

신경망을 이용한 ROBOT ARM의 디버링(Deburring)
작업에 관한 연구

A STUDY ON DEBURRING TASK OF ROBOT
ARM USING NEURAL NETWORK

⁰주진화*, 이경문*, 이장명*

*부산대학교 전자공학과(Tel: 051-510-1696; Fax: 051-515-5190; E-mail: yther@hyowon.cc.pusan.ac.kr)

Abstracts This paper presents a method of controlling contact force for deburring tasks. The cope with the nonlinearities and time-varying properties of the robot and the environment, a neural network control theory is applied to design the contact force control system. We show that the contact force between the hand and the contacting surface can be controlled by adjusting the command velocity of a robot hand, which is accomplished by the modeling of a robot and the environment as Mass-Spring-Damper system. Simulation results are shown.

Keywords Contact Force, Neural Net., DeburringTask, Command Velocity

1. 서론

산업현장에서 로봇의 활용은 생산증대와 고임금에 따른 인력난의 해소 등을 위해 더욱 증가되고 있으며, 산업이 발달함에 따라 더욱 더 복잡하고 고정밀도의 제 어기를 가진 로봇이 요구 되고 있다. 산업용 로봇이 투입되는 작업에는 로봇의 End-Effector와 주변환경(Environment) 사이에 접촉이 일어나지 않는 비접촉 작업으로 현재 산업현장에 투입된 로봇의 대다수가 이 부류에 속하는 작업을 하고 있다. 예를들면, Spot 혹은 Arc용 접로봇, Spray도장 로봇, 검사로봇등이다. 이와는 달리 조립, 가공등의 복잡하고 진보된 로봇이 필요한 작업을 로봇과 주변환경 간의 접촉이 불가피하다. 이러한 작업을 수행하는 로봇은 작업 수행 경로에 대한 위치제어는 물론, 주변환경과 작업 수행에 따른 접촉상황 등을 고려하여 제어되어야 한다[2]. 로봇 Arm과 작업환경간의 접촉에 따른 접촉력의 제어는 최근에 많이 연구되어 지고 있다[4],[5]. 이들은 로봇 Arm이 작업환경과 접촉할 때 접촉점에서의 접촉력의 변화를 제어하고 있으며, 경로 이동에 대한 접촉점에서의 힘의 변화는 고려하지 않고 있다.

본 논문에서는 로봇이 디버링(Deburring) 작업을 수행하기 위하여 작업 환경과 접촉했을 때를 수학적으로 모델링하고, 접촉 후, 로봇이 작업수행 경로를 따라 일정한 속도로 이동할 때, 작업환경의 변화에 따른 로봇 Arm과 작업 환경간의 접촉력(Contact Force)의 변화를 신경망을 이용하여 제어변수를 적절히 조절함으로써, 이들간의 접촉력을 일정하게 유지할 수 있음을 보인다.

2. 시스템 Model

1) 로봇과 작업환경과의 접촉

로봇이 주어진 작업을 수행하기 위해서는 일차적으로 작업 환경과의 접촉이 필요하다. 따라서, 먼저 로봇 Arm이 작업환경과 접촉했을때의 상태를 그림.1과 같이 Mass, Spring, Damper 시스템으로 나타내고, 그 상황을 수식적으로 표현한다.

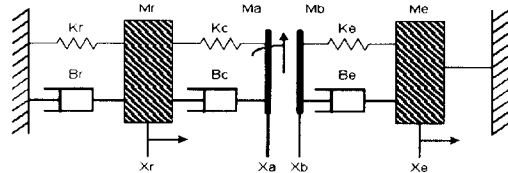


그림 1. 작업환경 접촉에 따른 모델.
Fig. 1. The model of contacting to the environment.

그림.1은 로봇이 주어진 작업을 수행하기 위해서 작업 환경에 접촉한 상태를 나타내고 있으며, 각 변수들은 다음과 같이 정의된다.

