

범용 로봇 시뮬레이션 패키지 개발에 관한 연구

강대희, 주광혁, 배한경, 김학표
 대우중공업, 기술연구소

Development of a General Purpose Software Package for Robot Simulation

Dae-hee Kang, Kwang-hyeok Ju, Han-kyoung Bae, Hak-pyo Kim
 DAEWOO Heavy Industries Technical Center

The simulation algorithm for all kinds of robots with arbitrary degrees of freedom which are combined with revolute joints or prismatic joints, or combinations was studied and implemented.

This simulation package is composed of trajectory planning routine, control routine, kinematics routine using Newton-Raphson method, dynamics based on Newton-Euler method with four-bar linkage analysis, input routine and output routine.

I. 서론

로봇 제어 알고리즘을 설계하여 적용하고자 할때, 시뮬레이션 패키지를 이용하면 그 성능을 미리 평가해 볼수 있으며, 로봇 본체 설계시에도 각 조인트에 부가되는 최대 토크를 계산하여 알맞는 모터를 선정할 수 있는 등, 그 용도가 다양하기 때문에 많은 시뮬레이션 패키지가 개발되어 왔다. 그러나 대부분의 시뮬레이션 패키지는 정역학, 동력학적인 측면이나 기구학적인 측면만을 고려하여 설계되었고, 특정한 로봇에 대해서만 시뮬레이션이 가능하므로 다양한 로봇에 적용시키기는 많은 어려움이 있다.

본 논문에서는 Revolute 조인트 또는 Prismatic 조인트가 조합되어 임의의 자유도를 갖는 모든 형태의 로봇에 대해서 정역학, 동력학, 기구학적인 분석이 가능하고 로봇 제어 및 궤적 계획을 포함하는 시뮬레이션 패키지의 컴퓨터 프로그램 구성방법과 그에 관한 알고리즘을 제시하였다.

II. 시스템의 구성 및 원리

전체적인 블록선도는 (그림 1)과 같다. 기구학적인 파라메타와 역학적인 파라메타 입력은 로봇 각 조인트의 형태 및 각 링크의 길이, 무게, 무게중심점, 혹은 Closed-chain 구조의 존재여부 및 그 형태등을 규정하는 값들로서, 시뮬레이션 하고자 하는 로봇의 고유 데이터가 입력된다. 로봇의 기구학 및 역학은 거의 모든 로봇에 적용할

수 있도록 하였다.

1. 로봇의 기구학적 분석

로봇의 기구학적 분석은 두가지 측면에서 병하여 지고 있다. 첫째는 Direct Kinematics로서, 어떤 매니퓰레이터에 대하여 조인트 변수 값들의 기하학적인 값이 주어졌을때, 기준 좌표계에 대한 기계손의 위치와 방향을 계산하는 것이고, 다른 하나는 Inverse Kinematics로서 기준 좌표계에 대한 기계손의 위치와 방향이 주어졌을 때 조인트 변수 값들을 계산하는 것이다.

1) Direct Kinematics

매니퓰레이터의 Base로부터 기계손까지 순차적으로 일련의 번호를 각 링크와 조인트에 부착하면, i 번째 조인트 J_i 는 i-1 번째 링크 L_{i-1} 과 i 번째 링크 L_i 를 연결하게 된다. L_{i-1} 좌표계와 L_i 좌표계 간의 관계는 다음과 같이 4×4 matrix로 표현된다.

$$A_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

또한 Base 좌표계에 대한 기계손의 위치와 방향은 위 식의 곱으로써 표현된다.

$${}^b T_e = A_1 A_2 \cdots A_n$$

2) Inverse Kinematics

임의의 매니퓰레이터에 대한 inverse kinematics를 계산해야 하기 때문에 그 해가 closed form으로 표현될수 없다. 따라서 수치해석 방법을 사용해야만 된다.

