รายการอ้างอิง

Ambrosino, G., Celentano, G., and Galofalo, F. Adaptive tracking control of industrial robots. Trans. ASME. J. Dynam. Syst. Measure. Contr. 110 (September 1988) : 215-220.

Deng, Z. -L., and Huang, X. -R. Multivariable decoupling pole assignment self-tuning feedforward controller. IEE Proc. D, Control Theory and Appl. 138 (January 1991) : 85-88.

Elmaraghy, H.A., Johns, B. An investigation into the compliance of SCARA robots. part 1: analytical model. Trans. ASME. J. Dynam. Syst. Measure. Contr. 110 (March 1988) : 8-22.

Elmaraghy, H.A., and Johns, B. An investigation into the compliance of SCARA robots. part 2 : experimental and numerical validation. Trans. ASME. J. Dynam. Syst. Measure. Contr. 110 (March 1988) : 23-30.

Fu, K.S., Gonzalez, R.C., and Lee, C.S.G. Robotics : control sensing,vision,and intelligence. New York : McGraw-Hill, 1987.

Goldenberg, A.A., Apkarian, J.A., and smith, H.W. An approach to adaptive control of robot manipulators using the computed torque technique. Trans. ASME. J. Dynam. Syst. Measure. Contr. 111 (March 1989) : 1-8.

Gu, K., and Tongue, B.H. A new strategy for adaptive motion control of robots. Trans. ASME. J. Dynam. Syst. Measure. Contr. 112 (September 1990) : 410-426.

Jayasuriya, S., and Hwang, C.H. Tracking controllers for robot manipulators: a high gain perspective. Trans. ASME. J. Dynam. Syst. Measure. Contr. 110 (March 1988) : 39-45.

Johansson, R. Quadratic optimization of motion coordination and control. IEEE Trans.Automat. Contr. 35 (November 1990) : 1197-1208.

Khosla, P.K., and Kanade, T. Real-time implementation and evaluation of computed-torque scheme. IEEE Trans.Automat. Contr. 5 (April 1989) : 245-253.

Koivo, A., J. Fundamentals for control of robotic manipulator. Singapore : John Willey and Son, 1989.

Leahy, M.B., Valavanis, K.P., and Saridis, G.N. Evaluation of dynamic models for PUMA robot control. IEEE Trans. Automat. Contr. 5 (April 1989) : 242-245.

Lee, C.S.G. Robot arm kinematics, dynamics, and control. IEEE Computer 15 no.12 (1982) : 62-80.

Liu, M.-H, and Lin, W. Multivariable self-tuning control with decoupling for robotic manipulators. IEE Proc. D, Control Theory and Appl. 135 (January 1988) : 43-48.

Middleton, R.H. Adaptive control for robot manipulators using discrete time identification. IEEE Trans. Automat. Contr. 35 (May 1990) : 633-637.

Neuman, C.P., and Murray, J.J. The complete dynamic model and customized algorithms of the PUMA robot. IEEE Trans. Syst., Man, Cybern. 17 (July 1987) : 635-654.

Novakovic, Z.R. The principle of self-support in robot control synthesis. IEEE Trans. Syst. Man, Cybern. 21 (January 1991) : 206-220.

Puthenpura, S.C., and MacGregor, J.F. Controllers and self-tuning regurators with better set-point tracking. IEE Proc. D, Control Theory and Appl. 134 (January 1987) : 26-30

Sadegh, N., Horowitz, R., Kao, W.W., and Tomizuka, M. A unified approach to the design of adaptive and repetitive controller for robotic manipulators. Trans. ASME. J. Dynam. Syst. Measure. Contr. K 110 (March 1988) : 94-96.

Sadegh, N., Horowitz, R., Kao, W.W., and Tomizuka, M. A unified approach to the design of adaptive and repetitive controller for robotic manipulators. Trans. ASME. J. Dynam. Syst. Measure. Contr. 112 (December 1990) : 618-629.

Seraji, H. Decentralized adaptive control of manipulators: theory, simulation, and experimentation. IEEE Trans. Robotics Automat. 5 no.2 (April 1989) : 183-201.

Slotine, J.-J., E., and Li, W. Applied nonlinear control. New Jersey : Prentice-Hall, 1991.

Spong, M.W., and Ortega, R. On adaptive inverse dynamics control of rigid robots. IEEE Trans. Automat. Contr. 35 (January 1990) : 92-95.

Stokic, D., and Vukobratovic, M. Practical stabilization of robotic systems by decentralized control. Automatica. 20 no.3 : 353-358.

Sugie, T., Yoshigawa, T., and Ono, T. Robust controller design for robot manipulators. Trans. ASME. J. Dynam. Syst. Measure. Contr. 110 (March 1988) : 94-96.

Tadikonda, S., and Baruh, H. Pointwise-optimal control of robotic manipulator. Trans. ASME. J. Dynam. Syst. Measure. Contr. 110 (June 1988) : 210-213.

Tamura, H., and Nagahama, K. Decentralized adaptive control of robotic manipulators. INT. J. Systems Sci. 19 (1988):2067-2078.

Tomizuka, M., Horowitz, R., Anwar, G., and Jia, Y.L. Implementation of adaptive techniques for motion control of robotic manipulators. Trans. ASME. J. Dynam. Syst. Measure. Contr. 110 (March 1988) : 62-69.

Wang, D., and Leondes, C.T. Robust tracking control of non-linear systems with uncertain dynamics : part 1. INT. J. Systems Sci. 20 no.12 (1989) : 2619-2641.

Wang, D., and Leondes, C.T. Robust tracking control of non-linear systems with uncertain dynamics : part 2. INT. J. Systems Sci. 20 no.12 (1989) : 2643-2661.

ภาคผนวก

ภาคผนวก ก.

หลักของ Hamiltion และ สมการของ Lagrange-Euler

หลักของ Hamiltion

พิจารณาการเคลื่อนที่ของมวล m ซึ่งเคลื่อนที่เนื่องจากแรง F และกำหนดเวคเตอร์ $\mathbf{r} = \mathbf{r}(t)$ เป็นเวคเตอร์ชี้ตำแหน่งของมวล m โดยอ้างอิงกับระบบพิกัดคงที่ จะได้กฎของ Newton อยู่ในรูป

$$m\frac{d\mathbf{r}(t)}{dt} - \mathbf{F} = 0 \qquad \text{(ก.1)}$$

สมมุติให้การเคลื่อนที่มีการขยับไปจากเส้นทางที่ระบุ (nominal path) เล็กน้อย ทำให้เกิดเส้นทางการเคลื่อนที่ใหม่คือ $\mathbf{r}(t) + \delta\mathbf{r}(t)$ โดยที่ $\mathbf{r}(t)$ และ $\delta\mathbf{r}(t)$ รวมทั้งอนุพันธ์ของเวคเตอร์ทั้งสองมีความต่อเนื่อง การเคลื่อนที่ของมวล m เริ่มต้นที่เวลา t_0 จนกระทั่งเวลา t_1 และกำหนดให้ตำแหน่งเริ่มต้น $\mathbf{r}(t_0)$ และตำแหน่งสุดท้าย $\mathbf{r}(t_1)$ เป็นตำแหน่งตายตัว กล่าวคือ $\delta\mathbf{r}(t_0)$ และ $\delta\mathbf{r}(t_1)$ มีค่าเป็นศูนย์

เมื่อแทนค่าเส้นทางการเคลื่อนที่ใหม่ และอินทิเกรชันสมการ (ก.1) เทียบกับเวลา จะได้

$$\int_{t_0}^{t_1}\left[m\frac{d^2\mathbf{r}^T}{dt^2}\delta\mathbf{r} - \mathbf{F}^T\delta\mathbf{r}\right]dt = 0 \qquad \text{(ก.2)}$$

โดยที่

$\mathbf{F}^T\delta\mathbf{r}$ คือ งานซึ่งทำโดยแรง F ทำให้เกิดการขจัด (displacement) $\delta\mathbf{r}$

