

두 대의 스카라로봇으로 구성된 시스템의 충돌회피에 대한 연구

A study on Dual-Arm SCARA robot system for collision avoidance

홍용준, 양승원, 황찬영, 김성권

삼성전자 생산기술센터 자동화연구소 (Tel:0331-200-2427 Fax:0331-200-2434)

Abstracts 본 논문은 작업공간을 공유하는 두 대의 스카라 로봇으로 이루어진 Dual-Arm SCARA 로봇 시스템과 여러 대의 로봇을 제어할 수 있는 로봇제어기로 동시에 독립적인 작업을 하는 경우 두 로봇의 동작에 따른 로봇 ARM의 모델링을 실시간으로 처리하여 상대 로봇과의 충돌이 예상되는지를 실시간 검색하여 두 로봇간의 충돌이 발생하지 않도록 하는 충돌검출 방법과, 두 대의 로봇이 충돌가능성으로 인하여 원하는 작업을 수행할 수 없는 상태가 되는 경우 작업을 원활하게 이루어지도록 하는 충돌회피를 위한 로봇의 궤적을 생성하여 로봇을 이동시키고 다시 원래의 위치로 돌아 올 수 있는 방법을 구현하였다.

Keywords Collision Avoidance, Robot Modeling, Trajectory

1. 개발 환경 및 배경

본 연구의 대상은 삼성전자에서 개발한 로봇제어기와 Dual-Arm SCARA 로봇으로 구성된다. 로봇제어기는 6축 이하의 로봇을 동시에 3대까지 제어할 수 있는 범용 로봇제어기이다. 로봇제어기의 운영체제는 Real Time Multi Tasking이 되는 LynxOS를 사용하였으며 사용자 인터페이스, 프로그램의 해석, 프로그램의 실행, 궤적생성 그리고, 시스템 환경 감시 등의 목적으로 Intel 80486 Processor를 사용하였으며, 모터를 구동하는 장치로 당사에서 개발한 내장형 디지털 서보를 사용하였다. 32bits DSP를 장착한 디지털 서보는 2개, 또는 4개의 모터를 동시에 제어 할 수 있다. Dual-Arm SCARA의 개발 배경은 로봇을 이용하여 제품을 생산하는 line을 구성하는데 있어서 단위작업을 하기위한 로봇 cell을 구성할 때 동작영역을 고려하여 cell의 크기를 결정하게 된다. 이러한 이유는 서로 독립적인 로봇 간에 대한 위치정보를 얻기가 어렵고, 중첩된 작업공간을 갖도록 로봇을 설치하는 경우 로봇의 동작교시, 프로그램 검증, 프로그램 실행시에 상대 로봇 위치에 대한 고려를 충분하게 하지 않는 경우 로봇간의 충돌을 피하면서 작업을 하는 것이 쉽지 않다. 따라서 따라서 당사에서는 Dual-Arm으로 구성된 SCARA와 여러 대의 로봇을 제어할 수 있는 범용제어기를 사용하여, 두개의 Arm이 서로 충돌을 일으키지 않고 작업을 할

수 있도록 하는 방법을 구현하였다. Dual-Arm SCARA의 구성은 그림 1.과 같다.

2. 로봇의 모델링

Dual-Arm SCARA의 모델링은 로봇의 1, 2ARM이 선택적인 평면에 대하여 동작을 하는 SCARA의 2차원적인 특징을 이용하여 X-Y의 2차원 평면에 대하여 이루어 졌다. 그림 1.의 로봇을 X-Y 평면에 대하여 modeling 한 형태는 그림 2.와 같다. 그림 2.에서 보듯이 Dual-Arm과 BASE부는 5개의 직사각형 형태로 표시되어 있다. 따라서 로봇의 모델링은 다섯 개의 직사각형에 대한 2차원 modeling으로 정리된다.

