로보트를 위한 Dynamics 방정식의 해 유도와 Controller설계 에관한 연구

○우 상 래 , 김 형 래 , 박 인 갑

건국대학교 공과대학 전자공학과

A Study on the Solution of Dynamics Equation and the Desin of Controller for ROBOT

Sang -Lae Woo, Hyung-Lae Kim, In-Kap Park

Kon-Kuk University

A B S T R A C T

In this paper, a servo controller for a robot manipulator is developed. It consists of a conventional MRAS controller for each joint together with a feedback control to Dynamics compensation. To generate the control torque effectively a current drive method is adoted.The speed of the actuator is measured by using a tachometer and the position of the link is measured by using a potentiometer. To show the effectiveness of the proposed control algorithm , the proposed controller is tested in real time.

I. 서 론

Dynamics를 해석하여 로보트 운동을 Simulation 하거나 알맞는 제어 방식을 구성할때,혹은 로보트의 구조를 해석 할 때 유용 하게 사용 된다.

본 논문에서는 로보트의 Dynamics 식을 세우고 이에 적합한 제어기를 실현하여 보았다.

그러나 Dynamics 방정식의 계산량이 방대하기 때문에 이를 수 msec 동안으로 나누어 계산하고 메모리에 저장한 후, 이것을 출력하여 실험을 진행하여 실시간 제어가 가능 하게 하였다.

Model Reference Adaptive System 을 소프트웨어로 실현하였다. 이방법은 고유Reference model과 실제 System에 대한 Feedback Gain을 바꾸어 줌으로 쉽게 수행 할 수 있다.

또 Potentiometer 와 Tachometer 로 위치와 속도를 감지하고 오차를 보완하여 출력하여 정확한 제어를 위해 Feedback제어를 하여 주었다.

II. 본 론

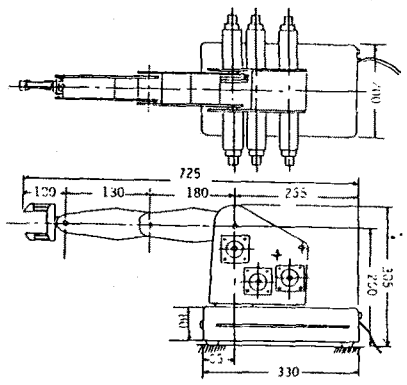

그림 1. KUN-KUK I 로보트의 외관

본 논문에서 사용한 KON-KUK I 로보트는 6축 5자유도인 Revolute Joint로 구성 되고, Wire로 동력을 전달하여 매우 부드럽게 작동한다.

II-1 . Lagrange-Euler(L-E) 의 운동 방정식

n자유도를 가진 로보트의 운동방정식을 유도할때 다음과 같은 식을 이용하여 방정식 의 해를 유도 한다.

1) i-1번째 좌표계에서 i번째 좌표계와 이웃하는 관절사이에 공간 전달이 일어나는 것을 4X4 Homogeneous Matrix로 나타내면 일반적으로 Ao은 다음과 같다.

$$A_o^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2\text{-}1)$$

논문에서는 KON-KUK I 로보트가 Revolute Joint로 만 이루어져 있으므로 식(2-1)을 이용 한다

2) Lagrange-Euler 방정식

일반적인 Lagrange-Euler 방정식은 다음과 같다.

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{q}_i}\right] - \frac{\partial L}{\partial q_i} = \tau_i \quad (2\text{-}2) \quad (i=1,2\ldots,n)$$

L = Lagrange 함수 = 운동 에너지(K) - 위치 에너지(P)

qi= 로보트의 일반적인 좌표계 ; Revolute qi=θ i,Prismatic qi=di

$\dot{q}_i$= 일반적인 좌표계 qi를 시간의 미분

τ i= 일반적인 로보트의 작용 되는 힘(토오크)

i) 로보트 팔의 속도

운동 L-E 방정식은 각 Joint의 속도에 관한 로보트의 운동 에너지를 알아야 한 다. 여기서는 Link i 점의 속도를 계산하고 이 Link에서의 모든 점의 영향을 계산 한다. Ri는 Link i에 고정된 점이라 하고 i번째 좌표계를 Homogeneous 좌표계로 설명 한다.

$$R_i = (x_i, y_i, z_i)^t$$

단 Superscript"t"는 벡터와 Matrix의 Transpose operation 이다.

