รายการอ้างอิง

1. Saeed B. Niku. **Introduction to Robotics Analysis, Systems, Applications.** Prentice Hall , 2001.
2. *Available from:* **http://sirius.cie.cau.ac.kr/class/robotics/notes/robotics-01.pdf**
3. Craig, J. J.; Raibert, M.H. A Systematic Method of Hybrid Position/Force Control of a Manipulator. **IEEE Transactions on Robotics and Automation.** 1979.
4. Hogan, N. Stable Execution of Contact Tasks Using Impedance Control. **Proceeding of IEEE International Conference on Robotics and Automation.** 1987.
5. Youcef-Toumi, K.; Li, D. Force Control of Direct-drive Manipulators for Surface Following. **IEEE Transactions on Robotics and Automation.** 1987.
6. Anderson, R. J.; Spong, M. W. Hybrid Impedance Control of Robotic Manipulators. **IEEE Transactions on Robotics and Automation.** (October 1988).
7. Spong, M. W. On the Force Control Problem for Flexible Joint Manipulators. **IEEE Transactions on Automatic Control.** Vol.34, No.1 (January 1989).
8. Karunkar, B. S.; Goldenberg, A. A. Contact Stability in Model-based Force Control Systems of Robot Manipulators. **IEEE Transactions on Robotics and Automation.** 1989.
9. Alici, G.; Daniel, R.W. Development and Experimental Verification of a Mathematical Model for Robot Force Control Design. **Proceeding of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems.** Yokohama, Japan, July 1993
10. Song, P.; Goldenberg A. Fundamental Principles of Design of Position and Force Controller for Robot Manipulators. **Proceeding of IEEE International Conference on Robotics and Automation.** Minneapolis, Minnesota, April 1996
11. Maase, R.; Zahn, V., Dapper, M.; Eckmiller, R. Hard Contact Surface Tracking for Industrial Manipulators with (SR) Position Based Force Control. **Proceeding of IEEE International Conference on Robotics and Automation.** Detroit, Michigan, May 1999

12. Kroger, K.; Finkemeyer, B.; Heuck, M.; Wahl, M. F. Adaptive Implicit Hybrid Force/Pose Control of Industrial Manipulators : Compliant Motion Experiments. **Proceeding of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems.** Sendal, Japan, September 2004

13. Tangpornprasert, P. **Hybrid Force-Position Control of a Robot Manipulator Arm.** Master's Thesis, Mechanical Engineering Department, Graduate School, Chulalongkorn University, 1996.

14. Panyavoravajn, B. **Adaptive Implicit Control.** Master's Thesis, Mechanical Engineering Department, Graduate School, Chulalongkorn University, 2000.

15. Seraji, H.; Bon, B. Real-time collision Avoidance for Position-Controlled Manipulators. **IEEE Transactions on Robotics and Automation.** (August 1999).

16. Inoue, Y.; Kitamura, S.; Kidawara Y. Force Feedback Control and Collision Avoidance of Redundant Manipulator. **Proceeding of the 1991 IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS91).**Osaka, Japan, November 1991

17. Seraji, H.; Colbaugh, R. Force Tracking in Impedance Control. **Proceeding of the 1993 IEEE International Conference on Robotics and Automation.** Vol.22, pp.499-506, 1993

18. Nemec, B.; Zlajpah, L. Implementation of Force Control on Redundant Robot. **Preceeding of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems.** Victoria, Canada, October 1998

19. John J. Craig. **Introduction to Robotics Machanics and Control.** Addision-Wesley Publishing Company,1989.

20. ผศ.ดร.รัชทิน จันทร์เจริญ เทคนิคการขับควบคุม End Effector ของหุ่นยนต์ด้วยเซนเซอร์ (A Technique for Sensor Based Servoing of Robotic End Effector) **บทความการประชุมวิชาการเครือข่ายวิศวกรรมเครื่องกลแห่งประเทศไทยครั้งที่ 17** (15-17 ตุลาคม 2546)

21. Chancharoen R., Sangveraphunsiri V., Sanguanpiyapan K., Chatchaisucha P., Dharachantra P., Nattarom S., and Pongparit S. Collision Avoidance Technique for Uncalibrated Visual Servoing for Industrial Robots. **2002 IEEE/RSJ International Conference on Industrial Robots,** Bangkok, Thailand, December 2002

22. Chatchaisucha P., Dharachantra P., Nattarom S., and Pongparit S. **Collision Avoidance Technique for Uncalibrated Visual Servoing for Industrial Robots.** Senior Project, Mechanical Engineering Department, Chulalongkorn University, 2002.

23. K. S. Fu; R. C. Gonzalez; C. S. G. Lee **Robotics Control, Sensing, Vision and Intelligence,** International Editions, McGraw-Hill Ch.3

ภาคผนวก

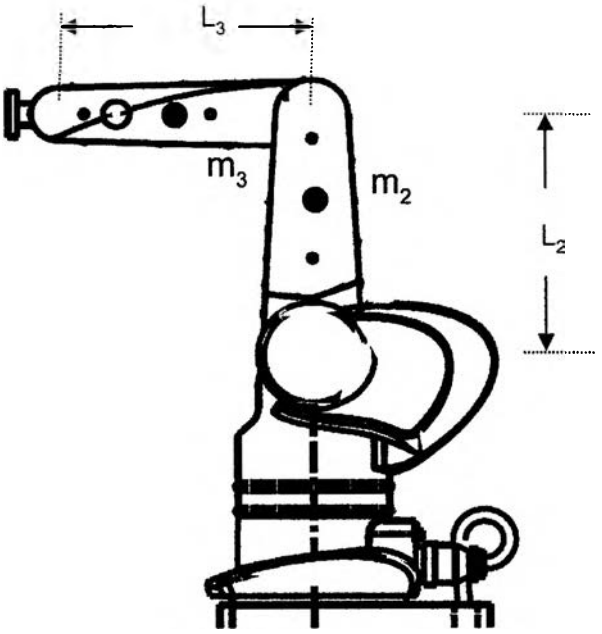
ภาคผนวก ก.

แบบจำลองทางคณิตศาสตร์ของหุ่นยนต์ CRS Robot

ก.1 นำเรื่อง

ในการควบคุมหุ่นยนต์อุตสาหกรรมสามารถนำแบบจำลองทางคณิตศาสตร์มาใช้อธิบายรูปแบบโครงสร้างและการเคลื่อนไหวลักษณะต่างๆ เพื่อนำสู่การหาตำแหน่งของปลายแขนกลในรูปแบบของจลน์ศาสตร์ไปข้างหน้า (Forward Kinematics) การหาตำแหน่งของข้อต่อจาก จลน์ศาสตร์ย้อนกลับ (Inverse Kinematics) หรือหาความเร็วและแรงกระทำที่ข้อต่อ หรือ ปลายแขนกลได้จาก Velocity Jacobian, Force Jacobian และหาพลศาสตร์ไปข้างหน้า/ย้อนกลับ (Forward/Inverse Dynamics) เพื่อใช้เป็นข้อมูลประกอบการควบคุมหุ่นยนต์อุตสาหกรรม โดยกล่าวถึงหุ่นยนต์ Articulated Robot

ก.2 จลน์ศาสตร์ไปข้างหน้าและย้อนกลับ (Forward and Inverse Kinematics)



รูปที่ ก.1 Kinematics Model ของแขนกลแบบ Articulated Robot

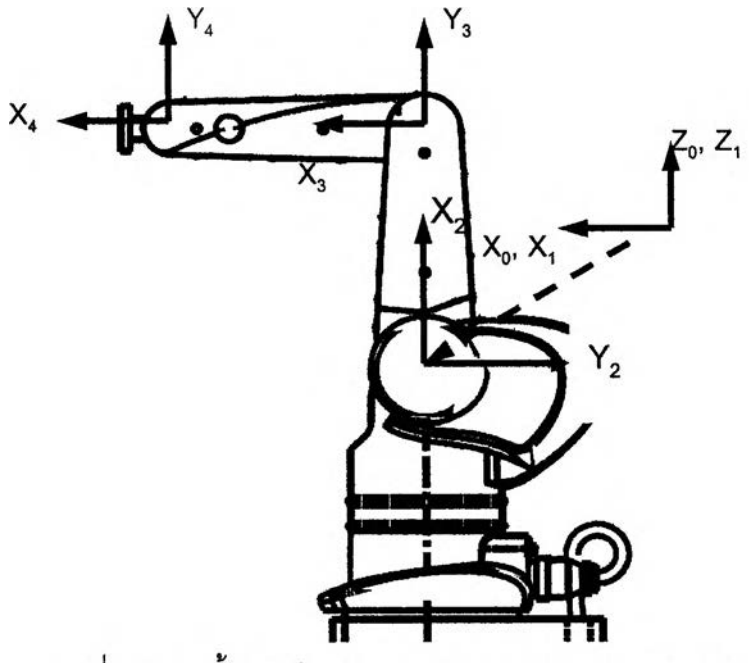
พารามิเตอร์ของหุ่นยนต์ (Robot Parameter) สามารถแสดงดังรูปที่ ก.1

กำหนดให้

ความยาวของ Link	$a_1 = a_2 = a_3 =$ 10 inch.
ระยะทางจุดศูนย์กลางมวลจากจุดหมุน	$l_1 = l_2 = l_3 =$ 5 inch.
มวลของLink	$m_{l1} = m_{l2} = m_{l3}$ 1 kg.
Inertia ของ Motor และ Link	$J_{ml1} = J_{ml2} = 200\text{x}10^{-6}$ kg.m.^{2}

ดี-เอชพารามิเตอร์ (D-H parameter)

วิธี D-H parameter ช่วยในการหาจลน์ศาสตร์ไปข้างหน้า (Forward Kinematics) โดยจะตั้งเฟรมอ้างอิงที่แต่ละข้อต่อแล้วหาค่าความสัมพันธ์ระหว่างแขนกลและข้อต่อ ดังนี้



รูปที่ ก.2 การตั้งแกนอ้างอิงของแขนกลแบบ Articulated Robot

ตารางที่ ก.1 D-H parameter ของ Articulated Robot

Link No.(i)	Link Twist (α_{i-1})	Link Length(a_{i-1})	Link Offset (d_i)	Joint Angle (θ_i)
1	0	0	0	θ_1
2	-90	0	0	θ_2
3	0	L_2	0	θ_3
4	0	L_3	0	θ_4

- Link Twist (α_{i-1})
- Link Length (a_{i-1})
- Link offset (d_i)
- Joint Angle (θ_i)

โดยที่ α_{i-1} , a_{i-1} เป็นค่าคงที่

θ_i , d_i เป็นตัวแปรตามขึ้นอยู่กับการเคลื่อนที่ว่าเป็น Prismatic หรือ Revolute Joint

โดยวิธีของ Denavit-Hartenberg ซึ่งในปี ค.ศ.1955 R. S. Hartenberg และ J. Denavit [19]ได้เสนอ **วิธี** D-H parameter จากรูปทั่วไปของ Homogeneous Transformation Matrix ดังนี้

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i \cos\alpha_{i-1} & \cos\theta_i \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i \sin\alpha_{i-1} \\ \sin\theta_i \sin\alpha_{i-1} & \cos\theta_i \sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i \cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