เมื่ออินทิเกรชัน บาย พาร์ท พจน์แรกทางซ้ายมือของสมการ (ก.2) จะได้เป็น

$$\int_{t_0}^{t_1} m\frac{d^2\mathbf{r}^T}{dt^2}\delta\mathbf{r}\,dt = \left[m\frac{d\mathbf{r}^T}{dt}\delta\mathbf{r}\right]_{t_0}^{t_1} - \int_{t_0}^{t_1} m\frac{d\mathbf{r}^T}{dt}\frac{d(\delta\mathbf{r})}{dt}dt \qquad \text{(ก.3)}$$

พจน์แรกทางขวามือของสมการ (ก.3) มีค่าเป็นศูนย์ เนื่องจาก $\delta\mathbf{r}(t_0)$ และ $\delta\mathbf{r}(t_1)$ มีค่าเป็นศูนย์

สำหรับพจน์ที่สองทางขวามือในเครื่องหมายอินทิเกรชันคือ การแปรผันอันดับหนึ่ง (first-order variation) ของพลังงานจลน์ (δK)

โดยที่

$$\delta K = m\dot{\mathbf{r}}^T\delta\dot{\mathbf{r}}$$
$$= m\frac{d\mathbf{r}^T}{dt}\frac{d(\delta\mathbf{r})}{dt} \qquad \text{\_\_\_\_\_(ก.4)}$$

เมื่อ

$$\text{พลังงานจลน์ } K = \frac{1}{2}m\dot{\mathbf{r}}^T\dot{\mathbf{r}}$$

ดังนั้นสมการ (ก.2) สามารถเขียนได้เป็น

$$-\int_{t_0}^{t_1}\left[\delta K + \mathbf{F}^T\delta\mathbf{r}\right]dt = 0 \qquad \text{\_\_\_\_\_(ก.5)}$$

สมการ (ก.5) คือ หลักของ Hamilton สำหรับมวลเดี่ยว

สำหรับระบบที่มีการอนุรักษ์พลังงาน (conservative system) พลังงานศักย์ P จะอยู่ในรูปความสัมพันธ์

$$\mathbf{F}^T\delta\mathbf{r} = -\delta P \qquad \text{\_\_\_\_\_(ก.6)}$$

ดังนั้นจากสมการ (ก.5) และ (ก.6) จะได้ หลักของ Hamilton ในรูปความสัมพันธ์ของพลังงานศักย์และพลังงานจลน์ คือ

$$\delta\int_{t_0}^{t_1}(K-P)\,dt = 0 \qquad \text{\_\_\_\_\_(ก.7)}$$

ผลต่างระหว่างพลังงานจลน์และพลังงานศักย์ ตามสมการ (ก.7) เรียกว่า ฟังก์ชันพลังงานของ Lagrange

$$L = K - P \qquad \text{\_\_\_\_\_(ก.8)}$$

ฟังก์ชันของ Lagrange ดังกล่าว พิจารณาจากมวลเดี่ยว แต่สามารถนำมาใช้กับระบบที่ประกอบด้วย วัตถุแข็งเกร็ง (rigid body) ได้

สมการของ Lagrange-Euler

พิจารณาระบบพลวัตที่มีตัวแปรอิสระทั้งหมด N ตัว คือ $q_1 = q_1(t)$, $q_2 = q_2(t)$, ... , $q_N = q_N(t)$ สามารถกำหนดตัวแปรดังกล่าวให้เป็นพิกัดทั่วไป (generalized coordinates) ซึ่งมีความต่อเนื่องตลอดเส้นทางที่ระบุ (nominal trajectory) ของการเคลื่อนที่ และกำหนดให้จุดปลายของเส้นทางที่ระบุมีค่าตายตัว ดังนั้นจะได้ฟังก์ชันของ Lagrange ตลอดเส้นทางการเคลื่อนที่คือ

$$L_1 \quad = \quad L\left[q_1(t) \cdots q_N(t) \;\; \dot{q}_1(t) \cdots \dot{q}_N(t)\right] \qquad \text{\_\_\_\_\_(ก.9)}$$

สมมุติให้การเคลื่อนที่มีการขยับไปจากเส้นทางที่ระบุเล็กน้อย เป็นผลให้เกิดเส้นทางใหม่คือ $q_i(t) + \delta q_i(t)$, $i = 1, 2, \ldots, N$ โดยที่ $\delta q_i(t)$ มีความต่อเนื่อง ทำให้ได้ฟังก์ชันของ Lagrange สำหรับเส้นทางใหม่ดังนี้

$$L_2 \quad = \quad L\left[q_1(t)+\delta q_1(t) \cdots q_N(t)+\delta q_N(t) \;\;\; \dot{q}_1(t)+\delta\dot{q}_1(t) \cdots \dot{q}_N(t)+\delta\dot{q}_N(t)\right]$$

_____(ก.10)

กระจายอนุกรมเทเลอร์ของผลต่าง $(L_2 - L_1)$ รอบๆเส้นทางที่ระบุ โดยถือว่าพจน์ที่มีการแปรผันอันดับสองขึ้นไปมีค่าน้อยมาก เมื่อเทียบกับพจน์ที่มีการแปรผันอันดับหนึ่ง สามารถตัดทิ้งได้ เมื่ออินทิเกรชันผลต่าง $(L_2 - L_1)$ เทียบกับเวลา โดยคงไว้แต่พจน์ที่มีการแปรผันอันดับหนึ่ง คือ $\int \delta L \, dt$ ทำให้ได้

$$\int_{t_0}^{t_1} \delta L\left[q_1 \cdots q_N \;\; \dot{q}_1 \cdots \dot{q}_N\right] dt$$

$$= \int_{t_0}^{t_1}\left[\frac{\partial L}{\partial q_1}\delta q_1 \;+\; \cdots \;+\; \frac{\partial L}{\partial q_N}\delta q_N \;+\; \frac{\partial L}{\partial \dot{q}_1}\delta \dot{q}_1 \;+\; \cdots \;+\; \frac{\partial L}{\partial \dot{q}_N}\delta \dot{q}_N\right]dt$$

_____(ก.11)

จากหลักของ Hamilton ตามสมการ (ก.7) แสดงว่าสมการ (ก.11) มีค่าเป็นศูนย์ และเมื่ออินทิเกรชัน บายพาร์ท สมการ (ก.11) และจัดรูปใหม่จะได้

$$\left[\frac{\partial L}{\partial \dot{q}_1}\delta q_1\right]_{t_0}^{t_1} + \cdots + \left[\frac{\partial L}{\partial \dot{q}_N}\delta q_N\right]_{t_0}^{t_1}$$
$$+ \int_{t0}^{t1}\left\{\left[\frac{\partial L}{\partial q_1} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right)\right]\delta q_1 + \cdots + \left[\frac{\partial L}{\partial q_N} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_N}\right)\right]\delta q_N\right\} dt = 0 \qquad \text{(ก.12)}$$

เนื่องจากจุดปลายของเส้นทางที่ระบุถูกกำหนดค่าตายตัว ทำให้ N พจน์แรกของสมการ (ก.12) มีค่าเป็นศูนย์ ดังนั้นเงื่อนไขจำเป็น (necessary condition) ที่ทำให้สมการ (ก.12) เป็นศูนย์คือ พจน์ในเครื่องหมายอินทิเกรชันต้องเป็นศูนย์ แต่เนื่องจากการแปรผัน $\delta q_i(t)$ มีค่าใดๆก็ได้ ดังนั้นจะได้เงื่อนไข

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0 \qquad i = 1, 2, \ldots, N \qquad \text{(ก.13)}$$

สมการ (ก.13) เรียกว่า สมการของ Lagrange-Euler สำหรับระบบที่มีการอนุรักษ์พลังงาน

ในกรณีที่มีแรงภายนอกมากระทำต่อระบบ จะปรากฏอยู่ทางขวามือของสมการ (ก.13) ดังนี้

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = F_i \qquad i = 1, 2, \ldots, N \qquad \text{(ก.14)}$$