수치해석 방법중 가장 접근 속도가 빠른 Newton-Raphson 방법을 사용하지만, 이 방법은 해를 구하지 못할 경우도 있으므로 modified Newton-Raphson 방법을 사용하고자 한다. 조인트 속도와 기계손 속도간의 관계는 Jacobian matrix로 표현 된다.

$$\dot{x} = \begin{bmatrix} \dot{\omega}_z \\ \dot{P}_e \end{bmatrix} = J \dot{q} \quad (1)$$

(1) 식으로 부터 다음의 식을 유추할 수 있다.

$$\begin{aligned} \Delta X &= J \Delta q \\ \therefore \Delta q &= J^{-1} \Delta X \end{aligned} \quad (2)$$

Base 좌표계에서 기계손의 위치와 방향이 다음과 같이 4 x 4 matrix로 주어졌다고 하자.

$$X_2 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}_2$$

또한 Newton-Raphson 방법에서는 초기치가 주어져야 하므로 다음과 같이 x 를 구할 수 있다.

$$X_i = \begin{bmatrix} n_{xi} & o_{xi} & a_{xi} & p_{xi} \\ n_{yi} & o_{yi} & a_{yi} & p_{yi} \\ n_{zi} & o_{zi} & a_{zi} & p_{zi} \\ 0 & 0 & 0 & 1 \end{bmatrix}_i$$

$$\begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \end{bmatrix} = \begin{bmatrix} p_{x2} - p_{xi} \\ p_{y2} - p_{yi} \\ p_{z2} - p_{zi} \\ (1/2)(n_{xi}n_{x2} + o_{xi}o_{x2} + a_{xi}a_{x2} - n_{yi}n_{y2} - o_{yi}o_{y2} - a_{yi}a_{y2}) \\ (1/2)(n_{xi}n_{x2} + o_{xi}o_{x2} + a_{xi}a_{x2} - n_{zi}n_{z2} - o_{zi}o_{z2} - a_{zi}a_{z2}) \\ (1/2)(n_{yi}n_{y2} + o_{yi}o_{y2} + a_{yi}a_{y2} - n_{xi}n_{x2} - o_{xi}o_{x2} - a_{xi}a_{x2}) \end{bmatrix}$$

(2) 식으로부터 q 의 값을 구할 수 있다.

$$q \leftarrow q + \Delta q$$

전체적인 알고리즘은 다음과 같다.

- (1) Initialize q_0
- (2) $K \leftarrow 0$
- (3) $X \leftarrow \text{CALL Direct-Kinematic}(q_0)$
- (4) Compute $\Delta X_k = [d, \delta, c]^T$
- (5) If $|\Delta X_k| < \text{Small}$, THEN END
- (6) If Iteration number $> \text{MAX}$, THEN Error
- (7) Compute Jacobian
- (8) Compute J^{-1}
- (9) $\Delta q \leftarrow J^{-1} \Delta X_k$
- (10) $j \leftarrow 1$
- (11) $q_{k+1} \leftarrow q_k + \Delta q$
- (12) $x_k \leftarrow \text{CALL Direct - Kinematics}(q_{k+1})$
- (13) $\Delta x_{kj} \leftarrow x_d - x_k$
- (14) If $|\Delta x_{kj}| > |\Delta x_k|$, THEN

$$q_{k+1} \leftarrow q_k + \Delta q / 2^j$$

$$j \leftarrow j+1$$
 GOTO (11)
- (15) $\Delta x_{k+1} \leftarrow \Delta x_{ki}$
- (16) $K \leftarrow K+1$
- (17) GOTO (5)

2. 로봇의 역학적 분석

로봇의 역학적 분석은 기구학적인 측면과 마찬가지로 다음의 두가지 측면에서의 분석이 행해지고 있다.

Direct dynamics는 로봇이 임의의 위치에서 임의의 속도로 움직이고 있을때 주어진 토크로 인하여 각 조인트에서 발생하는 가속도를 계산하며, inverse dynamics는 그와 반대로 주어진 위치와 속도에서 요구되는 가속도를 얻기위해서 필요한 토크를 계산한다.

1) Inverse Dynamics

시뮬레이션 패키지의 inverse dynamics 에서는 조인트의 변수(위치, 속도, 가속도)들을 이용하여 각 조인트에 걸리는 토크 또는 힘(force)을 구한다. 본 논문에서는 개루프 구조(Open-loop)를 가진 로봇과 폐루프(closed-loop) 구조를 가진 로봇 어느 것에 대해서도 inverse dynamics의 해석이 가능하도록 패키지가 구성되어 있으며 특히 대우로봇(NOVA-10)가 가진 fourbar linkage 구조를 모델로 하여 그 해를 전개하였다.