ก.2.1 จลน์ศาสตร์ไปข้างหน้า (Forward Kinematics)

เป็นการโอนย้ายความสัมพันธ์ของตำแหน่งแขนกลจากการอ้างอิงเชิงมุมของแต่ละ Joint ใน Joint Space มาเป็นการอ้างอิงเชิงเส้น 3 มิติแกน X ,Y, Z ใน Cartesian Space เพื่อบอกตำแหน่งปลายแขนกลเทียบกับเฟรมอ้างอิงเริ่มต้น (Base Frame)

Joint 1 จะได้ว่า Homogeneous Transformation Matrix

$$ {}^{0}_{1}T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Joint 2 จะได้ว่า Homogeneous Transformation Matrix

$$ {}^{1}_{2}T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin\theta_2 & -\cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Joint 3 จะได้ว่า Homogeneous Transformation Matrix ที่

$$ {}^{2}_{3}T = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Joint 4 จะได้ว่า Homogeneous Transformation Matrix

$$ {}^{3}_{4}T = \begin{bmatrix} 1 & 0 & 0 & L3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$\therefore {}^{0}_{4}T = \begin{bmatrix} \cos\theta_1\cos(\theta_2+\theta_3) & -\cos\theta_1\sin(\theta_2+\theta_3) & -\sin\theta_1 & T14 \\ \sin\theta_1\cos(\theta_2+\theta_3) & -\sin\theta_1\sin(\theta_2+\theta_3) & \cos\theta_1 & T24 \\ -\sin(\theta_2+\theta_3) & -\cos(\theta_2+\theta_3) & 0 & T34 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

โดยที่

$$L_2 = L_3 = l$$

$$T14 = (l\cos\theta_2 + l\cos(\theta_2+\theta_3))\cos\theta_1$$

$$\mathrm{T24} = (l\cos\theta_2 + l\cos(\theta_2+\theta_3))\sin\theta_1$$

$$\mathrm{T34} = -l\sin\theta_2 - l\sin(\theta_2+\theta_3)$$

ก.2.2 จลน์ศาสตร์ย้อนกลับ (Inverse Kinematics)

เป็นการโอนย้ายความสัมพันธ์ย้อนกลับของตำแหน่งแขนกล จากการอ้างอิงเชิงเส้น3 มิติ แกน x, y, z ใน Cartesian Space ของปลายแขนกล มาเป็นการอ้างอิงใน Joint Space เพื่อบอกตำแหน่งปลายแขนกลเทียบกับเฟรมอ้างอิงเริ่มต้น (Base Frame) ซึ่งจะพบว่าสามารถหาผลเฉลยได้หลายคำตอบขึ้นอยู่กับเงื่อนไขในการทำงานของแขนกล เช่น สิ่งกีดขวาง

วิธีหาผลเฉลยแบบ Geometric

Joint 1

$$\therefore \theta_1 = A\tan 2(y,x) \quad \text{(ต้องพิจารณาเงื่อนไขประกอบเนื่องจากคำตอบมี 2 ค่า)}$$

โดยที่ $\mathrm{x} = (l\cos\theta_2 + l\cos(\theta_2+\theta_3))\cos\theta_2$

$$\mathrm{y} = (l\cos\theta_2 + l\cos(\theta_2+\theta_3))\sin\theta_1$$

$$\mathrm{z} = l\sin\theta_2 + l\sin(\theta_2+\theta_3)$$

Joint 2

$$L_3^{\,2} = L_2^{\,2} + r^2 - 2L_2 r\cos\beta$$

$$\beta = \cos^{-1}\left[\frac{l^2 - l^2 - r^2}{-2lr}\right]$$

$$\psi = \sin^{-1}(z/r)$$

$$\therefore \theta_2 = \beta + \psi$$

โดยที่ $L_2 = L_3 = l$

Joint 3

$$L_2, L_3 = l$$

$$r^2 = l^2 + l^2 - 2ll\cos\Omega$$

$$\Omega = \cos^{-1}(\frac{l^2 + l^2 + r^2}{2ll})$$

$$\therefore \theta_3 = 180 - \cos^{-1}(\frac{l^2 + l^2 - r^2}{2ll})$$

ก.3 จาโคเบียน (Jacobian)

การที่หุ่นยนต์มีการเคลื่อนที่ปฏิสัมพันธ์กับสิ่งแวดล้อม คุณลักษณะต่างๆจะเปลี่ยนแปลงอย่างต่อเนื่องตลอดการเคลื่อนไหว ทำให้เกิดการเปลี่ยนแปลงของระยะทาง แรงในแนวเชิงเส้น เชิงมุมเมื่อเทียบกับแกนอ้างอิง จึงได้ทำการหาวิธีการโอนย้ายความสัมพันธ์ความเร็ว แรงของตำแหน่งต่างๆบนแขนกล เช่น ความเร็วเชิงมุม (ในกรณีที่เป็น Revolute Joint) ความเร็วเชิงเส้น (ในกรณีที่เป็น Prismatic Joint หรือปลายแขน)

ก.3.1 จาโคเบียนความเร็ว (Velocity Jacobian)

โอนย้ายความสัมพันธ์ความเร็วของตำแหน่งต่างๆบนแขนกล เช่น ความเร็วเชิงมุม (ในกรณีที่เป็น Revolute Joint) หรือความเร็วเชิงเส้น (ในกรณีที่เป็น Prismatic Joint หรือปลายแขน) ระหว่างการอ้างอิงที่ฐานของหุ่นยนต์และการอ้างอิงทีปลายแขนกล

Joint 1 :

$${}^1\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}$$

$${}^1v_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Joint 2 :

$${}^2\omega_2 = {}^2_1R({}^1\omega_1) + \dot{\theta}_2$$

$${}^2\omega_2 = \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ -\sin\theta_2 & 0 & -\cos\theta_2 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} -(\sin\theta_2)\dot{\theta}_1 \\ -(\cos\theta_2)\dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$${}^2v_2 = {}^2_1R({}^1v_1 + {}^1\omega_1 \times {}^1P_1)$$

$$= {}^{2}_{1}R(0 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix})$$

$$\therefore {}^{2}v_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Joint 3 : $${}^{3}\omega_3 = {}^{3}_{2}R\,{}^{2}\omega_2 + \dot{\theta}_3$$

$$= \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -(\sin\theta_2)\dot{\theta}_1 \\ -(\cos\theta_2)\dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix}$$

$$= \begin{bmatrix} -(\sin(\theta_2 + \theta_3))\dot{\theta}_1 \\ -(\cos(\theta_2 + \theta_3))\dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix}$$

$${}^{3}v_3 = {}^{3}_{2}R({}^{2}v_2 + {}^{2}\omega_2 \times {}^{2}P_3)$$

$$= {}^{3}_{2}R(0 + \begin{bmatrix} -(\sin\theta_2)\dot{\theta}_1 \\ -(\cos\theta_2)\dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \times \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix}); l = l$$

$$\therefore {}^{3}v_3 = \begin{bmatrix} l(\sin\theta_3)\dot{\theta}_2 \\ l(\cos\theta_3)\dot{\theta}_2 \\ l(\cos\theta_2)\dot{\theta}_1 \end{bmatrix}$$

Joint 4 : $${}^{4}\omega_4 = {}^{4}_{3}R\,{}^{3}\omega_3 + \dot{\theta}_4$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -(\sin(\theta_2 + \theta_3))\dot{\theta}_1 \\ -(\cos(\theta_2 + \theta_3))\dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -(\sin(\theta_2 + \theta_3))\dot{\theta}_1 \\ -(\cos(\theta_2 + \theta_3))\dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4 \end{bmatrix}$$

$${}^{4}v_4 = {}^{4}_{3}R({}^{3}v_3 + {}^{3}\omega_3 \times {}^{3}P_4)$$

$$= {}^{4}_{3}R({}^{3}v_3 + \begin{bmatrix} -(\sin(\theta_2 + \theta_3))\dot{\theta}_1 \\ -(\cos(\theta_2 + \theta_3))\dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix} \times \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix}); l = l$$

$$\therefore {}^{4}v_4 = \begin{bmatrix} l(\sin\theta_3)\dot{\theta}_2 \\ (l\cos\theta_3 + l)\dot{\theta}_2 + l\dot{\theta}_3 \\ (\cos\theta_2 + \cos(\theta_2 + \theta_3))l\dot{\theta}_1 \end{bmatrix}$$

ทำการโอนย้ายจากเฟรมที่สี่ไปยังเฟรมศูนย์

$${}^{0}v_4 = {}^{0}_{4}R\,{}^{4}v_4$$

$$ {}^0v_4 = \begin{bmatrix} v_{11} \\ v_{21} \\ v_{31} \end{bmatrix} $$

โดยที่

$$ v_{11} = l\sin\theta_3\cos\theta_1(\cos(\theta_2+\theta_3))\dot{\theta}_2 - (l\cos\theta_3 + l)\cos\theta_1(\sin(\theta_2+\theta_3))\dot{\theta}_2 - l\cos\theta_1(\sin(\theta_2+\theta_3))\dot{\theta}_3 $$

$$ = -(\cos\theta_2 + \cos(\theta_2+\theta_3))\sin\theta_1 l\dot{\theta}_1 - (\sin\theta_2 + \sin(\theta_2+\theta_3))\cos\theta_1 l\dot{\theta}_2 - \cos\theta_1(\sin(\theta_2+\theta_3))l\dot{\theta}_3 $$

$$ v_{21} = (\cos(\theta_2+\theta_3) + \cos\theta_2)\cos\theta_1 l\dot{\theta}_1 - \sin\theta_1\sin\theta_2 l\dot{\theta}_2 - \sin\theta_1(\sin(\theta_2+\theta_3))l\dot{\theta}_2 - l\sin\theta_1(\sin(\theta_2+\theta_3))\dot{\theta}_3 $$

$$ = (\cos(\theta_2+\theta_3) + \cos\theta_2)\cos\theta_1 l\dot{\theta}_1 - (\sin\theta_2 + (\sin(\theta_2+\theta_3))\sin\theta_1 l\dot{\theta}_2 - \sin\theta_1(\sin(\theta_2+\theta_3))l\dot{\theta}_3 $$

$$ v_{31} = -l\sin\theta_3(\sin(\theta_2+\theta_3))\dot{\theta}_2 - (\cos\theta_3 + 1)l(\cos(\theta_2+\theta_3))\dot{\theta}_2 - (\cos(\theta_2+\theta_3))l\dot{\theta}_3 $$

$$ = -(\cos\theta_2 + \cos(\theta_2+\theta_3))l\dot{\theta}_2 - \cos(\theta_2+\theta_3)l\dot{\theta}_3 $$