เมื่อ F_i คือ แรงภายนอกที่มากระทำต่อระบบ ซึ่งอ้างอิงทิศทางตามระบบพิกัด q_i

<u>สมการ Lagrange-Euler สำหรับการเคลื่อนที่ของแขนหุ่นยนต์</u>

กำหนดให้แขนหุ่นยนต์ ซึ่งประกอบด้วยลิงค์มาต่อกันแบบอนุกรม N ลิงค์ มีเมตริกซ์ทรานส์ฟอร์เมชัน คือ T_{i-1}^{i} โดยที่ $i = 1, 2, \ldots, N$ และกำหนดให้มวล dm_h เป็นมวลปริมาณเล็กๆ (differential mass) บนลิงค์ i ซึ่งระบุตำแหน่งโดยเวคเตอร์ p_{ih} ในระบบแกนพิกัด i ตำแหน่งมวล dm_h ที่เดียวกันนี้ สามารถระบุในระบบพิกัดฐาน ได้เป็น

$$\begin{aligned}\mathbf{p}_{0h} &= T_0^1 T_1^2 \cdots T_{i-1}^{i}\mathbf{p}_{ih} \\ &= T_0^i \mathbf{p}_{ih}\end{aligned} \qquad \text{(ก.15)}$$

เมื่อ เมตริกซ์ทรานส์ฟอร์เมชัน $T_0^i = T_0^1 T_1^2 \cdots T_{i-1}^{i}$ และทำให้ได้เวคเตอร์ความเร็ว ซึ่งสัมพันธ์กับเวคเตอร์ตำแหน่ง $\mathbf{p}_{ih}$ ดังนี้

$$\dot{\mathbf{p}}_{0h} = \frac{d}{dt}\left(T_0^i \mathbf{p}_{ih}\right)$$
$$= \sum_{j=1}^{i} \frac{\partial T_0^i}{\partial q_j} \dot{q}_j \mathbf{p}_{ih} \qquad \text{(ก.16)}$$

โดยที่ q_j คือ พิกัดทั่วไปของข้อที่ j

เมื่อกำหนดเวคเตอร์ระบุตำแหน่ง และความเร็ว ของมวล dm_h แล้ว สามารถหาพลังงานจลน์และพลังงานศักย์ของมวล dm_h ได้ดังต่อไปนี้

พลังงานจลน์ของมวล dm_h คือ

$$dK_h = \tfrac{1}{2}\dot{\mathbf{p}}_{0h}^T \dot{\mathbf{p}}_{0h} dm_h$$
$$= \tfrac{1}{2}\left(\sum_{k=1}^{i} \frac{\partial T_0^i}{\partial q_k} \dot{q}_k \mathbf{p}_{ih}\right)^T \left(\sum_{j=1}^{i} \frac{\partial T_0^i}{\partial q_j} \dot{q}_j \mathbf{p}_{ih}\right) dm_h \qquad \text{(ก.17)}$$
$$= \tfrac{1}{2}\,\mathrm{trace}\left\{\left(\sum_{k=1}^{i} \frac{\partial T_0^i}{\partial q_k} \dot{q}_k \mathbf{p}_{ih}\right)\left(\sum_{j=1}^{i} \frac{\partial T_0^i}{\partial q_j} \dot{q}_j \mathbf{p}_{ih}\right)^T\right\} dm_h$$

เมื่อ trace (.) คือผลบวกของสมาชิกตามแนวทแยงของเมตริกซ์ โดยที่ $\mathbf{q}^T\mathbf{q} = \mathrm{trace}\,(\mathbf{q}\mathbf{q}^T)$ เมื่อ $\mathbf{q}$ เป็นเวคเตอร์ซึ่งมีมิติใดๆ

ดังนั้นสามารถหาค่าพลังงานจลน์ของลิงค์ i โดยการอินทิเกรชันสมการ (ก.17) เทียบกับมวล dm_h ได้ดังนี้

$$K_i = \int_{\text{link } i} dK_h$$
$$= \tfrac{1}{2}\,\mathrm{trace}\left[\sum_{j=1}^{i}\sum_{k=1}^{i} \frac{\partial T_0^i}{\partial q_k}\left(\int_{\text{link } i} \mathbf{p}_{ih}\mathbf{p}_{ih}^T dm_h\right)\left(\frac{\partial T_0^i}{\partial q_j}\right)\dot{q}_j \dot{q}_k\right] \qquad \text{(ก.18)}$$
$$= \tfrac{1}{2}\,\mathrm{trace}\left[\sum_{j=1}^{i}\sum_{k=1}^{i} \frac{\partial T_0^i}{\partial q_j} I_i \left(\frac{\partial T_0^i}{\partial q_k}\right)^T \dot{q}_j \dot{q}_k\right]$$

เมื่อ I_i คือ เมตริกซ์ pseudo-inertia ของลิงค์ i ซึ่งเป็นเมตริกซ์ mass moment ของลิงค์ i รอบจุดกำเนิดของพิกัดอ้างอิงที่ i

โดยที่

$$I_i = \int_{\text{link } i} \mathbf{p}_{ih}\mathbf{p}_{ih}^T dm_h$$

$$= \begin{bmatrix} \frac{1}{2}\left(-I_{x_i} + I_{y_i} + I_{z_i}\right) & I_{x_iy_i} & I_{x_iz_i} & m_i\bar{p}_{x_i} \\ I_{x_iy_i} & \frac{1}{2}\left(I_{x_i} - I_{y_i} + I_{z_i}\right) & I_{y_iz_i} & m_i\bar{p}_{y_i} \\ I_{x_iz_i} & I_{y_iz_i} & \frac{1}{2}\left(I_{x_i} + I_{y_i} - I_{z_i}\right) & m_i\bar{p}_{z_i} \\ m_i\bar{p}_{x_i} & m_i\bar{p}_{y_i} & m_i\bar{p}_{z_i} & m_i \end{bmatrix}$$

_____(ก.19)

จุดศูนย์กลางมวลของลิงค์ i คือ m_i ถูกระบุตำแหน่งโดยเวคเตอร์ $\bar{\mathbf{p}}_i = \left[\bar{p}_{x_i}\bar{p}_{y_i}\bar{p}_{z_i}\,1\right]^T$ พจน์ I_{xi} I_{yi} และ I_{zi} คือ mass moment of inertia อันดับสอง ซึ่งหาค่าได้ดังนี้

$$I_{x_i} = \int (p_{y_i}^2 + p_{z_i}^2)\, dm_i \qquad I_{y_i} = \int (p_{x_i}^2 + p_{z_i}^2)\, dm_i$$

$$I_{z_i} = \int (p_{x_i}^2 + p_{y_i}^2)\, dm_i \qquad I_{wv} = \int p_w p_v\, dm_i\,,\ w \neq v\,;\, w, v = x_i, y_i, z_i$$

_____(ก.20)

องค์ประกอบในแถวและหลักที่สี่ของเมตริกซ์ I_i คือ mass moment อันดับหนึ่งของลิงค์

จากสมการ (ก.18) จะได้ค่าพลังงานจลน์ของระบบแขนหุ่นยนต์ N ลิงค์คือ

$$K = \sum_{i=1}^{N} K_i$$

$$= \frac{1}{2}\sum_{i=1}^{N} \text{trace}\left[\sum_{k=1}^{i}\sum_{j=1}^{i} \frac{\partial T_0^i}{\partial q_k} I_i \left(\frac{\partial T_0^i}{\partial q_j}\right)^T \dot{q}_j \dot{q}_k\right]$$

