하계학술회의
논문85-3-5

관절형 로보트의 연속경로 제어를 위한 Interpolator 의 실현에 관한 연구

여 인 택

한국기계연구소
제어공학연구실

1. 서 론

로보트의 운동을 나타내는 방정식들은 모두 비선형이므로 계산상에 많은 문제가 있다.

현재 로보트가 어떤 지시된 경로를 따라가게 하는 방법으로는 경로의 중간에 많은 수의 점을 선정하여 이위치들에 대한 로보트 관절각을 구하여 여러번의 위치제어를 연속적으로 수행하여 실현하고 있는 실정으로 이경우에 점들 간의 경로는 제어하고 있지 못하므로 점들 사이의 오차는 제어하지 못하여 점들 사이의 오차는 제어하지 못하여 점들 마다의 속도가 갑자기 변할 수 있으므로 운동방정식에 나타나는 중력항등의 영향을 더 크게 받게 된다. 본 연구에서는 위의 두가지 문제를 해결하기 위한 알고리즘을 제시하였으며 빗-슬라이스(Bit-slice)구성 방법을 제시하여 실용성을 입증해 보였다.

2. 본 론

문제의 구성 및 그 해결 방법을 설명해 보면 로보트 운동방정식의 해을 구하는 데에는 여러가지 방법이 있다. 그런데 근래에 로보트의 운동을 나타내는 위치와 자세를 분리해서 실현 시키려는 시도도 되고 있다. [1] 이러한 점들에 착안하여 본연구에서는 연속경로제어를 충실히 실행하기 위해서 각관절들의 움직임에 대한 실시간 명령을 내기 위한 방안으로 아래와 같은 방법을 구상하였으며 설명을 위해 6축 엘보우 타입(Elbow Type)을 선택하였다.

위치제어를 위해서 1,2,3축을 자세제어를 위해 4,5,6축을 쓰게 되면 우선 위치제어를 위해 $\theta 4, \theta 5, \theta 6$를 고정시키면 위치 PX,PY,PZ의 미소변화에 대한 $\theta 1, \theta 2, \theta 3$의 변화량은

$$\begin{matrix} \Delta\theta 1 \\ \Delta\theta 2 \\ \Delta\theta 3 \end{matrix} = \begin{matrix} IJK \\ LMN \\ PQ \end{matrix}^{-1} \begin{matrix} \Delta X \\ \Delta Y \\ \Delta Z \end{matrix}$$

로 부터 구할 수 있으며 자세제어에 있어서도 마찬가지의 방법으로 $\Delta\theta 4, \Delta\theta 5, \Delta\theta 6$를 구할 수 있으나 이들은 3×3 메트릭스의 인버스를 구해야하므로 PX,PY,PZ의 변화량을 아주 작게하여 경로제어에 주안을 두고자하는 경우에 이들의 계산시간은 너무 크므로 실제로 사용하는 데에는 부적합하므로 다른 방법을 모색하게 되었다.

본연구에서는 실시간적으로 기준이 되는 경로상의 위치를 만들고 로보트의 끝점이 경로상의 계속해서 변하는 위치에 있기 위해서 $\theta 1, \theta 2, \theta 3$를 $\delta\theta 1, \delta\theta 2, \delta\theta 3$를 기본단위로 자세도 $\theta 4, \theta 5, \theta 6$를 $\delta\theta 4, \delta\theta 5, \delta\theta 6$를 기본단위로 계속 구해나가는 일을 반복함으로써 원하는 위치와 자세제어를 가능하게 하는 것이다. 또한 기준위치를 발생시키는 문제는 직각좌표계에서의 문제로 한정되므로 3차원직선보간과 2차원원호보간에 대해서 DDA방법을 사용한다.[2]

