

로봇 매니퓰레이터의 동적방정식의 자동 생성에 관한 연구

°원 태현 • 황 창 선

부산대학교 공과대학 전기공학과

Automatic Generation of Dynamic Equations for Robotic Manipulators

°Tae-Hyun Weon • Chang-Sun Hwang

Dept. of Electrical Eng., Pusan National University

Abstract

A program is developed for generating the dynamic equations for robotic manipulators using the symbolic language muSIMP/MATH. The muSIMP/MATH is a LISP-based computer algebra package, devoted to the manipulation of algebraic expressions including number, variables, functions, and matrix. The muSIMP/MATH can operate on IBM-PC compatibles with MS-DOS.

The program is developed, based on the Lagrange formalism. This program is applicable to the manipulators of any number of degrees of freedom, maximum six degree of freedom in this program.

To control robotic manipulators by using dynamic equations is required a symbolic equations. The generated dynamic equation can be applied directly to the robotic manipulators, for the generated dynamic equation is a reduced form of symbolic expression.

I. 서론

공장 자동화의 필요성이 강조됨에 따라 로봇의 중요성이 크게 부각되었다. 특히, 로봇을 제어하기 위한 여러방법이 제시되고 있으며, force control 혹은 position control 하기 위하여는 Kinematics or Dynamics의 해석이 필요로 한다. Dynamics를 해석하기 위하여는 Newton-Euler 방법과 Lagrange-Euler 방법등이 잘 알려져 있으나, 이들 방법으로 Dynamic Equation을 유도하려면 지루하고, 많은 시간이 요구되므로 곤란한 점이 있다. 또한, 로봇의 기종이 바뀌면 이미 계산해 놓은 Dynamics

Equation을 바로 사용할 수 없는 단점이 있었다. 또한 로봇의 Dynamic Equation을 유도함에 있어서 임의의 n차 자유도를 가진 경우에 최소한 n개의 변수가 발생하므로 수치해석적 방법을 사용하기가 곤란하며 기호연산 (Symbolic Computation)이 필요하게 된다. 기호연산을 위한 Computer Language는 여러가지 개발이 되어 있으며, 이로서 Dynamic Equation을 유도한 Program은 LISP나 PROLOG 언어 및 LISP를 토대하여 만든 algebra package인 MACSYMA, REDUCE 및 ARM등이 Robot Dynamic Equation을 유도하는데 응용되었다. 그러나 이들 algebra package들은 대형 혹은 중형 Computer에서 운용 가능하므로 이들 Computer가 없을 경우 사용상의 문제가 있으므로 이를 개인용 Computer에 이식되어 진다면 바람직하다. 또한 LISP 나 PROLOG Language를 이용하여 Dynamic Equation을 구하는 program도 소개되어 있으나, 이들 언어를 사용하려면 Dynamic Equation을 유도하는데 필요한 모든 기능을 만들어야 함으로 Program이 크지는 문제점이 있다. 본 논문은 이러한 문제점을 개선 하기 위하여 개인용 Computer에서도 수, 변수, 함수 및 행렬을 포함한 기호연산 기능이 있는 muSIMP/MATH를 사용하여 IBM-PC 호환기종(OS는 MS-DOS, main memory 640K)에서 Robot Dynamic Equation을 유도할 수 있는 Program을 소개 하고자 한다.

본 Program의 입력은

- 1) 자유도
- 2) link의 Denavit-Hartenburg parameter인 $\theta_i, \alpha_i, d_i, a_i$
- 3) joint variables
 - i. R rotational joint인 경우
 - ii. P prismatic joint인 경우

- 4) Pseudo Inertia matrix의 elements인
Ixx, Iyy, Izz, Ixy, Ixz, Iyz
- 5) link의 질량 중심
Px, Py, Pz
- 6) 중력 벡터 (BASE 좌표계 기준)
Gx, Gy, Gz

이며.

출력은

- 1) Kinematic Transformation matrices
- 2) Forward Transformation matrices
- 3) Inertial coefficient matrix
- 4) Centrifugal and Coriolis force vector
- 5) Gravitational force vector

를 포함한 전체 Dynamic Equation이나 필요에 따라서 위에 나열한 각각의 matrix 및 vector를 단독으로 출력시킬수 있도록 되어 있다.

II. Lagrange Equation

로봇의 Kinematics는 Denavit-Hartenburg방법과 Homogeneous Transformation을 이용하여 표현 가능하다. link i 에 관련된 link의 기하학적 parameter는 $\theta_i, \alpha_i, d_i, a_i$ 의 4가지 이다. 로봇 joint 종류에 따라서, Rotational joint와 Prismatic joint로 구분이 가능하므로 Rotational joint인 경우 θ_i 가, Prismatic joint인 경우 a_i 가 변수가 된다.

link i 좌표계와 link i-1 좌표계 사이의 Homogeneous Transformation은 (4 x 4) 행렬이고 아래와 같다.