_____(ก.21)

สำหรับพลังงานศักย์ของลิงค์ i มีค่าดังนี้

$$P_i = -m_i \mathbf{g}^T T_0^i \bar{p}_i$$

_____(ก.22)

เมื่อ เวคเตอร์ $\mathbf{g} = [g_{0x}\ \ g_{0y}\ \ g_{0z}\ \ 0]^T$ คือ ความเร่งเนื่องจากความโน้มถ่วงของโลก ซึ่งแตกองค์ประกอบในแนวแกนพิกัด $x_0\, y_0\, z_0$ (พิกัดฐาน) และจะได้ค่าพลังงานศักย์ของระบบแขนหุ่นยนต์ N ลิงค์ คือ

$$P = \sum_{i=1}^{N} P_i$$

$$= -\sum_{i=1}^{N} m_i \mathbf{g}^T T_0^i \overline{p}_i \qquad \text{\_\_\_\_\_(ก.23)}$$

ดังนั้นฟังก์ชันพลังงานของ Lagrange คือ $L = K - P$ สามารถหาค่าได้จากสมการ (ก.21) และ (ก.23)

จากสมการ (ก.14) สมการการเคลื่อนที่ของ Lagrange-Euler สำหรับลิงค์ที่ n ของแขนหุ่นยนต์ คือ

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_n}\right) - \frac{\partial L}{\partial q_n} = F_n \quad ; n = 1, 2, \ldots, N \qquad \text{\_\_\_\_\_(ก.23)}$$

เมื่อ F_n คือ แรงที่กระทำต่อข้อที่ n ในทิศทางตามระบบแกนพิกัด q_n ค่าอนุพันธ์ของพจน์ในสมการ (ก.23) หาค่าได้ดังนี้

$$\frac{\partial L}{\partial q_n} = \sum_{i=1}^{N}\sum_{k=1}^{i}\sum_{j=1}^{i} \text{trace}\left[\frac{\partial T_0^i}{\partial q_k} I_i \left(\frac{\partial^2 T_0^i}{\partial q_j \partial q_n}\right)^T\right] \dot{q}_j \dot{q}_k + \sum_{i=1}^{N} m_i g^T \frac{\partial T_0^i}{\partial q_n} \overline{p}_i$$

_____(ก.24)

$$\frac{\partial L}{\partial \dot{q}_n} = \sum_{i=1}^{N}\sum_{k=1}^{i} \text{trace}\left[\frac{\partial T_0^i}{\partial q_k} I_i \left(\frac{\partial T_0^i}{\partial q_n}\right)^T\right] \dot{q}_k \qquad \text{\_\_\_\_\_(ก.25)}$$

เนื่องจาก $T_0^i = T_0^1 T_1^2 \ldots T_{i-1}^i$ ขึ้นอยู่กับ $q_1, q_2, \ldots q_i$ เท่านั้น นั่นคือ $\frac{\partial T_0^i}{\partial q_n}$ และ $\frac{\partial T_0^i}{\partial \dot{q}_n}$ เมื่อ $n > i$ มีค่าเป็นศูนย์ ดังนั้นลิมิตล่างของเครื่องหมายซัมเมชันตัวแรกในสมการ (ข.24) และ (ข.25) จึงเริ่มจาก $i = n$ และเมื่อแทนค่าลงในสมการ (ก.23) จะได้ดังสมการ (ก.26)

$$\sum_{i=n}^{N}\left\{\sum_{k=1}^{i} \text{trace}\left[\frac{\partial T_0^i}{\partial q_n} I_i \left(\frac{\partial T_0^i}{\partial q_k}\right)^T\right] \ddot{q}_k + \sum_{k=1}^{i}\sum_{j=1}^{i} \text{trace}\left[\frac{\partial T_0^i}{\partial q_n} I_i \left(\frac{\partial^2 T_0^i}{\partial q_k \partial q_j}\right)^T\right] \dot{q}_k \dot{q}_j - m_i g^T \frac{\partial T_0^i}{\partial q_n} \overline{p}_i\right\}$$

_____(ก.26)

กำหนดสัญญลักษณ์แทนตัวแปรในเครื่องหมายซัมเมชัน และจัดรูปสมการ (ก.26) ใหม่ได้เป็น

$$F_n = \sum_{k=1}^{N} D_{nk}\ddot{q}_k + \sum_{k=1}^{N}\sum_{j=1}^{N} D_{nkj}\dot{q}_k\dot{q}_j + G_n \qquad \text{(ก.27)}$$

เมื่อ

$$D_{nk} = \sum_{i=\max n,k}^{N} \mathrm{trace}\left[\frac{\partial T_0^i}{\partial q_n} I_i \left(\frac{\partial T_0^i}{\partial q_k}\right)^T\right] \qquad \text{(ก.28)}$$

$$D_{nkj} = \sum_{p=\max n,k,j}^{N} \mathrm{trace}\left[\frac{\partial T_0^i}{\partial q_n} I_i \left(\frac{\partial^2 T_0^i}{\partial q_k \partial q_j}\right)^T\right] \qquad \text{(ก.29)}$$

$$G_n = -\sum_{i=n}^{N} m_i \mathbf{g}^T \frac{\partial T_0^i}{\partial q_n} \overline{p}_i \qquad \text{(ก.30)}$$

สมการ (ก.27) ซึ่งเป็นสมการอนุพันธ์แบบไม่เป็นเชิงเส้นอันดับสอง คือแบบจำลองพลวัตของลิงค์ที่ n ของแขนหุ่นยนต์ในรูปสมการ Lagrange-Euler

ภาคผนวก ข

การจัดพารามิเตอร์สมการ Lagrange-Euler ตามวิธีของ Goldenberg et al.

การจัดพารามิเตอร์สมการ Lagrange-Euler สำหรับแขนหุ่นยนต์ N ข้อใดๆ สามารถจัดให้อยู่ในรูปของ dynamics operator $\mathbf{A}(\mathbf{q},\dot{\mathbf{q}})$ ได้ดังนี้

$$\tau = \mathbf{A}(\mathbf{q},\dot{\mathbf{q}})\mathbf{z} \qquad \text{(ข.1)}$$

เมื่อ

τ คือ เวคเตอร์แรงหรือแรงบิดของข้อ

Z คือ เวคเตอร์ Augmented joint coordinates มีค่าดังนี้

$$\mathbf{z}^T = \left(\ddot{\mathbf{q}}^T \ \dot{\mathbf{q}}^T \ \tilde{\mathbf{q}}^T \ 1\right) \qquad \text{(ข.2)}$$

โดยที่

$\ddot{\mathbf{q}}(t)$ คือ เวคเตอร์ความเร่งของข้อ

$\dot{\mathbf{q}}(t)$ คือ เวคเตอร์ความเร็วของข้อ

1 คือ ค่าคงที่ซึ่งเป็นปริมาณสเกลาร์

$\tilde{\mathbf{q}}(t)$ คือ เวคเตอร์ขยาย (extented vector) ของพิกัดของข้อ ซึ่งมีค่าดังนี้

$$\tilde{\mathbf{q}}^T(t) = \left(\tilde{\mathbf{q}}_1^T(t) \ \tilde{\mathbf{q}}_2^T(t) \ \cdots \ \tilde{\mathbf{q}}_N^T(t)\right)^T \qquad \text{(ข.3)}$$

เมื่อ

$$\tilde{\mathbf{q}}_i = \begin{cases} \begin{matrix} \sin\theta_i \\ \cos\theta_i \end{matrix} & \text{, สำหรับข้อชนิดเคลื่อนที่เชิงมุม , } i = 1, 2, \ldots, N \\ \\ d_i & \text{, สำหรับข้อชนิดเคลื่อนที่เชิงเส้น} \end{cases}$$

โดยที่

θ_i คือ มุมของข้อ i และ d_i คือ การขจัด (displacement) ของข้อ i

dynamics operator $\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}})$ ตามสมการ (ข.1) พิจารณาได้จากสมการพลวัต Lagrange-Euler สำหรับแขนหุ่นยนต์ N ข้อใดๆ ซึ่งมีรูปทั่วไปคือ

$$\tau_i = \sum_{j=1}^{N} D_{ij}\ddot{q}_j + I_{a_i}\ddot{q}_i + \sum_{j=1}^{N}\sum_{k=1}^{N} D_{ijk}\dot{q}_j\dot{q}_k + D_i \qquad \text{(ข.4)}$$