기준위치를 발생시키는 데 필요한 정보는 수행속도와 시점과 종점에 대한 직각좌표계로 나타낸 값이다. 수행속도를 OUR로 하고 시점과 종점을 각각 (X_s, Y_s, Z_s), (X_e, Y_e, Z_e)라 하면 3차원보간의 경우 각각의 차의 절대값을 IVPX, IVPY, IVPZ라 하면 OVR-IVPX에 비례한 속도로 PX를 시점 X_s에서 종점 X_e까지 변화시키고, PY는 OVR·IVRY에 PZ는 OUR·IVPZ에 비례한 속도로 각각 로보트 위치오차의 최소설정단위 만큼씩 더하기나 빼감으로써 기준위

차들을 발생시키며 이때 시점과 종점에 대해서 가감속을 행하는 것이 좋다.

식(1)에서 ΔX, ΔY, ΔZ가 생기는 원인은 기준위치가 계속해서 주어진 경로상에서 변하기 때문에 또한 자세제어를 위해 $\theta 4$를 움직이므로써 생기는 두가지의 원인이 있다.

이와같은 차이를 계속 보정해가며 경로제어를 하기 위하여 다음의 방법을 택하였다.

로보트의 원점에서 끝점까지의 거리를 R이라하면

$$R^2 = a_1^2 + a_2^2 + a_3^2 + 2a_1a_3c3 + 2a_1a_3c34 + 2a_2a_3c4$$

이므로 $\theta 4$가 일정하다면 $\theta 3$에만 의존하는 것을 알 수 있다. 매 시점에서의 기준위치를 X1, Y1, Z1 이라고 하고 $\theta 1, \theta 2, \theta 3, \theta 4$에 의해서 결정된 현재 로보트의 위치를 X2, Y2, Z2라 하고 X1, Y1, Z1에 의해 결정된 R을 CR, X2, Y2, Z2에 의해 결정된 R을 GR이라 하고

$$\Delta CR^2_{i+1} = CR^2_i - CR^2_{i+1}$$

$$\Delta GR^2_{i+1} = GR^2_i - GR^2_{i+1}$$

$$HCR_{i+1} = HCR_i + \Delta CR^2_{i+1} - \Delta GR^2_{i+1}$$

$HCR_0 = 0$ 이라 하면

$\theta 3$를 조정해 CR = GR이 되게 하는 일은 $HCR_{i+1} = 0$로 되게 하므로써 실현가능하다. $\theta 3$의 변화로 인한 R^2의 변화는

$\Delta R^2 = -2a_1(a_2 s3 + a_3 s34)\cdot\Delta\theta 3$ 이므로

$ARM = a_2 s3 + a_3 s34$ 라 하면

IF $HCR_i < 0$ .AND. $ARM_i > 0$
THEN $\theta 3_{i+1} \leftarrow \theta 3_i + \delta\theta 3$

IF $HCR_i < 0$ .AND. $ARM_i < 0$
THEN $\theta 3_{i+1} \leftarrow \theta 3_i - \delta\theta 3$

IF $HCR_i > 0$ .AND. $ARM_i > 0$
THEN $\theta 3_{i+1} \leftarrow \theta 3_i - \delta\theta 3$

IF $HCR_i > 0$ .AND. $ARM_i < 0$
THEN $\theta 3_{i+1} \leftarrow \theta 3_i + \delta\theta 3$

와 같이 하므로써 실현시킬 수 있다.

다음에 $\theta 2$를 사용해서 Z1 = Z2가 되게 하기 위해서 $\theta 2$의 미소변화로 인한 ΔPZ의 변화를 보면

$$\Delta PZ = (a_1 c2 + a_2 c23 + a_3 c234)\cdot\Delta\theta 2 = ZAM\cdot\Delta\theta 2$$

이므로 동작범위가 ZAM이 항상 양의 값을 갖도록 제한하면

IF Z2 .EQ. Z1 THEN $\theta 2_{i+1} \leftarrow \theta 2_i$
Z2 .GT. Z1 THEN $\theta 2_{i+1} \leftarrow \theta 2_i - \delta\theta 2$
Z2 .LT. Z1 THEN $\theta 2_{i+1} \leftarrow \theta 2_i + \delta\theta 2$

가 되도록 $\theta 2$를 증감시킨다.

