บรรณานุกรม

Dorf, Richard C. MODERN CONTROL SYSTEM USA, Addison-wesley 1980

J.Y.S.Luh "CONVENTIONAL CONTROLLER DESING FOR INDUSTRIAL ROBOT - A TUTORIAL" ,IEEE Trans.Sys., Man,Cybern,Vol.SMC,May 1983

Kuo,B.C, AUTOMATIC CONTROL SYSTEM Prentice-Hall of India Private Limited, New Delhi, 1985

Neuman,C.P.,and Marray,J.J.,"Linearization and Sensitivity Function of Dynamic Robot Model," IEEE Trans.Sys., Man,Cybern, Vol.SMC-14 Nov/Dec 1980

Ogata, K, MODERN CONTROL EGINEERING Prentice-Hall of India Private Limited, New Delhi, 1984

Palm,William J. MODELING ANALYSIS AND CONTROL OF DYNAMIC SYSTEMS New York, John Wiley&sons, 1983

Paul. Richard P. ROBOT MANIPULATORS: MATHEMATICS, PROGRAMMING, AND CONTROL , Cambridge, MA:MIT, 1981

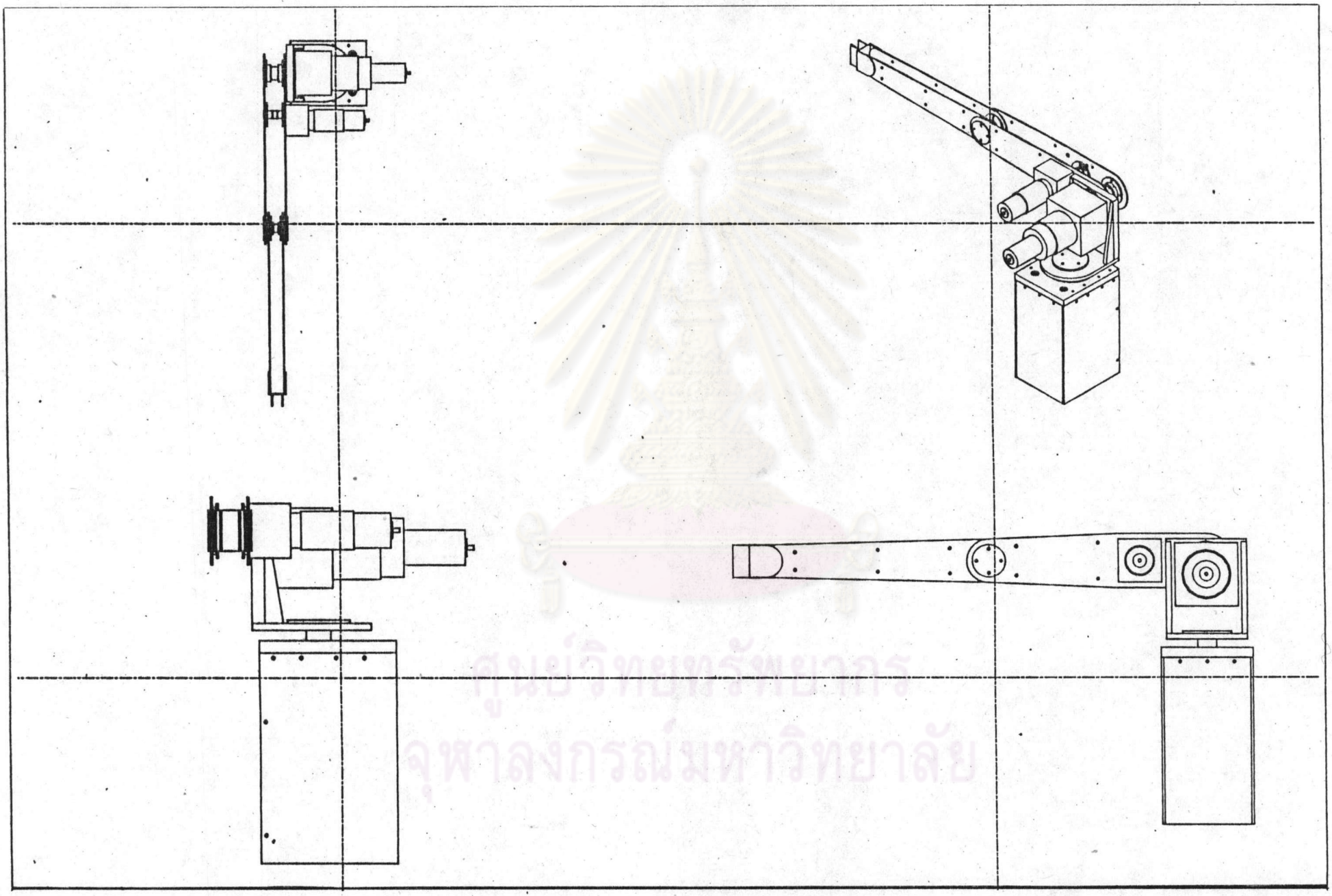
Yoram Koren ROBOTICS FOR ENGINEERS New York, McGraw-Hill, 1985

ภาคผนวก ก.

รายละเอียดของมอเตอร์และแบบโครงสร้างของแขนกล

มอเตอร์ที่ใช้ในโครงการวิทยานิพนธ์นี้ เป็นมอเตอร์กระแสตรงของบริษัท Electro Craft Model 586-022-113 และ Model 652-02-112 ด้านท้ายของมอเตอร์มีทาโคมิเตอร์ ด้านเพลามอเตอร์มีชุดเฟืองทดที่มีอัตราทด 100:1 และ 90:1 สำหรับรายละเอียดต่างๆ ของมอเตอร์ แสดงดังตารางต่อไปนี้

SPECIFICATIONS		MODELS	
		E652-MG 0652-02-XXX	**E586-MG**
Rated Voltage V_r	V	120	36
NO-LOAD SPEED at V_r N_o	rpm	3750	6200
MAX. RATED I at STALL I_r	A	4.4	4.6
STALL TORQUE at I_r T_o	oz-in	100	29
MAX. PULSE CURRENT I_{pk}	A	25.0	24.0
TORQUE CONSTANT K_t	oz-in/A	27	7.8
VOLTAGE CONSTANT K_e	volts/krpm	20	5.8
TERMINAL RESISTANCE R_t	Ωat 25°C	1.8	1.1
Armature Mom. Inertia J_m	oz-in-s^2	3.2×10^{-2}	5.5×10^{-3}
Rotational Loss Constant K_d	oz-in/krpm	0.68	0.10
Static Friction Torque T_f	oz-in	7.0	3.0
Thermal Resistance Arm./Amb. R_{th}	°C/W	2.8	5.0
ARMATURE INDUCTANCE L_a	mH	6.7	2.3
ELECTRICAL Time Constant τ_e	ms	3.7	2.1
MECHANICAL Time Constant τ_m	ms	10.0	14.0
TACH. Voltage Gradient K_g	V/krpm	21.0	14.2
TACH. Terminal Resistance R_g	Ωat 25°C	800	720
TACH. Armature Inductance L_g	mH	255	138
TACH. Load Resistance (optimum) R_l	Ω	10.000	5000
Ripple Amplitude	%pk-pk	5.0	5.0
LINEARITY	%	0.2	0.2
Temperature Coefficient	%/°C	-0.05	-0.05



ภาคผนวก ข.

Q-Matrix Lagrangian formulation

การหาความสัมพันธ์ไดนามิกตามวิธีลากรานจ์ มีการหาอนุพันธ์ของทรานสฟอร์เมตริก และหาทรานสโพสของเมตริกบ่อยๆ เพื่อลดความยุ่งยากในการหาความสัมพันธ์ และสะดวกต่อการคำนวณหาคำตอบด้วยคอมพิวเตอร์จึงได้มีการสร้างเมตริก Q ขึ้น โดยมีรายละเอียดและขั้นตอนทั้งหมดตามวิธีการ Q เมตริก รายละเอียดในตารางที่ 1-3และมีวิธีการนำไปใช้แสดงดังตัวอย่าง

TABLE I
MANIPULATOR KINEMATICS

	Coordinates
N	Number of degrees-of-freedom (DOF).
X_i, Y_i, Z_i	Right cartesian axes defining the ith coordinate frame (at the ith link).
r_i	Position four-vector of a point on line i, in the ith coordinate frame: $r_i = [r_{ix} r_{iy} r_{iz} 1]^T$.
	Kinematic Link Parameters
θ_i	Angle[a] between links $(i-1)$ and i; θ_i is the joint coordinate if joint i is revolute.
Δ_i	Distance[a] between links $(i-1)$ and i; Δ_i is the joint coordinate if joint i is prismatic.
a_i	Length[a] of link i; $a_i = 0$ if joint i is prismatic.
α_i	Twist[a] of link i.
	Homogeneous Transformations
A_i	Homogeneous (4×4) coordinate transformation matrix, *from* the ith coordinate frame *to* the $(i-1)$st coordinate frame; A_i, which is defined completely by the four kinematic link parameters (θ_i, Δ_i, a_i and α_i), is

$$A_i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & \Delta_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

${}^pT_k$	Homogeneous (4×4) coordinate transformation matrix from the kth coordinate frame to the pth coordinate frame; ${}^pT_k$ is calculate according to

$$ {}^pT_k = A_{p+1}A_{p+2}\cdots A_{k-1}A_k, \quad \text{for } 0 \leqslant p < k \leqslant N \quad (2)$$

and

$$ {}^pT_p \triangleq I.$$

${}^0T_N$	Forward solution (i.e., homogeneous (4×4) coordinate transformation matrix) from the end-effector (Nth) coordinate frame to the base (zeroth) coordinate frame.

[a]Angle, distance, length, and twist follow the definitions in [21].

TABLE II
MANIPULATOR DYNAMICS

Dynamic Link Parameters

J_i Pseudo-inertia (4 × 4) matrix of link i: $J_i = \int_{\text{link } i} [r_i r_i^T]\, dm_i$.

$\bar{r}_i$ Center of mass four-vector of link i, in the ith coordinate frame: $\bar{r}_i = [\bar{r}_{ix}\bar{r}_{iy}\bar{r}_{iz}1]^T$.

m_i Mass of link i.

Gravity Vector

g Gravity four-vector in the base coordinate frame: $g = [g_x g_y g_z 0]^T$.

Differential Kinematics

${}^pU_{ki}$ First partial-derivative (4 × 4) matrix of ${}^pT_k$ with respect to q_i; ${}^pU_{ki}$ is calculated according to

$$ {}^pU_{ki} = \frac{\partial {}^pT_k}{\partial q_i} = {}^pT_{(i-1)} Q_i^{(i-1)} T_k, \quad \text{for } 0 \leqslant p < i \leqslant k \leqslant N. \tag{3} $$

${}^pU_{kjm}$ Second partial-derivative (4 × 4) matrix of ${}^pT_k$ with respect to q_j and q_m; ${}^pU_{kjm}$ (which equals ${}^pU_{kmj}$) is calculated according to

$$ {}^pU_{kjm} = \frac{\partial^2 {}^pT_k}{\partial q_j \partial q_m} = {}^pT_{(j-1)} Q_j^{(j-1)} T_{(m-1)} Q_m^{(m-1)} T_k, \quad \text{for } 0 \leqslant p < j \leqslant m \leqslant k \leqslant N. \tag{4} $$

Q_i Bejczy constant (4 × 4) Q-matrix for joint i:

$$ Q_i = Q_R = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for a revolute } i\text{th joint, and} $$

$$ Q_i = Q_T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for a prismatic } i\text{th joint.} $$

TABLE III
DYNAMIC ROBOT MODEL

$$ D(q)\ddot{q} + C(q,\dot{q}) + G(q) = F(t) \tag{5a} $$

$$ \sum_{j=1}^{N} d_{ij}\ddot{q}_j + \sum_{j=1}^{N}\sum_{k=1}^{N} \dot{q}_j c_{jk}(i)\dot{q}_k + G_i = F_i(t), \quad \text{for } i = 1, 2, \cdots, N. \tag{5b} $$

Joint Coordinates

$q_i(q_i, \ddot{q}_i)$ Generalized joint coordinate (velocity, acceleration) of joint i: $q_i = \theta_i$ if joint i is revolute; and $q_i = d_i$ if joint i is prismatic.

$q(\dot{q}, \ddot{q})$ Generalized joint coordinate (velocity, acceleration) N-vector.