จะได้ว่า

$$ {}^0v_4 = {}^0J(\theta)\dot{\theta} $$

$$ {}^0v_4 = \begin{bmatrix} -(\cos\theta_2 + \cos(\theta_2+\theta_3))\sin\theta_1 l & -(\sin\theta_2 + \sin(\theta_2+\theta_3))\cos\theta_1 l & -\cos\theta_1(\sin(\theta_2+\theta_3))l \\ (\cos(\theta_2+\theta_3) + \cos\theta_2)\cos\theta_1 l & -(\sin\theta_2 + (\sin(\theta_2+\theta_3))\sin\theta_1 l & -\sin\theta_1(\sin(\theta_2+\theta_3))l \\ 0 & -(\cos\theta_2 + \cos(\theta_2+\theta_3))l & -\cos(\theta_2+\theta_3)l \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} $$

$$ \therefore {}^0J(\theta) = \begin{bmatrix} -(\cos\theta_2 + \cos(\theta_2+\theta_3))\sin\theta_1 l & -(\sin\theta_2 + \sin(\theta_2+\theta_3))\cos\theta_1 l & -\cos\theta_1(\sin(\theta_2+\theta_3))l \\ (\cos(\theta_2+\theta_3) + \cos\theta_2)\cos\theta_1 l & -(\sin\theta_2 + (\sin(\theta_2+\theta_3))\sin\theta_1 l & -\sin\theta_1(\sin(\theta_2+\theta_3))l \\ 0 & -(\cos\theta_2 + \cos(\theta_2+\theta_3))l & -\cos(\theta_2+\theta_3)l \end{bmatrix} $$

ก.3.2 จาโคเบียนแรง (Force Jacobian)

โอนย้ายความสัมพันธ์แรงของตำแหน่งต่างๆบนแขนกล เช่น แรงบิด (ในกรณีที่เป็น Revolute Joint) หรือแรงเชิงเส้น (ในกรณีที่เป็น Prismatic Joint หรือปลายแขน) ระหว่างการอ้างอิงที่ฐานของหุ่นยนต์และการอ้างอิงที่ปลายแขนกล

Joint 4 : $$ {}^4f_4 = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} $$

Joint 3 : $$ {}^3f_3 = {}^3_4R\,{}^4f_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} $$

$$^3n_3 = L_3\hat{x} \times \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 \\ -L_3 f_z \\ L_2 f_y \end{bmatrix}; L_3 = L$$

Joint 2 :

$$^2f_2 = {}^2_3R\,{}^3f_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$$

$$\therefore {}^2f_2 = \begin{bmatrix} (\cos\theta_3)f_x - (\sin\theta_3)f_y \\ (\sin\theta_3)f_x + (\cos\theta_3)f_y \\ f_z \end{bmatrix}$$

$$^2n_2 = {}^2_3R\,{}^3n_3 + {}^2P_3 \times {}^2f_2$$

$$= \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -Lf_z \\ Lf_y \end{bmatrix} + \begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} (\cos\theta_3)f_x - (\sin\theta_3)f_y \\ (\sin\theta_3)f_x + (\cos\theta_3)f_y \\ f_z \end{bmatrix}; L_2 = L$$

$$= \begin{bmatrix} L(\sin\theta_3)f_z \\ -Lf_z(\cos\theta_3 + 1) \\ L(\sin\theta_3)f_x + (\cos\theta_3 + 1)Lf_y \end{bmatrix}$$

Joint 1 :

$$^1f_1 = {}^1_2R\,{}^2f_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ 0 & 0 & 1 \\ -\sin\theta_2 & -\cos\theta_2 & 0 \end{bmatrix} \begin{bmatrix} (\cos\theta_3)f_x - (\sin\theta_3)f_y \\ (\sin\theta_3)f_x + (\cos\theta_3)f_y \\ f_z \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_2(\cos\theta_3)f_x - \cos\theta_2(\sin\theta_3)f_y - \sin\theta_2(\sin\theta_3)f_x - \sin\theta_2(\cos\theta_3)f_y \\ f_z \\ -\sin\theta_2(\cos\theta_3)f_x + \sin\theta_2(\sin\theta_3)f_y - \cos\theta_2(\sin\theta_3)f_x - \cos\theta_2(\cos\theta_3)f_y \end{bmatrix}$$

$$= \begin{bmatrix} (\cos(\theta_2 + \theta_3))f_x - (\sin(\theta_2 + \theta_3))f_y \\ f_z \\ -(\sin(\theta_2 + \theta_3))f_x - (\cos(\theta_2 + \theta_3))f_y \end{bmatrix}$$

$$^1n_1 = {}^1_2R\,{}^2n_2 + {}^1P_2 \times {}^1f_1$$

$$= \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ 0 & 0 & 1 \\ -\sin\theta_2 & -\cos\theta_2 & 0 \end{bmatrix} \begin{bmatrix} L(\sin\theta_3)f_z \\ -L(\cos\theta_3 + 1)f_z \\ L(\sin\theta_3)f_x + L(\cos\theta_3 + 1)f_y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} L\cos\theta_2(\sin\theta_3)f_z + L(\sin\theta_2)f_z(\cos\theta_3+1) \\ L(\sin\theta_3)f_x + Lf_y(\cos\theta_3+1) \\ L\sin\theta_2(\sin\theta_3)f_z + L(\cos\theta_2)_z(\cos\theta_3+1) \end{bmatrix}$$

$$= \begin{bmatrix} L(\sin(\theta_2+\theta_3))f_z + L(\sin\theta_2)f_z \\ L(\sin\theta_3)f_x + L(\cos\theta_3+1)f_y \\ L(\cos(\theta_2+\theta_3))f_z + L(\cos\theta_2)f_z \end{bmatrix}$$

$$\therefore \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & L(\sin(\theta_2+\theta_3)+\sin\theta_2) \\ L\sin\theta_3 & L(\cos\theta_3+1) & 0 \\ 0 & 0 & L(\cos(\theta_2+\theta_3)+\cos\theta_2) \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$$

ก.4 พลศาสตร์ของหุ่นยนต์ (Robot Dynamics)

สามารถเขียนให้อยู่ในรูปสมการการเคลื่อนที่ (Equation of Motion) ของข้อต่อของแขนกล ดังนี้

$$\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta})\dot{\theta} + G(\theta)$$

โดยที่

$M(\theta)$	คือ	Mass Matrix ของแขนกล
$V(\theta,\dot{\theta})$	คือ	Centifugal และ Coriolis Vector
$G(\theta)$	คือ	Gravity Vector
τ	คือ	แรงบิดที่ข้อต่อของแขนกล
θ	คือ	ตำแหน่งของข้อต่อของแขนกล

พลศาสตร์ของแขนกล เป็นความสัมพันธ์ของแรงบิด (Joint Torque) ที่กระทำที่ข้อต่อ กับการเคลื่อนที่ของแขนกลใน Cartesian Space ซึ่งจะแสดงในรูปแบบของสมการการเคลื่อนที่ของระบบ (Equation of Motion) โดยสามารถหาได้จากวิธี 2 วิธี

1. Newton-Euler's Formulation

เป็นวิธีที่ใช้กฎข้อที่ 2 ของนิวตัน (Newton's Second Law of Motion) ในการหา ซึ่งแสดงถึงพลศาสตร์ของระบบในรูปของแรง (Force) และโมเมนต์ (Moment) เพื่อพยายามหาความสัมพันธ์ของ Joint Torque กับการเคลื่อนที่ของแขนกลในรูปของ Joint Displacement

2. Lagrangian Formulation

เป็นวิธีที่แสดงถึงพลศาสตร์ของระบบในรูปของงาน (Work) และพลังงาน (Energy) และใช้ Generalized Coordinates ซึ่ง Equation of Motion ที่ได้จะอยู่ในรูป Close-Form ที่แสดงความสัมพันธ์ของ Joint Torque และ Joint Displacement

แต่ในที่นี้จะใช้วิธีการของ Newton-Euler's formulation ในการหา Equation of Motion เนื่องจากลักษณะของสมการจะเหมาะสำหรับการเขียนโปรแกรมคอมพิวเตอร์มากกว่า จากนั้นเป็นการแก้ปัญหา Inverse Dynamics เพื่อที่จะหา Input Joint Torque ที่เหมาะสมเพื่อทำให้ Output เป็นไปตามที่ต้องการ

จากนั้นใช้ Newton-Euler Equation คำนวณหาค่า Joint Torque ของแต่ละ Joint โดยเริ่มจาก Link สุดท้ายย้อนกลับไป Link ศูนย์ ซึ่งขั้นตอนการคำนวณนี้จะใช้วิธีของ Luh-Walker-Paul's Algorithm (Recursive Computation of Kinematics and Dynamic Equation)

เนื่องจากเรามี Link อยู่ 2 ชนิด ฉะนั้น ความเร็วและความเร่งของแต่ละ Link ของแขนกลจึงขึ้นอยู่กับชนิดของ Link และในที่นี้จะเขียนความเร็วและความเร่งของ Link ใดๆเทียบกับ Link นั้นๆ ดังนี้ [23]

Forward Equations : i = 1, 2, 3, 4

$${}^{i}R_0\omega_i = {}^{i}R_{i-1}({}^{i-1}R_0\omega_{i-1} + z_0\dot{q}_i)$$ if link i is rotational

$${}^{i}R_0\dot{\omega}_i = {}^{i}R_{i-1}[{}^{i-1}R_0\dot{\omega}_{i-1} + z_0\ddot{q}_i + ({}^{i-1}R_0\omega_{i-1}) \times z_0\dot{q}_i)]$$ if link i is translational

$${}^{i}R_0\dot{v}_1 = [({}^{i}R_0\dot{\omega}_i) \times ({}^{i}R_0p_i^*) + ({}^{i}R_0\omega_i) \times ({}^{i}R_0\omega_i \times {}^{i}R_0p_i^*)] + {}^{i}R_{i-1}({}^{i-1}R_0\dot{v}_{i-1})$$ if link i is rotational

$${}^{i}R_0\,\bar{a}i = ({}^{i}R_0\dot{\omega}_i) \times ({}^{i}R_0\,\bar{s}_i) + ({}^{i}R_0\omega_i) \times [({}^{i}R_0\omega_i) \times ({}^{i}R_0\,\bar{s}_i)] + {}^{i}R_0\dot{v}_i$$

$\bar{a}_i$ = accumulate of CM.