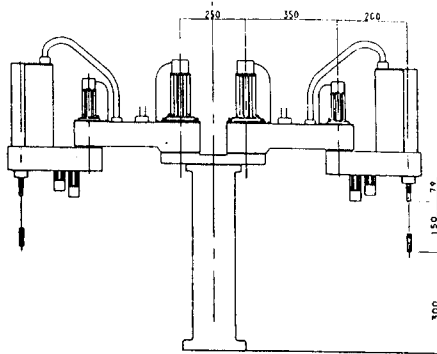


그림 1. Dual-Arm SCARA 로봇

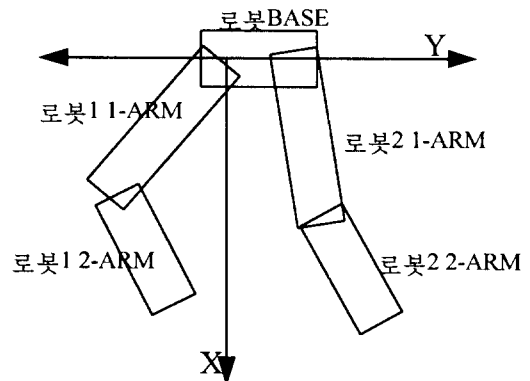


그림 2. Dual-Arm SCARA를 X-Y 평면에서 모델링한 결과

2.1. 2차원에서의 직사각형 모델링

X-Y평면으로 표현되는 2차원에서 직사각형의 모델링은 직사각형을 이루는 네 개의 직선 식의 표현으로 이루어진다. 그림 3.과 같이 X-Y평면 상에 직사각형을 이루는 네 개의 직선은 식 (1)과 같이 일반적인 형태의 직선 식으로 표현할 수 있다.

$$ax + by + c = 0 \quad (1)$$

식 (1)의 a, b, c 를 구하고, X-Y 좌표계상에서 각 직선의 범위 $X_{min}, X_{max}, Y_{min}, Y_{max}$ 를 구하여 직사각형의 Modeling을 하게 된다. 네 개의 직선 식은 인접한 두점 $(p_1, p_2), (p_2, p_3),$

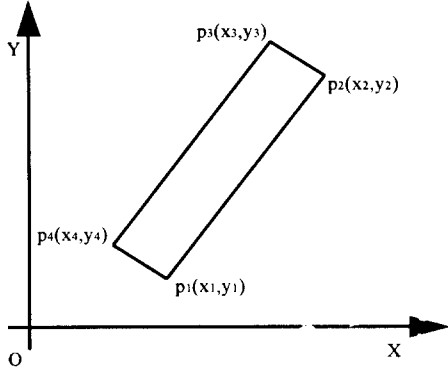


그림 3. X-Y평면에 있는 직사각형의 modeling

$(p_1, p_2), (p_2, p_3)$ 으로 이루어진다. 두점 p_1, p_2 가 이루는 직선의 a, b, c 와 범위는 다음과 같이 구한다. 먼저 $p_1(x_1, y_1), p_2(x_2, y_2)$ 를 식 (1)에 대입하면 식 (2) (3)과 같다.

$$ax_1 + by_1 + c = 0 \quad (2)$$

$$ax_2 + by_2 + c = 0 \quad (3)$$

식 (2)를 c 에 대하여 정리하여 식 (3)에 대입하면 식 (4)를 구할 수 있다.

$$\frac{a}{b} = \frac{y_1 - y_2}{x_2 - x_1} \quad (4)$$

식 (4)에서 a 를 y_1, y_2 라고 하면 b 는 $x_2 - x_1$ 이됨을 알 수 있다. 식 (4)에서 구한 a, b 를 식 (2)에 대입하여 c 를 구하면 식 (5)와 같이 된다.

$$c = x_1 \times y_2 - x_2 \times y_1 \quad (5)$$

그리고, X-Y좌표계 상에서 직선의 구간은 식 (6)과 같다.

$$\begin{aligned} x_{min} &= x_1, & x_{max} &= x_2 \\ y_{min} &= y_1, & y_{max} &= y_2 \end{aligned} \quad (6)$$

식 (4)~(6)과 같은 방법으로 네 개의 직선을 구하고 X-Y평면에 대한 각 직선의 구간을 구하여 직사각형 Modeling을 한다. 이러한 방법을 사용하여 두 대의 로봇에 대한 Modeling을 한다.

2.2 로봇 몸체에 대한 Modeling

먼저 그림 2.와 같이 두 대의 스카라 로봇 중 하나의 로봇 BASE부를 X-Y 좌표에 대한 원점으로 하여 로봇 Modeling의 원점으로 정한다. 이 원점을 기준으로 하여 각 스카라 로봇의 1, 2 ARM에 대한 Arm 길이와 두께를 가지고 직사각형 Modeling을 한다. 먼저 X-Y평면에 대한 각 Arm의 직사각형 Modeling을 위하여 각 Joint의 위치를 구해야 한다.