다음에 $\theta 1$을 사용해서 IVPX ≥ IVPY인 경우에는 X1 = X2로 IVPX < IVPY인 경우에는 Y1 = Y2로 되게 하기 위해 $\theta 1$의 미소변화에 따른 ΔPX, ΔPY의 변화를 보면

$$\Delta PX = -PY\cdot\Delta\theta 1$$

$$\Delta PY = PX\cdot\Delta\theta 1$$

이므로 만약 IVPX < IVPY인 경우라면

IF Y2 = Y1 THEN $\theta 1_{i+1} \leftarrow \theta 1_i$
Y2 > Y1, X1 > 0 THEN $\theta 1_{i+1} \leftarrow \theta 1_i - \delta\theta 1$
Y2 > Y1, X1 < 0 THEN $\theta 1_{i+1} \leftarrow \theta 1_i + \delta\theta 1$
Y2 < Y1, X1 > 0 THEN $\theta 1_{i+1} \leftarrow \theta 1_i + \delta\theta 1$
Y2 < Y1 - X1 < 0 THEN $\theta 1_{i+1} \leftarrow \theta 1_i - \delta\theta 1$

이 되도록 $\theta 3$를 증감시켜 간다.

다음에 자세제어를 하는 방법을 생각하자.

경우의 수를 줄여 설명을 간단히 하기 위해

$-90° < \theta 2 + \theta 3 + \theta 4 < 90°$

$0 < \theta 5, \theta 6 < 180°$로 제한하면

자세를 나타내는 식들로 부터

$C5 = S1\cdot ax - c1\cdot ay$

$ag = S234\cdot S5$

$C6 = C234\cdot Oy - S234(c1\cdot ox + s1\cdot oy)$

를 얻고 이를 만족하도록 θS의 경우는

$SMC = S1\cdot a_y$로 한 경우에

IF C5 = SMC THEN $\theta 5_{i+1} \leftarrow \theta 5_i$
C5 > SMC THEN $\theta 5_{i+1} \leftarrow \theta 5_i + \delta\theta 5$
C5 < SMC THEN $\theta 5_{i+1} \leftarrow \theta 5_i - \delta\theta 5$

를 만족하도록 $\theta 5$를 조정하고

$\theta 4$의 경우는 $SS = S234\cdot S5$라 하면

IF $SS = a_y$ THEN $\theta 4_{i+1} \leftarrow \theta 4_i$
$SS = a_y$ THEN $\theta 4_{i+1} \leftarrow \theta 4_i - \delta\theta 4$
$SS < a_y$ THEN $\theta 4_{i+1} \leftarrow \theta 4_i + \delta\theta 4$

를 만족하도록 $\theta 4$를 조정하며

$\theta 6$의 경우는 $CSC = C234\cdot O_y - S234\cdot(C1\cdot O_x + S1\cdot O_y)$라 하면

IF C5 = CSC THEN $\theta 6_{i+1} \leftarrow \theta 6_i$
C5 > CSC THEN $\theta 6_{i+1} \leftarrow \theta 6_i + \delta\theta 6$
C5 < CSC THEN $\theta 6_{i+1} \leftarrow \theta 6_i - \delta\theta 6$

가 되도록 하면 된다.