Backward Equations : i = 4, 3, 2, 1

$${}^{i}R_0f_i = {}^{i}R_{i+1}({}^{i+1}R_0f_{i+1}) + m_i\,{}^{i}R_0\,\bar{a}_i$$

$${}^{i}R_0n_i = {}^{i}R_{i+1}[{}^{i+1}R_0n_{i+1} + ({}^{i+1}R_0p_i^*) \times ({}^{i+1}R_0f_{i+1})] + [({}^{i}R_0p_i^* + {}^{i}R_0\,\bar{s}_i) \times ({}^{i}R_0F_i)] + ({}^{i}R_0I_i\,{}^{0}R_i)({}^{i}R_0\dot{\omega}_i) + ({}^{i}R_0\omega_i) \times [({}^{i}R_0I_i\,{}^{0}R_i)({}^{i}R_0\omega_i)]$$

Torque apply for Link i (for Link i is rotational)

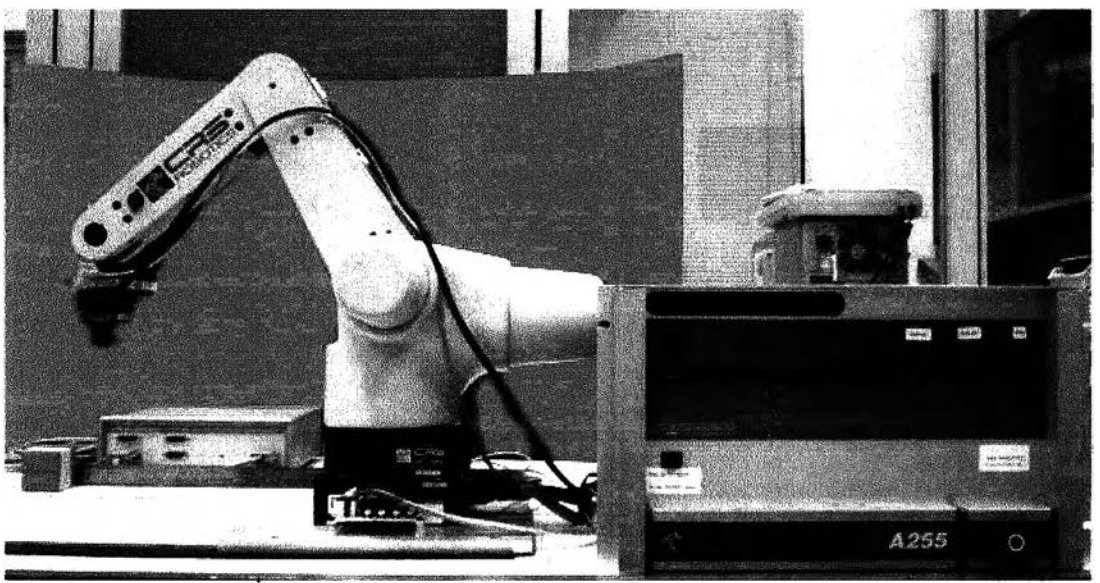
$$\tau_i = ({}^{i}R_0n_i)^T({}^{i}R_{i-1}z_0 + b_iq_i)$$ if link i is rotational

ภาคผนวก ข.

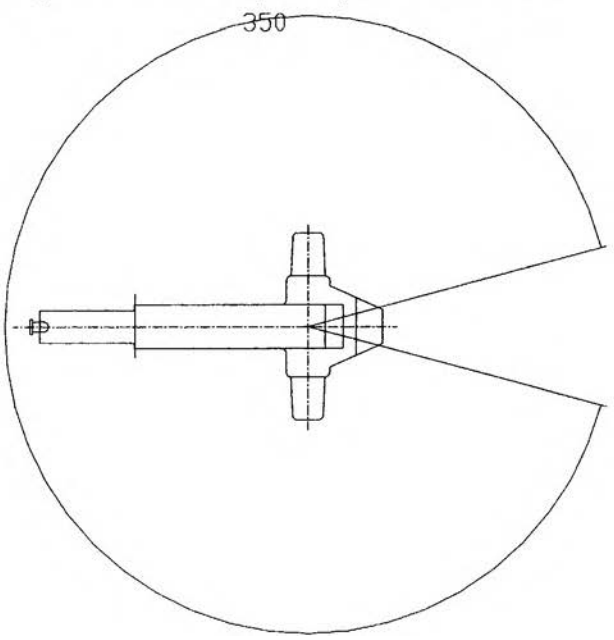
รายละเอียดข้อมูลเทคนิคของชุดทดลองหุ่นยนต์ CRS Robot

ข.1 แขนหุ่นยนต์ CRS Robot

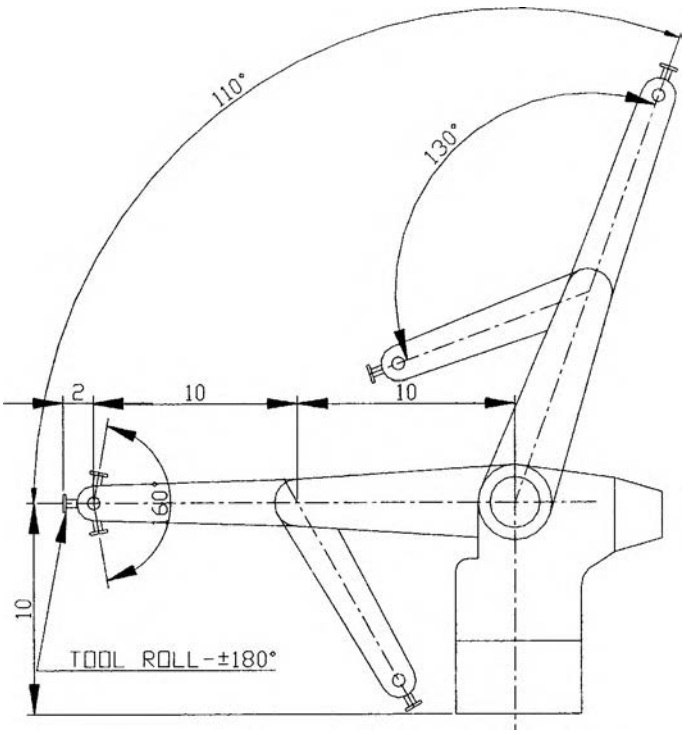
ในการทดลองควบคุมทางเดินของแขนหุ่นยนต์เพื่อหลบหลีกสิ่งกีดขวาง ได้ทำการเลือกแขนหุ่นยนต์ CRS Robot เนื่องจากติดตั้งอุปกรณ์ตรวจวัดทางตำแหน่งที่มีค่าความละเอียดสูง จึงทำให้ผลการทดลองออกมาถูกต้อง โดยชุดทดลองและระบบควบคุมจะมีลักษณะดังรูปที่ ข.1



รูปที่ ข.1 ชุดการทดลองควบคุมแขนหุ่นยนต์ CRS Robot



รูปที่ ข.2 ด้านบนของแขนหุ่นยนต์แบบ Articulated ของบริษัท CRS Robotics รุ่น 255



รูปที่ ข.3 ด้านข้างของแขนหุ่นยนต์แบบ Articulated ของบริษัท CRS Robotics รุ่น 255

ชุดการทดลองเป็นแขนหุ่นยนต์แบบ Articulated ของบริษัท CRS Robotic รุ่น 255 จะประกอบด้วยอุปกรณ์ต่างๆ ดังนี้ :

- แขนหุ่นยนต์แบบ Articulated ของบริษัท CRS Robotics รุ่น 255 : มี 5 แกนหมุน และ 5 องศาอิสระโดยแต่ละแกนจะขับด้วย มอเตอร์กระแสตรงขนาดแรงดัน ±25 V. ส่งผ่านกำลังขับไปสู่แกนด้วยระบบ Harmonic Drive มีค่าความละเอียดในการเคลื่อนที่ 0.005 องศา และมีความเร็วสูงสุด 3.05 rad/s และมีคุณสมบัติอื่นๆ ดังตาราง ข.1

ตารางที่ ข.1 รายละเอียดแขนหุ่นยนต์ CRS Robot

รายการ	รายละเอียด
Structure	**articulated 5 DOF**
Drive motor	**Permanent magnet DC Servo**
Bearings	ABEC Class 1- 0.375" ID
Max voltage	+/- 25 Vdc
Mac Current	10.8 amps
Mech. Time const.	11.62 msec
Max speed @ 25 V	3600 rpm
Peak torque	100 oz-in
Brush life	8000 hours @ 1200 rpm
Transmission	
Waist rotate	Size 20 cup type harmonic drive
Shoulder	Size 20 cup type harmonic drive
Elbow	Size 20 cup type harmonic drive/chain
Wrist bend (pitch)	Bevel-/spur-gear/chain
Tool roll	Bevel-/spur-gear/chain/gear
Payload	**Kg**
Max design	2.0
Full speed/acc	1.0
Reach – Waist to tool flange	**22 inches**
Reach (by link)	**Inches**
Base to shoulder	10
Shoulder to elbow	10
Elbow to wrist pivot	10
Wrist pivot to tool flange	2

Joint travel ranges	Degrees
Waist rotate	+/-175
Shoulder	+110,-0
Elbow	+0,-130
Wrist bend (pitch)	+/-110
Tool roll	+/-180
Joint speeds at 100 % program speed	**Rad/sec**
A150 Series:	
Waist rotate	1.74
Shoulder3	1.08
Elbow	1.74
Wrist bend (pitch)	3.14
Tool roll	6.28
A250 Series:	
Waist rotate	3.05
Shoulder 3	2.18
Elbow	3.05
Wrist bend (pitch)	3.14
Tool roll	6.28
Joint default acceleration rates	**Rad/sec^2**
A150 Series:	
Waist rotate	5.45
Shoulder	5.45
Elbow	5.45
Wrist bend (pitch)	24.54
Tool roll	49.09
A250 Series:	

Waist rotate	12.93
Shoulder 3	12.93
Elbow	12.93
Wrist bend (pitch)	58.18
Tool roll	116.36
Position Feedback	**Optical incremental encoders**
Resolution	1000 pulse/rev
Index	marker pulse 1 per rev
Output	Chnls A,B,Z sq.wave TTL
Joint Resolution	**Deg**
Waist rotate	0.005
Shoulder	0.005
Elbow	0.005
Wrist bend (pitch)	0.023
Tool roll	0.045
Joint Resolution	**Inches @ tool flange**
Waist rotate	0.0019
Shoulder	0.0009
Elbow	0.0009
Wrist bend (pitch)	0.0008
Tool roll	0.0016

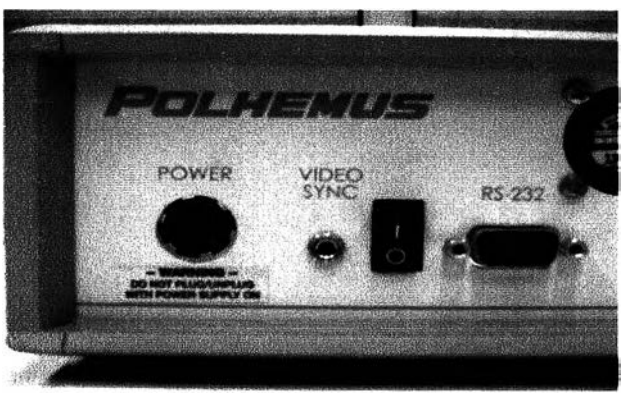
ข.2 ชุดอุปกรณ์ตรวจรู้ตำแหน่งพิกัดใน 3 มิติ (Fastrak®)

ข.2.1 การติดตั้งชุดอุปกรณ์ตรวจรู้ตำแหน่งพิกัดใน 3 มิติ (Fastrak®)

1) ติดตั้งข้อต่อแบบ D (15pin,ตัวผู้)ของตัวส่งสัญญาณ (Transmitter) ไปที่ด้านหน้าของกล่องควบคุม
2) ติดตั้งข้อต่อแบบ D (15 pin,ตัวเมีย)ของตัวรับสัญญาณ (Receiver) ไปที่ด้านหน้าของกล่องควบคุม

รูปที่ ข.4 ข้อต่อและช่องสำหรับต่ออุปกรณ์รับส่งสัญญาณ

3) ตรวจสอบสวิตช์ (Power Switch ON/OFF) อยู่ในตำแหน่ง OFF
4) เชื่อมข้อต่อสำหรับไฟฟ้า (Power, 5 pin) ไปที่ด้านหลังของกล่องควบคุม