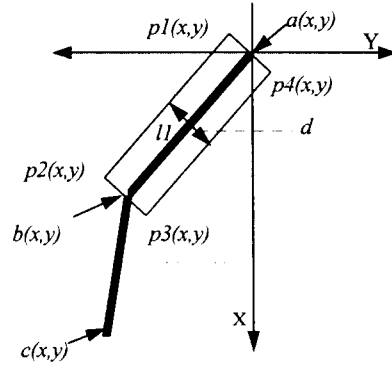


그림 4. SCARA1-Arm의 직사각형 Modeling

그림 3.에서 보듯이 X-Y좌표에 대한 각 Joint의 위치는 1, 2-Arm에 대한 Forward Kinematics를 응용하여 구하게 되는데 입력으로 로봇의 1, 2관절에 대한 각도 θ_1, θ_2 와 관절의 길이 l_1, l_2 , Arm의 두께 d_1, d_2 를 입력으로 식 (7)을 구한다.

$$a_1 = 0.0$$

$$a_2 = 0.0$$

$$b_1 = l_1 \times \cos(\theta_1)$$

$$b_2 = l_1 \times \sin(\theta_1)$$

$$c_1 = b_2 + l_2 \times \cos(\theta_1 + \theta_2)$$

$$c_2 = b_1 + l_2 \times \sin(\theta_1 + \theta_2) \quad (7)$$

위의 식 (7)의 값과 d_1 의 값으로 직사각형의 모서리의 x, y 값인 p_1, p_2, p_3, p_4 의 x, y 를 구하기 위하여 점 a, b 가 이루는 직선의 기울기를 구하면 식 (8)과 같다.

$$s = \frac{b_2 - a_2}{b_1 - a_1} \quad (8)$$

두께 l_1 은 로봇 a, b 가 이루는 직선과 직각을 이루므로 Arm의 두께를 계산하기위한 기울기는 식 (8)의 기울기와 직각을 이룬다. 이러한 사항을 고려하여 X-Y좌표계 상에서 Arm 두께 l_1 의 x, y 성분 l_{1x}, l_{1y} 를 구하면 식 (9)와 같이 구할 수 있다.

$$l_s = -\sqrt{\frac{\left(\frac{l_1}{2}\right)^2}{s^2 + 1}}$$

$$l_a = -l_s \times s \quad (9)$$

1-Arm의 두께 l_1 과 기울기 s , $a(x, y)$, $b(x, y)$ 를 이용하여 다음과 같이 네 개의 모서리에 대한 X-Y좌표계 상의 값 $p_1(x, y)$, $p_2(x, y)$, $p_3(x, y)$, $p_4(x, y)$ 를 구한다.

$$\begin{aligned} p1_x &= a_x + l_{ax}, p1_y = a_y + l_{ay} \\ p2_x &= b_x + l_{bx}, p2_y = b_y + l_{by} \\ p3_x &= b_x - l_{bx}, p3_y = b_y - l_{by} \\ p4_x &= a_x - l_{ax}, p4_y = a_y - l_{ay} \end{aligned} \quad (10)$$

3. 충돌검출 알고리즘

Dual-Arm SCARA의 충돌검출방법은 각 Arm에 대한 직사각형 Model이 다른 Arm의 직사각형 Modeling 결과와 식(6)의 주어진 범위 내에서 중첩되는 지로 판단한다. 그림 2.에서 보면 "로봇1 1-ARM", "로봇1 2-ARM", "로봇2 1-ARM", "로봇2 2-ARM", "로봇 BASE"가 각ARM의 직사각형 범위 내에서 중첩되는지를 검사하는 것으로 이루어진다. 표 1.은 Modeling된 Arm과, BASE간에 충돌이 일어나는지를 검사해야하는 관계를 표시하였다.