점 Ri 뿐만 아니라 다른점을 Link i에 고정시키고 i번째 좌표계에 관한 zero 속도를 갖는다.

기본 좌표계에 대한 Link i의 한점 Ri의 속도는 다음과 같다.

$$Vi = \frac{dRo}{dt} \quad (2-3)$$

기본 좌표계에 대한 Ri = Ro 이다.

$$Vo \equiv Vi = \frac{d}{dt}\begin{bmatrix} i \\ Ro \end{bmatrix} = \frac{d}{dt}\begin{bmatrix} i \\ Ao\ Ri \end{bmatrix}$$

$$= \begin{bmatrix} i \\ \sum_{j=1} \frac{\partial Ao}{\partial qj}\dot{q}j \end{bmatrix} Ri \quad (2-4)$$

단 i=1,2,3...,n

qi에 관해 Ao 를 풀때 Matrix Qi를 대입하여 용이하게 계산 할수 있다.

Revolute Joint인경우;

$$Qi = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2-5)$$

$$\frac{Ai-1}{qi}\Big|_{i} = Qi.Ai-1 \quad (2-6)$$

이다.

KON-KUK I 로보트 팔에 적용 하면 i=1 그리고 q1=θ 1 이다.

$$\frac{1}{qi}\begin{bmatrix} Ao \end{bmatrix} = \begin{bmatrix} -SIN\theta\ 1 & -COS\alpha\ |SIN\theta\ 1 & SIN\alpha\ |COS\theta\ 1 & -a1COS\theta\ 1 \\ COS\theta\ 1 & -COS\alpha\ |SIN\theta\ 1 & SIN\alpha\ |COS\theta\ 1 & -a1COS\theta\ 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= Q1Ao \quad (2-7)$$

때문에 Uij를 정의 하면 다음과 같이 된다.

$$Uij \equiv \frac{\partial Ao}{\partial qj} = \begin{cases} Ao\ Qj\ Aj-1 & ;단\ j \le i \\ 0 & ;단\ j > i \end{cases} \quad (2-8)$$

이를 이용하여 로보트 팔의 속도 Vi를 구하면 다음과 같다.

$$Vi = \sum_{j=1}^{i} Uij\ qj\ Ri \quad ;단\ ,i= 1,2,...,n \quad (2-9)$$

식(2 -9)로부터 속도를 구할수 있다.
다음은 Joint 상호간의 내부운동 효과(Interaction effect)가 필요하다.

$$\frac{\partial Uij}{\partial Qk} \equiv Uijk = \begin{cases} Ao\ Qj\ Aj-1\ Qk\ Ak-1 & ;i \ge k \ge j \\ Ao\ Qk\ Ak-1\ Qj\ Aj-1 & ;i \ge j \ge k \\ 0 & ;i>.j\ or\ i<k \end{cases} \quad (2-10)$$

ii) 로보트 팔의 운동 에너지

Link i에서의 모든점에서 속도를 얻은후,모든 Link 에서의 운동 에너지를 구할 필요가 있다. Ki를 기본 좌표에서 해석되는 Link i의 운동 에너지라 하면, Link i에서 질량 m 일때 운동 에너지는 ½ mv² 이다.한점에서의 질량을d m,이때 이점 에서의 운동 에너지 dKi라 하면 dKi는

$$dKi = \tfrac{1}{2}(xi^2 + yi^2 + zi^2)dm = \tfrac{1}{2}\,Trace(ViVi^{t})dm$$

$$= \tfrac{1}{2}\,Tr(ViVi^{t})dm$$

Tr은 정방 행렬에서의 Trace operator 이다.
식 (2-20)에서의 Vi를 대입하면 다음과 같다.

$$dKi = \tfrac{1}{2}\,Tr\left[\left(\sum_{p=1}^{i} Uip\ qp\ Ri\right)\left(\sum_{r=1}^{i} Uir\ qr\ Ri\right)^{t}\right]dm$$

$$= \tfrac{1}{2}\,Tr\left[\sum_{i}\sum_{i} Uip(RiRidm)Uir^{t}\ qp\ qr\right]$$

모든점에 운동 에너지는 합하고 적분해야한다.

$$Ki = \int dKi = \tfrac{1}{2}\,Tr\left[\sum_{i}\sum_{i} Uip\left(\int RiRidm\right)Uir^{t}\ qp\ qr\right].$$

괄호 안의 적분항은 i번째 좌표계에 대한 Link i 의 모든점의 관성이다.
Ri =(xi,yi,zi,1) 는 먼저 정의되었고 Ji는 inertia tensor matrix 라고 설명 할수 있다.