รูปที่ ข.5 ตำแหน่งของการต่อสาย Power และสวิตช์เปิดปิด

5) เลือก I/O Select Switch (1-8) โดยที่ตำแหน่งของสวิตช์ที่ 1 ถึง 8

รูปที่ ข.6 ตำแหน่งของสวิตช์เลือกคุณสมบัติของชุดควบคุม

โดยที่

Switch	1	2	3	4	5	6	7	8
Position	ON	ON	ON	OFF	ON	OFF	OFF	ON

Switch 1-3 : Baud Rate

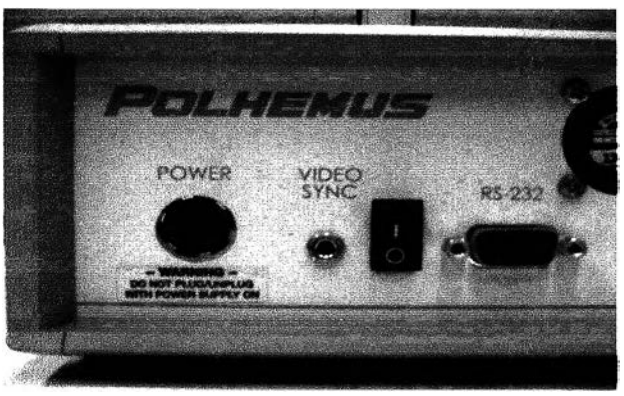
Switch 4 : Hardware Handshake(RTS/CTS)

Switch 5 : Character Width

Switch 6-7 : Parity

Swithc 8 : Serial Operation

6) เชื่อมต่อสาย RS-232 แบบ D Female ไปที่ช่อง I/O ด้านหลังของกล่องควบคุม และเชื่อมต่อสาย RS-232 แบบ D Male ไปที่ช่อง Com1 ของ PC หลัก



รูปที่ ข.7 ตำแหน่งของช่องต่อสัญญาณแบบ RS-232

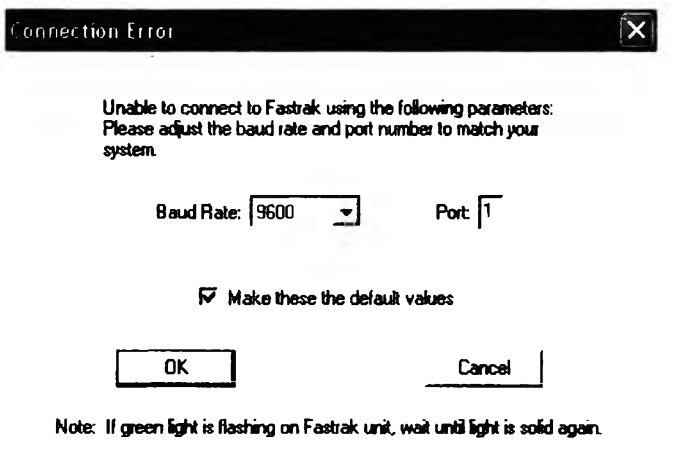
7) เปิดสวิตช์ไฟฟ้าเข้ากล่องควบคุม (จะมีไฟสีเขียวกระพริบประมาณ 10 วินาที)

ข.2.2 การใช้งานโปรแกรม FTGui ของชุดอุปกรณ์ตรวจรู้ตำแหน่งพิกัดใน 3 มิติ (Fastrak®)

การเชื่อมต่อ Software FTGui กับ ชุดควบคุม Fastrak®

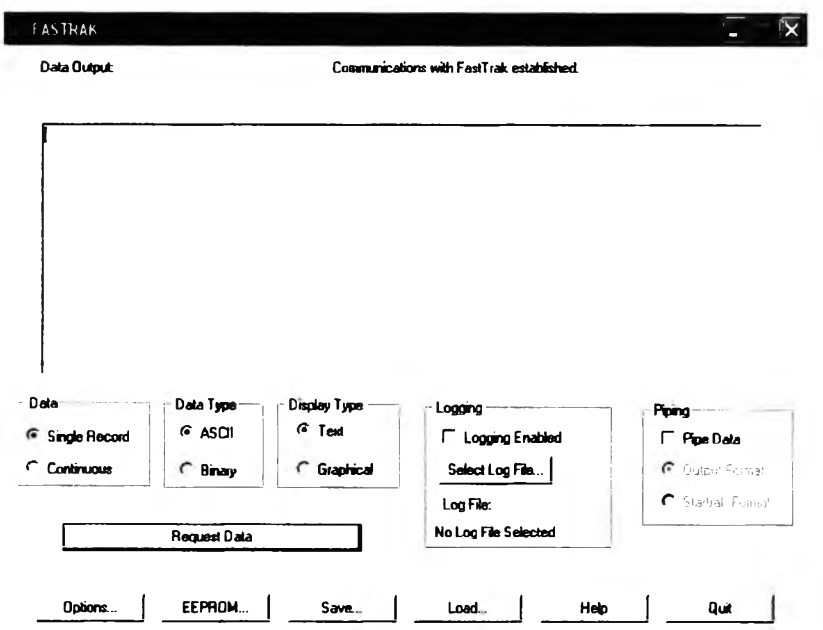
1) เลือก Baud Rate : 9600

2) เลือก Port 1



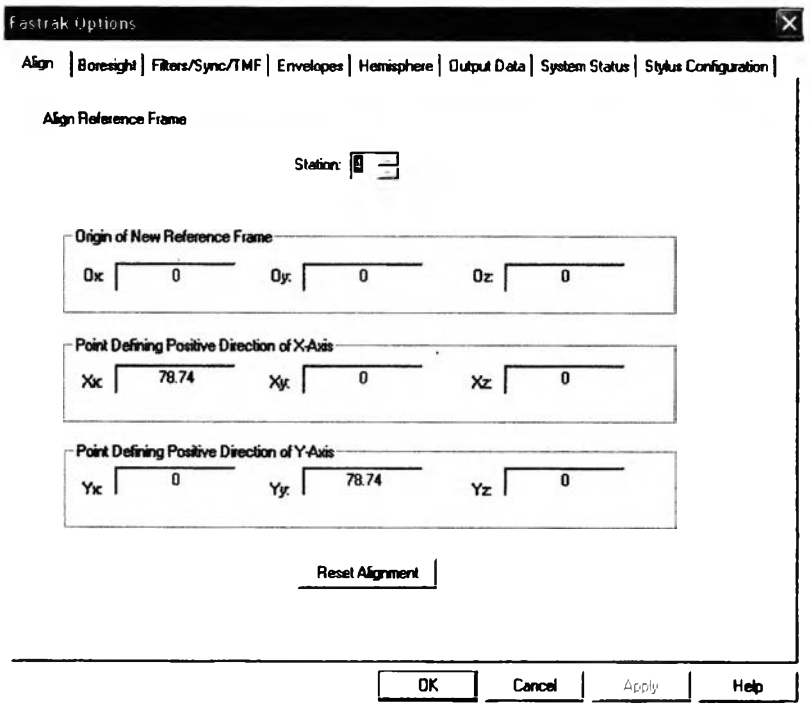
รูปที่ ข.8 การเลือก Baud Rate ของชุดควบคุม Fastrak® ที่ใช้งาน