표 1. Arm의 충돌검출 관계

	로봇1 1-ARM	로봇1 2-ARM	로봇 BASE
로봇2 1-ARM	○	○	×
로봇2 2-ARM	○	○	○
로봇 BASE	×	○	×

3.1 직사각형의 충돌검출 방법

각 직사각형 Modeling에 대한 충돌 검출방법은 직사각형을 이루는 각 직선이 주어진 범위 내에서 교점이 있는 지로 판단한다. 먼저 직사각형을 이루는 직선 식 (1)의 a , b , c 를 식(4)~(5)와 같이 구하고, 각 직선의 범위를 식(6)과 같이 구하였을 때, 두개의 직선을 식(11) (12)와 같이 표현 할 수 있는데, 이 두 직선의 두 직선의 교점을 구하고 그 교점이 직선의 범위 내에 존재하는지를 판별하여 충돌을 검출하게 된다.

$$a_1x + b_1y + c_1 = 0, x_{min} \leq x \leq x_{max}, y_{min} \leq y \leq y_{max} \quad (11)$$

$$a_2x + b_2y + c_2 = 0, x_{min} \leq x \leq x_{max}, y_{min} \leq y \leq y_{max} \quad (12)$$

먼저 식(11)을 x 에 대하여 정리한 후 식(12)에 대입하여 식(13)을 구한다.

$$y = \frac{a_2c_1 - a_1c_2}{a_1b_2 - a_2b_1} \quad (13)$$

식 (13)에서 분모에 해당하는 부분이 0에 근접하는 값인 경우에는 두 직선이 평행을 이루는 것으로 판단하여 계산을 하지 않고 다음 직선에 대한 계산을 한다. 식(13)에서 구한 y 값을 식(11) (12)의 범위 내에 있는지 검사하여 두개의 구간 내에 동시에 존재하면 두 직선이 교점이 있는 것이므로, Arm의 Modeling형태인 직사각형이 충돌 되는 것으로 판단한다. 위와 같은 방법으로 직사각형을 이루는 직선을 모두 검사하여 직사각형의 충돌을 검사한다.

3.2 Dual-Arm SCARA의 충돌검출

3.1에서와 같은 방법을 사용하여 표 1.의 조건에 따라 두개의 Arm이 상호 충돌이 발생하는지를 검사하게 된다. Dual-Arm SCARA의 충돌검출 알고리즘에 대한 플로우는 그림 5.와 같다. 먼저 두 Arm은 원점복귀를 하고 난후 로봇의 현재위치를 읽어서 Arm과 BASE에 대한 직사각형 Modeling 정보를 두 로봇이 공유하고 있는 기억장치에 저장한다. 다음으로 로봇을 구동하기위하여 단위시간마다 로봇의 궤적을 생성할 때 새로 계산되어진 로봇의 궤적에 대한 직사각형 Modeling을 하여 공유 기억장치에 있는 다른 로봇의 1, 2-Arm 및 BASE와 3.1에 제시한 방법에 의하여 충돌이 일어나는지를 검사하게 된다.

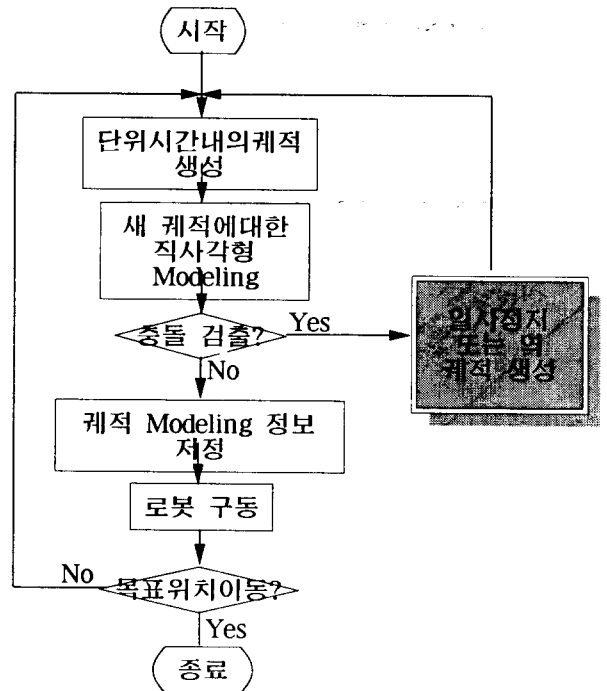


그림 5. 궤적생성과 충돌감지

이때 충돌이 발생하지 않는 경우에는 공유기억장치에 자신의 직사각형 Modeling정보를 저장하여, 상대 로봇이 사용할 수 있도록 한후, 로봇을 구동시켜서 이동한다. 반면, 충돌한다는 결과가 계산되면 생성한 궤적을 무시하고 로봇을 감속정지시켜서

로봇이 충돌하는 것을 방지한다. 두 대의 로봇이 공유 기억장치를 사용하기 때문에 공유기억장치에 대한 정보의 입출력은 상호배제하여 잘못된 정보의 입출력이 발생하지 않도록 한다. 그림5.의 기능은 Lynx OS에서 두개의 쓰레드로 생성되어 각각의 로봇에 대응된다.