เมื่อ

$$D_{ij} = \sum_{p=\max i,j}^{N} \operatorname{trace}\left(\frac{\partial T_p}{\partial q_j} J_p \frac{\partial T_p^T}{\partial q_i}\right) \qquad \text{(ข.5)}$$

$$D_{ijk} = \sum_{p=\max i,j,k}^{N} \operatorname{trace}\left(\frac{\partial^2 T_p}{\partial q_j \partial q_k} J_p \frac{\partial T_p^T}{\partial q_i}\right) \qquad \text{(ข.6)}$$

$$D_i = -\sum_{p=i}^{N} m_p \mathbf{g}^T \frac{\partial T_p}{\partial q_i}\, {}^p\mathbf{r}_p \qquad \text{(ข.7)}$$

โดยที่

τ_i คือ แรงหรือแรงบิดของข้อ i

q_i คือ ตัวแปรของข้อ i ซึ่งแทนตำแหน่งของข้อ

g คือ เวคเตอร์ของความเร่งเนื่องจากแรงโน้มถ่วงของโลก

${}^p\mathbf{r}_p$ คือ เวคเตอร์ชี้ตำแหน่งศูนย์กลางมวลของลิงค์ p เมื่อเทียบกับพิกัดอ้างอิงของลิงค์ p

m_p คือ มวลของลิงค์ p

I_{a_i} คือ actuator inertia ของข้อ i

J_p คือ เมตริกซ์ psudo-inertia ของลิงค์ p

T_p คือ เมตริกซ์ทรานส์ฟอร์เมชัน ระหว่างพิกัดฐานและพิกัดอ้างอิงของลิงค์ p และกำหนดให้ T_{i-1}^i คือ เมตริกซ์ทรานส์ฟอร์เมชัน ระหว่างพิกัดอ้างอิงที่ i-1 และพิกัด อ้างอิงที่ i มีค่าดังนี้

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(ข.8)}$$

เมื่อ α_i คือ มุมบิด (twist angle) ของลิงค์ i และ a_i คือความยาวของลิงค์ i

เมตริกซ์ T_p สามารถแสดงในรูปความสัมพันธ์ระหว่างเมตริกซ์ T_{i-1}^i ได้ดังนี้

$$T_p = T_0^1 \, T_1^2 \, T_2^3 \cdots T_{p-1}^p \qquad \text{(ข.9)}$$

เมื่อแทนค่าเวคเตอร์ $\boldsymbol{z}$ จากสมการ (ข.2) ในสมการ (ข.4) สามารถจัดรูปสมการ (ข.4) ให้อยู่ในรูปสมการ (ข.10) ดังนี้

$$\tau_i = \sum_{j=1}^{N} A_{ij}^1 \ddot{q}_j + \sum_{j=1}^{N} A_{ij}^2 \dot{q}_j + \sum_{j=1}^{N} A_{ij}^3 \tilde{q}_j + A_i^0 \qquad \text{(ข.10)}$$

เมื่อกำหนดค่าต่างๆดังนี้

$$A_{ij}^1 \triangleq \begin{cases} D_{ij} + I_{a_i} & \text{เมื่อ } i = j \\ D_{ij} & \text{เมื่อ } i \neq j \end{cases} \qquad \text{(ข.11)}$$

และ

$$A_{ij}^2 \triangleq \sum_{k=1}^{N} D_{ijk} \dot{q}_k \qquad \text{(ข.12)}$$

แทนค่าสมการ (ข.11) และ (ข.12) ในสมการ (ข.4) ทำให้ได้

$$\tau_i = \sum_{j=1}^{N} A_{ij}^1 \ddot{q}_j + \sum_{j=1}^{N} A_{ij}^2 \dot{q}_j + D_i \qquad \text{(ข.13)}$$

จากสมการ (ข.8) และ (ข.9) จะได้ค่า

$$\frac{\partial T_p}{\partial q_i} = T_0^1 T_1^2 T_2^3 \cdots T_{i-2}^{i-1} \frac{\partial T_{i-1}^i}{\partial q_i} T_i^{i+1} T_{i+1}^{i+2} \cdots T_{p-1}^p \qquad \text{(ข.14)}$$

เมื่อแทนค่าสมการ (ข.14) ในสมการ (ข.7) จะได้

$$D_i = -\sum_{p=i}^{N} m_p \mathbf{g}^T \, T_0^1 T_1^2 T_2^3 \cdots T_{i-2}^{i-1} \frac{\partial T_{i-1}^i}{\partial q_i} T_i^{i+1} T_{i+1}^{i+2} \cdots T_{p-1}^p \; {}^p\mathbf{r}_p \qquad \text{(ข.15)}$$

กำหนดนิยามให้

$$\mathbf{b}_i \triangleq -\mathbf{g}^T T_0^1 T_1^2 T_2^3 \cdots T_{i-1}^i \qquad \text{(ข.16)}$$

และ

$$\widetilde{v}_{ip} \quad \underline{\Delta} \quad m_p\, T_i^{i+1} T_{i+1}^{i+2} T_{i+2}^{i+3} \cdots T_{p-1}^{p}\ {}^{p}\mathbf{r}_p \qquad \text{\_\_\_\_\_(ข.17)}$$

แทนค่าสมการ (ข.16) และ (ข.17) ในสมการ (ข.15) จะได้

$$D_i \quad = \quad \sum_{p=i}^{N} \mathbf{b}_i \frac{\partial T_{i-1}^{i}}{\partial q_i} \widetilde{v}_{ip} \qquad \text{\_\_\_\_\_(ข.18)}$$

$$D_i \quad = \quad \mathbf{b}_i \frac{\partial T_{i-1}^{i}}{\partial q_i} \sum_{p=i}^{N} \widetilde{v}_{ip} \qquad \text{\_\_\_\_\_(ข.19)}$$

และกำหนดเวคเตอร์

$$\begin{aligned} V_i &= \sum_{p=i}^{N} \widetilde{v}_{ip} \\ &= \begin{bmatrix} v_{i1} & v_{i2} & v_{i3} & v_{i4} \end{bmatrix}^T \end{aligned} \qquad \text{\_\_\_\_\_(ข.20)}$$

$$\mathbf{b}i \quad = \quad [\ b_{i1} \ \ b_{i2} \ \ b_{i3} \ \ b_{i4}\ ] \qquad \text{\_\_\_\_\_(ข.21)}$$

แทนค่าสมการ (ข.20) และ (ข.21) ในสมการ (ข.19) จะได้

$$D_i \quad = \quad \begin{bmatrix} b_{i1} & b_{i2} & b_{i3} & b_{i4} \end{bmatrix} \ \frac{\partial T_{i-1}^{i}}{\partial q_i} \begin{bmatrix} v_{i1} \\ v_{i2} \\ v_{i3} \\ v_{i4} \end{bmatrix} \qquad \text{\_\_\_\_\_(ข.22)}$$

จากสมการ (ข.8) ถ้า $q_i \ = \ \theta_i$ (ข้อแบบเคลื่อนที่เชิงมุม)

$$\frac{\partial T_{i-1}^{i}}{\partial q_i} \quad = \quad \begin{bmatrix} -\sin\theta_i & -\cos\theta_i\cos\alpha_i & \cos\theta_i\sin\alpha_i & -a_i\sin\theta_i \\ \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \text{\_\_\_\_\_(ข.23)}$$