Parameters

$C(q,\dot{q})$ Centrifugal and Coriolis force N-vector with elements

$$ C_i = \dot{q}^T C(i) \dot{q}. \tag{6} $$

$C(i)$ Coupling ($N \times N$) matrix for joint i with elements

$$ c_{jk}(i) = \sum_{m=\max(i,j,k)}^{N} Tr\{{}^pU_{mjk} J_m {}^pU_{mi}^T\}, \quad \text{where } p \triangleq \min(i; j, k) - 1. \tag{7} $$

$D(q)$ Inertial coefficient ($N \times N$) matrix with elements

$$ d_{ij} = \sum_{k=\max(i,j)}^{N} Tr\{{}^pU_{kj} J_k {}^pU_{ki}^T\}, \quad \text{where } p \triangleq \min(i, j) - 1. \tag{8} $$

$G(q)$ Gravitational force N-vector with elements

$$ G_i = -\sum_{k=i}^{N} m_k g^T {}^0U_{ki} \bar{r}_k. \tag{9} $$

$F(t)$ Generalized external joint force N-vector with elements $F_i(t)$

ตัวอย่างการหาไดนามิกโมเดลของแขนกลดังรูปที่ 4.3 บทที่ 4 ข้างต้น โดยวิธี Q-matrix Lagrangian formulation

จากสมการที่ 1 และตารางในรูปที่ 4.3 ได้

$$ {}_{0}T_{1} = A_1 = \begin{vmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad \ldots 10 $$

$$ {}_{1}T_{2} = A_2 = \begin{vmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad \ldots 11 $$

$$ {}_{2}T_{3} = A_3 = \begin{vmatrix} C_2 & 0 & S_3 & a_3C_3 \\ S_3 & 0 & -C_3 & a_3S_3 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad \ldots 12 $$

เมื่อ $S_i = \sin\theta_i$ $\quad S_{ij} = \sin(\theta_i+\theta_j)$

$C_i = \cos\theta_i$ $\quad C_{ij} = \cos(\theta_i+\theta_j)$

จากตารางที่ 2 Pseudo-inertia matrix คือ

$$ J_i = \begin{vmatrix} -I_{ixx}+I_{iyy}+I_{izz} & I_{ixy} & I_{ixz} & m_ix_i \\ I_{ixy} & I_{ixx}-I_{iyy}+I_{izz} & I_{iyz} & m_iy_i \\ I_{ixz} & I_{iyz} & I_{ixx}+I_{iyy}-I_{izz} & m_iz_i \\ m_ix_i & m_iy_i & m_iz_i & m_i \end{vmatrix} $$

หรือ

$$ J_i = \begin{vmatrix} J_{ixx} & I_{ixy} & I_{ixz} & m_ix_i \\ I_{ixy} & J_{iyy} & I_{iyz} & m_iy_i \\ I_{ixz} & I_{iyz} & J_{izz} & m_iz_i \\ m_ix_i & m_iy_i & m_iz_i & m_i \end{vmatrix} \qquad \ldots 13 $$

และ Q-Matrix คือ

$$Q_1 = \begin{vmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} \qquad \ldots 14$$

จากสมการที่ 8 หาสัมประสิทธิ์แรงเฉื่อยได้

$$D_{ij} = \sum \mathrm{Tr}\{{}^{p}U_{kj} J_k {}^{p}U_{ki}{}^{T}\} \qquad p = \min(i,j)-1$$

$$D_{11} = \mathrm{Tr}\{{}^{0}U_{11} J_1 {}^{0}U_{11}{}^{T}\} + \mathrm{Tr}\{{}^{0}U_{21} J_2 {}^{0}U_{21}{}^{T}\} + \mathrm{Tr}\{{}^{0}U_{31} J_3 {}^{0}U_{31}{}^{T}\}$$

$$D_{12} = \mathrm{Tr}\{{}^{0}U_{22} J_2 {}^{0}U_{21}{}^{T}\} + \mathrm{Tr}\{{}^{0}U_{32} J_3 {}^{0}U_{31}{}^{T}\}$$

$$D13 = \mathrm{Tr}\{{}^{0}U_{33} J_3 {}^{0}U_{31}{}^{T}\}$$

$$D_{33} = \mathrm{Tr}\{{}^{2}U_{33} J_3 {}^{2}U_{33}{}^{T}\}$$

$$D_{21} = \mathrm{Tr}\{{}^{0}U_{21} J_2 {}^{0}U_{22}{}^{T}\} + \mathrm{Tr}\{{}^{0}U_{31} J_3 {}^{0}U_{32}{}^{T}\}$$

$$= D_{12}$$

$$D_{22} = \mathrm{Tr}\{{}^{1}U_{22} J_2 {}^{1}U_{22}{}^{T}\} + \mathrm{Tr}\{{}^{1}U_{32} J_3 {}^{1}U_{32}{}^{T}\}$$

$$D_{23} = \mathrm{Tr}\{{}^{1}U_{33} J_3 {}^{1}U_{32}{}^{T}\}$$

$$D_{31} = \mathrm{Tr}\{{}^{0}U_{31} J_3 {}^{0}U_{33}{}^{T}\}$$

$$= D_{13}$$

$$D_{32} = \mathrm{Tr}\{{}^{1}U_{32} J_3 {}^{1}U_{33}{}^{T}\}$$

$$= D_{23}$$

$$D_{33} = \mathrm{Tr}\{{}^{2}U_{33} J_3 {}^{2}U_{33}{}^{T}\}$$

การหา D_{13}

จากสมการที่ 2 หา ${}^{p}T_k$ ได้

$${}^{0}T_2 = A_1 A_2 = \begin{vmatrix} C_1C_2 & -C_1S_2 & -S_1 & a_2C_1C_2 \\ S_1C_2 & -S_1S_2 & C_1 & a_2S_1C_2 \\ -S_2 & -C_2 & 0 & -a_2S_2 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad \ldots 15$$

$${}^{0}T_3 = A_1 A_2 A_3 = \begin{vmatrix} C_1C_{23} & -S_1 & C_1S_{23} & a_3C_1C_{23}+a_2C_1C_2-d_3S_1 \\ S_1C_{23} & C_1 & S_1S_{23} & a_3S_1C_{23}+a_2S_1C_2+d_3C_1 \\ -S_{23} & 0 & C_{23} & -a_3S_{23}-a_2S_2 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad \ldots 16$$

จากสมการที่ 3 หา ${}^{P}U_{ki}$ ได้

$$ {}^{o}U_{3i} = {}^{o}T_{o}Q_{i}{}^{o}T_{3} $$

$$ {}^{o}U_{3i} = \begin{vmatrix} -S_1C_{23} & -C_1 & -S_1S_{23} & -a_3S_1C_{23}-a_2S_1C_2-d_3C_1 \\ C_1C_{23} & -S_1 & C_1S_{23} & a_3C_1C_{23}+a_2C_1C_2-d_3S_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} \quad \ldots 17 $$

$$ {}^{o}U_{3i}{}^{T} = \begin{vmatrix} -S_1C_{23} & C_1C_{23} & 0 & 0 \\ -C_1 & -S_1 & 0 & 0 \\ -S_1S_{23} & C_1S_{23} & 0 & 0 \\ -a_3S_1C_{23}-a_2S_1C_2-d_3C_1 & a_3C_1C_{23}+a_2C_1C_2-d_3S_1 & 0 & 0 \end{vmatrix} $$

$$ {}^{o}U_{33} = {}^{o}T_{2}Q_{i}{}^{2}T_{3} $$

$$ {}^{o}U_{33} = \begin{vmatrix} -C_1S_{23} & 0 & C_1C_{23} & -a_3C_1S_{23} \\ -S_1S_{23} & 0 & S_1C_{23} & -a_3S_1S_{23} \\ -C_{23} & 0 & -S_{23} & -a_3C_{23} \\ 0 & 0 & 0 & 0 \end{vmatrix} \quad \ldots 18 $$

$$ J_3{}^{o}U_{3i}{}^{T} = $$

$$ \begin{vmatrix} [-J_{3xx}S_1C_{23}-I_{3xy}C_1-I_{3xz}S_1S_{23}-m_3x_3(a_3S_1C_{23}+a_2S_1C_2+d_3C_1)] \\ [-I_{3xy}S_1C_{23}-J_{3yy}C_1-I_{3yz}S_1S_{23}-m_3y_3(a_3S_1C_{23}+a_2S_1C_2+d_3C_1)] \\ [-I_{3xz}S_1C_{23}-I_{3yz}C_1-J_{3zz}S_1S_{23}-m_3z_3(a_3S_1C_{23}+a_2S_1C_2+d_3C_1)] \\ [-m_{3x3}S_1C_{23}-m_{3y3}C_1-m_3z_3S_1S_{23}-m_3(a_3S_1C_{23}+a_2S_1C_2+d_3C_1)] \end{vmatrix} $$

$$ \begin{vmatrix} [-J_{3xx}C_1C_{23}-I_{3xy}S_1-I_{3xz}C_1S_{23}-m_3x_3(a_3C_1C_{23}+a_2C_1C_2+d_3S_1)] & 0 & 0 \\ [-I_{3xy}C_1C_{23}-J_{3yy}S_1-I_{3yz}C_1S_{23}-m_3y_3(a_3C_1C_{23}+a_2C_1C_2+d_3S_1)] & 0 & 0 \\ [-I_{3xz}C_1C_{23}-I_{3yz}S_1-J_{3zz}C_1S_{23}-m_3z_3(a_3C_1C_{23}+a_2C_1C_2+d_3S_1)] & 0 & 0 \\ [-m_{3x3}C_1C_{23}-m_{3y3}S_1-m_{3z3}C_1S_{23}-m_3(a_3C_1C_{23}+a_2C_1C_2+d_3S_1)] & 0 & 0 \end{vmatrix} $$

$$ Tr\{{}^{o}U_{33}J_3{}^{o}U_{3i}{}^{T}\} = I_{3xy}S_{23}+m_3x_3d_3S_{23}-I_{3yz}C_{23}-m_3z_3d_3C_{23}+ m_3y_3a_3S_{23}+m_3d_3a_3S_{23} \quad \ldots 19 $$

$$ D_{13} = I_{3xy}S_{23}-I_{3yz}C_{23}-m_3d_3(x_3S_{23}-z_3C_{23})+m_3a_3S_{23}(y_3+d_3) $$

ในทำนองเดียวกันหา D_{11}-D_{33} ได้

$$ D_{11} = I_{1yy}+I_{2xx}S_2S_2+I_{2yy}C_2C_2+m_2a_2C_2(2x_2C_2-2y_2S_2+a_2C_2)+ I_{3xx}S_{23}S_{23}+I_{3zz}C_{23}C_{23}+m_3(a_3C_{23}+a_2C_2)(a_3C_{23}+a_2C_2+ 2x_3C_{23}+2z_3S_{23})+m_3d_3(2y_3+d_3) $$

$$D_{12} = m_2 z_2 a_2 S_2 + m_3 d_3 (x_3 S_{23} - z_3 C_{23}) + m_3 (y_3 + d_3)(a_3 S_{23} + a_2 S_2)$$

$$D_{13} = I_{3xy} S_{23} - I_{3yz} C_{23} - m_3 d_3 (x_3 S_{23} - z_3 C_{23}) + m_3 a_3 S_{23} (y_3 + d_3)$$

$$D_{21} = D_{12}$$

$$D_{22} = I_{2zz} + m_2 a_2 (a_2 + 2x_2) + I_{3yy} + 2m_3 (a_2 C_3 + a_3)(a_3 + x_3) + 2m_3 z_3 a_2 S_3)$$

$$D_{23} = I_{3yy} + m_3 a_2 (x_3 C_3 + a_3 C_3 + z_3 S_3) + m_3 a_3 (2x_3 + a_3)$$

$$D_{31} = D_{13}$$

$$D_{32} = D_{23}$$

$$D_{33} = I_{3yy} + m_3 a_3 (a_3 + 2x_3)$$

จากสมการที่ 7 หาสัมประสิทธิ์แรงคอริโอลิสและเซนทิฟิวเกิลได้

$$D_{ijk} = c_{jk}(i) = \sum Tr\{{}^{p}U_{mjk} J_m {}^{p}U_{mi}{}^{T}\} \quad :p = \min(ijk)-1$$

$${}^{p}U_{mjk} = {}^{p}T_{(j-1)} Q_j^{(j-1)} T_{(m-1)} Q_m^{(m-1)} T_k \qquad 0<p<j<k<m<N$$

$$D_{111} = Tr\{{}^{0}U_{111} J_1 {}^{0}U_{11}{}^{T}\} + Tr\{{}^{0}U_{211} J_2 {}^{0}U_{21}{}^{T}\} + Tr\{{}^{0}U_{311} J_3 {}^{0}U_{31}{}^{T}\}$$