$$Ji = \begin{bmatrix} \tfrac{1}{2}(-Ixx+Iyy+Izz)+\bar{mi}xi^2 & Ixy & Ixz & \bar{mi}xi \\ Ixy & \tfrac{1}{2}(Ixx-Iyy+Izz)+\bar{mi}yi^2 & Iyz & \bar{mi}yi \\ Ixz & Iyz & \tfrac{1}{2}(Ixx+Iyy-Izz)+\bar{mi}zi^2 & \bar{mi}zi \\ \bar{mi}xi & \bar{mi}yi & \bar{mi}zi & mi \end{bmatrix}$$

때문에 모든 Link의 운동 에너지 는 로보트 팔의 총 운동 에너지 K로 주어진다.

$$K = \sum_{i=1}^{n} Ki = \tfrac{1}{2}\sum_{i=1}^{n}\sum_{1}^{i}\sum_{r=1}^{i}\left[Tr(Uip\ Ji\ Uir^{t})qp\ qr\right]$$

Ji는 Link i의 질량 분포에 독립적이다.

iii) 로보트 팔의 위치 에너지

로보트 팔의 위치 에너지를 P라 하고 Link i의 위치 에너지를 Pi라 하면 다음과 같다.

$$Pi = -mi\ G\ Ro = -mi\ G(Ao\ \bar{Ri});i=1,2,...,n$$

여기서 G 는 중력 low vector (gx,gy,gz,0)이고 기준 좌표계에서 설명 한다. 로보트 는 해면을 기준으로 G=(0,0,-|g|,0)이고 G는 중력가속도(9.8062 M/sec²)이다 로보트의 총 위치 에너지는 각 관절의 위치 에너지의 합으로 구할수 있다.

$$P = \sum_{i=1}^{n} Pi = \sum_{i=1}^{n} -mi\ g\ (Ao\ \bar{Ri})\ (2-11)$$

iv) 운동의 Lagrange-eluer 함수

$$\tau i = \sum_{k=1}^{n}\sum_{m=1}^{i} Tr(Uji\ Jj\ Uji^{t})\ddot{q}k +$$

$$\sum_{j=1}^{n}\sum_{k=1}^{j}\sum_{m=1}^{j} Tr(Ujkm\ Ji\ Uji^{t})\dot{q}k\ \dot{q}m$$

$$- \sum_{j=1}^{n} mj\ G\ Uji\ \bar{Ri}\ ;i= 1,2,...,n \quad (2-12)$$

식(2 - 12)를 일반적 으로 간단하게 하면 다음과 같다.

$$\tau = D(\theta)\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta)\ (2-13)$$

여기서
G(θ) = N X 1 중력 부하 벡터

H(θ ,θ̇)= N X 1 Coriolis 그리고 원심력 벡터
D (θ) = N X N Inertial accelation - related 등방 Matrix
τ = joint 모터의 토오크 혹은 힘 이다.

II-2 . KON-KUK I로보트의 2관절의 Largrange-Euler Dynamics의 해

i 관절의 Kinematic 구조해석에서 Homogeneous Transformation matrix Ai-1 을 얻는다.

$$Ai-1 = \begin{bmatrix} Ci & -Si & 0 & LCi \\ Si & Ci & 0 & LSi \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

단; Si=sinθ i
Ci=cosθ i

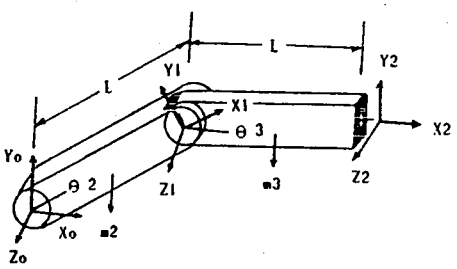

그림 2. KON-KUK I 로보트의 2,3 관절

Joint 변수;θ 2,θ 3 관절의 무게;m2,m3 관절의 변수 ;α 2=α 3=0 ;d2=d3=0;a2=a3=180 mm =L

식 (2-8)을 사용하면 다음과 같다.

$$U22=Q2\,A1=\begin{bmatrix} -S2 & -C2 & 0 & -LS2 \\ C2 & S2 & 0 & LS2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

같은 방법으로 U32 과 U33 를 구하면 다음과 같다.