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

임의의 link j-1 좌표계로 부터 link i 좌표계의 Transformation은

$${}^{j-1}T_i = A_j A_{j+1} \dots A_i \quad (2)$$

이다. 로봇 motion에 대한 동적방정식 (Dynamic Equation)은 아래와 같다.

$$F_i = \sum_{k=1}^m D_{ik} \ddot{q}_k + \sum_{j=1}^m \sum_{k=1}^m H_{ijk} \dot{q}_j \dot{q}_k + G_i \quad (3)$$

여기서

- i. F_i 는 joint actuator에 적용되는 generalized force (or torque) 이다.

- ii. D_{ik} 는 inertia term으로서 식(4)와 같이 나타낼 수 있다.

$$D_{ik} = \sum_{p=\max(i,k)}^m \text{Tr} \left(\frac{\partial T_p}{\partial q_k} J_p \frac{\partial T_p^T}{\partial q_i} \right) \quad (4)$$

- iii. H_{ijk} 는 centrifugal과 Coriolis force term으로서 식(5)와 같이 표현된다.

$$H_{ijk} = \sum_{p=\max(i,j,k)}^m \text{Tr} \left(\frac{\partial T_p}{\partial q_i \partial q_k} J_p \frac{\partial T_p^T}{\partial q_j} \right) \quad (5)$$

- iv. G_i 는 gravity loading force term으로서 식(6)과 같이 쓸 수 있다.

$$G_i = \sum_{p=i}^m -m_p g^T \frac{\partial T_p}{\partial q_i} r_p \quad (6)$$

- v. 벡터 r_i 는 link i 에 관한 질량중심의 위치를 표시되며, link i 좌표계를 기준으로 하여 표시하면 식(7)과 같다.

$$r_i = [x_i \ y_i \ z_i \ 1]^T \quad (7)$$

- vi. 벡터 g 는 중력장을 표현하며 이를 base좌표계로 표시하면 식(8)과 같다.

$$g = [g_x \ g_y \ g_z \ 0]^T \quad (8)$$

- vii. J_i 는 link i에 관한 pseudo-inertia matrix로서 식(9)와 같이 나타낼 수 있다.

$$J_i = \begin{bmatrix} \frac{-I_x^i + I_y^i + I_z^i}{2} & I_{xy}^i & I_{xz}^i & m_i x_i \\ I_{xy}^i & \frac{I_x^i - I_y^i + I_z^i}{2} & I_{yz}^i & m_i y_i \\ I_{xz}^i & I_{yz}^i & \frac{I_x^i + I_y^i - I_z^i}{2} & m_i z_i \\ m_i x_i & m_i y_i & m_i z_i & m_i \end{bmatrix} \quad (9)$$

III. muSIMP/MATH를 기초로한 Lagrange Equation의 SYMBOLIC Computation

muSIMP/MATH은 기호연산 기능을 가지고 있으므로 이를 이용하여 Dynamic Equation을 구하는 program을 작성할수 있다. 표1.은 muSIMP/MATH의 이러한 기능을 도식적으로 나타내고 있다. 즉 LISP나 PROLOG language로 Dynamic Equation을 유도할 경우 Programmer가 기호 연산이 가능하도록 하는

FUNCTION들을 정의해야하나, muSIMP/MATH에서는 이러한 문제점을 고려하지 않아도 된다.

i. 두 matrix A, B를 곱하기 위하여 간단하게

$$A \cdot B$$

로서 구할수 있다.

ii. matrix A의 transpose를 구하려면 아래와 같이 하면 된다.

$$A'$$

iii. matrix A를 변수 θ 으로 미분하려면 다음과 같이 구할수 있다.

$$DIF(A, \theta)$$

iv. muSIMP/MATH에 정의되어 사용 가능한 algebraic simplification function은 EXPD, EXPAND, FCTR, TRGEXPD등이 있으며, 각 function들의 기능은 표2.와 같다.

muSIMP/MATH의 수식	결과
DIF(SIN θ COS θ , θ)	1 - 2 SIN 2 θ
DIF(SIN 2 θ , θ , 2)	2 - 4 SIN 2 θ
$\theta^4 * \theta^{(1/2)}$	$\theta^{(9/2)}$
$\theta * \theta$	θ^2

표. 1 muSIMP/MATH의 기호 연산

함 수	기능
EXPD	다항식을 전개
EXPAND	다항식을 공분분모로 전개
FCTR	다항식을 공통인수로 전개
TRGEXPD	삼각함수 다항식을 전개

표 2. 수식단순화 기능

이를 기초로하여 그림1과 같은 순서도를 가지는 program을 개발할수 있다.