3) เลือกชนิดของข้อมูลที่ต้องการรับเป็น ASCII Type

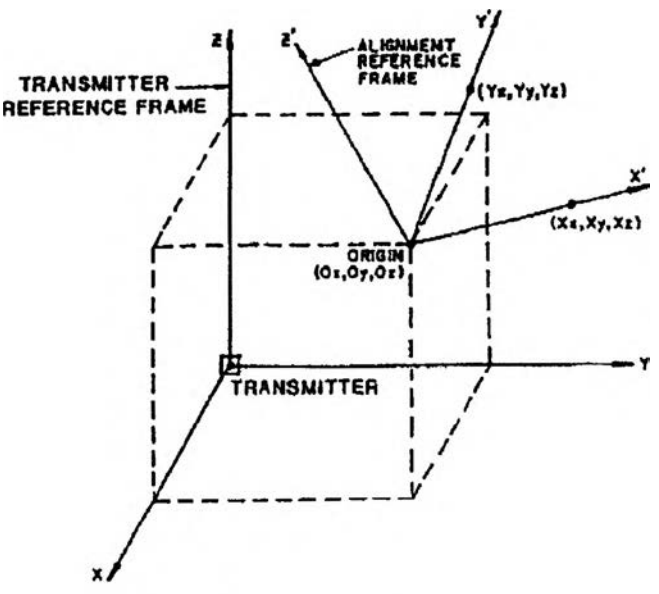


รูปที่ ข.9 ประเภทของข้อมูลที่ต้องการรับค่าจาก Fastrak®

4) เลือก Options -> Align เพื่อเลือก Reference Frame ของ Transmitter

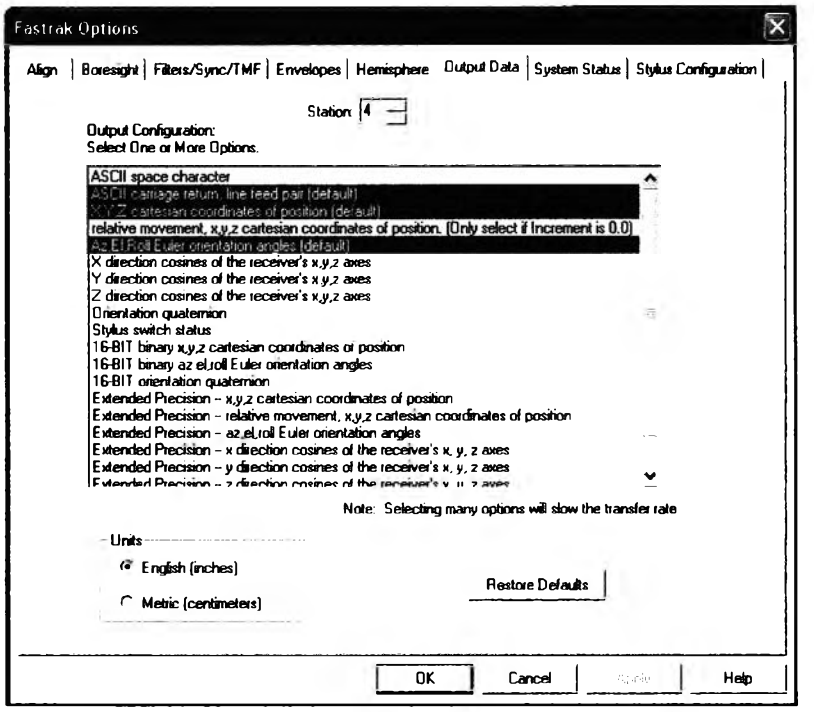


รูปที่ ข.10 การ Alignment ของตัวส่งสัญญาณ



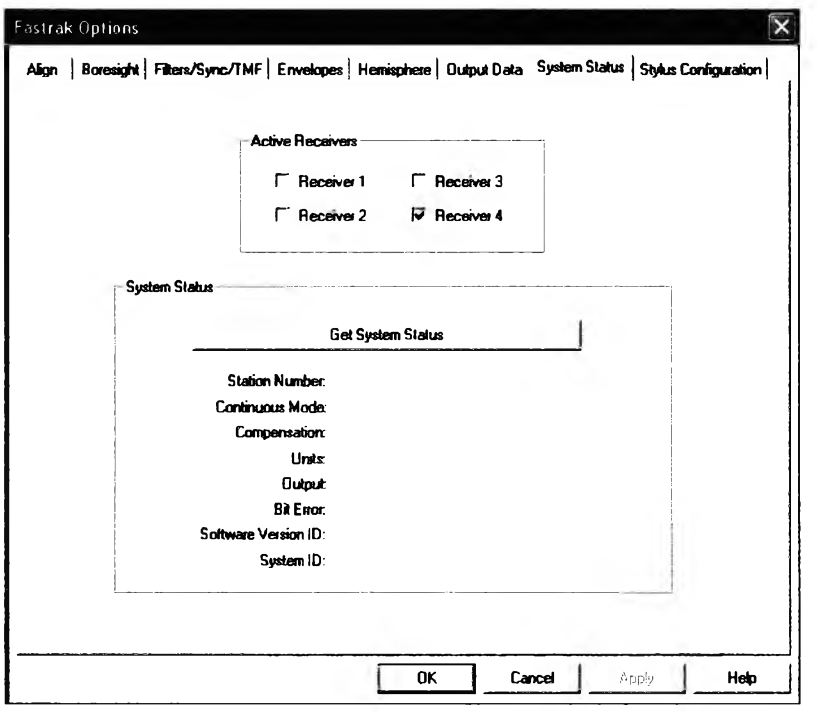
รูปที่ ข.11 การตั้งแกนอ้างอิงของตัวส่งสัญญาณ

5) เลือก Options -> Output Data



รูปที่ ข.12 ประเภทของข้อมูลส่งค่าออกมาจากชุดควบคุม Fastrak®

6) เลือก Options -> System Status (ตรวจสอบสถานะ)



รูปที่ ข.13 ตรวจสอบสถานะของระบบรับส่งสัญญาณ Fastrak®

ภาคผนวก ค.

โปรแกรมที่ใช้ในการทดลอง

ค.1 นำเรื่อง

โปรแกรมที่พัฒนาขึ้นมานี้ใช้การเขียนในรูปแบบของ m-file ซึ่งเป็นส่วนหนึ่งของโปรแกรม Matlab® รูปแบบการใช้โปรแกรมจะเข้าได้ง่ายเนื่องจากเป็นการเขียนด้วยภาษาชั้นสูงและยังมีฟังค์ชั่นภายในให้เลือกใช้มากมาย จากโปรแกรมที่จะนำเสนอนี้ ผู้จัดทำคาดหวังว่าจะสามารถยังประโยชน์แก่ผู้ที่นำวิทยานิพนธ์นี้ไปศึกษาได้ต่อไป

ค.2 โปรแกรมที่ใช้ในการจำลองควบคุม

โปรแกรมที่ใช้งานแบ่งออกเป็นสองส่วนคือ ส่วนที่ใช้ควบคุมทางเดินของหุ่นยนต์ 2-Link Planar Arm และส่วนของหุ่นยนต์ Articulated

ค.2.1 โปรแกรมควบคุมทางเดินของหุ่นยนต์ 2-Link Planar Arm

การแบ่งวิธีการควบคุมออกเป็นสองแบบ โดยใช้ควบคุมทางเดินวงกลมบนระนาบและรูปสี่เหลี่ยมซึ่งจะขอยกตัวอย่างเฉพาะโปรแกรมควบคุมทางเดินวงกลม เนื่องจากใช้หลักการเดียวกัน

ค.2.1.1 วิธี Global Collision Avoidance

```
% Global : Create Modified Trajectory for Avoidance already %
clear all
w=pi/8;
DT=0.05;timeN=16;
t=0:DT:timeN;
% Create Circular path and Modified %
wall = 0.3;
for i=1:length(t)
   X(i)=0.5+0.25*cos(w*t(i));
   Y(i)=0.25*sin(w*t(i));
      if X(i)<wall
```

```
            X(i)=wall;
            Y(i)=Y(i);
               Xda(i)=(X(i)-X(i-1))/DT;      % Backward Divided Difference %
               Yda(i)=(Y(i)-Y(i-1))/DT;
               Xd(i)=Xda(i);
               Yd(i)=Yda(i);
                  Xdda(i)=(Xd(i)-Xd(i-1)/DT);
                  Ydda(i)=(Yd(i)-Yd(i-1)/DT);
                  Xdd(i)=Xdda(i);
                  Ydd(i)=Ydda(i);
         else
            Xd(i)=-0.25*w*sin(w*t(i));
            Yd(i)=0.25*w*cos(w*t(i));
            Xdd(i)=-0.25*w*w*cos(w*t(i));
            Ydd(i)=-0.25*w*w*sin(w*t(i));
         end
      Z(i)=0.3;Zd(i)=0;Zdd(i)=0;
end
plot(X,Y)
axis([0.25 0.8 -0.4 0.4]);
% Robot Model : Two-link planar arm %
twolink
i=1;R=[1 0 0;0 1 0;0 0 1];
   T=[R [0.5 Y(i) Z(i)]';[0 0 0 1]];
   q(1:2,i)=ikine(tl,T,[0 0],[1 1 0 0 0 0])';
   J=jacob0(tl,q(1:2,i));
   qd(1:2,i)=inv(J(1:2,1:2))*[Xd(i) Yd(i)]';
for i=2:length(t)
   T=[R [X(i) Y(i) Z(i)]';[0 0 0 1]];
   q(1:2,i)=ikine(tl,T,q(1:2,i-1)',[1 1 0 0 0 0])';
   J=jacob0(tl,q(1:2,i));
```

```
    qd(1:2,i)=inv(J(1:2,1:2))*[Xd(i) Yd(i) ]';
end
% Find Joint Acceleration (Ideal)
i=2;
qdd(1:2,i)=(qd(1:2,i)-qd(1:2,i-1))/(DT); % Backward Divided Difference %
qdd(1:2,1)=qdd(1:2,2);
for i=2:length(t)-1
    qdd(1:2,i)=(qd(1:2,i+1)-qd(1:2,i-1))/(2*DT);%CentralDivided Difference %
end
% Actual Status % Dynamic Simulation on Robot Control Law %
Kp=100
Kv=2*sqrt(Kp);
clear qdda qda qa
%%Initial Position%%
qa(1:2,1)=0; qda(1:2,1)=0; qdda(1:2,1)=0;
T=fkine(tl,qa(1:2,1));
% Robot Arm Control  %
for i=1:length(t)-1
    T=fkine(tl,qa(1:2,i));
    Txc=T(1,4);Tyc=T(2,4);Tzc=T(3,4);
    Xc(i)=Txc;Yc(i)=Tyc;Zc(i)=Tzc;
    c=coriolis(tl,qa(1:2,i)',qda(1:2,i)');
    g=gravload(tl,qa(1:2,i)');
    beta=c+g;
    alpha=inertia(tl,qa(1:2,i));
    Kp_term(i,1:2)=q(1:2,i)'-qa(1:2,i)';
    Kv_term(i,1:2)=qd(1:2,i)'-qda(1:2,i)';
    ft=qdd(1:2,i)+Kv*(qd(1:2,i)'-qda(1:2,i)')'+Kp*(q(1:2,i)'-qa(1:2,i)')';
% Actual total Joint Torque
    f=alpha*ft+beta' ;
    qdda(:,i+1)=accel(tl,qa(:,i),qda(:,i),f);
```

```
   qda(:,i+1)=qda(:,i)+qdda(:,i+1)*DT;
   qa(:,i+1)=qa(:,i)+qda(:,i+1)*DT+0.5*qdda(:,i+1)*DT^2;
end
figure(1)
axis([0.3 1.5 -0.3 0.3])
plot(X,Y)
hold on
plot(Xc,Yc,'r.-')
title('2link planar Arm: Path Trajectory Avoidance Kp=100');
xlabel('X axis');ylabel('Y axis');
LEGEND('Avoidance Trajectory','Moving Path')
```

ค.2.1.2 วิธี Local Collision Avoidance

กรณี Circular Path

```
% Local : Create Modified Trajectory for Avoidance already %
clear all
w=pi/8; DT=0.05;timeN=16;
t=0:DT:timeN;
% Create Circular path %
for i=1:length(t)
   X(i)=0.5+0.25*cos(w*t(i));
   Y(i)=0.25*sin(w*t(i));
   Xd(i)=-0.25*w*sin(w*t(i));
   Yd(i)=0.25*w*cos(w*t(i));
   Xdd(i)=-0.25*w*w*cos(w*t(i));
   Ydd(i)=-0.25*w*w*sin(w*t(i));
   Z(i)=0.3;Zd(i)=0;Zdd(i)=0;
end
% Robot Model : Two-link planar arm %
twolink
i=1;R=[1 0 0;0 1 0;0 0 1];
```

```
    T=[R [X(i) Y(i) Z(i)]';[0 0 0 1]];
    q(1:2,i)=ikine(tl,T,[0 0],[1 1 0 0 0 0])';
    J=jacob0(tl,q(1:2,i));
    qd(1:2,i)=inv(J(1:2,1:2))*[Xd(i) Yd(i)]';
for i=2:length(t)
    T=[R [X(i) Y(i) Z(i)]';[0 0 0 1]];
    q(1:2,i)=ikine(tl,T,q(1:2,i-1)',[1 1 0 0 0 0])';
    J=jacob0(tl,q(1:2,i));
    qd(1:2,i)=inv(J(1:2,1:2))*[Xd(i) Yd(i) ]';
end
% Find Joint Acceleration (Ideal)
i=2;
qdd(1:2,i)=(qd(1:2,i)-qd(1:2,i-1))/(DT);    % Backward Divided Difference %
qdd(1:2,1)=qdd(1:2,2);
for i=2:length(t)-1
    qdd(1:2,i)=(qd(1:2,i+1)-qd(1:2,i-1))/(2*DT); % Central Divided Difference %
end
% Actual Status % Dynamic Simulation on Robot Control Law %
Kp=95
Kv=2*sqrt(Kp);
clear qdda qda qa
%%Initial Position%%
qa(1:2,1)=0; qda(1:2,1)=0; qdda(1:2,1)=0;
T=fkine(tl,qa(1:2,1));
Xc=T(1,4);Yc=T(2,4);Zc=T(3,4);
barrier(1,1)=0.3;
 % Robot Arm Control  %
for i=1:length(t)-1
    c=coriolis(tl,qa(1:2,i)',qda(1:2,i)'); g=gravload(tl,qa(1:2,i)');
    beta=c+g; alpha=inertia(tl,qa(1:2,i));
    Kp_term(i,1:2)=q(1:2,i)'-qa(1:2,i)';
```

```
    Kv_term(i,1:2)=qd(1:2,i)'-qda(1:2,i)';
    ft=qdd(1:2,i)+Kv*(qd(1:2,i)'-qda(1:2,i)')'+Kp*(q(1:2,i)'-qa(1:2,i)')';
% Actual total Joint Torque
    f=alpha*ft+beta' ;
    qdda(:,i+1)=accel(tl,qa(:,i),qda(:,i),f);
    qda(:,i+1)=qda(:,i)+qdda(:,i+1)*DT;
    qa(:,i+1)=qa(:,i)+qda(:,i+1)*DT+0.5*qdda(:,i+1)*DT^2;
       T=fkine(tl,qa(1:2,i+1));
       Xa(i+1)=T(1,4); Ya(i+1)=T(2,4);Za(i+1)=T(3,4);
%% Collision Avoidance %%
T=fkine(tl,qa(1:2,i+1));
Xc(i+1)=T(1,4); Yc(i+1)=T(2,4);Zc(i+1)=T(3,4);
wall=0.5;
barrier(1,i+1)=wall;
 if Xc(i+1)<=wall
    Xcc = wall;
    Xc(i+1)=Xcc;
    Yc(i+1)=T(2,4);
    Xdc(i+1)=(Xc(i)-Xc(i-1))/DT;
    Ydc(i+1)=(Yc(i)-Yc(i-1))/DT;
    Jc=jacob0(tl,qa(1:2,i+1));
    qda(1:2,i+1)=inv(Jc(1:2,1:2))*[Xdc(i+1) Ydc(i+1)]';
    qa(1:2,i+1)= qa(1:2,i)+ (qda(1:2,i+1)*DT);
    qdda(1:2,i+1)=(qda(1:2,i+1)-qda(1:2,i))/(DT);
 end
end
figure(1)
axis([0 2 -0.3 0.3])
plot(X,Y)
hold on
plot(Xa(2:i),Ya(2:i),'r.-')
```

```
title('2link planar Arm: Path Trajectory Avoidance Kp=50');
xlabel('X axis');ylabel('Y axis');
LEGEND('Avoidance Trajectory','Moving Path','Wall=0.3')
```