$$D_{112} = Tr\{{}^{0}U_{212} J_2 {}^{0}U_{21}{}^{T}\} + Tr\{{}^{0}U_{312} J_3 {}^{0}U_{31}{}^{T}\}$$

$$D_{113} = Tr\{{}^{0}U_{313} J_3 {}^{0}U_{31}{}^{T}\}$$

$$D_{121} = Tr\{{}^{0}U_{221} J_2 {}^{0}U_{21}{}^{T}\} + Tr\{{}^{0}U_{321} J_3 {}^{0}U_{31}{}^{T}\} = \emptyset$$

$$D_{122} = Tr\{{}^{0}U_{222} J_2 {}^{0}U_{21}{}^{T}\} + Tr\{{}^{0}U_{322} J_3 {}^{0}U_{31}{}^{T}\}$$

$$D_{123} = Tr\{{}^{0}U_{323} J_3 {}^{0}U_{31}{}^{T}\}$$

$$D_{131} = Tr\{{}^{0}U_{331} J_3 {}^{0}U_{31}{}^{T}\} = \emptyset$$

$$D_{132} = Tr\{{}^{0}U_{332} J_3 {}^{0}U_{31}{}^{T}\} = \emptyset$$

$$D_{133} = Tr\{{}^{0}U_{333} J_3 {}^{0}U_{31}{}^{T}\}$$

$$D_{211} = Tr\{{}^{0}U_{211} J_2 {}^{0}U_{22}{}^{T}\} + Tr\{{}^{0}U_{311} J_3 {}^{0}U_{32}{}^{T}\}$$

$$D_{212} = Tr\{{}^{0}U_{212} J_2 {}^{0}U_{22}{}^{T}\} + Tr\{{}^{0}U_{312} J_3 {}^{0}U_{32}{}^{T}\}$$

$$D_{213} = Tr\{{}^{0}U_{313} J_3 {}^{0}U_{32}{}^{T}\}$$

$$D_{221} = Tr\{{}^{0}U_{221} J_2 {}^{0}U_{22}{}^{T}\} + Tr\{{}^{0}U_{321} J_3 {}^{0}U_{32}{}^{T}\} = \emptyset$$

$$D_{222} = Tr\{{}^{1}U_{222} J_2 {}^{1}U_{22}{}^{T}\} + Tr\{{}^{1}U_{322} J_3 {}^{1}U_{32}{}^{T}\}$$

$$D_{223} = Tr\{{}^{1}U_{323} J_3 {}^{1}U_{32}{}^{T}\}$$

$$D_{231} = Tr\{{}^{0}U_{331} J_3 {}^{0}U_{32}{}^{T}\} = \emptyset$$

$$D_{232} = Tr\{{}^{1}U_{332} J_3 {}^{1}U_{32}{}^{T}\} = \emptyset$$

$$D_{233} = Tr\{{}^{1}U_{333} J_3 {}^{1}U_{32}{}^{T}\}$$

$$D_{311} = Tr\{{}^{0}U_{311} J_3 {}^{0}U_{33}{}^{T}\}$$

$$D_{312} = Tr\{{}^{0}U_{312}J_3{}^{0}U_{33}{}^{T}\}$$

$$D_{313} = Tr\{{}^{0}U_{313}J_3{}^{0}U_{33}{}^{T}\}$$

$$D_{321} = Tr\{{}^{0}U_{321}J_3{}^{0}U_{33}{}^{T}\} = 0$$

$$D_{322} = Tr\{{}^{1}U_{322}J_3{}^{1}U_{33}{}^{T}\}$$

$$D_{323} = Tr\{{}^{1}U_{323}J_3{}^{1}U_{33}{}^{T}\}$$

$$D_{331} = Tr\{{}^{0}U_{331}J_3{}^{0}U_{33}{}^{T}\} = 0$$

$$D_{332} = Tr\{{}^{1}U_{332}J_3{}^{1}U_{33}{}^{T}\} = 0$$

$$D_{333} = Tr\{{}^{2}U_{333}J_3{}^{2}U_{33}{}^{T}\}$$

การหา D_{113}

จากสมการที่ 4 หา ${}^{p}U_{kjm}$ ได้

$${}^{0}U_{313} = {}^{0}T_0Q_1{}^{0}T_2Q_1{}^{2}T_3$$

$$= \begin{bmatrix} S_1S_{23} & 0 & -S_1C_{23} & -a_3S_1S_{23} \\ -C_1S_{23} & 0 & C_1C_{23} & a_3C_1S_{23} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \ldots 20$$

$$Tr\{{}^{0}U_{313}J_3{}^{0}U_{31}{}^{T}\} = (I_{3xx}-I_{3zz})S_{23}C_{23}-I_{3xz}(S_{23}{}^{2}-C_{23}{}^{2})+$$
$$m_3(a_3C_{23}+a_2C_2)(z_3C_{23}-x_3S_{23}-a_3S_{23})-$$
$$m_3a_3S_{23}(x_3C_{23}+z_3S_{23})$$
$$= D_{113}$$

ในทำนองเดียวกันหา D_{111}-D_{333} ได้

$$D_{111} = 0$$

$$D_{112} = (I_{2xx}-I_{2yy})S_2C_2+(m_2y_2a_2)(S_2{}^{2}-C_2{}^{2})-m_2a_2S_2C_2(2x_2+a_2)$$
$$+(I_{3xx}-I_{3zz})S_{23}C_{23}-m_3(x_3C_{23}+z_3S_{23})(a_2S_2+a_3S_{23})$$
$$+m_3(a_2C_2+a_3C_{23})(z_3C_{23}-x_3S_{23}-a_3S_{23}-a_2S_2)$$

$$D_{113} = (I_{3xx}-I_{3zz})S_{23}C_{23}-m_3a_3S_{23}(x_3C_{23}+z_3S_{23})$$
$$+m_3(a_2C_2+a_3C_{23})(z_3C_{23}-x_3S_{23}-a_3S_{23})$$

$$D_{122} = m_2z_2a_2C_2+m_3d_3(x_3C_{23}+z_3S_{23})+m_3(y_3+d_3)(a_2C_2+a_3C_{23})$$

$$D_{123} = m_3d_3(x_3C_{23}+z_3S_{23})+m_3a_3C_{23}(y_3+d_3)$$

$$D_{133} = D_{123}$$

$$D_{211} = -D_{112}$$

$$D_{223} = -m_3a_2(a_3S_3+x_3S_3-z_3C_3)$$

$$D_{233} = D_{223}$$

$$D_{311} = -D_{113}$$

$$D_{322} = -D_{223}$$

$$D_{212} = D_{213} = D_{222} = D_{312} = D_{313} = D_{323} = D_{333} = 0$$

จากสมการที่ 9 หาแรงโน้มถ่วงที่ทำกับลิงค์ได้

$$D_i = G_i = -\Sigma m_k g^{T0} U_{ki} r_k$$

$$D_1 = m_1 g^{T0} U_{11} r_1 + m_2 g^{T0} U_{21} r_2 + m_3 g^{T0} U_{31} r_3$$

$$D_2 = m_2 g^{T0} U_{22} r_2 + m_3 g^{T0} U_{32} r_3$$

$$D_3 = m_3 g^{T0} U_{33} r_3$$

การหา D_3

$$g^T = [0 \quad 0 \quad -g \quad 0]$$

$$g^{T0} U_{33} = [gC_{23} \quad 0 \quad gS_{23} \quad ga_3 C_{23}]$$

$$D_3 = -m_3 g(C_{23}(x_3 + a_3) + z_3 S_{23})$$

ในทำนองเดียวกันหา $D_1 - D_2$ ได้

$$D_2 = m_2 g(C_2(-x_2 - a_2) + y_2 S_2) - m_3 g((x_3 + a_3)C_{23} + z_3 S_{23} + a_2 C_2)$$

$$D_1 = 0$$

จาก Di, Dij, Dijk นำมาหาแรงรวมที่ทำให้ลิงค์ด้วยความเร็ว θ ความเร่ง θ ณ ตำแหน่ง θ ได้

$$F_1 = D_{11}\theta_1 + D_{21}\theta_2 + D_{31}\theta_3 + Ia_1\theta_1 + D_1 + D_{112}\theta_1\theta_2 + D_{113}\theta_1\theta_3 + D_{122}\theta_2^2 + D_{123}\theta_2\theta_3 + D_{133}\theta_3^2 \quad \ldots 21$$

$$F_2 = D_{21}\theta_1 + D_{22}\theta_2 + D_{32}\theta_3 + Ia_2\theta_2 + D_2 + D_{223}\theta_2\theta_3 + D_{233}\theta_3^2 \quad \ldots 22$$

$$F_3 = D_{31}\theta_1 + D_{32}\theta_2 + D_{33}\theta_3 + Ia_3\theta_3 + D_3 + D_{311}\theta_1^2 + D_{322}\theta_2^2 \quad \ldots 23$$

ภาคผนวก ค
การจำลองแบบของแขนกลด้วยแบบจำลองไดนามิก

จากสมการของลากรานจ์ในภาคผนวก ข.มีรูปแบบทั่วไปเป็น

$$F_i = \Sigma D_{ij}\theta_i + Ia_i\theta_i + \Sigma\Sigma D_{ijk}\theta_j\theta_k + D_i \quad \ldots ค-1$$

กำหนดให้ $x_1 = \theta_1 \quad x_3 = \theta_2 \quad x_5 = \theta_3$ $\ldots ค-2$

$x_2 = x_1 \quad x_4 = x_3 \quad x_6 = x_5$ $\ldots ค-3$

แทนสมการ(ค-2),(ค-3)ลงในสมการ ข.21-23 ได้

$$F_1 = D_{11}x_2 + D_{21}x_4 + D_{31}x_6 + Ia_1x_2 + D_1 + D_{112}x_2x_4 + D_{113}x_2x_6 + D_{122}x_{42} + D_{123}x_4x_6 + D_{133}x_{62} \quad \ldots ค-4$$

$$F_2 = D_{21}x_2 + D_{22}x_4 + D_{32}x_6 + Ia_2x_4 + D_2 + D_{223}x_4x_6 + D_{233}x_{62} \quad \ldots ค-5$$

$$F_3 = D_{31}x_2 + D_{32}x_4 + D_{33}x_6 + Ia_3x_6 + D_3 + D_{311}x_{22} + D_{322}x_{42} \quad \ldots ค-6$$

การจำลองแบบต้องการให้สมการมีรูปแบบเป็น $\dot{x} = f(x,u,t)$ หรือ $\dot{X} = A(t)X + B(t)u$ โดยที่ u เป็นฟังก์ชั่นของแรงที่กระทำต่อระบบจึงสามารถจัดรูปสมการได้ดังนี้

(ค-4) x D_{32}

$$D_{32}F_1 = (D_{32}D_{11} + D_{32}Ia_1)x_2 + D_{32}D_{21}x_4 + D_{32}D_{31}x_6 + D_{32}D_1 + D_{32}D_{112}x_2x_4 + D_{32}D_{113}x_2x_6 + D_{32}D_{122}x_{42} + D_{32}D_{123}x_4x_6 + D_{32}D_{133}x_{62} \quad \ldots ค-7$$

(ค-6) x D_{21}

$$D_{21}F_3 = D_{31}D_{21}x_2 + D_{32}D_{21}x_4 + (D_{33}D_{21} + D_{21}Ia_3)x_6 + D_3D_{21} + D_{21}D_{311}x_{22} + D_{21}D_{322}x_{42} \quad \ldots ค-8$$

(ค-7)-(ค-8)

$$D_{32}F_1 - D_{21}F_3 = (D_{32}D_{11} - D_{32}Ia_1 - D_{31}D_{21})x_2 + (D_{32}D_{31} - D_{33}D_{21} - D_{21}Ia_3)x_6 + D_{32}D_1 - D_{21}D_3 - D_{21}D_{311}x_{22} + D_{32}D_{112}x_2x_4 + D_{32}D_{113}x_2x_6 + (D_{32}D_{122} - D_{21}D_{322})x_{42} + D_{32}D_{123}x_4x_6 + D_{32}D_{133}x_{62} \quad \ldots ค-9$$