(1) 개 루프(open-loop) 구조를 가진 로봇의 inverse dynamics; 본 논문에서는 간단하면서도 계산시간이 적게 걸려서 실시간 제어가 가능한 Recursive form의 Newton-Euler(이 후 N-E라 약칭) 방정식을 사용하였다. 그런데 N-E 방정식은 토크 또는 힘, 각 링크의 inertia와 무게중심의 위치 등이 기준 좌표계에 대하여 설정되기 때문에 로봇이 움직임에 따라 그 값들이 순간순간마다 변하므로 계산의 어려움이 있어 해당 링크의 좌표계를 기준으로 모든값들을 고쳐서 계산이 간단하도록 하였다.

$$\begin{aligned} R_i F_i &= m_i R_i^0 a_i \\ R_i N_i &= (R_i^0 I_i R_i^0) (R_i^0 \dot{w}_i) + (R_i^0 w_i) \times [(R_i^0 I_i R_i^0) (R_i^0 w_i)] \\ R_i f_i &= R_i^{i-1} (R_{i+1}^0 f_{i+1}) + R_i^0 F_i \\ R_i n_i &= R_i^{i-1} [R_{i+1}^0 n_{i+1} + (R_{i+1}^0 p_i^*) \times (R_{i+1}^0 f_{i+1})] \\ &\quad + (R_i^0 p_i^* + R_i^0 s_i) \times (R_i^0 F_i) + R_i^0 N_i \\ \tau_i &= \begin{cases} (R_i^0 n_i)^T (R_i^{i-1} z_0) + b_i \dot{q}_i & (\text{링크 } i \text{가 revolute}) \\ (R_i^0 f_i)^T (R_i^{i-1} z_0) + b_i \dot{q}_i & (\text{링크 } i \text{가 prismatic}) \end{cases} \end{aligned}$$

(2) 폐 루프(closed-loop) 구조를 가진 로봇의 inverse dynamics; 폐 루프 구조에서 구동원(Actuator)에 의해 구동되지 않는 루프상의 한 조인트를 분리하여(cut open) 로봇이 tree-structure를 가진 폐 루프 구조를 가지도록 가정한다.

그 다음 각각의 폐 루프 구조에 대해 N-E방정식을 이용, 각 조인트에 걸리는 토크 또는 힘을 구한 다음 holonomic constraint 방정식과 Lagrangian multiplier 벡터를 이용하여 구동원에 의해 직접 구동되지 않는 조인트에 걸리는 토크 또는 힘을 구한다.

그리고 이때 구한 토크 또는 힘으로 실제 구동원에 의해 직접 구동되는 조인트에 걸리는 토크 또는 힘을 보상해 준다.

$$\begin{bmatrix} \tau_1^c \\ \tau_2^c \\ \vdots \\ \tau_{n-m}^c \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_{n-m} \end{bmatrix} - \left[\left(\frac{\partial C^*}{\partial q} \right)^T \right] \lambda$$

2) Direct Dynamics

각 조인트에 걸리는 토크 또는 힘의 함수와 위치 및 속도의 함수를 이용하여 가속도를 계산하는 루틴으로, 다음과 같은 알고리즘으로 구현하였다.

$$\tau = H(q) \cdot \ddot{q} + B(q, \dot{q})$$

- (1) $\ddot{q} \leftarrow 0$
- (2) $\tau \leftarrow$ CALL Inverse - Dynamics(\ddot{q})
- (3) $B \leftarrow \tau$
- (4) $k \leftarrow 1$
- (5) $\ddot{q} \leftarrow e_k$ (e_k : k -th element가 1이고 나머지는 0 인 $n \times 1$ vector)
- (6) $H^k \leftarrow$ CAL Inverse - Dynamics(\ddot{q})
(H^k : H matrix 의 k -th column)
- (7) If k No. of d.o.f. Then $k \leftarrow k+1$, GOTO(5)
- (8) $\ddot{q} \leftarrow H^{-1}(\tau_d - B)$

3. 궤적 계획

궤적계획은 속도와 가속도의 연속성을 보장하여, 매니퓰레이터의 jerky motion을 방지하는데 중점을 두고 있다. 즉 선형 MOVE 명령의 보간 함수는 직교좌표계에서 5차 함수이고, 조인트 좌표계에서 3차 함수이다.

