cost]=h4[robot]
END
```

ระบบเป็นแขนหุ่นยนต์แบบทรงกลม

```
CONTINUOUS SYSTEM robot
"simple Spherical robot
STATE x1 x2 x3 x4 x5 x6
DER dx1 dx2 dx3 dx4 dx5 dx6
INPUT t1 t2 t3
OUTPUT p1 p2 p3 v1 v2 v3 a1 a2 a3 a4 a5 a6 h1 h2 h3
OUTPUT kf1 kf2 kf3
TIME t
"abbreviate notation
s1=sin(x1)
s2=sin(x2)
c1=cos(x1)
c2=cos(x2)
```

```
"constant parameters
l=l1-l2+lc2
a1=j1+j2+(m1*lc1*lc1)+(m2*l*l)
a2=m2*l
a3=m2
a4=j1p+j2p
a5=j3
a6=m1*lc1
"inertia matrix elements
h11=a1+(2*a2*x2)+(a3*x2*x2)
h22=a3
h33=(a1+2*a2*x2+a3*x2*x2)*s1*s1+(a4*c1*c1)+a5
"viscous-friction force
kf1=75
kf2=10
kf3=1
fv1=kf1*x4
fv2=kf2*x5
fv3=kf3*x6
"temporary variable
temp11=(k1*t1)-fv1-(a2*x5*x4)-(a3*x2*x5*x4)-(a2*x4*x5)
temp12=(a1+2*a2*x2+a3*x2*x2-a4)*s1*c1*x6*x6-(a3*x2*x4*x5)
temp13=(a6+a2+a3*x2)*s1*g
temp21=(k2*t2)-fv2+(a2*x4*x4)+(a3*x2*x4*x4)
temp22=(a2*s1*s1*x6*x6)+(a3*x2*s1*s1*x6*x6)
temp23=-(a3*c1*g)
temp31=(k3*t3)-fv3-(a1+2*a2*x2+a3*x2*x2-a4)*s1*c1*x6*x4
```

```
temp32=-(a2*s1*s1*x6*x5+a3*x2*s1*s1*x6*x5)
temp33=-((a1+2*a2*x2+a3*x2*x2-a4)*s1*c1*x4*x6)
temp34=-(a2*s1*s1*x5*x6+a3*x2*s1*s1*x5*x6)
temp1=temp11+temp12+temp13
temp2=temp21+temp22+temp23
temp3=temp31+temp32+temp33+temp34
"state equaions
dx1=x4
dx2=x5
dx3=x6
dx4=temp1/h11
dx5=temp2/h22
dx6=temp3/h33
"convert radians to degrees
pp1=((180*7)/22)*x1
pp2=x2
pp3=((180*7)/22)*x3
vv1=((180*7)/22)*x4
vv2=x5
vv3=((180*7)/22)*x6
aa1=((180*7)/22)*dx4
aa2=dx5
aa3=((180*7)/22)*dx6
"output
p1=x1
p2=x2
p3=x3
```

```
v1=x4
v2=x5
v3=x6
h1=h11
h2=h22
h3=h33
"nominal values
l1:1
l2:1
lc1:0.5
lc2:0.5
m1:10
m2:10
j1:0.5
j1p:0.01
j2:0.5
j2p:0.01
j3:1
k1:40
k2:20
k3:40
g:9.81
x1:0
x2:0
x3:0
x4:0
x5:0
```

```
x6:0
END

CONTINUOUS SYSTEM contr
"adaptive controller for spherical robot
INPUT p1 p2 p3 v1 v2 v3 pd1 pd2 pd3 vd1 vd2 vd3 ad1 ad2 ad3
OUTPUT t1 t2 t3 perr1 perr2 perr3 verr1 verr2 verr3
OUTPUT u1 u2 u3 aa1 aa2 aa3 aa4 aa5 aa6 kff1 kff2 kff3
STATE a1 a2 a3 a4 a5 a6 kf1 kf2 kf3
DER da1 da2 da3 da4 da5 da6 dkf1 dkf2 dkf3
"abbreviate notation
c1=cos(p1)
c2=cos(p2)
s1=sin(p1)
s2=sin(p2)
"errors
perr1=p1-((22*pd1)/(7*180))
perr2=p2-pd2
perr3=p3-((22*pd3)/(7*180))
err1=v1-((22*vd1)/(7*180))
err2=v2-vd2
verr3=v3-((22*vd3)/(7*180))
perr1d=((180*7)/22)*perr1
perr2m=perr2
perr3d=((180*7)/22)*perr3
verr1d=((180*7)/22)*verr1
verr2m=verr2
```