หรือเขียนใหม่ได้

$$D_{32}F_1 - D_{21}F_3 = -Bx_2 + Mx_6 + D_{32}D_1 - D_{21}D_3 - D_{21}D_{311}x_{22} + D_{32}D_{112}x_2x_4 + D_{32}D_{113}x_2x_6 + Sx_{42} + D_{32}D_{123}x_4x_6 + D_{32}D_{133}x_{62} \quad \ldots ค-10$$

$$B = D_{31}D_{21} - D_{32}D_{11} - D_{32}Ia_1$$

$$M = D_{32}D_{31} - D_{33}D_{21} - D_{21}Ia_3$$

$$S = D_{32}D_{122} - D_{21}D_{322}$$

(ค-5) x D_{32},

$$D_{32}F_2 = D_{32}D_{21}x_2+(D_{32}D_{22}+D_{32}Ia_2)x_4+D_{32}D_{32}x_6+D_{32}D_2+D_{32}D_{211}x_{22}+ D_{32}D_{223}x_4x_6+D_{32}D_{233}x_{62} \quad \text{...ค-11}$$

(ค-6) x $(D_{22}+Ia_2)$

$$(D_{22}+Ia_2)F_3 = D_{31}(D_{22}+Ia_2)x_2+D_{32}(D_{22}+Ia_2)x_4+(D_{33}D_{22}+D_{22}Ia_3+ D_{33}Ia_2+Ia_3Ia_2)x_6+(D_{22}+Ia_2)D_3+(D_{22}D_{311}+Ia_2D_{311})x_{22} +(D_{22}D_{322}+Ia_2D_{322})x_{42} \quad \text{...ค-12}$$

(ค-11)-(ค-12)

$$D_{32}F_2-(D_{22}+Ia_2)F_3 = -Ax_2+Cx_6+D_{32}D_2-(D_{22}+Ia_2)D_3+Ex_{22}+Lx_{42}+ D_{32}D_{223}x_4x_6+D_{32}D_{233}x_{62} \quad \text{...ค-13}$$

$$A = D_{31}D_{22}+D_{31}Ia_2-D_{32}D_{21}$$

$$C = D_{32}D_{32}-D_{33}D_{22}-D_{22}Ia_3-D_{33}Ia_2-Ia_3Ia_2$$

$$E = D_{32}D_{211}-D_{22}D_{311}-Ia_2D_{311}$$

$$L = -D_{22}D_{322}-Ia_2D_{322}$$

(ค-10) x C

$$D_{32}CF_1-D_{21}CF_3 = -BCx_2+MCx_6+D_{32}CD_1-D_{21}CD_3-D_{21}D_{311}Cx_{22}+ D_{32}D_{112}Cx_2x_4+D_{32}D_{113}Cx_2x_6+SCx_{42}+ D_{32}D_{123}Cx_4x_6+D_{32}D_{133}Cx_{62} \quad \text{...ค-14}$$

(ค-13) x M

$$D_{32}MF_2-(D_{22}+Ia_2)MF_3 = -AMx_{22}+MCx_6+D_{32}MD_2-(D_{22}+Ia_2)MD_3+EMx_{22}+ LMx_{42}+D_{32}D_{223}Mx_4x_6+D_{32}D_{233}Mx_{62} \quad \text{...ค-15}$$

(ค-14)-(ค-15)

$$D_{32}CF_1-D_{32}MF_2+[(D_{22}+Ia_2)M-D_{21}C]F_3 = (AM-BC)x_2+D_{32}CD_1-D_{32}MD_2+ [(D_{22}+Ia_2)M-D_{21}C]D_3-(D_{21}D_{311}C-EM)x_{22}+D_{32}D_{112}Cx_2x_4+ D_{32}D_{113}Cx_2x_6+(SC-LM)x_{42}+(D_{32}D_{123}C-D_{32}D_{223}M)x_4x_6+ (D_{32}D_{133}C-D_{32}D_{233}M)x_{62} \quad \text{...ค.16}$$

จัดรูปใหม่ได้

$$x_2 = VA_2x_2^2+VB_2x_2x_4+VC_2x_2x_6+VD_2x_4^2+VE_2x_4x_6+VF_2x_6^2+UA_2(F_1-D_1)+ UB_2(F_2-D_2)+UC_2(F_3-D_3) \quad \text{...ค.17}$$

$VH_2 = AM-BC$ $\qquad$ $UA_2 = D_{32}C/VH_2$

$VA_2 = (D_{21}D_{311}C-EM)/VH_2$ $\qquad$ $UB_2 = -D_{32}M/VH_2$

$VB_2 = -D_{32}D_{112}C/VH_2$ $\qquad$ $UC_2 = -[(D_{22}+Ia_2)M-D_{21}C]/VH_2$

$VC_2 = -D_{32}D_{113}C/VH_2$

$VD_2 = -(SC-LM)/VH_2$

$VE_2 = -(D_{32}D_{123}C-D_{32}D_{223}M)/VH_2$

$VF_2 = -(D_{32}D_{133}C-D_{32}D_{233}M)/VH_2$

(ค-5) x D_{31}

$$D_{31}F_2 = D_{21}D_{31}x_2+(D_{22}D_{31}+D_{31}Ia_2)x_4+D_{32}D_{31}x_6+D_{31}D_2+D_{31}D_{211}x_{22}+ D_{31}D_{223}x_4x_6+D_{31}D_{233}x_{62} \quad \text{...ค-18}$$

(ค-7)-(ค-18)

$$D_{32}F_1-D_{31}F_2 = -Bx_2-Ax_4+D_{32}D_1-D_{31}D_2-D_{31}D211_{x22}+D_{32}D_{112}x_2x_4+ D_{32}D_{113}x_2x_6+D_{32}D_{122}x_{42}+Px_4x_6+Qx_{62} \quad \text{...ค-19}$$

$$P = D_{32}D_{123}-D_{31}D_{223}$$

$$Q = D_{32}D_{133}-D_{31}D_{233}$$

(ค-5) x $(D_{33}+Ia_3)$

$$(D_{33}+Ia_3)F_2 = (D_{33}+Ia_3)D_{21}x_2+(D_{33}D_{22}+D_{22}Ia_3+D_{33}Ia_2+Ia_2Ia_3)x_4+ (D_{33}+Ia_3)D_{32}x_6+(D_{33}Ia_3)D_2+(D_{33}+Ia_3)D_{211}x_{22}+ (D_{33}+Ia_3)D_{223}x_4x_6+(D_{33}+Ia_3)D_{233}x_{62} \quad \text{...ค-20}$$

(ค-6) x D_{32}

$$D_{32}F_3 = D_{32}D_{31}x_2+D_{32}D_{32}x_4+(D_{33}+Ia_3)D_{32}x_6+D_{32}D_3+D_{32}D_{311}x_{22}+ D_{32}D_{322}x_{42} \quad \text{...ค-21}$$

(ค-20)-(ค-21)

$$(D_{33}+Ia_3)F_2-D_{32}F_3 = -Mx_2-Cx_4+(D_{33}+Ia_3)D_2-D_{32}D_3+Rx_{22}-D_{32}D_{322}x_{42} +Tx_4x_6+Ox_{62} \quad \text{...ค-22}$$

$$R = D_{33}D_{211}+D_{211}Ia_3-D_{32}D_{311}$$

$$T = D_{33}D_{223}+D_{223}Ia_3$$

$$O = D_{33}D_{233}+D_{233}Ia_3$$

(ค-19) x M

$$D_{32}MF_1-D_{31}MF_2 = -BMx_2-AMx_4+D_{32}MD_1-D_{31}MD_2-D_{31}D_{211}Mx_{22}+ D_{32}D_{112}Mx_2x_4+D_{32}D_{113}Mx_2x_6+D_{32}D_{122}Mx_{42}+ PMx_4x_6+QMx_{62} \quad \text{...ค-23}$$

(ค-22) x B

$$(D_{33}+Ia_3)BF_2-D_{32}BF_3 = -BMx_2-CBx_4+(D_{33}+Ia_3)BD_2-D_{32}BD_3+RBx_{22}- D_{32}D_{322}Bx_{42}+TBx_4x_6+OBx_{62} \quad \text{...ค-24}$$

(ค-23)-(ค-24) แล้วจัดสมการในทำนองเดียวกับสมการ (ค-16),(ค-17)

$$x_4 = VA_4x_2^2+VB_4x_2x_4+VC_4x_2x_6+VD_4x_4^2+VE_4x_4x_6+VF_4x_6^2+UA_4(F_1-D_1)+$$
$$UB_4(F_2-D_2)+UC_4(F_3-D_3) \qquad \text{...ค-25}$$

$VH_4 = BC-AM$ $\qquad UA_4 = D_{32}M/VH_4$

$VA_4 = (D_{31}D_{211}M-RB)/VH_4$ $\qquad UB_4 = -[(D_{33}+Ia_3)+D_{31}M]/VH_4$

$VB_4 = -D_{32}D_{112}M/VH_4$ $\qquad UC_4 = D_{32}B/VH_4$

$VC_4 = -D_{32}D_{113}M/VH_4$

$VD_4 = -(D_{32}D_{122}M+D_{32}D_{322}B)/VH_4$

$VE_4 = -(PM-TB)/VH_4$

$VF_4 = (OB-QM)/VH_4$

(ค-10) x A

$$D_{32}AF_1-D_{21}AF_3 = -ABx_2+MAx_6+D_{32}AD_1-D_{21}AD_3-D_{21}D_{311}Ax_{22}+$$
$$D_{32}D_{112}Ax_2x_4+D_{32}D_{113}Ax_2x_6+SAx_{42}+$$
$$D_{32}D_{123}Ax_4x_6+D_{32}D_{133}Ax_{62} \qquad \text{...ค-26}$$

(ค-13) x B

$$D_{32}BF_2-(D_{22}+Ia_2)BF_3 = -ABx_2+CBx_6+D_{32}BD_2-(D_{22}+Ia_2)BD_3+EBx_{22}+$$
$$LBx_{42}+D_{32}D_{223}Bx_4x_6+D_{32}D_{233}Bx_{62} \qquad \text{...ค-27}$$

(ค-26)-(ค-27) แล้วจัดสมการในทำนองเดียวกับสมการ (ค-16),(ค-17)

$$x_6 = VA_6x_2^2+VB_6x_2x_4+VC_6x_2x_6+VD_6x_4^2+VE_6x_4x_6+VF_6x_6^2+UA_6(F_1-D_1)+$$
$$UB_6(F_2-D_2)+UC_6(F_3-D_3) \qquad \text{...ค-28}$$

$VH_6 = MA-BC$ $\qquad UA_6 = D_{32}A/VH_6$

$VA_6 = (D_{21}D_{311}A-EB)/VH_6$ $\qquad UB_4 = -D_{32}B/VH_6$

$VB_6 = -D_{32}D_{112}A/VH_6$ $\qquad UC_4 = [(D_{22}+Ia_2)B-D_{21}A]/VH_6$

$VC_6 = -D_{32}D_{113}A/VH_6$

$VD_6 = -(SA+LB)/VH_6$

$VE_6 = -(D_{32}D_{223}B-D_{32}D_{123}A)/VH_6$

$VF_6 = -(D_{32}D_{233}B-D_{32}D_{133}A)/VH_6$

จากสมการ(ค-3),(ค-18),(ค-29) และ(ค-33)จะได้ความสัมพันธ์ของ dynamic Model เป็นสมการ non-linear ของอนุพันธ์อันดับที่ 1 ที่มีความซับซ้อน ดังนั้นการหาคำตอบให้กับสมการเหล่านี้จะอาศัยคอมพิวเตอร์เข้าช่วยในการอินทิเกรทด้วยวิธีทาง numerical ซึ่งได้แสดงวิธีการของ Runge-Kutta ไว้ในภาคผนวก ง.

ภาคผนวก ง.