IV. Computer Simulation

Example 1. 2 자유도 joint 로봇

link NO.	θ_i	α_i	d_i	a_i	변 수
1	θ_1	0°	0	L	$Q_1 = \theta_1$
2	θ_2	0°	0	L	$Q_2 = \theta_2$

inertia term

$$I_{yyi} = I_{zzi} = 1/3 * m_i * L^2$$

$$I_{xxi} = I_{xyi} = I_{yzi} = I_{xzi} = 0 \quad \text{for } i=1,2$$

mass

$$M_1, M_2$$

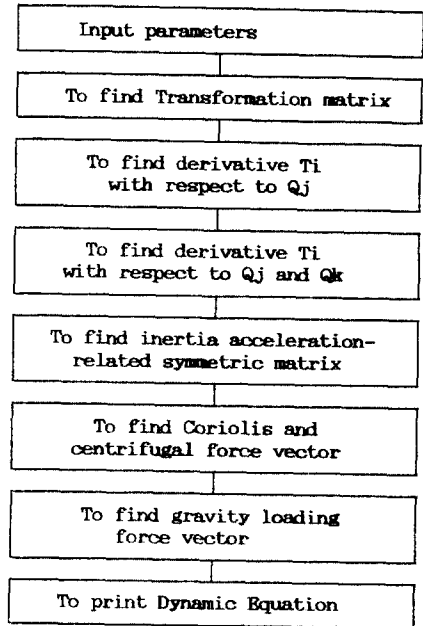


Fig. 1 Flow Chart of program

그림 1 프로그램의 순서도

질량 중심

$$r = [-1/2 * L_i \ 0 \ 0 \ 1]^T \quad \text{for } i = 1,2$$

중력장

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

위의 파라미터를 입력하여 program을 수행시키면

아래와 같은 결과를 출력한다.

$$P_1 = (L^2 (M_1 + M_2 (4 + 3 \cos Q_2)) / 3 + I_{a1}) * ddQ_1 + (L^2 M_2 (2 + 3 \cos Q_2) / 6) * ddQ_2 + (-L^2 M_2 \sin Q_2) * dQ_2 * dQ_1 + (-L^2 M_2 \sin Q_2 / 2) * dQ_2^2 + L G (M_2 (2 \cos Q_1 + \cos (Q_1 + Q_2)) + M_1 \cos Q_1) / 2$$

$$P_2 = (L^2 M_2 (2 + 3 \cos Q_2) / 6) * ddQ_1 + (L^2 M_2 / 3 + I_{a2}) * ddQ_2 + (L^2 M_2 \sin Q_2 / 2) * dQ_1^2 + L M_2 G \cos (Q_1 + Q_2) / 2$$

V. 결론

Robot Dynamic Equation을 기호연산을 사용하여 풀어 냄으로서 Dynamic Equation을 이용하여 제어하는데 필요한 방정식을 간단히 구할 수 있음으로서 편리한 점이 많아 졌으며, 기존 개발된 program의 문제점을 해결 할 수 있었다. 이후 Inverse Kinematics와 Inverse Dynamics를 유도할 수 있는 program을 개발함으로써 본 program으로 해결 할 수 없는 문제를 해결 해야한다.

VI. 참고 문헌

1. Paul, R., " Robot manipulators : Mathematics, Programming, and Control ", M.I.T. Press, 1981
2. Luh, J.Y.S., Walker, M.W., and Paul, R.P., " On-Line Computation scheme for Mechanical Manipulators", ASME Journal of Dynamic Systems, Measurement, and Control, Vol.102, June 1980, pp.69-76
3. Walker, M. W., and Orin, D. E., " Efficient Dynamic Computer Simulation of Robotic Mechanisms", ASME Journal of Dynamic Systems, Measurement, and Control, Vol.104, Sept. 1982, pp.205-211
4. Hollerbach, J. M., " A Recursive Formulation of Lagrangian Manipulator Dynamics", IEEE Transactions on Systems, Man and Cybernetics, Vol.SMC-10, No.11, 1980, pp.730-736
5. Vecchio, L , Nicosia, S., Nicolo, F., and Lentini, D., " Automatic Generation of Dynamic Models of Manipulators", 10th International Symposium on Industrial Robotics, Milan, Italy, Mar. 1980, pp.5-7
6. Leu, M. C. and Hemati, N. , "Automated Symbolic Derivation of Dynamic Equations of Motion for Robotic Manipulators", ASME of Dynamic Systems, Measurement, and Control, Vol.108, Sept. 1986, pp.172-179
7. J.Y.S. Luh, C.S. Lin, " Automatic Generation of Dynamic Equations for Mechanical Manipulators", Proceeding of JACC, June 17-19, 1981,
8. Neto, J.L., Pereira, A.B.C., Alves, J.B., " Symbolic Computation applied to Robot Dynamic Modeling", Proceeding of the 16th International Symposium on Industrial Robots, 1986, pp.389-400
9. 원 태현, 황 창선, 최 영규, " LISP를 이용한 로보트 동역방정식의 생성 ", 제어계측 연구회 논문집, MAY, 1987, pp.16-19
10. Paul, R., Izaguirre, A. " Automatic Generation of the Dynamic Equations of The Robot Manipulators using a LISP program " , IEEE Conference on Robotics Automation , 1986, pp.220-226
11. Stoutemyer, D. R. " LISP Based Symbolic Math Systems ", BYTE, Aug. 1979, pp.176-192
12. " muSIMP/MATH Reference Manual", The Soft Warehouse , Sept., 1983