4. Dual-Arm SCARA의 충돌회피 및 궤적수정

Dual-Arm SCARA를 이용하여 조립작업을 하는 경우 충돌검출이 되었을 경우 두 Arm의 궤적 생성방향이 충돌을 일으키는 방향인 경우 두 로봇 모두 감속정지하여 대기하는 상태에 놓이게 된다. 이러한 경우에 두 로봇 모두 원하는 목표위치로 이동하지 못하게 되는데 이러한 상태에서 어느 한 로봇의 궤적을 변경시켜서 다른 로봇이 동작 할 수 있도록 하는 방법이 고안되어야 한다. 그 방법으로 Modeling의 원점을 기준으로 로봇의 End-Effector가 원점에 가까이 있는 로봇을 Master로봇으로 설정하고, 다른 로봇에는 Slave로봇으로 설정하여 Master로봇은 현재위치에서 대기하고, Slave로봇이 현재 진행하던 궤적의 역으로 이동하여 단위동작의 시작위치까지 이동하는 방법을 사용하였다.

5. 결 론

본 논문에서는 SCARA로봇의 특성을 이용하여 궤적 생성시 실시간으로 충돌을 감지할 수 있는 방법을 구현하였다. 충돌감지 후 두 로봇이 서로 이동하지 못하는 상태인 경우에 역 궤적을 생성하는 방법이 모든 경우에 적용될 수 없기 때문에 로봇의 작업공정을 고려하여 동작 프로그램을 작성해야 한다. 이런 경우 로봇에 대한 우선순위를 설정할 수 있는 기능을 추가로 고안하여 충돌회피 알고리즘과 결합시킨다면 보다 향상된 기능을 낼 것으로 생각한다.

6. 참고문헌

- [1] S. Fu, R. C. Gonzalez, C. S. G. Lee, *Robotics Control, Sensing, Vision, and Intelligence*, McGraw-Hill Book Company, 1987, pp. 40-42
- [2] 삼성전자, SRC기술자료, , 1995

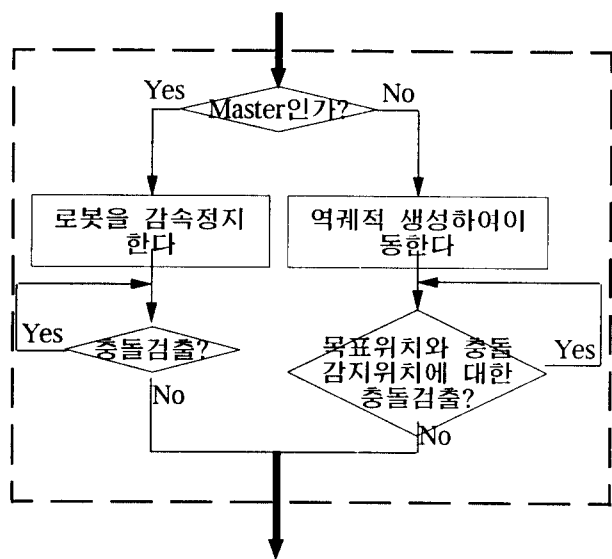


그림 6. 역궤적 생성

임의의 궤적으로 이동하는 방법보다 역 궤적으로 이동하는 방법의 장점은 로봇이 역 궤적을 따라 이동하면서 주변장치와의 충돌을 염두에 두지 않아도 되기 때문에 주변장치에 대한 추가 정보가 필요하지 않다는 것이다. 그림 6.은 그림 5.의 “일시정지 또는 역 궤적생성”에 대한 플로우차트이다.

그림 6.에서 보듯이 Master로 선택된 로봇은 감속정지하고 충돌이 해소될 때까지 대기하지만, Slave로봇은 역 궤적으로 이동하여 로봇이 가고자 하는 최종위치와 충돌이 검출된 위치가 충돌상태에서 해제되었는지를 검사하여 충돌이 해소된 후에 다시 그림 5.의 궤적생성 및 충돌검출을 하면서 목표위치로 이동한다.