Runge-Kutta Method

Fifth-order Runge-kutta method เป็นวิธีการทาง numerical ที่ใช้กันอย่างแพร่หลายในการหาคำตอบให้กับสมการอนุพันธ์ เพราะว่าวิธีนี้สามารถหาคำตอบได้โดยอาศัยข้อมูล ณ ตำแหน่งเริ่มต้นหรือตำแหน่งปัจจุบันเพื่อหาคำตอบของตำแหน่งต่อไป ไม่ต้องการข้อมูลของตำแหน่งก่อนหน้า(Self-start) หรือหาค่า y_{i+1} ได้เมื่อทราบค่าของ y_i โดยหาคำตอบทีละลำดับขั้น(single step)ซึ่งทำให้ง่ายต่อการเขียนโปรแกรมคำนวณด้วยเครื่องคอมพิวเตอร์ แต่ก็มีข้อเสียที่ต้องหาคำตอบตามความสัมพันธ์ของฟังก์ชั่น f(x,y) ในทุกๆครั้ง เพื่อจะได้คำตอบของขั้นต่อๆไป ซึ่งทำให้ต้องใช้เวลาในการคำนวณมาก แต่ถ้าเวลาไม่ใช่สิ่งสำคัญมากแล้ววิธีนี้เป็นวิธีที่ดีมีรูปแบบดังสมการ 1 และ 2

$$y' = f(x,y) \qquad \ldots 1$$

$$y_{i+1} = y_i + 1/90(7k_1+32k_3+12k_4+32k_5+7k_6) \qquad \ldots 2$$

เมื่อ

$$k_1 = (h)f(x_i, y_i)$$

$$k_2 = (h)f(x_i+h/4, y_i+k_1/4)$$

$$k_3 = (h)f(x_i+h/4, y_i+k_1/8+k_2/8)$$

$$k_4 = (h)f(x_i+h/2, y_i+k_2/2+k_3)$$

$$k_5 = (h)f(x_i+3h/4, y_i+3k_1/16+9k_4/16)$$

$$k_6 = (h)f(x_i+h, y_i+(3k_1+2k_2+12k_3+12k_4+8k_5)/7)$$

h เป็นช่วงของการเพิ่มค่าของ x (step size)

การคำนวณด้วยเครื่องคอมพิวเตอร์สามารถเขียนเป็นโปรแกรมย่อยด้วยภาษา C ได้ดังรายละเอียดตามภาคผนวก จ.

ภาคผนวก จ.

รายละเอียดโปรแกรม

โปรแกรมทั้งหมดเขียนด้วยภาษา C โดยใช้คอมไพลเลอร์ ของบริษัทคอมพิวเตอร์ อินโนเวชั่น จำกัด Optimizing C86 เวอร์ชั่น 2.20J ซึ่งมีรายละเอียดของโปรแกรมดังต่อไปนี้