4. 로봇트 제어

간단하게 P.I.D 제어 방법을 사용하였다.

$$\tau = \tau_d + K_p(\theta_d - \theta) + K_v(\dot{\theta}_d - \dot{\theta}) + K_i \int (\theta_d - \theta) dt$$

III. 결 론

본 시뮬레이션 패키지를 그림 2의 구조를 갖는 대우중공업의 로봇트 NOVA-10에 적용시켜 보았다. 그림 3, 그림 4와 그림 5는 각각 궤적계획 루틴에서 발생시킨 위치, 속도와 가속도이며, 앞에서 예상한 바와 같이 속도와 가속도의 연속성을 볼 수 있다. Inverse kinematics 루틴에서는 3-4회의 반복으로 10⁻⁴ rad이내로 수렴하고 있으며, inverse dynamics 루틴에서는 궤적계획에 따른 토크가 그림 6과 같이 발생하였다. PID 제어기에 의해서 계산된 토크를 dynamics 루틴에서 매니퓰레이터의 실제 움직임으로 변환시키고 있는데, 각 조인트의 위치를 그림 7에 나타 내었다.

이와같이 시뮬레이션 패키지를 사용하면 로봇트 제어기와 매니퓰레이터 설계에 필요한 모든 자료를 일괄적으로 플로팅하여 볼 수 있으므로, 이들의 설계에 큰 도움을 주는 도구로 사용할 수 있을 것이다.

참 고 문 헌

- (1) Parviz E. Nikrassh "Computer aided analysis

of mechanical system", The University of Iowa, Feb., 1982

- (2) R.P.Paul, "Robot Manipulators : Mathematics, Programming, and Control," MIT Press, 1981.
- (3) J.J.Craig, "Introduction to robotics : Mechanics & Control,"
- (4) J.Y.S. Luh, M.W.Walker, and R.P.C. Paul, "On-line computational scheme for mechanical manipulators," ASME Trans.: J. Dynamic Syst., Meas., Contr., Vol. 102, PP. 468-474, June 1980.
- (5) J.Y.S. Luh and YUAN-FANG ZHENG, "Computation of Input Generalized Forces for Robots with Closed kinematic Chain Mechanisms," IEEE J. Robotics and Automation, Vol. RA-1, No.2, PP. 95-103, June 1985.
- (6) S.Y.Oh, Orin, and Bach, "An Inverse Kinematics Solution for kinematically Redundant Robot Manipulators," J. Robotic Systems, PP.235-249, 1984.
- (7) J.Furusko and S.Onishi, "An Efficient Approach for solving the Inverse kinematics of Manipulators," Proceedings of 15th ISIR, Vol.2, PP.1051-1058, 1985.
- (8) Y.Nakamura, H.Hanafusa, Y.Yokokohji, and T. Yoshikawa, "Efficient Computation and Kinematic Representation for Robot Manipulator Simulation," Proceedings of 15th ISIR, Vol. 2, PP.1059-1066, 1985.
- (9) D.E. Orin and W.W. Schrader, "Efficient Computation of the Jacobian for Robot Manipulators," International Journal of Robotics Research, Vol.3, No.4, PP.66-75, 1984.
- (10) M.W. Walker and D.E. Orin, "Efficient Dynamic Computer Simulation of Robot Mechanisms," ASME J. Dynamic Syst., Meas., Contr., Vol. 104, PP. 205-211, Sept. 1982.