แทนค่าสมการ (ข.23) ใน (ข.22) ทำให้ได้ดังสมการ (ข.24)

$$D_i \;=\; -v_{i1}b_{i1}\sin\theta_i \;+\; v_{i1}b_{i2}\cos\theta_i \;-\; v_{i2}b_{i1}\cos\alpha_i\cos\theta_i \;-\; v_{i2}b_{i2}\cos\alpha_i\sin\theta_i$$
$$+v_{i3}b_{i1}\sin\alpha_i\cos\theta_i \;+\; v_{i3}b_{i2}\sin\alpha_i\sin\theta_i \;-\; v_{i4}b_{i1}a_i\sin\theta_i \;+\; v_{i4}b_{i2}a_i\cos\theta_i$$

$$= \begin{bmatrix} -v_{i1}b_{i1} - v_{i2}b_{i2}\cos\alpha_i + v_{i3}b_{i2}\sin\alpha_i - v_{i4}b_{i1}a_i \\ -v_{i1}b_{i2} - v_{i2}b_{i1}\cos\alpha_i + v_{i3}b_{i1}\sin\alpha_i - v_{i4}b_{i2}a_i \end{bmatrix}^T \begin{bmatrix} \sin\theta_i \\ \cos\theta_i \end{bmatrix}$$

$$\triangleq \quad A_i^3\tilde{\mathbf{q}}_i$$

______(ข.24)

จากสมการ (ข.8) ถ้า $q_i = d_i$ (ข้อแบบเคลื่อนที่เชิงเส้น)

$$\frac{\partial T_{i-1}^i}{\partial q_i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \text{\_\_\_\_\_\_(ข.25)}$$

แทนค่าสมการ (ข.25) ในสมการ (ข.22)

$$D_i \;=\; b_{i3}v_{i4} \;\triangleq\; A_i^0 \qquad \text{\_\_\_\_\_\_(ข.26)}$$

ดังนั้นสามารถกำหนด A^3_{ij} ในสมการ (ข.10) ได้ดังนี้

$$A_{ij}^3 \;\triangleq\; \begin{cases} A_i^3 & \text{ถ้า } i = j \\ 0 & \text{ถ้า } i \neq j \end{cases} \qquad \text{\_\_\_\_\_\_(ข.27)}$$

และจากสมการ (ข.24), (ข.26) และ (ข.27) จะได้

$$\tau_i \;=\; \sum_{j=1}^{N} A_{ij}^3\tilde{q}_j \;+\; A_i^0 \qquad \text{\_\_\_\_\_\_(ข.28)}$$

โดยที่

$A^3_{ij} = 0$ ถ้า $q_i = d_i$ และ $A^0_i = 0$ ถ้า $q_i = \theta_i$

ดังนั้นสามารถจัดรูปสมการ (ข.10) ให้อยู่ในรูปสมการ (ข.1) โดยใช้สมการ (ข.24) (ข.26) และ (ข.27) ดังนี้

$$\boldsymbol{\tau} = \left[\begin{array}{ccc:ccc:ccccc:c} A_{11}^{1} & \cdots & A_{1N}^{1} & A_{11}^{2} & \cdots & A_{1N}^{2} & A_{1}^{3} & & & & \mathbf{0} & A_{1}^{0} \\ & & & & & & & A_{2}^{3} & & & & \\ \vdots & & \vdots & \vdots & & \vdots & & & A_{3}^{3} & & & \vdots \\ & & & & & & & & & \ddots & & \\ A_{N1}^{1} & \cdots & A_{NN}^{1} & A_{N1}^{2} & \cdots & A_{NN}^{2} & \mathbf{0} & & & & A_{N}^{3} & A_{N}^{0} \end{array}\right]\mathbf{z}$$

_____(ข.29)

เมื่อ $A^3_i = 0$ ถ้า $q_i = d_i$ และ $A^0_i = 0$ ถ้า $q_i = \theta_i$

ความสัมพันธ์นี้ใช้ได้สำหรับแขนหุ่นยนต์ N ลิงค์ทั่วไป แต่การจัดความสัมพันธ์ลักษณะนี้ไม่เป็นแบบเดียวตายตัว (non-uniqueness) เป็นผลจากการจัดพจน์ A_{ij}^{2} และ A_i^{3} ซึ่งเลือกได้มากกว่าหนึ่งแบบตามความเหมาะสม

ภาคผนวก ค.

โปรแกรมจำลองการทำงานของระบบควบคุมแขนหุ่นยนต์

/* MAIN PROGRAM */

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define  N  2              /*** number of joints ***/
#define  Pi  3.14159265358979323846264 3
#define  r_d  180/Pi
#define  Ts  0.01
#define  Tint 0.002
#define  MAX_step  300
#define  L  4
#define  L1  4*N

float  t, tf, gamma1, gamma2, qi[N], qf[N], q[N], qq[N], qqq[N], qqqr[2][N],
     qqr[2][N], qr[2][N], F[2*N], Xx[2*N];
float  Phi1[N], Z[4*N], A1[N][4*N], P11[4*N][4*N];
float  Vtemp01[4*N], Vtemp11[4*N];
float  U[N], Un[N];
float  Phi[N], dU[N][3], dY[N][3], r01, r11, q00, q10, qtemp[2],
       Theta[N][L], Alpha[N][L], Vtemp20[L], Vtemp21[L], P2[N][L][L];

main( )
{
```

```
int  i, j, k, step, count;
float  temp;
float  D[N][N], Di[N][N], C[N][N], G[N][N], V[N];  FILE  *fq, *fqq, *fu;
fq   = fopen("c:q.gia", "w");
fqq  = fopen("c:qq.gia", "w");
fu   = fopen("c:u.gia", "w");
/****************************************************************/
 printf("\n\nEnter forgetting factor 1:  ");
 scanf("%g", &gamma1);
 printf("\nEnter forgetting factor 2 :  ");
 scanf("%g", &gamma2);
 printf("\nEnter final time :  ");
 scanf("%g", &tf);
 printf("/nPRIMARY CONTROLLER");
 printf("\nSymmetric matrix P1 = pI ,I : identity matrix");
 printf("\nEnter p : ");
 scanf("%g", &temp);
 for(i = 0 ; i < 4*N ; i++)   P11[i][i] = temp;
 for(i = 0 ; i < N ; i++)
   for(j = 0 ; j < L1 ; j++)  A1[i][j] = 1.0;
 printf("/nSECONDARY CONTROLLER (GMV)");
 printf("\nEnter r01, r11 :");
 scanf("%g,%g", &r01, &r11);
 printf("\nEnter q00, q10 :");
 scanf("%g,%g", &q00, &q10);
 printf("P2=pI, enter p = ");
 scanf("%g", &temp);

 for(i = 0 ; i < N ; i++)  {
   for(j = 0 ; j < L ; j++)  {
      P2[i][j][j] = temp;
```

```
            Theta[i][j] = 1.0;
        }
    }
    printf("\nEnter initial positions qi[0],qi[1] (degrees) : ");
    scanf("%g,%g", &qi[0], &qi[1]);
    printf("\nEnter final positions qf[0],qf[1] (degrees) : ");
    scanf("%g,%g", &qf[0], &qf[1]);
    /* degree -> radian */
    qi[0]  /= r_d;   qi[1]  /= r_d;
    qf[0]  /= r_d;   qf[1]  /= r_d;
    t = 0;      Traj(t);
    q[0]     = qi[0];   q[1]      = qi[1];
    qr[0][0] = qi[0];   qr[0][1] = qi[1];
    Xx[0]    = qi[0];   Xx[1] = qi[1];
    Z[4] = sin(q[0]);       Z[5] = cos(q[0]);
    Z[6] = sin(q[1]);       Z[7] = cos(q[1]);
    count = 0;
/****************************************************/


for(step = 0 ; step < MAX_step ; step++)  {
    Traj(t);      /* traj(t+Ts) */


    /*** PRIMARY CONTROLLER ***/
    for(i = 0 ; i < N ; i++)  {
        Phi1[i]      = U[i];
        Z[i]         = qqq[i];
        Z[N+i]       = qq[i];
        Z[2*(N+i)]   = sin(q[i]);
        Z[2*(N+i)+1] = cos(q[i]);
    }
        /*** estimate A(q,qq) ***/
```

```
RLS11();