โปรแกรมการจำลองแบบระบบหุ่นยนต์อุตสาหกรรม

```
/* program simulation  14:50:15  11/14/1987  */

#include "stdio.h"

float smp,vmx,amx,Kp[3],Ki[3],Kd[3],data[500][16];

main()
{
int chk,i;

chk = 0;
do {
     edit_gain(&chk);
     ctrl_sim(&chk);
     graph_menu(chk);
     crt_mode(2);
     puts(" DO YOU WANT TEST CONTINUE ?  ");
     do { i = key_getc();
        } while(i != 5497 && i != 12654);
     chk = 1;
   } while(i != 12654);
crt_mode(2);
puts(" good bye ! ");
}

       /******* generate reference signal *******/
refsnl (zs,zf,nl)

float zs[3],zf[3];
int *nl;

{
extern float smp,amx,vmx,sqrt(),fabs(),data[][16];
float ttovm,ztovm,dz[3],Tam[3],Tvm[3],tt[3],Tofm,tts[3],
      tmv[3],zvc[3],tvc[3],time,tme;
int i,l,md[3];

crt_srcp(1,64,0);
puts(" No. ref. ");
Tofm = 0;
ttovm = 0;
```

```
            /* find dz,dt start to vmax */
      while (amx*ttovm < vmx)
            { ztovm = amx*ttovm*ttovm/2.0;
              ttovm += 0.001;
            };
for (i = 0; i < 3; i++)
    {
      dz[i] = fabs(zf[i]-zs[i]);
      Tam[i] = 4.0*dz[i]/amx;
      Tam[i] = sqrt(Tam[i]);
      Tvm[i] = 2.0*dz[i]/vmx;
       /**** set condition to mode of motion ****/
      if (Tvm[i] <= Tam[i])
         {
         md[i] = 1;
         tt[i] = Tam[i];
         tmv[i] = tt[i]/2.0;
         zvc[i] = tvc[i] = 0;
         }
      else
         {
         md[i] = 0;
         tt[i] = ttovm*2.0+(dz[i]-2.0*ztovm)/vmx;
         tmv[i] = vmx/amx;
         ztovm = amx*tmv[i]*tmv[i]/2.0;
         zvc[i] = dz[i]-2.0*ztovm;
         tvc[i] = zvc[i]/vmx;
         };
       /**** find time of motion ****/
      if (Tofm < tt[i])
         Tofm = tt[i];
    };

       /**** create input.ref file ****/
    time = 0;    l = 0;
while (time <= Tofm+5.0*smp)
      {
for(i = 0; i < 3; i++)
   {
     if (time <= tmv[i])
     {
       if (zs[i] <= zf[i])
       {
       data[l][i+1] = zs[i]+amx*time*time/2.0;
       data[l][i+7] = amx*time;
       }
       else
       {
       data[l][i+1] = zs[i]-amx*time*time/2.0;
       data[l][i+7] = -amx*time;
       };
     }
     else if (time <= tmv[i]+tvc[i] && md[i] == 0)
          {
            if (zs[i] <= zf[i])
            {
            data[l][i+1] = zs[i]+((time-tmv[i])*vmx+ztovm);
            data[l][i+7] = vmx;
            }
            else
```

```
            {
            data[1][i+1] = zs[i]-((time-tmv[i])*vmx+ztovm);
            data[1][i+7] = -vmx;
            };
          }
          else if (time <= tt[i])
              {
                tts[i] = tt[i]-tvc[i];
                tme = time-tvc[i];
                if (zs[i] <= zf[i])
                {
                data[1][i+1] = zs[i]+(amx*(tts[i]*tme-
                   tme*tme/2.0-tts[i]*tts[i]/4.0)+zvc[i]);
                data[1][i+7] = amx*(tts[i]-tme);
                }
                else
                {
                tts[i] = tt[i]-tvc[i];
                tme = time-tvc[i];
                data[1][i+1] = zs[i]-(amx*(tts[i]*tme-
                   tme*tme/2.0-tts[i]*tts[i]/4.0)+zvc[i]);
                data[1][i+7] = -amx*(tts[i]-tme);
                };
              }
              else
              {
                data[1][i+1] = zf[i];
                data[1][i+7] = 0.0;
              };
   };  /* end of for */
        crt_srcp(1,74,0);
        printf(" %4d ",1);
        time += smp;  1 += 1;
      };  /* end of while */
      *n1 = 1-1;
}

/* control loop with P I D controller */
float ca[4],sa[4],d[4],a[4],xb[4],yb[4];
float Ixx[4],Iyy[4],Izz[4],Ixy[4],Ixz[4];
float zb[4],ms[4],Iyz[4],Jzz[4];
float ml,motr,Ia1,Ia2,Ia3;

ctrl_sim(chk)
int *chk;

{

extern float smp,vmx,amx,Kp[],Ki[],Kd[],data[][16];
float pi = 3.1415927;
int i,l,lnk,nl,n[3];
float zs[3],f[4],time,err[3],itgerr[3],Km[3],xact[7];

        /*  spec. of motor  */
   n[0] = 90;   n[1] = 90;     n[2] = 100;
   Km[0] = 0.0741;    Km[1] = 0.0741;     Km[2] = 0.05;   /*

  nl = *chk;
  itgerr[0]=itgerr[1]=itgerr[2]=0;
  time = 0; l = 0;
```

```
   for(l=0;l<3;l++)
      { xact[2*l+1] = data[0][l+1];
        xact[2*l+2] = data[0][l+7];
      }
      /*** control loop ***/
for(l=0;l<=nl-1;l++) /* No. of cycle */
   { crt_srcp(1,74,0);
     printf(" %4d ",l);
     data[l][0] = time;
     for(lnk=0;lnk<3;lnk++)
        {
          err[lnk]=data[l+1][lnk+1]-data[l][lnk+4];
          itgerr[lnk] += smp*err[lnk];
          f[lnk+1] = Kp[lnk]*err[lnk]+Ki[lnk]*itgerr[lnk]+
                                              Kd[lnk]*data[l][lnk
          f[lnk+1] = Km[lnk]*f[lnk+1]*n[lnk]/2.0; /* assume
          data[l][lnk+13] = f[lnk+1];
        } /* end of each link */
     xact[0] =0;    f[0] = 0;
     romech(6,&time,xact,smp,f);
     for(lnk=0;lnk<3;lnk++)
        { data[l+1][lnk+4] = xact[2*lnk+1] ;
          data[l+1][lnk+10] = xact[2*lnk+2];
        }
    } /* end of control loop */
data[nl][0] = time;
for(lnk=0;lnk<3;lnk++)
   zs[lnk] = data[nl][lnk+4]*57.2957795;
   crt_srcp(18,31,0);
   printf("%7.2f",zs[0]);
   crt_srcp(18,46,0);
   printf("%7.2f",zs[1]);
   crt_srcp(18,61,0);
   printf("%7.2f",zs[2]);
putchar('\007');

crt_srcp(22,25,0);
puts(" PRESS ANY KEY TO CONTINUE ");
i = key_getc();

}
   /******* Runge-Kutta sixth-order *******/
#define NofE 6

romech (nofe,x,y,h,f)

int nofe;
float *x,y[],f[],h;

{
  int i;
  float k1[NofE+1],k2[NofE+1],k3[NofE+1],k4[NofE+1],xfcn;
  float k5[NofE+1],k6[NofE+1],ydot[NofE+1],yfcn[NofE+1];

k1[0] = k2[0] = k3[0] = 0;
k4[0] = k5[0] = k6[0] = 0;
ydot[0] = yfcn[0] = 0;
  for (i = 1; i < nofe+1; i++)
      yfcn[i]=y[i];
    xfcn=*x;
    fcn (nofe,&xfcn,yfcn,ydot,f);
```

```
  for (i = 1; i < nofe+1; i++)
      k1[i]=h*ydot[i];
  for (i = 1; i < nofe+1; i++)
      yfcn[i]=y[i]+k1[i]/4.0;
    xfcn=*x+h/4.0;
    fcn (nofe,&xfcn,yfcn,ydot,f);
  for (i = 1; i < nofe+1; i++)
      k2[i] = h*ydot[i];
  for (i = 1; i < nofe+1; i++)
      yfcn[i]=y[i]+k1[i]/8.0+k2[i]/8.0;
    xfcn=*x+h/4.0;
    fcn (nofe,&xfcn,yfcn,ydot,f);
  for (i = 1; i < nofe+1; i++)
      k3[i]=h*ydot[i];
  for (i = 1; i < nofe+1; i++)
      yfcn[i]=y[i]-k2[i]/2.0+k3[i];
    xfcn=*x+h/2.0;
    fcn (nofe,&xfcn,yfcn,ydot,f);
  for (i = 1; i < nofe+1; i++)
      k4[i]=h*ydot[i];
  for (i = 1; i < nofe+1; i++)
      yfcn[i]=y[i]+k1[i]*3.0/16.0+k4[i]*9.0/16.0;
    xfcn=*x+h*0.75;
    fcn (nofe,&xfcn,yfcn,ydot,f);
  for (i = 1; i < nofe+1; i++)
      k5[i]=h*ydot[i];
  for (i = 1; i < nofe+1; i++)
      yfcn[i]=y[i]-(k1[i]*3.0+k2[i]*2.0+k3[i]*12.0-
              k4[i]*12.0+k5[i]*8.0)/7.0;
    xfcn=*x+h;
    fcn (nofe,&xfcn,yfcn,ydot,f);
  for (i = 1; i < nofe+1; i++)
      k6[i]=h*ydot[i];
  for (i = 1; i < nofe+1; i++)
      y[i]=y[i]+(7.0*k1[i]+32.0*k3[i]+12.0*k4[i]+32.0*k5[i]
           +7.0*k6[i])/90.0;
      *x=*x+h;
}

/*  complex form of fcn()  */

#define ROW 4
#define CLM 4
#define N   3

fcn (nofe,time,xfcn,xdot,f)
int nofe;
float *time,xfcn[],xdot[],f[];

{

extern int fclose();
extern float sin(),cos();
extern float ca[N+1],sa[N+1],d[N+1],a[N+1],xb[N+1],yb[N+1];
extern float Ixx[N+1],Iyy[N+1],Izz[N+1],Ixy[N+1],Ixz[N+1];
extern float zb[N+1],ms[N+1],Iyz[N+1],Jzz[N+1];
extern float ml,motr,Ia1,Ia2,Ia3;
float A[N][ROW][CLM],T[N][N+1][ROW][CLM],g;
float TQ[N][N][ROW][CLM],Q[ROW][CLM],sz[N+1],cz[N+1];
```

```
float U[N][N+1][N+1][ROW][CLM],D[N+1][N+1];
float UC[N][N+1][N+1][N+1][ROW][CLM];
float J[N+1][ROW][CLM],JU[N][N+1][N+1][ROW][CLM],rtr;
float R[N+1][ROW],gUR,G[N+1],C[N+1][N+1][N+1],gU[ROW];
float xbn[N+1],ybn[N+1],zbn[N+1],Jxx[N+1],Jyy[N+1];
float ha[N+1][N+1],la[N+1][N+1],pa[N+1][N+1],qa[N+1][N+1];
float VF[N+1][N+1][N+1],UF[N+1][N+1],wa,wb,wc,wm;
int i,j,k,l,p,m,maxij,minij,maxijk,minijk;
FILE *spc;

      /**** open file for read data ****/
spc = NULL;
if (*time == 0)
   { if ((spc=fopen("spec.dat","r"))==NULL)
        { printf("cannot open spec.dat \n");
          exit(1);
        }
for (i=1; i < N+1; i++)       /*** read sin(alpha) ***/
    fscanf(spc," %f ",&sa[i]);
for (i=1; i < N+1; i++)       /*** read cos(alpha) ***/
    fscanf(spc," %f ",&ca[i]);
for (i=1; i < N+1; i++)       /*** read d(i) ***/
    fscanf(spc," %f ",&d[i]);
for (i=1; i < N+1; i++)       /*** read a(i) ***/
    fscanf(spc," %f ",&a[i]);
for (i=1; i < N+1; i++)       /*** read mass(i) ***/
    fscanf(spc," %f ",&ms[i]);
for (i=1; i < N+1; i++)       /*** read CG.of x-axis ***/
    fscanf(spc," %f ",&xb[i]);
for (i=1; i < N+1; i++)       /*** read CG.of y-axis ***/
    fscanf(spc," %f ",&yb[i]);
for (i=1; i < N+1; i++)       /*** read CG.of z-axis ***/
    fscanf(spc," %f ",&zb[i]);
for (i=1; i < N+1; i++)       /*** read Ixx ***/
    fscanf(spc," %f ",&Ixx[i]);
for (i=1; i < N+1; i++)       /*** read Iyy ***/
    fscanf(spc," %f ",&Iyy[i]);
for (i=1; i < N+1; i++)       /*** read Izz ***/
    fscanf(spc," %f ",&Izz[i]);
fscanf(spc,"%f %f %f",&Ia1,&Ia2,&Ia3); /* gear inertia */
fscanf(spc,"%f %f ",&motr,&ml);  /* motor mass payload */

    /**** move cg. for payload ****/
xbn[3] = ms[3]*xb[3]/(ms[3]+ml);
ybn[3] = ms[3]*yb[3]/(ms[3]+ml);
zbn[3] = ms[3]*zb[3]/(ms[3]+ml);
Ixx[3]=Ixx[3]+ms[3]*((ybn[3]-yb[3])*(ybn[3]-yb[3])+(zbn[3]
 -zb[3])*(zbn[3]-zb[3]))+ml*(ybn[3]*ybn[3]+zbn[3]*zbn[3]);
Iyy[3]=Iyy[3]+ms[3]*((xbn[3]-xb[3])*(xbn[3]-xb[3])+(zbn[3]
 -zb[3])*(zbn[3]-zb[3]))+ml*(xbn[3]*xbn[3]+zbn[3]*zbn[3]);
Izz[3]=Izz[3]+ms[3]*((xbn[3]-xb[3])*(xbn[3]-xb[3])+(ybn[3]
 -yb[3])*(ybn[3]-yb[3]))+ml*(xbn[3]*xbn[3]+ybn[3]*ybn[3]);
ms[3] = ms[3]+ml;
xb[3] = xbn[3];
yb[3] = ybn[3];
zb[3] = zbn[3];
```

```
    /**** move cg. for motor mass ****/
xbn[2] = ms[2]*xb[2]/(ms[2]+motr);
ybn[2] = ms[2]*yb[2]/(ms[2]+motr);
zbn[2] = ms[2]*zb[2]/(ms[2]+motr);
Ixx[2]=Ixx[2]+ms[2]*((ybn[2]-yb[2])*(ybn[2]-yb[2])+(zbn[2]
-zb[2])*(zbn[2]-zb[2]))+motr*(ybn[2]*ybn[2]+zbn[2]*zbn[2]);
Iyy[2]=Iyy[2]+ms[2]*((xbn[2]-xb[2])*(xbn[2]-xb[2])+(zbn[2]
-zb[2])*(zbn[2]-zb[2]))+motr*(xbn[2]*xbn[2]+zbn[2]*zbn[2]);
Izz[2]=Izz[2]+ms[2]*((xbn[2]-xb[2])*(xbn[2]-xb[2])+(ybn[2]
-yb[2])*(ybn[2]-yb[2]))+motr*(xbn[2]*xbn[2]+ybn[2]*ybn[2]);
ms[2] = ms[2]+motr;
xb[2] = xbn[2];
yb[2] = ybn[2];
zb[2] = zbn[2];
for (i=1; i<N+1; i++)
    {  Jxx[i] = (-Ixx[i]+Iyy[i]+Izz[i])/2.0;
       Jyy[i] = (Ixx[i]-Iyy[i]+Izz[i])/2.0;
       Jzz[i] = (Ixx[i]+Iyy[i]-Izz[i])/2.0;
       Ixy[i] = Ixz[i] = Iyz[i] = 0;
    };
   } /* if time for open */
   if(spc) fclose(spc);
      /****** assemble g(1,4) *******/
    g = 9.81;
      /******** assemble A-matrices *********/
/*** A[0] = A1 ***/
for(i = 1; i < N+1; i++)
   { sz[i] = sin(xfcn[2*i-1]);  cz[i] = cos(xfcn[2*i-1]);
   A[i-1][0][0] = cz[i];       A[i-1][0][1] = -ca[i]*sz[i];
   A[i-1][0][2] = sa[i]*sz[i];  A[i-1][0][3] = a[i]*cz[i];
   A[i-1][1][0] = sz[i];       A[i-1][1][1] = ca[i]*cz[i];
   A[i-1][1][2] = -sa[i]*cz[i]; A[i-1][1][3] = a[i]*sz[i];
   A[i-1][2][0] = 0;            A[i-1][2][1] = sa[i];
   A[i-1][2][2] = ca[i];        A[i-1][2][3] = d[i];
   A[i-1][3][0] = A[i-1][3][1] = A[i-1][3][2] = 0;
   A[i-1][3][3] = 1;
    };
             /****** assemble ri(4) *******/
for (i = 1; i < N+1; i++)
    {  R[i][0] = xb[i];   R[i][1] = yb[i];
       R[i][2] = zb[i];   R[i][3] = 1;
    };
            /******* assemble J-matrix *******/
for (i = 1; i < N+1; i++)
    {
    J[i][0][0] = Jxx[i]; J[i][0][1] = J[i][1][0] = Ixy[i];
    J[i][1][1] = Jyy[i]; J[i][0][2] = J[i][2][0] = Ixz[i];
    J[i][2][2] = Jzz[i]; J[i][1][2] = J[i][2][1] = Iyz[i];
    J[i][3][3] = ms[i];
    J[i][0][3] = J[i][3][0] = ms[i]*xb[i];
    J[i][1][3] = J[i][3][1] = ms[i]*yb[i];
    J[i][2][3] = J[i][3][2] = ms[i]*zb[i];
    };
         /********* assemble I-matrix *********/
for (k = 0; k < N; k++)
    {  for (i = 0; i < 4; i++)
          {  for (j = 0; j < 4; j++)
                if (i == j)
                  T[k][k][i][j] = 1;
```

```
                    else
                      T[k][k][i][j] = 0;
                };
        };
            /********** assemble Q-matrix **********/
    for (i = 0; i < 4; i++)
            {  for (j = 0; j < 4; j++)
                    if (i == 0 && j == 1)
                       Q[i][j] = -1;
                    else { if (i == 1 && j == 0)
                              Q[i][j] = 1;
                           else
                              Q[i][j] = 0;
                         };
            };
            /********** compute T-matrices **********/
    for (p = 0; p < N; p++)
        {  for (k = p; k < N; k++)
              Mtxmul (T[p][k],A[k],4,4,4,T[p][k+1]);
        };
            /********** compute Uij-matrices **********/
     for (p = 0; p < N; p++)
        {   for (i = p; i < N; i++)
              Mtxmul (T[p][i],Q,4,4,4,TQ[p][i]);
        };

     for (p = 0; p < N; p++)
        { for (i = p+1; i < N+1; i++)
              { for (k = i; k < N+1; k++)
           Mtxmul (TQ[p][i-1],T[i-1][k],4,4,4,U[p][k][i]);
              };
        };
            /******** compute Uijk-matrices *********/
    for (p = 0; p < N; p++)
       { for (j = p+1; j < N+1; j++)
            { for (m = j; m < N+1; m++)
                 { for (k = m; k < N+1; k++)
                      {
      Mtxmul (TQ[p][j-1],U[j-1][k][m],4,4,4,UC[p][k][j][m]);
      if (j != m)
         for (i=0; i<ROW; i++)
            for (l=0; l<CLM; l++)
               UC[p][k][m][j][i][l] = UC[p][k][j][m][i][l];
                      };
                 };
            };
       };
            /******** compute Dij *********/
     for (p = 0; p < N; p++)
        { for (i = p+1; i < N+1; i++)
            { for (k = i; k < N+1; k++)
                 { Tsp (U[p][k][i]);
                Mtxmul (J[k],U[p][k][i],4,4,4,JU[p][k][i]);
                      Tsp (U[p][k][i]);
                 };
            };
        };
  for (i = 0; i < N+1; i++)
      for (j = 0; j < N+1; j++)
          D[i][j] = 0;
```

```
    for (i = 1; i < N+1; i++)
        { for (j = i; j < N+1; j++)
              { rtr = 0;
                if (i < j)  {  maxij = j;
                               minij = i;  }
                else  {  maxij = i;
                         minij = j;  };
                for (k = maxij; k < N+1; k++)
                    { p = minij-1;
                   Trace (U[p][k][j],JU[p][k][i],4,4,&rtr);
                      D[i][j] += rtr;
                    };
                D[j][i] = D[i][j];
              };
        };
             /******* compute Dijk ********/
for (i = 0; i < N+1; i++)
    for (j = 0; j < N+1; j++)
        for (k = 0; k < N+1; k++)
            C[i][j][k] = 0;
    for (i = 1; i < N+1; i++)
        { for (j = i; j < N+1; j++)
              {for (k = j; k < N+1; k++)
                    { rtr = 0;
                      if (i < j)  { maxij = j;
                                    minij = i;  }
                      else  { maxij = i;
                              minij = j;  };
                      if (maxij < k)  maxijk = k;
                         else  maxijk = maxij;
                      if (minij < k)  minijk = minij;
                         else  minijk = k;
                      for (m = maxijk; m < N+1; m++)
                          { p = minijk-1;
                Trace (UC[p][m][j][k],JU[p][m][i],4,4,&rtr);
                            C[i][j][k] += rtr;
                          };
                      C[i][k][j] = C[i][j][k];
                    };
              };
        };
             /****** compute Gi *******/
for (i=0; i<N+1; i++)
    G[i] = 0;
    for (i = 1; i < N+1; i++)
        { gUR = 0;
          for (k = i; k < N+1; k++)
              { for (j = 0; j < ROW; j++)
                    {
                      gU[j] = g*U[0][k][i][2][j];
                      gUR += gU[j]*R[k][j];
                    };
              G[i] += -ms[k]*gUR;
              };
         };
        /**** creat function xdot = f(xfcn,time) ****/
      wa = D[3][1]*D[2][2]+D[3][1]*Ia2-D[3][2]*D[2][1];
      wb = D[3][1]*D[2][1]-D[3][2]*D[1][1]-D[3][2]*Ia1;
      wc = D[3][2]*D[3][2]-D[3][3]*D[2][2]-D[2][2]*Ia3
           -D[3][3]*Ia2-Ia2*Ia3;
      wm = D[3][2]*D[3][1]-D[3][3]*D[2][1]-D[2][1]*Ia3;
```

```
for (i=1; i<N+1; i++)
    for (j=i; j<N+1; j++)
        {ha[i][j] = D[3][2]*C[1][i][j]-D[2][1]*C[3][i][j];
          if (i != j)
             ha[i][j] = 2.0*ha[i][j];
        };

for (i=1; i<N+1; i++)
    for (j=i; j<N+1; j++)
        {
   la[i][j] = D[3][2]*C[2][i][j]-(D[2][2]+Ia2)*C[3][i][j];
          if (i != j)
             la[i][j] = 2.0*la[i][j];
        };

for (i=1; i<N+1; i++)
    for (j=i; j<N+1; j++)
        {pa[i][j] = D[3][2]*C[1][i][j]-D[3][1]*C[2][i][j];
          if (i != j)
             pa[i][j] = 2.0*pa[i][j];
        };

for (i=1; i<N+1; i++)
    for (j=i; j<N+1; j++)
        {
   qa[i][j] = (D[3][3]+Ia3)*C[2][i][j]-D[3][2]*C[3][i][j];
          if (i != j)
             qa[i][j] = 2.0*qa[i][j];
        };

      VF[1][0][0] = wa*wm-wb*wc;
for(i=1; i<N+1; i++)
   for(j=i; j<N+1; j++)
      VF[1][i][j] = (ha[i][j]*wc-la[i][j]*wm)/VF[1][0][0];
     UF[1][1] = D[3][2]*wc/VF[1][0][0];
     UF[1][2] = -D[3][2]*wm/VF[1][0][0];
     UF[1][3] = ((D[2][2]+Ia2)*wm-D[2][1]*wc)/VF[1][0][0];

      VF[2][0][0] = wb*wc-wa*wm;
for (i=1; i<N+1; i++)
   for (j=i; j<N+1; j++)
      VF[2][i][j] = (pa[i][j]*wm-qa[i][j]*wb)/VF[2][0][0];
    UF[2][1] = D[3][2]*wm/VF[2][0][0];
    UF[2][2] = -((D[3][3]+Ia3)*wb+D[3][1]*wm)/VF[2][0][0];
    UF[2][3] = D[3][2]*wb/VF[2][0][0];

      VF[3][0][0] = wa*wm-wb*wc;
for (i=1; i<N+1; i++)
    for (j=i; j<N+1; j++)
      VF[3][i][j] = (ha[i][j]*wa-la[i][j]*wb)/VF[3][0][0];
     UF[3][1] = D[3][2]*wa/VF[3][0][0];
     UF[3][2] = -D[3][2]*wb/VF[3][0][0];
     UF[3][3] = ((D[2][2]+Ia2)*wb-D[2][1]*wa)/VF[3][0][0];

for (k=1; k<N+1; k++)
    { xdot[2*k-1] = 0;
      xdot[2*k] = 0;
      for (i=1; i<N+1; i++)
          { xdot[2*k] += UF[k][i]*(f[i]-G[i]);
            for (j=i; j<N+1; j++)
```

```
                xdot[2*k] += VF[k][i][j]*xfcn[2*i]*xfcn[2*j];
            };
                xdot[2*k-1] = xfcn[2*k];
    };
}
        /******** Transpose a(m,n)t ********/
Tsp (a)

float a[][CLM];

{
 int i,j;
 float temp[ROW][CLM];

  for (i = 0; i < CLM; i++)
      {  for (j = 0; j < ROW; j++)
             temp[i][j] = a[i][j];
      };
  for (i = 0; i < CLM; i++)
      {  for (j = 0; j < ROW; j++)
             a[i][j] = temp[j][i];
      };
}
          /******* a(m,n)*b(n,j) = c(m,j) ********/
Mtxmul (a,b,m,n,j,c)

int m, n, j;
float a[][CLM], b[][CLM], c[][CLM];

{
int clu, row, l;
float cij;

 for (clu = 0; clu < j ; clu++)
     {  for (row = 0; row < m; row++)
            {  cij = 0;
               for (l = 0; l < n; l++)
                   cij += a[row][l]*b[l][clu];
                   c[row][clu] = cij;
            };
     };
}
          /******** Tr[a(m,n)*b(n,m)] = tr ********/
Trace (a,b,m,n,tr)

int m, n;
float a[][CLM], b[][CLM], *tr;

{
int row, l;
float cij;

          *tr = 0;
     for (row = 0; row < m; row++)
         { cij = 0;
             for (l = 0; l < n; l++)
                 cij += a[row][l]*b[l][row];
           *tr += cij;
         };
}
```