- Mr : robot endffector에서 본 robot의 mass
- Kr : robot endffector의 stiffness
- Br : robot endffector의 friction
- Ks : robot body의 stiffness
- Bs : robot body 와 지면과의 friction 및 로봇 각 조인트의 friction
- Kb : robot 이 environment에 contact한 경우의 stiffness

Bb : robot 이 environment에 contact한 경우의 friction

Me : environment의 mass

Ke : environment의 stiffness, Me가 무한대인 경우 무시할 수 있다.

Be : environment의 friction, Me가 무한대인 경우 무시할 수 있다.

위에서, Ke와 Be의 경우 작업환경의 질량 Me가 무한대인 경우 무시할 수 있다는 것은 작업환경이 가진 질량이 무한대인 경우는 지면에 놓인 물체나 혹은 벽면에 고정된 물체를 예들 들 수 있을 것이다. 이들의 경우는 Ke의 값은 Kb로 포함한 값으로 대신 모델할 수 있고, Be의 값도 Bb의 값에 포함시켜 모델이 가능한 것이다. 모델의 운동방정식은 다음과 같이 표현할 수 있다. 이 경우 중력의 영향은 고려하지 않았다.

$$\begin{aligned} [M_r S^2 + (B_r + B_b)S + (K_r + K_b)]X_r(s) - (B_r S + K_r)X_a(s) &= F(s) \\ (M_a S^2 + B_r S + K_r)X_a(s) - (B_r S + K_r)X_r(s) &= 0 \end{aligned} \quad (1)$$

윗식을 벡터 Matrix형태로 표현하면,

$$\begin{bmatrix} M_r S^2 + (B_r + B_b)S + (K_r + K_b) & -(B_r S + K_r) \\ -(B_r S + K_r) & (M_a S^2 + B_r S + K_r) \end{bmatrix} \begin{bmatrix} X_r(s) \\ X_a(s) \end{bmatrix} = \begin{bmatrix} F(s) \\ 0 \end{bmatrix} \quad (2)$$

식(2)에서 역행렬을 구해 곱해주면

$$\begin{bmatrix} X_r(s) \\ X_a(s) \end{bmatrix} = \begin{bmatrix} \frac{M_a S^2 + B_r S + K_r}{\Delta} & \frac{B_r S + K_r}{\Delta} \\ \frac{B_r S + K_r}{\Delta} & M_r S^2 + (B_r + B_b)S + (K_r + K_b) \end{bmatrix} \begin{bmatrix} F(s) \\ 0 \end{bmatrix} \quad (3)$$

여기서,

$$\begin{aligned} \Delta &= [M_r S^2 + (B_r + B_b)S + (K_r + K_b)](M_a S^2 + B_r S + K_r) - (B_r S + K_r)^2 \neq 0 \\ &= M_r M_a S^4 + (M_a B_r + M_r B_b + M_r B_r)S^3 + (M_a K_r + M_r K_b + B_r B_r + M_r K_r)S^2 \\ &\quad + (B_r K_r + K_r B_b)S + K_r K_b \neq 0 \end{aligned} \quad (4)$$

이다. 윗식(1),(2),(3)은 로봇 Arm에 대한 운동방정식이며 작업환경에 대해서도 동일한 방법으로 유도가 가능하다.

2) 로봇이 작업 경로를 따라 이동할 때의 모델

로봇이 작업을 수행하기 위해서는 작업환경과 접촉한 후, 작업환경에 적절한 힘을 가하면서 주어진 작업경로를 따라 이동이 필요하다. 그림.2에서는 로봇이 디버링(Deburring) 작업을 수행하기 위해서 그림.1에서보인 것처럼 일차적으로 작업환경과의 접촉 후, 작업경로를 따라 이동하는데 따른 상태를 보이고 있다.

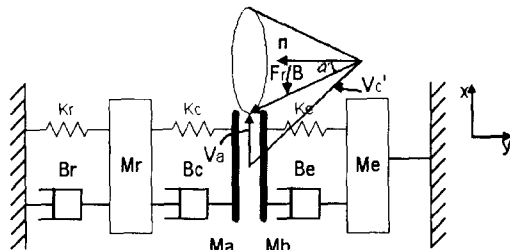


그림 2 작업수행을 위한 경로이동에 대한 모델.
Fig. 2. The model of trajectory tracking for tasks.