```
verr3d=((180*7)/22)*verr3
err1=(verr1)+(t12*perr1/t11)
err2=(verr2)+(t12*perr2/t11)
err3=(verr3)+(t12*perr3/t11)
"controller variables
dqr1=((22*vd1)/(7*180))-((t12*perr1)/t11)
dqr2=vd2-((t12*perr2)/t11)
dqr3=((22*vd3)/(7*180))-((t12*perr3)/t11)
ddqr1=((22*ad1)/(7*180))-((t12*verr1)/t11)
ddqr2=ad2-((t12*verr2)/t11)
ddqr3=((22*ad3)/(7*180))-((t12*verr3)/t11)
y11=ddqr1-(s1*c1*v3*dqr3)
y12=2*p2*ddqr1+v2*dqr1+v1*dqr2-2*p2*s1*c1*v3*dqr3-s1*g
temp1=p2*s1*g
y13=p2*p2*ddqr1+p2*v2*dqr1+p2*v1*dqr2-p2*p2*s1*c1*v3*dqr3-temp1
y14=s1*c1*v3*dqr3
y16=-s1*g
y22=-(v1*dqr1+s1*s1*v3*dqr3)
y23=ddqr2-p2*v1*dqr1-p2*s1*s1*v3*dqr3+c1*g
y31=s1*s1*ddqr3+s1*c1*v3*dqr1+s1*c1*v1*dqr3
temp2=2*p2*s1*c1*v1*dqr3+s1*s1*v2*dqr3
y32=2*p2*s1*s1*ddqr3+2*p2*s1*c1*v3*dqr1+s1*s1*v3*dqr2+temp2
temp3=p2*p2*s1*c1*v1*dqr3+p2*s1*s1*v2*dqr3
y33=p2*p2*s1*s1*ddqr3+p2*p2*s1*c1*v3*dqr1+p2*s1*s1*v3*dqr2+temp3
y34=c1*c1*ddqr3-s1*c1*v3*dqr1-s1*c1*v1*dqr3
y35=ddqr3
"adaptive equations
```

```
da1=-(r1*t11*t11*(y11*err1+y31*err3))
da2=-(r2*t11*t11*(y12*err1+y22*err2+y32*err3))
da3=-(r3*t11*t11*(y13*err1+y23*err2+y33*err3))
da4=-(r4*t11*t11*(y14*err1+y34*err3))
da5=-(r5*t11*t11*(y35*err3))
da6=-(r6*t11*t11*(y16*err1))
dkf1=-(r7*t11*t11*v1*err1)
dkf2=-(r8*t11*t11*v2*err2)
dkf3=-(r9*t11*t11*v3*err3)
"output
u1=(kv1*verr1)+(kp1*perr1)
u2=(kv2*verr2)+(kp2*perr2)
u3=(kv3*verr3)+(kp3*perr3)
t1=(y11*a1+y12*a2+y13*a3+y14*a4+y16*a6+kf1*v1-u1)/k1
t2=(y22*a2+y23*a3+kf2*v2-u2)/k2
t3=(y31*a1+y32*a2+y33*a3+y34*a4+y35*a5+kf3*v3-u3)/k3
aa1=a1
aa2=a2
aa3=a3
aa4=a4
aa5=a5
aa6=a6
kff1=kf1
kff2=kf2
kff3=kf3
"nominal values
g:9.81
```

```
r1:0.2
r2:0.2
r3:0.5
r4:0.01
r5:0.1
r6:0.2
r7:5
r8:1
r9:0.2
t11:4
t12:5
kv1:1600
kv2:1600
kv3:1600
kp1:2000
kp2:2000
kp3:2000
k1:40
k2:20
k3:40
END

CONTINUOUS SYSTEM path
"Path generation for sphericl robot
OUTPUT pd1 pd2 pd3 vd1 vd2 vd3 ad1 ad2 ad3
TIME t
pd1=30*(1-COS(2*22*t/7))
```

```
pd2=0.5*(1-COS(2*22*t/7))
pd3=45*(1-COS(2*22*t/7))
vd1=(30*2*22/7)*SIN(2*22*t/7)
vd2=(0.5*2*22/7)*SIN(2*22*t/7)
vd3=(45*2*22/7)*SIN(2*22*t/7)
ad1=(30*(2*22/7)*(2*22/7))*COS(2*22*t/7)
ad2=(0.5*(2*22/7)*(2*22/7))*COS(2*22*t/7)
ad3=(45*(2*22/7)*(2*22/7))*COS(2*22*t/7)
END