/*** form vecter Z(qqqr,qqr,qr) ***/
for(i = 0 ; i < N ; i++)  {
   Z[i]          = qqqr[1][i];
   Z[N+i]        = qqr[1][i];
   Z[2*(N+i)]    = sin(qr[1][i]);
   Z[2*(N+i)+1] = cos(qr[1][i]);
}


for(i = 0 ; i < N ; i++)  {
   Un[i] = A1[i][0] * Z[0];
   for(j = 1 ; j < 4*N ; j++)  Un[i] += A1[i][j] * Z[j];
}
/************************************************************/


/*** SECONDARY CONTROLLER ***/


dY[0][0] = q[0] - qr[1][0];
dY[1][0] = q[1] - qr[1][1];


/* parameter estimation */
Alpha[0][0] = dU[0][1];    Alpha[0][1] = dU[1][2];
Alpha[0][2] = dY[0][1];    Alpha[0][3] = dY[0][2];
Alpha[1][0] = dU[1][1];    Alpha[1][1] = dU[0][2];
Alpha[1][2] = dY[1][1];    Alpha[1][3] = dY[1][2];


Phi[0]   = dY[0][0] + r01*dY[0][1] + q00*dU[0][1];
Phi[1]   = dY[1][0] + r11*dY[1][1] + q10*dU[1][1];


RLS2();
```

```
/* control law */

dU[0][0] = (Theta[0][1]*dU[1][1]  + Theta[0][2]*dY[0][0]
          + Theta[0][3]*dY[0][1]) / (-Theta[0][0]);
dU[1][0] = (Theta[1][1]*dU[0][1]  + Theta[1][2]*dY[1][0]
          + Theta[1][3]*dY[1][1]) / (-Theta[1][0]);


for(i=0 ; i<N ; i++)  {
  for(j=2 ; j>0 ; j--)  dU[i][j]  = dU[i][j-1];
  for(j=2 ; j>0 ; j--)  dY[i][j]  = dY[i][j-1];
}
/*****************************************************************/
/* joint input torques = nominal torques + perturbation torques */


for(i = 0 ; i < N ; i++)   U[i] = Un[i] + dU[i][0];


RKF5( );   /* t = t+Ts */


if(count == 0)   {
fprintf(fq,"%g,%g,%g,%g,%g,%g,%g,%g\n", t,qr[0][0]*r_d,
         t,q[0]*r_d,t,qr[0][1]*r_d,t,Xx[1]*r_d);
fprintf(fqq,"%g,%g,%g,%g,%g,%g,%g,%g\n", t,qqr[0][0]*r_d,
         t,qq[0]*r_d,t,qqr[0][1]*r_d,t,qq[1]*r_d);
fprintf(fu,"%g,%g,%g,%g,%g,%g,%g,%g\n",t,Un[0],t,Un[1],
         t,dU[0][0],t,dU[1][0]);
count = 2;
}
count--;


qr[0][0]   = qr[1][0];    qr[0][1]   = qr[1][1];
qqr[0][0]  = qqr[1][0];   qqr[0][1]  = qqr[1][1];
qqqr[0][0] = qqqr[1][0];  qqqr[0][1] = qqqr[1][1];
```

```
  }
  fclose(fq);
  fclose(fqq);
  fclose(fu);
}
/********************************************************/
```

/* RUNG-KUTTA NUMERICAL INTEGRATION */

```
#define step   Ts/Tint    /* 5 */

RKF5( )
{
   int    i, k;
   float  k1[2*N], k2[2*N], k3[2*N], k4[2*N], k5[2*N], k6[2*N],
          F[2*N], Xn[2*N], tn;

   for(k = 0 ; k < step ; k++)  {
    Model(Xx, F);
    for(i = 0 ; i < 2*N ; i++)  {
       k1[i] = F[i];
       Xn[i] = Xx[i] + Tint*0.25*k1[i];
    }
    tn = t + Tint/4;
    Model(Xn, F);
    for(i = 0 ; i < 2*N ; i++)  {
       k2[i] = F[i];
       Xn[i] = Xx[i] + Tint*(3*k1[i] + 9*k2[i])/32;
    }
    tn = t + 3*Tint/8;
    Model(Xn, F);
```

```
    for(i = 0 ; i < 2*N ; i++)  {
        k3[i] = F[i];
        Xn[i] = Xx[i] + Tint*(1932*k1[i] - 7200*k2[i] + 845*k3[i])/2197;
    }
    tn = t + 12*Tint/13;
    Model(Xn, F);
    for(i = 0 ; i < 2*N ; i++)  {
        k4[i] = F[i];
        Xn[i] = Xx[i] + Tint*(439*k1[i]/216 - 8*k2[i] + 3680*k3[i]/513
               - 845*k4[i]/4104);
    }
    tn = t + Tint;
    Model(Xn, F);
    for(i = 0 ; i < 2*N ; i++)  {
        k5[i] = F[i];
        Xn[i] = Xx[i] + Tint*(-8*k1[i]/27 + 2*k2[i] - 3544*k3[i]/2565
               + 1859*k4[i]/4104 - 11*k5[i]/40);
    }
    tn = t + Tint/2;
    Model(Xn, F);
    for(i = 0 ; i < 2*N ; i++)  {
        k6[i] = F[i];
        Xx[i] += Tint*(16*k1[i]/135 + 6656*k3[i]/12825 + 28561*k4[i]/56430
               - 9*k5[i]/50 + 2*k6[i]/55);
    }
  t += Tint;
  }
  Model(Xx, F);
}
/*******************************************************************/
```

/* ROBOT MODEL */

```
#define l0  0.5     /* length of link 0 */
#define l1  0.5     /* length of link 1 */
#define m0  1.5     /* mass of link 0 */
#define m1  1.5     /* mass of link 1 */
#define g   9.81    /* gravity */

Model(Xt,F)
   float  Xt[ ], F[ ];
{
   int  i, j;
   float  D[N][N], Di[N][N], C[N][N], G[N][N], B[N],
          Temp[2], delta, m11, c, g0, g1;

   if(t>=0.5)  m11 = m1 + 0.5;
   else m11=m1;

  /* D(q)*qqq + C(q,qq)*qq + G(q) + B(qq) = Torque */
   q[0] = Xt[0]; q[1] = Xt[1]; qq[0] = Xt[2]; qq[1] = Xt[3];

   /* matrix D(q) */
   D[0][0] = (m0+m11)*l0*l0 + m11*l1*l1 + 2*m11*l0*l1*cos(q[1]);
   D[0][1] = m11*l1*l1 + m11*l0*l1*cos(q[1]);
   D[1][0] = D[0][1];
   D[1][1] = m11*l1*l1;
   c   = -m11*l0*l1*sin(q[1]);
   g0  = (m0+m11)*g*l0;
   g1  = m11*g*l1;

   /* matrix C(q,qq) */
```

```
    C[0][0] = c*qq[1];
    C[0][1] = c*qq[0] + c*qq[1];
    C[1][0] = C[0][1];
    C[1][1] = c*qq[0];

    /* matrix G(q) */
    G[0][0] = g0 + g1*cos(q[1]);
    G[0][1] = g1*sin(q[1]);
    G[1][0] = g1*cos(q[0]);
    G[1][1] = g1*sin(q[0]);

    /* viscous friction vector B(qq) */
    B[0] = 20*qq[0];
    B[1] = 20*qq[1];

    /* find D(q)*qqq = Torque - B(qq) - C(q,qq)*qq - G(q) */
    Temp[0] = U[0]  - C[0][0]*qq[0] - C[0][1]*qq[1]
            - G[0][0]*sin(q[0]) - G[0][1]*cos(q[0]) - B[0];
    Temp[1] = U[1]  - C[1][0]*qq[0] - C[1][1]*qq[1]
            - G[1][0]*sin(q[1]) - G[1][1]*cos(q[1]) - B[1];

    delta =  D[0][0]*D[1][1] - D[0][1]*D[1][0];