$$U32=\begin{bmatrix} -S23 & -C23 & 0 & -L(S23+S2) \\ C23 & -S23 & 0 & L(S23+C2) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$U32=\begin{bmatrix} -S23 & -C23 & 0 & -LS23 \\ C23 & -S23 & 0 & LS23 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

J inertia tensor matrix를 구하면 다음과 같다.

$$J2=\begin{bmatrix} 1/3m2L^2 & 0 & 0 & -\tfrac{1}{2}m2L \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\tfrac{1}{2}m2\,L & 0 & 0 & m2 \end{bmatrix}$$

$$J3=\begin{bmatrix} 1/3m3L^2 & 0 & 0 & -\tfrac{1}{2}m3L \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\tfrac{1}{2}m3L & 0 & 0 & m3 \end{bmatrix}$$

Initial Accelation related 을 구합면 다음과 같다.

$$D22 = Tr\ U22\ J2\ {}^{t}U22 + Tr\ U32\ J3\ {}^{t}U32$$
$$= 1/3m2\,L^2 + 4/3m3\,L^2 + m3\,C3\,L^2$$
$$D23 = D32 = Tr\ U33\ J3\ {}^{t}U32$$
$$= 1/3m3\,L^2 + \tfrac{1}{2}m3\,L^2\,C3$$
$$D33 = Tr\ U33\ J3\ {}^{t}U33 = 1/3m3L^2$$

Corioly 힘과 원심력 항을 계산하면 다음과 같다.

i=2 일때

$$H2 = \sum_{k=2}^{3}\sum_{m=2}^{3} H2km\ \dot{\theta}k\,\dot{\theta}m = H222\ \dot{\theta}1^2 + H223\ \dot{\theta}2\dot{\theta}3 + H232\dot{\theta}2\dot{\theta}3 + H233\dot{\theta}3^2$$

식(2-10)에서 Ujkm 의 정의를 사용하여 Hjkm 을 구한다. 그러므로 joint 2에서

$$H2 = \tfrac{1}{2}m3S3L^2\,\dot{\theta}2^2 - m3S3L^2\,\dot{\theta}2\dot{\theta}3$$

i=3 에서는

$$H3 = \sum_{k=2}^{3}\sum_{m=2}^{3} H3km\ \dot{\theta}k\,\dot{\theta}m = H233\ \dot{\theta}2^2 + H323\,\dot{\theta}2\dot{\theta}3 + H332\dot{\theta}2\dot{\theta}3 + H333\dot{\theta}3^2 = \tfrac{1}{2}m3\,S3\,L^2\,\dot{\theta}2^2$$

중력항은 다음과 같다.

$$G = (G2, G3)^{t}$$

$$G2 = m2\,G\,U22\,\bar{R}2 + m3\,G\,U32\,\bar{R}23$$
$$= -\tfrac{1}{2}m2\,G\,L\,C2 - \tfrac{1}{2}m3\,G\,L\,C23 - m3\,G\,L\,C2$$

$$G3 = m3\,G\,U33\,\bar{R}3$$
$$= -\tfrac{1}{2}m3\,G\,L\,C23$$

마지막으로 2관절 로보트의 운동 lagrange-Euler방정식은 다음과 같다.

$$\tau = D(\theta)\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta)$$

$$\begin{bmatrix} \tau 2 \\ \tau 3 \end{bmatrix} = \begin{bmatrix} 1/3m2L^2 + 4/3m3L^2 + m3C3L^2 & 1/3m3L^2 + \tfrac{1}{2}m3C3L^2 \\ 1/3m3\,L^2 + \tfrac{1}{2}m3C3\,L^2 & 1/3m3\,L^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}2 \\ \ddot{\theta}3 \end{bmatrix} + \begin{bmatrix} -\tfrac{1}{2}m3\,S3\,L^2\,\dot{\theta}^2 3 - m3\,S3\,L^2\,\dot{\theta}2\dot{\theta}3 \\ \tfrac{1}{2}m3\ S3\,L^2\,\dot{\theta}^2 2 \end{bmatrix}$$

$$+ \begin{bmatrix} -\tfrac{1}{2}m2\,G\,L\,C2 - \tfrac{1}{2}m3\,G\,L\,C23 - m3\,GL\ C2 \\ -\tfrac{1}{2}m3GLC23 \end{bmatrix}$$

II-3. Controller의 설계

본 논문은 정확한 위치및 속도를 제어하기 위하여 M R A S(Model-Reference Adaptive System)을 채택하였다.