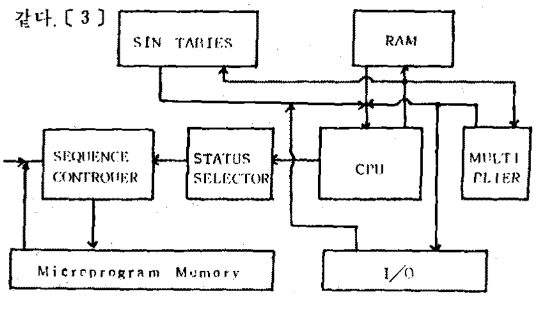
본 연구에서 주장하고자 하는 알고리즘의 입증을 위해 프로그램을 짜서 여러가지 경우에 대한 시뮬레이션을 하였다. 시뮬레이션에 있어서는 하드웨어 구

성을 염두에 두었고 $a1 = 500$ ㎜, $a2 = 400$ ㎜, $a3 = 40$ ㎜, $-180° < \theta 1 < 180°$로 가정하고 $\delta\theta 1 \sim \delta\theta 6 = 0.006°$로 하였다.

이러한 값들을 가지고 위치오차는 0.2 ㎜ 이하의 오차만을 허용하고 자세도 실지로 사용하는 데 무리가 없는 것을 목표로 하였다. 예상했던 대로 오차는 반복횟수가 많을수록, $\delta\theta 1 \sim \delta\theta 6$가 작을수록 적었다. 그러나 수행시간을 짧게할 경우에는 기준위치의 변화율이 크게 되어 $\delta\theta 1 \sim \delta\theta 6$의 크기를 크게 해야하며 너무 크게 되면 이 알고리즘을 쓸 수 없게 된다. 그러나 실제 로보트가 움직이는 상황을 생각해서 위의 가정을 가지고 경로의 중간 어디에서나 0.2 ㎜이하의 위치오차를 가지고 50 ㎝/sec 이 속도로 수행할 수 있음을 알 수 있었다. 왜냐하면 알고리즘의 전체 수행 단계(STEP) 수가 충분히 잡아서 100단계라 하고 하드웨어로 구성할 때 동작할 수 있는 주파수를 5M Hg로 생각하고 직각 좌표계에서 0.1 ㎜ 변화할 때마다 계속 반복해서 1초에 50 ㎝를 움직이는 일을 한다고 하면 직각좌표계에서 설정한 최소단위를 0.1 ㎜로 생각할 경우 0.1 ㎜의 증분이 발생하는 율은

500 ㎜ × 100 stop × 1/0.1 ㎜ = 500 k stop 이므로 매 0.1 ㎜의 변화당 10회의 반복이 가능한데 본 연구에서는 5회로 하였으므로 50 ㎝/sec의 수행이 가능하게 된다.

이 알고리즘을 하드웨어로 구성하는 데 있어서 고려해야 할 점은 어느 디지탈 회로 구성에서와 마찬가지로 신뢰성과 정확성을 생각해야 한다. 특히 어떤 계산을 반복하는 경우에 오차의 누적을 방지해야 하는 것은 무엇 보다도 중요하다. 이를 위해서 가능하면 SIN과 COS Term들은 원칙적으로 Table로 구성해서 처리하도록 하였으나 실제의 구성에 있어서는 설계자 마다 다른 방법을 모색할 수 있으리라 생각된다. 설계를 위한 기본 블럭도는 아래와 같다. [3]



3. 결 론

본 연구에서는 6축 관절형 로보트의 연속경로제어를 위한 보간기(Interpolator)의 알고리즘과 이의 실현가능성을 시뮬레이션을 통하여 입증하였으며 이 알고리즘의 실현을 위한 하드웨어 블럭도를 제시하였다. 본 연구에서 목표로 삼은 경로오차는 0.1 ㎜이며 수행속도는 50 ㎝/sec이다. 특히 이와 같은 방법을 택할 경우 메인 프로세서(Main Processor)의 부담을 상당히 줄일 수 있는 효과가 있다.

참 고 문 헌

(1) M. Takano, "A new method of Sdution of synthesis problem of a robot and its application to computer simulatiow system", 14 th ISIR, P261 ~ 268, 1985

(2) Yoram Koren, "Computer Control of Manufacturing Systems", McGraw-Hill, 1983, P115 ~ 141

(3) 여인택, 노태석, "마이크로 프로세서를 이용한 인터플레이터", 전기학회논문지, Vol. 33 No. 2 1984. 2