    /* state equations */
    F[0] = Xt[2];
    F[1] = Xt[3];
    F[2] = (D[1][1]*Temp[0] - D[0][1]*Temp[1])/delta;
    F[3] = (D[0][0]*Temp[1] - D[1][0]*Temp[0])/delta;
    qqq[0] = F[2];   qqq[1] = F[3];
}
```

/* TRAJECTORY GENERATOR*/

```
Traj(t)
   float  t;
{
   float  w, Ttraj;
   Ttraj =  tf;

   w = 2*Pi/Ttraj;

   if(t<=tf)  {
   qr[1][0] = qi[0] + ( qf[0]-qi[0] )/(2*Pi) * ( w*t - sin(w*t) );
   qr[1][1] = qi[1] + ( qf[1]-qi[1] )/(2*Pi) * ( w*t - sin(w*t) );

   qqr[1][0] = ( qf[0]-qi[0] )/Ttraj * ( 1 - cos(w*t) );
   qqr[1][1] = ( qf[1]-qi[1] )/Ttraj * ( 1 - cos(w*t) );

   qqqr[1][0] = 2*Pi*( qf[0]-qi[0] )/Ttraj/Ttraj * sin(w*t);
   qqqr[1][1] = 2*Pi*( qf[1]-qi[1] )/Ttraj/Ttraj * sin(w*t);
   }
   else  {
     qqr[1][0] = 0; qqr[1][1] = 0;
     qqqr[1][0] = 0; qqqr[1][1] = 0;
   }
}
/***********************************************************/
```

/* RECURSIVE LEAST-SQUARES ESTIMATION */

```
RLS2( )
{
   int  i, j, k;
   float  temp;

   for(i = 0 ; i < N ; i++)  {

    /**************** find  Mi(k) ****************/
    temp = 0.0;
    for(j = 0 ; j < L ; j++)  {
       Vtemp20[j] = 0.0;
       for(k = 0 ; k < L ; k++)
        Vtemp20[j] += P2[i][j][k] * Alpha[i][k];
      temp += Alpha[i][j] * Vtemp20[j];
    }
    temp = 1.0/(temp + gamma2);
    for(j = 0 ; j < L ; j++)
       Vtemp21[j] = Vtemp20[j] * temp;

    /**************** find  Pi(k) ****************/
    for(j = 0 ; j < L ; j++)  {
       for(k = 0 ; k < L ; k++)
        P2[i][j][k] = (P2[i][j][k] - Vtemp20[j] * Vtemp21[k])/gamma2;
    }
       /************** find  Thetai(k) **************/
    temp = 0.0;      for(j = 0 ; j < L ; j++)
       temp += Alpha[i][j] * Theta[i][j];
    temp = Phi[i] - temp;
    for(j = 0 ; j < L ; j++)
```

```
          Theta[i][j] += Vtemp21[j] * temp;
      }
  }
/*****************************************************************/

RLS1( )
{
    int  i, j, k;
    float  temp;

    /**************** find  M(k) ****************/
    temp = 0.0;
    for(j = 0 ; j < L1 ; j++)  {
      Vtemp01[j] = P11[j][0] * Z[0];
      for(k = 1 ; k < L1 ; k++)   Vtemp01[j] += P11[j][k] * Z[k];
      temp += Z[j] * Vtemp01[j];
    }
    temp += gamma1;
    for(j = 0 ; j < L1 ; j++)
    Vtemp11[j] = Vtemp01[j] / temp;

    /**************** find  P(k) ****************/
    for(j = 0 ; j < L1 ; j++)  {
      for(k = 0 ; k < L1 ; k++)
          P11[j][k] = (P11[j][k] - Vtemp01[j] * Vtemp11[k])/gamma1;
    }

   /************** find  A(k) **************/
    for(i = 0 ; i < N ; i++)  {
      temp = 0.0;
      for(j = 0 ; j < L1 ; j++)     temp += Z[j] * A1[i][j];
```

```
        temp = Phi1[i] - temp;
        for(j = 0 ; j < L1 ; j++)
            A1[i][j] += Vtemp11[j] * temp;
    }
}

/*********************************************************/
```

```
/* ONE-STEP OPTIMAL CONTROLLER */

#define  n   2*N   /* colA */
#define  p   N     /* colB */
#define  L2  n+p

float  dU[p], A[n][n], B[n][p], R[p][p], Q[n][n],
                X[2][n], Mpp[p][p], Mpn0[p][n], Mpn1[p][n], S[p][p];
float  qqqr[2][N], qqr[2][N], qr[2][N], Phi[n], Alpha[L2],
                Theta[n][L2], P1[L2][L2], Vtemp0[L2], Vtemp1[L2];

/*** SECONDARY CONTROLLER ***/
  X[1][0] = qr[1][0] - q[0];    X[1][1] = qr[1][1] - q[1];
  X[1][2] = qqr[1][0] - qq[0];  X[1][3] = qqr[1][1] - qq[1];

  /* parameter estimation */
  Alpha[0] = X[0][0];      Alpha[1] = X[0][1];
  Alpha[2] = X[0][2];      Alpha[3] = X[0][3];
  Alpha[4] = dU[0];        Alpha[5] = dU[1];
  Phi[0]   = X[1][0];      Phi[1]   = X[1][1];
  Phi[2]   = X[1][2];      Phi[3]   = X[1][3];

  RLS12();

  for(i=0 ; i<n ; i++)  {
    for(j=0 ; j<n ; j++)   A[i][j] = Theta[i][j];
    for(j=0 ; j<n ; j++)   B[i][j] = Theta[i][j+n];
  }

  /* Bt*Q */
  for(i=0 ; i<p ; i++)  {
```

```
  for(j=0 ; j<n ; j++)  {
      Mpn0[i][j] = B[0][i]*Q[0][j];
      for(k=1 ; k<n ; k++)   Mpn0[i][j] += B[k][i]*Q[k][j];
  }
}

/* Bt*Q*B */
for(i=0 ; i<p ; i++)  {
  for(j=0 ; j<p ; j++)  {
      Mpp[i][j] = Mpn0[i][0]*B[0][j];
      for(k=1 ; k<n ; k++)   Mpp[i][j] += Mpn0[i][k]*B[k][j];
  }
}

/* Bt*Q*A */
for(i=0 ; i<p ; i++)  {
  for(j=0 ; j<n ; j++)  {
      Mpn1[i][j] = Mpn0[i][0]*A[0][j];;
      for(k=1 ; k<n ; k++)   Mpn1[i][j] += Mpn0[i][k]*A[k][j];
  }
}

/* R + Bt*Q*B */
for(i=0 ; i<p ; i++)  {
  for(j=0 ; j<p ; j++)   Mpp[i][j] += R[i][j];
}

/* [R + Bt*Q*B]invert */

Matinv(Mpp);  /* matrix inversion */
```

```
/* feedback gain K = [R + Bt*Q*B]invert * Bt*Q*A */
for(i=0 ; i<p ; i++)  {
  for(j=0 ; j<n ; j++)  {
      Mpn0[i][j] = Mpp[i][0]*Mpn1[0][j];
      for(k=1 ; k<p ; k++)   Mpn0[i][j] += Mpp[i][k]*Mpn1[k][j];
  }
}

/* control U = -KX */
for(i=0 ; i<p ; i++)  {
  dU[i] = -Mpn0[i][0]*X[0][0];
  for(j=1 ; j<n ; j++)    dU[i] -= Mpn0[i][j]*X[0][j];
}
/****************************************************************/
/* joint input torques = nominal torques + perturbation torques */
for(i = 0 ; i < N ; i++)  {
  Un[i] = A1[i][0] * Z[0];
  for(j = 1 ; j < 4*N ; j++)  Un[i] += A1[i][j] * Z[j];
}
```

ประวัติผู้เขียน

นาย ศิริพล สุวรรณโรจน์ เกิดวันที่ 2 สิงหาคม พ.ศ. 2509 ที่กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ในปีการศึกษา 2531 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2532