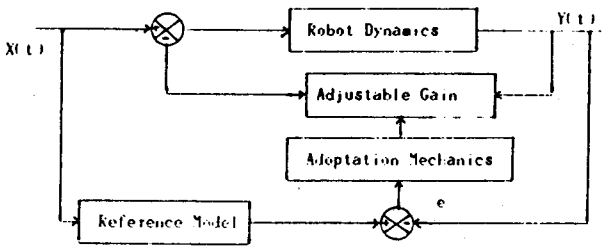

그림 3 .MRAS 블록도

본 논문에서 채택한 MRAS방식은 수행하기 쉽기 때문에 가장 널리 사용되고 있다.
먼저 계산된 값과 실제로 얻은 System값을 비교하여 그의 오차를 다시 System으로 출력하여 실제 원하는 응답을 효과적으로 쉽게 구할 수 있다.
이 실험에서는 Dynamics방정식으로 계산된 위치와 속도 값과 Feedback된 값을 비교 하여 원하는 Gain을 쉽게 로보트에 전달할 수 있었다.
본 논문에서는 Sensor를 사용하지 않는 대신 모터의 전류를 제어하므로써 원하는 토오크를 발생시켜 보다 경제적이고 쉽게 제어 했다.
이를 선택한 이유는 모터에서 많은 전류를 소비하므로 D/A 변환기의 전류로는 전류 가 미세하기 때문에 D669 라는 transistor을 사용하여 최대 1.5Ampere까지 증폭하여 모터의 동작 전원으로 사용하였다. 이때 전류는 D/A 변환기출력에 따라 변하게 된다.

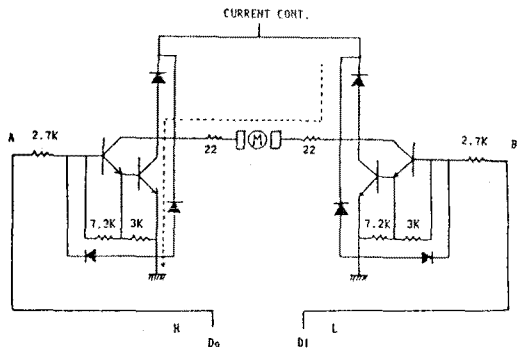

그림 4. 모터 구동 회로

Tachometer 출력 전압이 0보다 작을 경우 level up 시켜 0보다 크게 만들어 준다. 속도가 0일 경우 2.5 V를 유지하게 하고 시계 반대방향의 최대 속도일때 +5 V, 시계 방향의 최대 속도일때 0 V가 되도 록 하여 속도와 방향을 알수 있게 속도 검출기를 제작하였다. 본 논문에서 사용한 위치 검출은 Potentiometer를 사용하여 궤환을 시켰다. 여기서 Potentiometer 를 관절에 부착하여 기어와 wire 에서 생기는 오차가 있 더라도 관절이 움직인 위치 정보를 정확히 궤환하게 제작되어 있다.

II-4. Interface 의 설계 와 지원 Software 의 설계

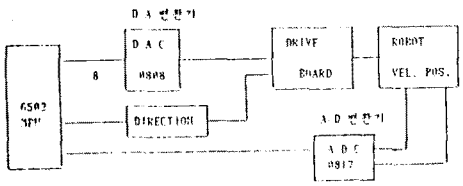

그림 5. 전체 시스템 블록도

그림5 는 본 논문에서 사용한 전체 시스템 블럭도이다. 여기서D/A 변환기, A/D 변환기,구동 보드 Interface 가 필요했다. APPLE II+ 컴퓨터에서 명령 데이타가 출력되면 D/A 변환기를 거쳐 Analog신호로 바뀌고 또,데이타에 따라 방향 제어 기에서 방향 성분이 주어지면 출력된 속도와 위치를 A/D 변환기 를 거쳐 오차를 다시 D/A 변환기 로 출력시켜 error 를 보상한다. 여기서 방향제어기는 I/O Selector 를 사용하여 $C300에 Buffer 를 사용하여 방향 데이타를 출력시켜 방향 성분을 결정한다.