로봇이 작업 수행을 위해 주어진 경로 X축을 따라 이동할 때는 작업환경 y축에 대하여 작업에 따른 적절한 힘을 가해야 하고, 또한, 경로 이동에 따른 작업경로,x축,에 대해서도 일정한 힘이 필요하다. 따라서, 로봇과 작업환경과의 접촉에 따른 강성도(Stiffness)와 작업수행을 위해 경로를 따라 이동할 때 야기되는 점성마찰(Viscous Friction)이 동시에 존재하게 된다. 물론 이 경우에는 점성마찰에 비해 강성도를 상대적으로 무시할 수 있을 정도의 양으로 추측된다. 그러므로, 이 경우의 모델에 있어서는 강성도 자체를 무시한 상태에서의 모델이 가능하지만, 보다 더 정밀한 제어를 위해서는 실제 존재하는 강성도항을 포함하는 것이 실험할때의 데이터 에러의 편차를 줄일수 있을것이라고 생각된다. 따라서, 이들관계를 적절하게 표현 하기 위해서는 그림.2에서 나타낸 것처럼 Friction Cone, [1]을 이용하므로서 보다 더 적절한 표현이 가능하다. 그림.2에서 Vc'는 로봇이 작업을 수행하면서 경로를 따라 이동하기 위해서 로봇 Arm에서 가해지는 속도에 따른 접촉점에서의 속도를 나타내고, Va는 Vc'의해서 발생하는 접촉점에서 작업경로,x축,를 따라 이동하는데 대한 실제 속도(Actual velocity)를 나타낸다. 그리고, Fr은 로봇 Arm에서 작업환경에 가해지는 힘에 대한 반작용력(Reactive Force)으로 접촉점에서 발생하는 실제의 접촉력,Fa,과 등가의 관계가 있다. 여기서 접촉점에서의 힘 ,Fa,은 적절한 검출기(Sensor)를 이용하므로서 측정할 수 있다. 그림.2를 이용하여 로봇 Arm과 작업환경간의 접촉점에서의 힘,Fa,은 다음과 같다.

$$F_a = B \cdot (V_a - V_c') \quad (5)$$

여기서, B는 Damping Matrix를 나타낸다.

식(5)는 로봇 Arm과 작업환경간에 발생하는 힘은 작업 경로 이동에 따른 로봇 Arm에서 가해지는 속도를 적절하게 조절하므로서 제어할 수 있음을 보이고 있다. 여기서, Vc'는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} V_c'(k+1) &= V_c'(k) + c_1 \cdot [V_s - V_a(k)] \\ &\quad + c_2 \cdot [F_d - F_a(k)] \end{aligned} \quad (6)$$

여기서, Fd는 작업실행에 따른 접촉점에서 요구되는 힘,Fc,이 주어졌을 때 접촉점에서 나타나는 힘으로 다음과 같이 표현할 수 있다.

$$F_d = F_c + [\tan(\alpha) \cdot |F_c| / |V_c|] \cdot V_c \quad (7)$$

그리고 c1과 c2는 가중 계수, k는 반복횟수, Vs는 슬라이딩(Sliding)속도, Vc는 로봇 Arm에서 작업환경으로 가해진 실제의 속도를 각각 나타낸다. 그리고 α는 Fa와 Friction cone 표면의 법선(Normal)벡터 n사이의 Friction 각으로서 다음식과 같이 나타낼 수 있다.

$$\alpha = \cos^{-1}(F_a \cdot n / |F_a|) \quad (8)$$

여기서,

$$\begin{aligned} n &= a \cdot V_a + b \cdot F_a \\ a &= -1 / [|V_a| \cdot \sqrt{r^2 - 1}] \\ b &= |V_a| / [(F_a \cdot V_a) \cdot \sqrt{r^2 - 1}] \\ r &= |F_a| |V_a| / (F_a \cdot V_a) \end{aligned} \quad (9)$$

로 표현 된다.