โปรแกรมการควบคุมระบบหุ่นยนต์อุตสาหกรรม

```
        /*  model control  11:10:07  11/9/1987  */

#include "stdio.h"

float smp,vmx,amx,Kp[3],Ki[3],Kd[3],data[500][19],zeta[6]

main()
{
int chk,i;
extern float smp,vmx,amx,Kp[],Ki[],Kd[],data[][19],zeta[];
float pos[3];

chk = 1;
smp = .01;      vmx = 0.5;      amx = 0.25;
Kp[0] = 315;    Kp[1] = 585;    Kp[2] = 275;
Ki[0] = 145;    Ki[1] = 275;    Ki[2] = 125;
Kd[0] = 3;      Kd[1] = 7;      Kd[2] = 2;
zeta[0] = zeta[3] = 90;  zeta[1] = zeta[4] = 85;
zeta[2] = zeta[5] = 70;
printf("Enter maxout[0..2] :");
scanf(" %f %f %f",&maxout[0],&maxout[1],&maxout[2]);
do { zero();
     edit_gain(&chk);
     crt_srcp(22,25,0);
     printf(" MOVE ARM TO START POSITION ");
     for(i=0;i<3;i++)
        pos[i] = data[0][i+1];
     go_desire(pos);
     crt_srcp(22,25,0);
     printf(" MOVE ARM WITH THE PROGRAM  ");
     control(&chk);
     crt_srcp(22,25,0);
     printf("  MOVE ARM TO HOME POSITION ");
     pos[0] = 1.571;  pos[1] = 1.4835;  pos[2] = 1.2217;
     go_desire(pos);
     graph_menu(chk);
     zero();
     crt_mode(2);
     puts(" DO YOU WANT TEST CONTINUE ?  ");
     do { i = key_getc();
        } while(i != 5497 && i != 12654);
     chk = 1;
   } while(i != 12654);
crt_mode(2);
puts(" good bye ! ");

}
```

```
edit_gain(chck)

int *chck;

{

extern float smp,vmx,amx,Kp[],Ki[],Kd[],zeta[];
float zs[3],zf[3],val[13];
unsigned int pvlu;
int chk,nl,lnk,i,ii,iii;

chk = *chck;
crt_mode(2);
block2(0,0,23,79);
crt_srcp(1,20,0);
puts(" Test manipulator arm by PID controller ");
crt_srcp(4,10,0);
puts(" Moving data ");
crt_srcp(4,24,0);
puts(" vmx : ");
crt_srcp(4,31,0);
if(chk==0) scanf(" %f ",&vmx);
crt_srcp(4,31,0);
printf("%7.2f",vmx);
crt_srcp(4,39,0);
puts(" amx : ");
crt_srcp(4,46,0);
if(chk==0) scanf(" %f ",&amx);
crt_srcp(4,46,0);
printf("%7.2f",amx);
crt_srcp(4,54,0);
puts(" smp : ");
crt_srcp(4,61,0);
if(chk==0) scanf(" %f ",&smp);
crt_srcp(4,61,0);
if(chk!=0) smp = smp*1000;
printf("%7.2f",smp);
crt_srcp(7,25,0);
puts(" PID controller gain ");
crt_srcp(9,10,0);
puts(" Waist ");
crt_srcp(10,10,0);
puts(" Shoulder ");
crt_srcp(11,10,0);
puts(" Elbow ");
for(lnk=0;lnk<3;lnk++)
   {
   crt_srcp(9+lnk,25,0);
   puts(" Kp : ");
   crt_srcp(9+lnk,31,0);
   if(chk==0) scanf(" %f ",&Kp[lnk]);
   crt_srcp(9+lnk,31,0);
   printf("%7.2f",Kp[lnk]);
   crt_srcp(9+lnk,40,0);
   puts(" Ki : ");
   crt_srcp(9+lnk,46,0);
   if(chk==0) scanf(" %f ",&Ki[lnk]);
   crt_srcp(9+lnk,46,0);
   printf("%7.2f",Ki[lnk]);
   crt_srcp(9+lnk,55,0);
```

```
      puts(" Kd : ");
      crt_srcp(9+lnk,61,0);
      if(chk==0) scanf(" %f ",&Kd[lnk]);
      crt_srcp(9+lnk,61,0);
      printf("%7.2f",Kd[lnk]);
    }
  crt_srcp(22,4,0);
  printf(" Use      move cursor to change data          PRESS
  crt_srcp(22,9,0);
  printf("%c %c ",'\033','\032');
  val[0] = 0;
  for(lnk=0;lnk<3;lnk++)
     {  i = 4+lnk;  ii = 5+lnk;  iii = 6+lnk;
        val[i+2*lnk] = Kp[lnk];   val[ii+2*lnk] = Ki[lnk];
        val[iii+2*lnk] = Kd[lnk];
     }  val[1] = vmx;  val[2] = amx;  val[3] = smp;
  adj_data(val,1);
  for(lnk=0;lnk<3;lnk++)
     {  i = 4+lnk;  ii = 5+lnk;  iii = 6+lnk;
        Kp[lnk] = val[i+2*lnk];    Ki[lnk] = val[ii+2*lnk];
        Kd[lnk] = val[iii+2*lnk];
     }  vmx = val[1];  amx = val[2];  smp = val[3];
    smp = smp/1000;
  crt_srcp(22,4,0);
  printf("

  crt_srcp(22,25,0);
  puts(" INPUT POSITION TO MOVE ARM ");
  putchar('\007');
  crt_srcp(14,25,0);
  puts(" Position of motion (deg) ");
  crt_srcp(16,10,0);
  puts(" Move from ");
  crt_srcp(17,10,0);
  puts("         To ");
  crt_srcp(18,10,0);
  puts("   Stop at ");
  for(lnk=0;lnk<3;lnk++)
     {
     crt_srcp(16+lnk,25,0);
     puts(" wa : ");
     crt_srcp(16+lnk,31,0);
     if(lnk==0)  printf("  (>0)  ");
     if(lnk==1)  printf(" (<180) ");
     crt_srcp(16+lnk,40,0);
     puts(" sh : ");
     crt_srcp(16+lnk,46,0);
     if(lnk==0)  printf("  (>0)  ");
     if(lnk==1)  printf(" (<165) ");
     crt_srcp(16+lnk,55,0);
     puts(" el : ");
     crt_srcp(16+lnk,61,0);
     if(lnk==0)  printf("  (>0)  ");
     if(lnk==1)  printf(" (<145) ");
     }
  crt_srcp(16,31,0);
  if(chk==0) scanf(" %f ",&zeta[0]);
  if(zeta[0]<0 || zeta[0]>180)   { putchar('\007');    zeta[0]
  crt_srcp(16,31,0);
  printf("%7.2f",zeta[0]);
  crt_srcp(16,46,0);
```

```
if(chk==0) scanf(" %f ",&zeta[1]);
if(zeta[1]<0 || zeta[1]>165)
  { putchar('\007');    zeta[1] = 85; }
crt_srcp(16,46,0);
printf("%7.2f",zeta[1]);
crt_srcp(16,61,0);
if(chk==0) scanf(" %f ",&zeta[2]);
if(zeta[2]<0 || zeta[2]>145)
  { putchar('\007');    zeta[2] = 75; }
crt_srcp(16,61,0);
printf("%7.2f",zeta[2]);
crt_srcp(17,31,0);
if(chk==0) scanf(" %f ",&zeta[3]);
if(zeta[3]<0 || zeta[3]>180)
  { putchar('\007');   zeta[3] = zeta[0]; }
crt_srcp(17,31,0);
printf("%7.2f",zeta[3]);
crt_srcp(17,46,0);
if(chk==0) scanf(" %f ",&zeta[4]);
if(zeta[4]<0 || zeta[4]>165)
  { putchar('\007');   zeta[4] = zeta[1]; }
crt_srcp(17,46,0);
printf("%7.2f",zeta[4]);
crt_srcp(17,61,0);
if(chk==0) scanf(" %f ",&zeta[5]);
if(zeta[5]<0 || zeta[5]>165)
  { putchar('\007');   zeta[5] = zeta[2]; }
crt_srcp(17,61,0);
printf("%7.2f",zeta[5]);
crt_srcp(22,4,0);
printf(" Use      move cursor to change data          PRESS
crt_srcp(22,9,0);
printf("%c %c ",'\033','\032');
for(lnk=0;lnk<6;lnk++)
   val[1+lnk] = zeta[lnk];
adj_data(val,2);
for(lnk=0;lnk<6;lnk++)
   zeta[lnk] = val[lnk+1];
       /*  generate input singal   */
for (lnk=0;lnk<3;lnk++)
    {
    zs[lnk] = zeta[lnk]*0.017453292;  /* rad */
    zf[lnk] = zeta[lnk+3]*0.017453292;
    }
crt_srcp(22,4,0);
printf("
crt_srcp(22,25,0);
puts("    Creating Referance     ");
   refsnl (zs,zf,&nl);
   *chck = nl;
putchar('\007');
crt_srcp(22,25,0);
puts(" PRESS ANY KEY TO MOVE ARM ");
i = key_getc();
putchar('\007');

}

adj_data(value,sec)
   int sec;
   float value[];
```

```
    {

    extern key_getc();
    extern double atof();
    int    j,check,dachk,i,index,point,sect;
    char   s_data[15];
    double temp,old_temp,val[13];

sect = sec;
for(j=1;j<13;j++)
    val[j] = value[j];
    point  = 0;
    index = 1;
    i = 0;
    cursor_pos(index,sect);
    s_data[i] = '\0';
    old_temp = temp = val[index];
    do
     {
        check = key_getc();
         switch(check)
           {

            case 19200 :   /* left arrow */
               if (i==0)
                 val[index] = temp = old_temp ;
               cursor_pos(index,sect);
               printf("%7.2f",temp);
               if (index == 1 && sect == 1)   index = 13;
               if (index == 1 && sect == 2)   index = 7;
               index--;
               old_temp = temp = val[index];
               cursor_pos(index,sect);
               point = i = 0;
               break;
            case 19712 :   /* right arrow */
               if (i==0)
                 val[index] = temp = old_temp ;
               cursor_pos(index,sect);
               printf("%7.2f",temp);
               if (index == 12 && sect == 1)  index = 0;
               if (index == 6 && sect == 2)  index = 0;
               index++;
               old_temp = temp = val[index];
               cursor_pos(index,sect);
               point = i = 0;
               break;
            case 7181 :  /* Return */
               if (i==0)
                 val[index] = temp = old_temp ;
               cursor_pos(index,sect);
               printf("%7.2f",temp);
               cursor_pos(index,sect);
               old_temp = temp;
               point = i = 0;
               break;
           }
        dachk  = check << 8;
        dachk  = dachk >> 8;
        if (dachk <=57 && dachk >=48 || dachk == 46)
         {
           if (i == 0)
```

```
                {
                printf("         ");
                cursor_pos(index,sect);
                }
                 s_data[i] = dachk;
                 s_data[i+1] = '\0';
                 printf("%c",s_data[i]);
                 if (point > 0) point++;
                 i++;
                 temp = atof(s_data);
                 val[index] = temp;
          }

     } while(check != 14624);  /* Space */
     for(j=1;j<13;j++)
     value[j] = val[j];
 }

cursor_pos(index,sect)
int index,sect;
{
 int r,c,i,j,chk,idex,sec;

if (sect == 1)
    {
    if(index>0 && index<4)  r = 4;
    if(index>3 && index<7)  r = 9;
    if(index>6 && index<10) r = 10;
    if(index>9 && index<13) r = 11;
    for(j=1;j<=3;j++)
       { chk = j;
         for(i=1;i<=4;i++)
            { if(index == chk)   c = 16+15*j;
              chk += 3;
            }
       }
    }
else
    {  if(index>0 && index<4)  r = 16;
       if(index>3 && index<7)  r = 17;
    if(index == 1 || index == 4)   c = 31;
    if(index == 2 || index == 5)   c = 46;
    if(index == 3 || index == 6)   c = 61;
    }
    crt_srcp(r,c,0);
}



control(chk)
int *chk;

{

extern float fabs(),smp,vmx,amx,Kp[3],Ki[3],Kd[3],data[]
float pi = 3.1415927;
unsigned int pvlu,vvlu,dtoa,selc;
int dl,dly,i,l,lnk,n1,n[3],reset;
float zs[3],zf[3],time,verr[3],err[3],itgerr[3],opvo,Km[3]

zero();
```

```
   nl = *chk;  dly = (smp-8.355)/0.022 ;
   itgerr[0]=itgerr[1]=itgerr[2]=0;
   time = 0; l = 0;   reset = 1;
      /*** control loop ***/
for (l=0;l<nl;l++) /* No. of cycle */
    {   data[l][0] = time;
       for(dl=0;dl<=dly;dl++); /* for delay */
    for (lnk=0;lnk<3;lnk++)
        {
        in(13+lnk,&pvlu);  /* read actual position (rad) */
        if(lnk == 0) data[l][4] = 0.001369*pvlu-1.110758 ;
        if(lnk == 1) data[l][5] = 4.140573-0.001367*pvlu ;
        if(lnk == 2) data[l][6] = 4.954108-0.001376*pvlu ;
        in(0+lnk,&vvlu);  /* read actual velocity (rad/sec)
        if(lnk == 0) data[l][10] = 0.002268*vvlu-4.766;
        if(lnk == 1) data[l][11] = 0.002268*vvlu-4.714;
        if(lnk == 2) data[l][12] = 0.002751*vvlu-6.000;
        verr[lnk]=data[l+1][lnk+7]-data[l][lnk+10];
        err[lnk]=data[l+1][lnk+1]-data[l][lnk+4];
        itgerr[lnk] += smp*err[lnk];
        if(data[l+1][lnk+1] == data[l][lnk+1] && reset == 1)
          { itgerr[lnk] = 0;  reset = 0; }
        data[l][lnk+13]=Kp[lnk]*err[lnk]+Ki[lnk]*itgerr[lnk]
                        +Kd[lnk]*verr[lnk];
        opvo = data[l][lnk+13];
        if(lnk == 1) dtoa = opvo*57.365913+2057.54;   /* amp
        if(lnk == 0) dtoa = opvo*69.209770+2053.23;   /* amp
        if(lnk == 2) dtoa = opvo*69.209770+2053.23;   /* amp
        if(dtoa < 0) dtoa = 5;
        if(dtoa > 4095) dtoa = 4090;
        selc = 4 >> 2-lnk ;
        out(selc,dtoa);
        } /* end of each link */
      time += smp;
    } /* end of control loop */
zero();
putchar('\007');
 /** read stop position (a/d ch13-15) **/
   in(13,&pvlu);  /* waist */
    zs[0] = 0.078429*pvlu-63.641750 ;
   in(14,&pvlu);  /* shoulder */
    zs[1] = 237.237335-0.078343*pvlu ;
   in(15,&pvlu);  /* elbow */
    zs[2] = 283.849487-0.078855*pvlu ;
   crt_srcp(18,31,0);
   printf("%7.2f",zs[0]);
   crt_srcp(18,46,0);
   printf("%7.2f",zs[1]);
   crt_srcp(18,61,0);
   printf("%7.2f",zs[2]);
data[nl][0] = time;
data[nl][4] = zs[0]*pi/180;
data[nl][5] = zs[1]*pi/180;
data[nl][6] = zs[2]*pi/180;
data[nl][10] = data[nl][11] = data[nl][12] = 0;
hold_sh();
zero();
putchar('\007');
crt_srcp(22,25,0);
puts(" PRESS ANY KEY TO CONTINUE ");
i = key_getc();
}
```

```
     /*  a/d in routine  */
  in (chnl,value)
unsigned int chnl,*value;

{
unsigned int i,a,b,c;

    outportb(635,0); /* clear A/D regiter */
    outportb(632,chnl); /* select A/D channel */

       for (i=1;i<=7;i++)
           a = inportb(636);
       for (i=1;i<=7;i++)
           a = inportb(637);

    b =inportb(634);
    c =inportb(633);

    a = b << 12;
    b = a >> 4;
    *value = b+c;
}

      /*   d/a out routine   */
  out (lnkslc,value)
unsigned int lnkslc,value;

{
unsigned int i,a,b,c;

    b = value >> 8 ;
    a = value << 8 ;
    c = a >> 8 ;
    outportb(638,c);    /* d/a out */
    outportb(639,b);
    outportb(638,c);
    outportb(639,b);
    outportb(888,lnkslc);    /* link selected */
    for(i=0;i<=10;i++);
    outportb(888,0);
}
   /*  exit routine  */
zero()

{
unsigned int lnk,dtoa,selc;

   for(lnk=0;lnk<3;lnk++)
      {
        if(lnk == 0) dtoa = 2053;        /* amp 40/2 w */
        if(lnk == 1) dtoa = 2057;        /* amp 150 w */
        if(lnk == 2) dtoa = 2053;        /* amp 40 w */
      selc = 4 >> 2-lnk ;
      out(selc,dtoa);
      }
   hold_sh();
}
```

```
hold_sh()
{

extern float fabs(),cos();
unsigned int pvlu,dtoa,selc;
int i;
float zs[3],opvo,c2,c23,d2,hv;
      /* read actual position (rad) */
  in(14,&pvlu);       zs[1] = 4.14057-0.001367*pvlu ;
  in(15,&pvlu);       zs[2] = 4.95411-0.001376*pvlu;
  c2 = cos(-0.1308997-zs[1]);
  c23 = cos(-zs[1]+1.05-zs[2]);
  d2 = 9.3187*c2+1.34456*c23;
  opvo = d2/2.5;
  dtoa = opvo*19.036495+2028.65; /* amp2 sw 120 */
  if(dtoa < 0 || dtoa > 4095) zero();
  selc =  2 ;
  out(selc,dtoa);
}

go_desire(zd)
float zd[];
{
extern int key_scan();
extern float fabs(),vmx,Kp[],Ki[],Kd[],maxout[];
float zt[3],mvo[3],err[3],itgerr[3],opvo,vel[3],oldp[3],verr
unsigned int pvlu,dtoa,selc;
int lnk,chk;
float smp = 0.015;

 itgerr[0] = itgerr[1] = itgerr[2] = 0;
 do {
    for (lnk=0;lnk<3;lnk++)
        {
        in(13+lnk,&pvlu);  /* read actual position (rad) */
        if(lnk == 0) zt[0] = 0.001369*pvlu-1.110758 ;
        if(lnk == 1) zt[1] = 4.140573-0.001367*pvlu ;
        if(lnk == 2) zt[2] = 4.954108-0.001376*pvlu ;
        vel[lnk] = (zt[lnk]-oldp[lnk])/smp;
        oldp[lnk] = zt[lnk];
        verr[lnk] = 0-vel[lnk];
        err[lnk]=zd[lnk]-zt[lnk];
        if(fabs(err[lnk])>0.05)  itgerr[lnk] = 0;
        itgerr[lnk] += smp*err[lnk];
 mvo[lnk]=Kp[lnk]*err[lnk]+Ki[lnk]*itgerr[lnk]+Kd[lnk]*verr
        if(mvo[lnk]>0 && mvo[lnk]>maxout[lnk])
           mvo[lnk] = maxout[lnk];
        if(mvo[lnk]<0 && mvo[lnk]<-1.0*maxout[lnk])
           mvo[lnk] = -1.0*maxout[lnk];
        opvo = mvo[lnk];
        if(lnk == 0) dtoa = opvo*69.209770+2053.23;  /* amp
        if(lnk == 1) dtoa = opvo*57.365913+2057.54;  /* amp
        if(lnk == 2) dtoa = opvo*69.209770+2053.23;  /* amp
        if(dtoa < 0 || dtoa > 4095) zero();
        selc = 4 >> 2-lnk ;
        out(selc,dtoa);
        } /* end of each link */
        chk = key_scan();
        if(chk != -1)  { err[0] = err[1] = err[2] = 0; }
    }while(fabs(err[0])>0.001 || fabs(err[1])>0.001
                              || fabs(err[2])>0.001);
```