ค.2.2 โปรแกรมควบคุมทางเดินของหุ่นยนต์ Articulated Robot

กรณี Circular Path

```
clear L
L{1}=link([-pi/2 0 0 10 0]);
L{2}=link([0 10 0 0 0]);
L{3}=link([0 10 0 0 0]);
L{1}.m=1;  L{2}.m=1;  L{3}.m=1;
 L{1}.r=[0 0 0]; L{2}.r=[10 0 0]; L{3}.r=[10 0 0];
L{1}.I=[0 0 0 0 0 0];
L{2}.I=[0 0 0 0 0 0];
L{3}.I=[0 0 0 0 0 0];
L{1}.Jm=200e-6;
L{2}.Jm=200e-6;
L{3}.Jm=200e-6;
L{1}.G=1;  L{2}.G=1;  L{3}.G=1;
%viscous friction (motor referenced)
L{1}.B=0;  L{2}.B=0;  L{3}.B=0;
%Coulomb friction (motor referenced)
L{1}.Tc=[0 0];
L{2}.Tc=[0 0];
L{3}.Tc=[0 0];
%some useful poses
q0=[0 0 0];   %zero angles , L shaped pose
ArticulatedRobot=robot(L,'3D articulated Arm', 'Chulalongkorn');
%clear L
ArticulatedRobot.name='3D Articulated Arm';
```

```
ArticulatedRobot.manuf='Chulalongkorn';
w=pi/6;
DT=0.01;timeN=12;
t=0:DT:timeN;
for i=1:length(t)
   X(i)=10+2.5*cos(w*t(i));
   Y(i)=2.5*sin(w*t(i));
   Xd(i)=-2.5*w*sin(w*t(i));
   Yd(i)=2.5*w*cos(w*t(i));
   Xdd(i)=-2.5*w*w*cos(w*t(i));
   Ydd(i)=-2.5*w*w*sin(w*t(i));
   Z(i)=5;Zd(i)=0;Zdd(i)=0;
end
%Articulated3D;
   R=[1 0 0;0 1 0;0 0 1];
i=1;
   T=[R [X(i) Y(i) Z(i)]';[0 0 0] 1];
   qr(1:3,i)=ikine(ArticulatedRobot,T,[0 0 0],[1 1 1 0 0 0])';
   J=jacob0(ArticulatedRobot,qr(1:3,i));
   qdr(1:3,i)=inv(J(1:3,1:3))*[Xd(i) Yd(i) Zd(i)]';
for i=2:length(t)
   T=[R [X(i) Y(i) Z(i)]';[0 0 0] 1];
   qr(1:3,i)=ikine(ArticulatedRobot,T,qr(1:3,i-1),[1 1 1 0 0 0])';
   J=jacob0(ArticulatedRobot,qr(1:3,i));
   qdr(1:3,i)=inv(J(1:3,1:3))*[Xd(i) Yd(i) Zd(i)]';
end
for i=2:length(t)-1
   qddr(1:3,i)=(qdr(1:3,i+1)-qdr(1:3,i-1))/(2*DT);
end
qddr(1:3,1)=qddr(1:3,2);
i=i+1;
```

```
qddr(1:3,i)=(qdr(1:3,i)-qdr(1:3,i-1))/(DT);
%% Dynamic Simulation on Robot Control %%
Kp=100
Kv=2*sqrt(Kp);
%%Initial Dummy%%
DQ=[1 0 0;0 1 0;0 0 1];
DX=[0 0 0];
%%Initial Position%%
q(1:3,1)=0; qd(1:3,1)=0; qdd(1:3,1)=0;
T=fkine(ArticulatedRobot,q(1:3,1));
Xc(1)=T(1,4);Yc(1)=T(2,4);Zc(1)=T(3,4);
XXe=0;
for i=1:length(t)
  beta=coriolis(ArticulatedRobot,q(1:3,i)',qd(1:3,i)')+gravload(ArticulatedRobot,q(1:3,i)');
   alpha=inertia(ArticulatedRobot,q(1:3,i));
   ft=qddr(1:3,i)+Kv*(qdr(1:3,i)'-qd(1:3,i)')'+Kp*(qr(1:3,i)'-q(1:3,i)')';
   f=alpha*ft+beta';
   qdd(:,i+1)=accel(ArticulatedRobot,q(:,i),qd(:,i),f);
   qd(:,i+1)=qd(:,i)+qdd(:,i+1)*DT;
   q(:,i+1)=q(:,i)+qd(:,i+1)*DT+0.5*qdd(:,i+1)*DT^2;
%% Collision Avoidance %%
T=fkine(ArticulatedRobot,q(1:3,i+1));
Xc(i+1)=T(1,4);Yc(i+1)=T(2,4);Zc(i+1)=T(3,4);
Jxx=(DX*inv(DQ));
XXe=(Jxx*qd(1:3,i+1))*DT+Xc(i);
wall=3;
barrier(i,1)=wall;
if Xc(i+1)<wall
   qdp=[0 0 1]';
   A=[Jxx;qd(1:3,i)';qdp'];
   B=[(XXe-wall)/DT 0 0]';
```

```
   dqdr(1:3,i+1)=-inv(A)*B;
   qdr(1:3,i+1)=qd(1:3,i+1)+dqdr(1:3,i+1);
   qdr(1:3,i+1)=0.05/norm(qdr(1:3,i+1))* qdr(1:3,i+1);
   dqdr(1:3,i+1)=(qdr(1:3,i+1)-qdr(1:3,i))/DT;
   dq(1:3,i+1)= qdr(1:3,i+1)*DT;
   qr(1:3,i+1)= q(1:3,i)+ dq(1:3,i+1);
 end
DQ(1:3,1)=DQ(1:3,2);DQ(1:3,2)=DQ(1:3,3);DQ(1:3,3)=q(1:3,i+1)-q(1:3,i);
DX(1)=DX(2);DX(2)=DX(3);DX(3)=Xc(i+1)-Xc(i);
end
figure(1)
axis([0 20 -5 20]);
plot(Xc(1,1:i),Yc(1,1:i))
title('ArticulatedRobot: Path Trajectory Avoidance Kp=10');
xlabel('X axis');ylabel('Y axis');
hold on
plot(X(1,1:i),Y(1,1:i),'r-.')
plot(barrier,Y(1,1:i),'m:')
LEGEND('Path Avoidance','Path Command','Barrier x=3')
```

ค.2.3 ฟังก์ชันใน Robotic Toolbox ที่ใช้งาน

1) Forward Kinematics :

T = fkine(robot,q)

โดยที่

T คือ Homogeneous Transformation Matrix

robot คือ **ชนิดของหุ่นยนต์** (2 Link Planar Arm = twolink = tl)

q คือ **เวคเตอร์ขนาดของมุมข้อต่อในแต่ละข้อต่อ ขณะที่ทำ** การหา Homogeneous Transformation Matrix

2) Inverse Kinematics :

q = ikine (robot, T, q0, M)

โดยที่

q	คือ	เวคเตอร์ขนาดของมุมข้อต่อในแต่ละข้อต่อ ขณะที่ทำการหา Inverse Transformation Matrix
robot	คือ	ชนิดของหุ่นยนต์ (2 Link Planar Arm = twolink = tl)
T	คือ	Homogeneous Transformation Matrix
q0	คือ	เวคเตอร์ขนาดของมุมข้อต่อเริ่มต้น
M	คือ	เมตริกซ์บอกความอิสระในการเคลื่อนที่ของหุ่นยนต์ (1x6) ในแนวเลื่อนตามแนวแกนและหมุนรอบแกน x, y, z M=1 สามารถเคลื่อนที่ได้ในแนวแกน M=0 ไม่สามารถเคลื่อนที่ได้ในแนวแกน

3) Jacobian :

J0 = jocob0 (robot,q)

Jn = jacobn (robot,q)

โดยที่

J0	คือ	Jacobian เทียบกับฐานของหุ่นยนต์
Jn	คือ	Jacobian เทียบกับปลายแขนกลของหุ่นยนต์
robot	คือ	ชนิดของหุ่นยนต์ (2 Link Planar Arm = twolink = tl)
q	คือ	เวคเตอร์ขนาดของมุมข้อต่อในแต่ละข้อต่อ ขณะที่ทำการหา Jacobian

4) Forward Dynamics :

tau = itorque (robot,q,qdd)

โดยที่

tau	คือ	เวคเตอร์ของแรงขับของข้อต่อในแต่ละข้อต่อ
robot	คือ	ชนิดของหุ่นยนต์ (2 Link Planar Arm = twolink = tl)
q	คือ	เวคเตอร์ขนาดของมุมข้อต่อในแต่ละข้อต่อ
qdd	คือ	เวคเตอร์ขนาดความเร่งของข้อต่อในแต่ละข้อต่อ

5) Inverse Dynamics :

qdd = accel (robot,q,qd,f)