3. 신경망의 구조 및 학습알고리즘

본 연구에서 사용된 신경망의 구조로는 비선형 문제를 해결하는데 우수한 특성을 지닌 Rumelhart[3]가 제안한 오차 역전파 다층 신경회로망을 사용하였다. 그림.3에서는 기본적인 신경망의 구조를 나타내고 있다.

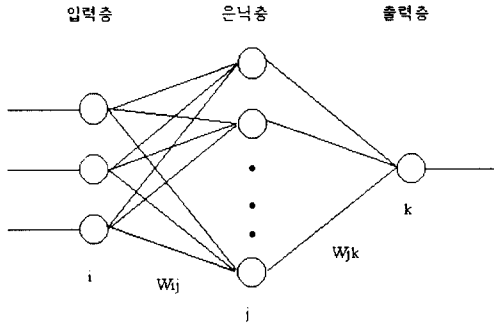


그림 3. 신경회로망의 구조.
Fig. 3. Structure of neural network.

그림.3에서 신경망에서 각각의 입력은 은닉층에 있는 뉴런이 웨이트를 통하여 연결되었다. 그리고 출력층에서의 출력 뉴런도 웨이트를 통하여 연결된다. 여기서, j 번째 뉴런의 출력은 다음과 같다.

$$h_j = a(Z_j) \quad (10)$$

여기서, $a(\cdot)$ 는 활동함수(activation function)이다. j번째 뉴런의 활동함수(activation function)는

$$Z_j = \sum_i W_{ji} X_i + U_j \quad (11)$$

와 같이 표현된다. j번째 뉴런은 입력 X_i 와 threshold U_j 의 웨이트 합성이다. 활동함수는 시그모이드 함수로서 다음과 같다.

$$h_j = a(Z_j) = (1 + e^{-Z_j})^{-1} \quad (12)$$

또한, 출력층의 m번째 뉴런의 출력은

$$y_m = \sum_j W_{mj} h_j \quad (13)$$

로 나타낼 수 있다. 그리고 요구값과 실제출력값의 오차는

$$e(k) = y(k) - \hat{y}_m(k) \quad (14)$$

와 같이 표현된다. 식(14)에서 학습 알고리즘의 목적은 시간에 대한 2차방정식 $Q(k) = e^T(k) \cdot e(k)$ 을 최소화 하기 위하여 ANNs의 웨이트 조정에 $e(k)$ 가 이용된다. 이 최소한도를 이루기 위해 간단한 알고리즘은 오차 역전파법 알고리즘으로 ANNs 응용에서 광범위하게 이용되고 있다. 오차는 ANNs에서 각각의 뉴런에 의하여 계산되어지며, 출력층에서 m번째 뉴런의 오차는 다음과 같다.

$$\delta_m(k) = e_m(k) = y_m(k) - \hat{y}_m(k) \quad (15)$$

은닉층에 있는 j번째 뉴런의 오차 $\delta_j(k)$ 는 다음 식과 같이 나타낼 수 있다.

$$\delta_j(k) = \frac{da(Z_j(k))}{dZ_j(k)} \sum_m \delta_m(k) W_{mj}(k) h_j(k) [1 - h_j(k)] \sum_m \delta_m(k) W_{mj}(k) \quad (16)$$

은닉층과 출력층을 연결하는 웨이트는 다음 식과 같이 수정된다.

$$\Delta W_{mj}(k) = \gamma_m \Delta W_{mj}(k-1) + \gamma_g \delta_m(k) h_j(k) \quad (17)$$

$$W_{mj}(k+1) = W_{mj}(k) + \Delta W_{mj}(k)$$

여기서, γ_m 은 모멘트 상수, γ_g 는 이득변수를 나타내고, 입력에서 은닉층으로 연결되는 웨이트는 다음과 같이 수정된다.

$$\Delta W_{ji}(k) = \gamma_m \Delta W_{ji}(k-1) + \gamma_g \delta_m(k) h_j(k)$$

$$W_{ji}(k+1) = W_{ji}(k) + \Delta W_{ji}(k) \quad (18)$$

출력층 오차는 식(15)에 의하여 계산 되어지고, 은닉층 오차는 식(16)에 의하여 계산되어 ANNs의 웨이트는 식(17)과 식(18)로 수정 되어진다.