จากรายละเอียดของโปรแกรมทั้งหมดที่แสดงข้างต้น ตัวแปรที่สำคัญและความหมายของตัวแปรมีดังต่อไปนี้

smp : sampling period

vmx : maximum velocity

amx : maximum acceleration

Kp[0] : proportional gain of waist

Ki[1] : integral gain of shoulder

Kd[2] : delivative gain of elbow

data[][i]

i = 0 : time

i = 1-3 : reference position

i = 4-6 : actual position

i = 7-9 : reference velocity

i = 10-12 : actual velocity

i = 13-15 : drived torque

ประวัติผู้เขียน

วิทยานิพนธ์ฉบับนี้จัดทำโดย นายไพรัช บุพพวงค์ เกิดวันที่ 21 ตุลาคม 2504 ที่จังหวัดสุพรรณบุรี สำเร็จการศึกษาขั้นปริญญาบัณฑิตวิศวกรรมศาสตร์ สาขาวิศวกรรมเครื่องกล (วศบ. เครื่องกล) จากมหาวิทยาลัยสงขลานครินทร์ จังหวัดสงขลา เมื่อปี พศ.2527 ปัจจุบัน ผู้เขียนทำงานด้านการวิจัยและพัฒนาเครื่องปรับอากาศและเครื่องจักรในการผลิต ที่บริษัท ยูนิแพ้บอิควิปเมนท์ จำกัด