그림 1. 전체 블록 선도

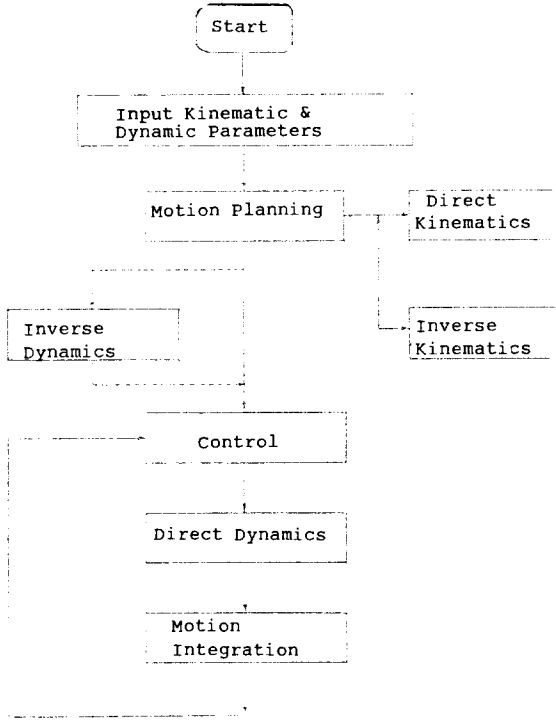


그림 2. 대우로봇 NOVA 10

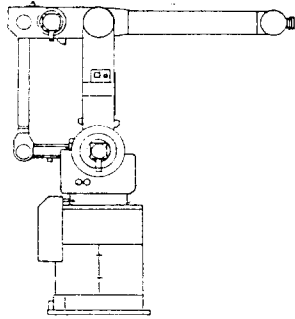


그림 3. 궤적계획에 의한 위치

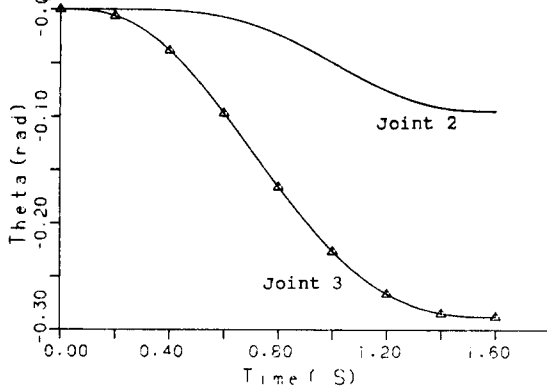


그림 4. 궤적계획에 의한 속도

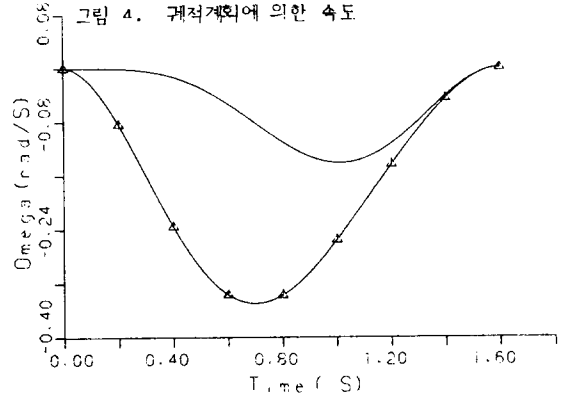


그림 5. 궤적계획에 의한 가속도

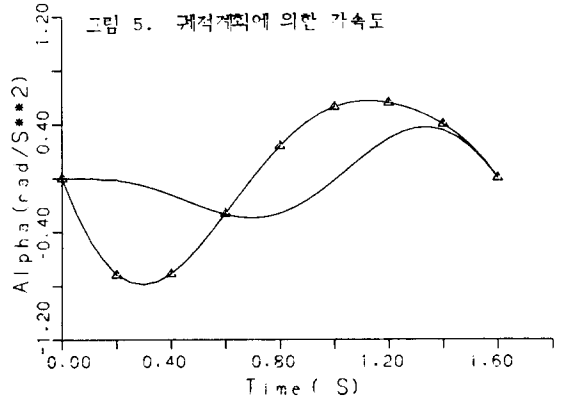


그림 6. 시간에 따른 토크 변화

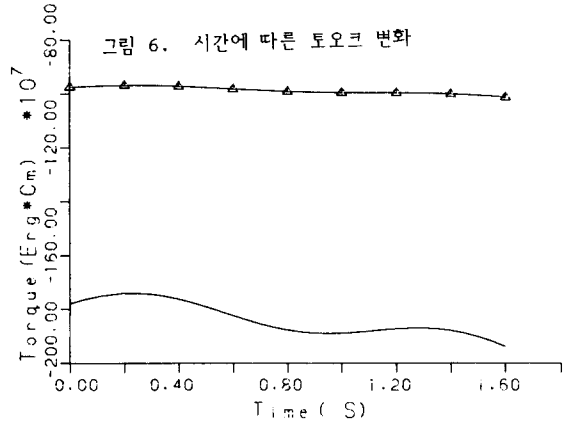


그림 7. 시뮬레이터를 통한 위치