โดยที่

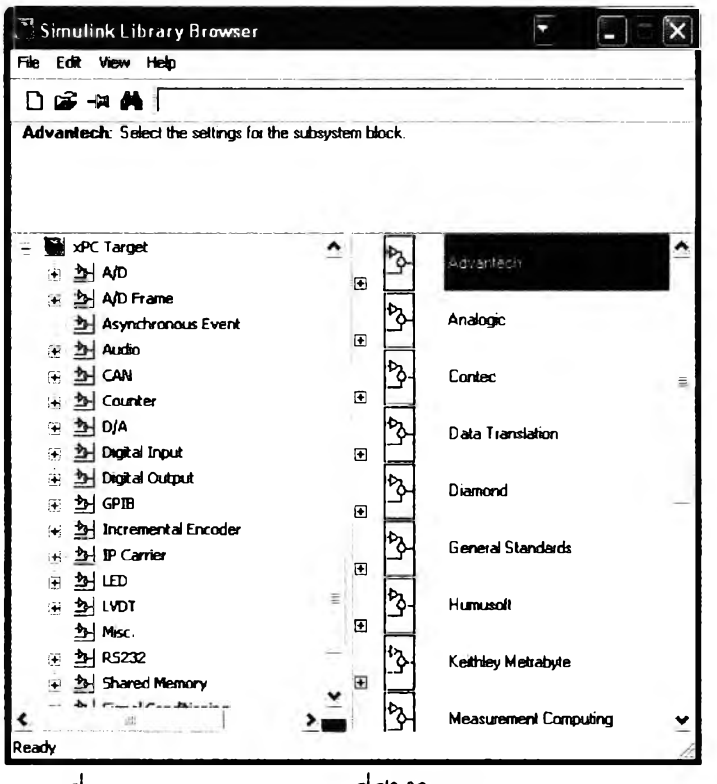
qdd	คือ	เวคเตอร์ขนาดความเร่งของข้อต่อในแต่ละข้อต่อ
robot	คือ	ชนิดของหุ่นยนต์ (2 Link Planar Arm = twolink = tl)
q	คือ	เวคเตอร์ขนาดของมุมข้อต่อในแต่ละข้อต่อ
qd	คือ	เวคเตอร์ขนาดความเร็วของข้อต่อในแต่ละข้อต่อ
f	คือ	เวคเตอร์ของแรงขับของข้อต่อในแต่ละข้อต่อ

ค.3 โปรแกรมที่ใช้ในการทดลองควบคุมทางเดินของแขนหุ่นยนต์ CRS Robot เพื่อหลบหลีกสิ่งกีดขวาง

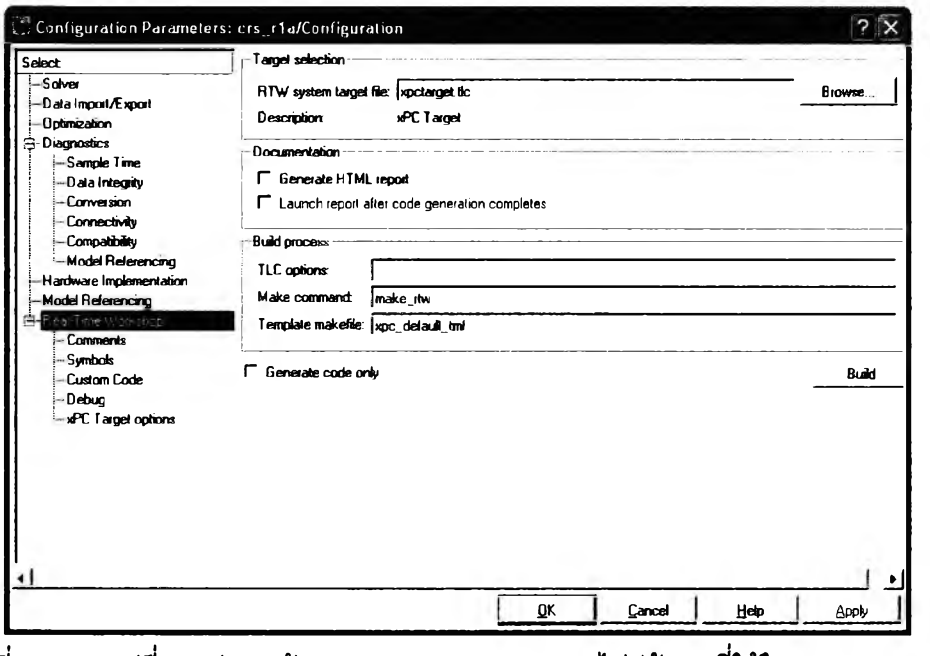
ค.3.1 ความรู้เบื้องต้นก่อนการใช้งาน Matlab® xPC

ก่อนการใช้งานของระบบ xPC เครื่อง Host PC จะต้องติดตั้งระบบปฏิบัติการ Window และโปรแกรมต่างๆ ดังนี้

- Matlab® ใช้ในการควบคุมและติดต่อกับระบบ xPC โดยผ่านบรรทัดคำสั่ง หรือหน้าต่างควบคุม สามารถใช้ในการ บันทึกข้อมูลจาก Target PC สั่งเริ่มและหยุดการทำงานของ Target เปลี่ยนแปลงค่าตัวแปรที่ใช้ รวมไปถึงการรวบรวมและวิเคราะห์ข้อมูลที่ได้จาก Target PC
- Simulink Library ใช้ในการสร้าง Block Diagram เพื่อควบคุมและจำลองระบบพลศาสตร์ที่สนใจ และสามารถสร้าง Block ที่มีคุณลักษณะเฉพาะตามที่ต้องการเพิ่มเติมจากที่ Simulink Library มีอยู่ โดยการใช้ C-Code S-Function เพื่อขยายความสามารถของโปรแกรมออกไป และอีกหนึ่งคุณลักษณะที่น่าสนใจใน Simulink Library คือ IO Diver Block Library ซึ่งเป็นการจัดเตรียม Diver สำหรับการติดต่อ IO ชนิดต่างๆ ที่นิยมใช้มากกว่า 400 ชนิด ดังแสดงในรูป ค.1 เป็นต้น
- Real-Time Workshop ใช้ในการเปลี่ยนรูปแบบข้อมูลจาก Block Diagram ไปสู่รหัสภาษา C ด้วยคำสั่ง Build ดังแสดงในรูปที่ ค.2
- C Compiler ใช้ในการสร้าง code เพื่อใช้ในการปฏิบัติการของ xPC kernel ในการใช้งานต้องติดตั้งที่ Host PC โดยโปรแกรมที่สามารถใช้ได้คือ Microsoft Visual C++ version 5 6 หรือ 7



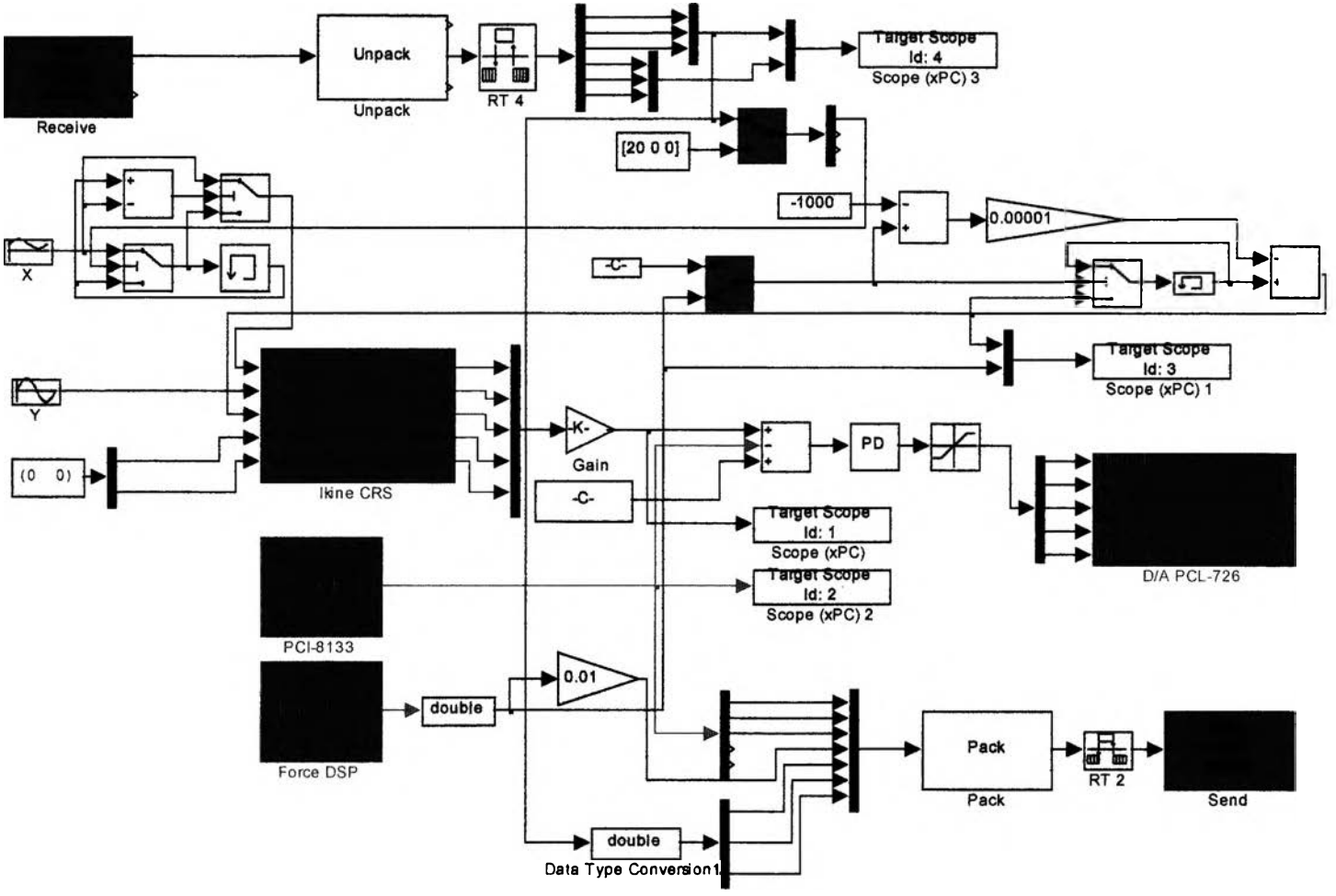
รูปที่ ค.1 IO Block Library ที่มีใช้ใน Simulink Library



รูปที่ ค.2 การเปลี่ยนรูปแบบข้อมูลจาก Block Diagram ไปสู่ข้อมูลที่ใช้ใน xPC Target

ค.3.2 โปรแกรมที่ใช้ทดลองควบคุมหุ่นยนต์ CRS Robot

โปรแกรมที่ใช้ทดลองควบคุมในรูปแบบของ Simulink ที่นำข้อมูลอุปกรณ์ตรวจรู้แรงและรู้ตำแหน่งพิกัดใน 3 มิติ มาใช้ป้อนกลับเพื่อใช้ประกอบการควบคุมหุ่นยนต์ CRS Robot



รูปที่ ค.3 ระบบควบคุมแขนหุ่นยนต์ CRS Robot ที่ติดตั้งอุปกรณ์ตรวจรู้แรงและรู้ตำแหน่งพิกัดใน 3 มิติ

ประวัติผู้เขียนวิทยานิพนธ์

นายมนตรี บุญยะผลานันท์ เกิดเมื่อวันที่ 6 กันยายน พ.ศ.2522 ที่กรุงเทพมหานคร สำเร็จการศึกษาในระดับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2543 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2545