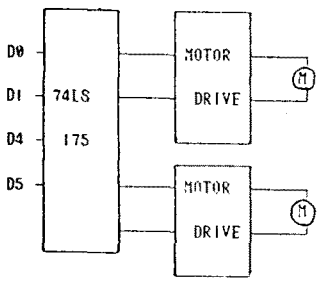

그림 6. 방향 제어기

HEX값		Link2	Link3
1	1	+	+
1	2	+	-
2	1	-	+
2	2	-	-

표 1. 방향 성분

본 논문에서 사용한 D/A 변환기는 DEVICE SELECT 신호가 떨어지면 LSB 와 MSB 가 latch를 통하여 DAC 0808 에 들어와 들어온 데이타를 0-225 level로 나누어 출력한 고, 이 출력을 Op.Amp.로 증폭하여 최대 +15 V까지 출력되어 전류 제어기에 들 어간다. Dynamics 방정식을 프로그램하여 계산하고 이 결과치를 $2000 - $3FFF 에 저장하고 1 번지에 2.5msec 마다 한번씩 데이타를 출력시켜야 하고 2.5msec 마다 위치,속도 검출및 error의 보상을 출력 시키고 각 메모리에 저장하여야 한다.

III. 실험 및 고찰

Dynamics 방정식에 소요되는 시간을 미리 계산하여 메모리에 저장하여 실시간 제어를 용이하게 하였다. 그러나 2 관절을 수 십초 움직이기위해 메모리를 4K X 3bit 사용하므로 메모리의 확장 및 좁더 계산 속도가 빠른 마이므로 프로세서 가 필요하다. 여기서 Dynamics 방정식을 해석하므로 관절이 움직이 는 속도를 일정하게 할수 있었다. 또한 MRAS 제어 방식으 로 error를 보정하여 이의 신뢰도를 높였다. 위치 감지기는 위 치에 따라 0 - 5 V의 전압으로 감지하여 A/D 변환기를 거쳐 메모리에 저장되며 속도 감지기도 속도에 따라 0 -5 V로 변화하 는데 2.5 V를 기준으로 전압이 높을수록 반 시계방향, 낮을수록 시계방향으로 관절이 움직이 는 것을 알수있게 제작되었다.

IV. 결 론

본 논문에서는 본 연구실에서 제작된 KON - KUK I 로보트를 사용하여 전류 제어에 의한 Dynamics Controller를 실현하였다. 각 관절의 운동 속도는 매우 일정하였고 MRAS제어 방식으로 동작도 매우 부드러웠다. 이것을 직접 이용하면 Trajectory등에 매우유용할 것으로 사료된다. 그러나 전류 제어 능력에 한계가 있고 모터 전류에 한계가있어 고속의 작업을 수행하기 힘들었다. 또한 계산기의 계산 속도 때문에 로보트Dynamics 제어를 실시간 제어가 불가능 하므로 이를 위하여 계산기의 속도를 올 려야 하고 Multy-processor등을 사용한 컴퓨터의 도입이 필요하다. 이 실험에서 Dynamics방정식을 시간별로 나누어 계산하고 데이타를 저장하여 수행 하였 으므로 실시간 제어가 가능하였고, 많은 메모리를 사용 하였으므로 그보다 더오래 작동 하기위해서는 더많은 양의 메모리가 필요 하다. 때문에 컴퓨터의 메모리 확장이 필요하다고 사료 된다. 본 논문에서 사용한 MRAS 제어 방식의 Algorithm은 간단하고 응답이 매우 좋았으므로 이 제어방식이 널리 사용될 것으로 생각된다.

참 고 문 헌

[1] James E. poulin " Practical Considerations in DC Motor and Amplifier Selection," IEEE TRANSACTIONS on Industry Applications, Vol. IA-20, No.5, September/October 1984.

[2] Symon,K.R."Mechaanics,"3rd Addition Addison-Wesley 1971. pp.107-130.

[3] Whiteney,D.E." The Mathematics of Coordinated Control of Porsthetic Arms and Mammipulator," ASEM. Jounal of Dynamics Systems,Measulment and Control. December,1972, pp.303-309

[4] C.S.George Lee" Robot Arm Kinematics, Dynamics,and Control," Computer, December, 1982, pp. 62-80.

[5] S.Dubowsky, D.T.Deforges " The Application of Model Referenced Adaptive Control to Robotic Manipulator," ASME. Jounal of Dynamics Systems, Measulment and Control. Septenber,1979, pp.193-200

[6] Richard P. Paul " Robot Manipulators Mathematics, Programing, and Control," The MIT Press 1982.

[7] R.P.Paul "Modeling, Trajectory Calculation, and Servoing of a Computer Controlled Arm," Stanford A.I.Lab. Memo AM-177