CONTINUOUS SYSTEM cost
"performance index for sphericl robot
INPUT perr1 perr2 perr3 verr1 verr2 verr3 u1 u2 u3 h1 h2 h3
INPUT a1 a2 a3 a4 a5 a6 aa1 aa2 aa3 aa4 aa5 aa6
INPUT kf1 kf2 kf3 kff1 kff2 kff3
STATE x
DER dx
TIME t
"temporary variable
tem1=q11*SQR(verr1)
tem2=q22*SQR(verr2)
tem3=q33*SQR(verr3)
tem4=q44*SQR(perr1)
tem5=q55*SQR(perr2)
tem6=q66*SQR(perr3)
tem7=r11*SQR(u1)
tem8=r22*SQR(u2)
```

```
tem9=r33*SQR(u3)
error1=(t11*verr1)+(t12*perr1)
error2=(t11*verr2)+(t12*perr2)
error3=(t11*verr3)+(t12*perr3)
dif1=rr1*SQR(aa1-a1)
dif2=rr2*SQR(aa2-a2)
dif3=rr3*SQR(aa3-a3)
dif4=rr4*SQR(aa4-a4)
dif5=rr5*SQR(aa5-a5)
dif6=rr6*SQR(aa6-a6)
dif7=rr7*SQR(kff1-kf1)
dif8=rr8*SQR(kff2-kf2)
dif9=rr9*SQR(kff3-kf3)
dif10=h1*SQR(error1)
dif11=h2*SQR(error2)
dif12=h3*SQR(error3)
dif13=p11*SQR(perr1)
dif14=p22*SQR(perr2)
dif15=p33*SQR(perr3)
temp1=tem1+tem2+tem3+tem4+tem5+tem6+tem7+tem8+tem9
temp2=(dif1+dif2+dif3+dif4+dif5+dif6+dif7+dif8)
temp3=(dif9+dif10+dif11+dif12+dif13+dif14+dif15)
"sum and integrate of cost functions
dx=temp1
sum=temp2+temp3
"nominal values
q11:25600
```

```
q22:25600
q33:25600
q44:40000
q55:40000
q66:40000
r11:0.01
r22:0.01
r33:0.01
rr1:5
rr2:5
rr3:2
rr4:100
rr5:10
rr6:5
rr7:0.2
rr8:1
rr9:5
p11:32000
p22:32000
p33:32000
t11:4
t12:5
END

CONNECTING SYSTEM conn
"connecting system for spherical robot and adaptive contr
t1[robot]=t1[contr]
```

t2[robot]=t2[contr]

t3[robot]=t3[contr]

u1[cost]=u1[contr]

u2[cost]=u2[contr]

u3[cost]=u3[contr]

p1[contr]=p1[robot]

p2[contr]=p2[robot]

p3[contr]=p3[robot]

v1[contr]=v1[robot]

v2[contr]=v2[robot]

v3[contr]=v3[robot]

pd1[contr]=pd1[path]

pd2[contr]=pd2[path]

pd3[contr]=pd3[path]

vd1[contr]=vd1[path]

vd2[contr]=vd2[path]

vd3[contr]=vd3[path]

ad1[contr]=ad1[path]

ad2[contr]=ad2[path]

ad3[contr]=ad3[path]

perr1[cost]=perr1[contr]

perr2[cost]=perr2[contr]

perr3[cost]=perr3[contr]

verr1[cost]=verr1[contr]

verr2[cost]=verr2[contr]

verr3[cost]=verr3[contr]

a1[cost]=a1[robot]

```
a2[cost]=a2[robot]
a3[cost]=a3[robot]
a4[cost]=a4[robot]
a5[cost]=a5[robot]
a6[cost]=a6[robot]
kf1[cost]=kf1[robot]
kf2[cost]=kf2[robot]
kf3[cost]=kf3[robot]
aa1[cost]=aa1[contr]
aa2[cost]=aa2[contr]
aa3[cost]=aa3[contr]
aa4[cost]=aa4[contr]
aa5[cost]=aa5[contr]
aa6[cost]=aa6[contr]
kff1[cost]=kff1[contr]
kff2[cost]=kff2[contr]
kff3[cost]=kff3[contr]
h1[cost]=h1[robot]
h2[cost]=h2[robot]
h3[cost]=h3[robot]
END
```

ประวัติผู้เขียน

นายศิริ ศุภพัฒน์ เกิดวันที่ 17 สิงหาคม พ.ศ. 2509 ที่อำเภอเมือง จังหวัดจันทบุรี สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ในปีการศึกษา 2531 และเข้าศึกษาต่อในหลักสูตรปริญญาวิศวรรมศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อพ.ศ. 2533