4. 신경망을 이용한 접촉력 제어기의 구성.

그림.4에서는 신경망을 이용한 접촉력 제어기의 구조를 나타내고있다.

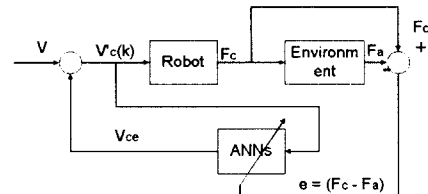


그림.4 신경망을 이용한 접촉력 제어기의 구조.
Fig. 4. Structure of the contact force control system.

작업에 필요한 접촉력 F_c 가 인가되면, 로봇 Arm은 작업환경에 접촉된 상태에서 그에 대응하는 속도로서 경로를 따라 이동하게 된다. 이때 로봇 Arm에 부착된 센서는 작업환경에 가해진 힘에 대한 정보를 출력하게 되고, 그 출력과 요구한 접촉력 F_c 에 대한 편차가 신경망에서 학습되어지며, 학습에 의한 신경망의 출력은 경로이동에 따른 로봇 Arm의 속도를 제어하게 된다.

5. 시뮬레이션 및 결과

로봇 Arm이 디버링 작업을 위해서 작업환경과 접촉하여 일정한 경로를 따라 작업에 필요한 힘을 유지하면서 이동할 경우에, 로봇 Arm과 작업환경간의 접촉력의 변화를 신경망을 이용하여 수정하므로써 작업에 요구된 힘을 일정하게 유지하면서 작업할 수 있음을 보이고자 한다. 시뮬레이션에서는 5[Kg]의 질량을 가진 로봇이 질량 100[Kg]의 작업환경에 접촉하여 0.1[mm/s]의 속도로 디버링 작업을 하는 경우를 나타내고 있다. 이때 작업에 필요한 힘을 3[N]으로 하고 작업경로를 따라 800[mm]를 이동하였을 경우에 신경망을 통하여 제어된 로봇과 작업환경간의 접촉력은 그림 5와 같이 나타났다. 그리고 그림 6에서는 이때 신경망을 이용하여 제어된 로봇 Arm의 속도 변화값을 나타내고 있다. 그리고 그림 7에서는 이때의 로봇 속도와 접촉력사이의 에러값을 나타내고 있다.

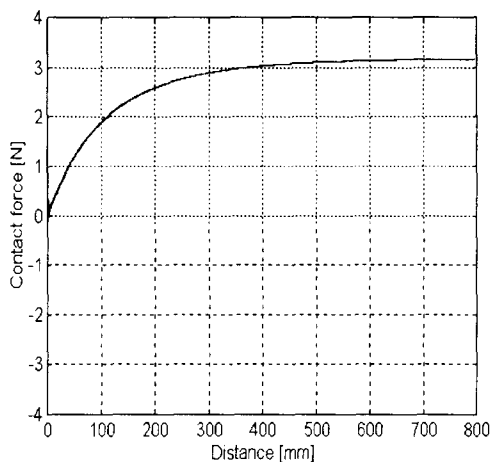


그림 5. 접촉점에서의 제어된 힘.
Fig. 5. Controlled force to the contact point.

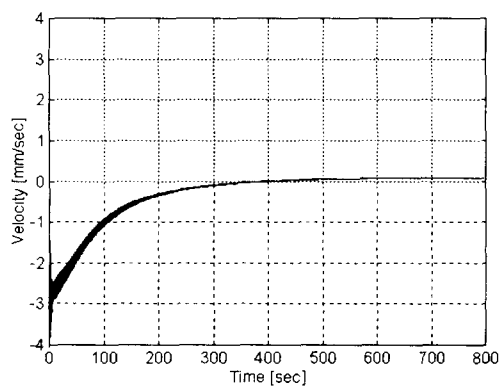


그림 6. 신경망으로 제어된 로봇 Arm의 속도.
Fig. 6. Controlled velocity of robot arm through neural network.

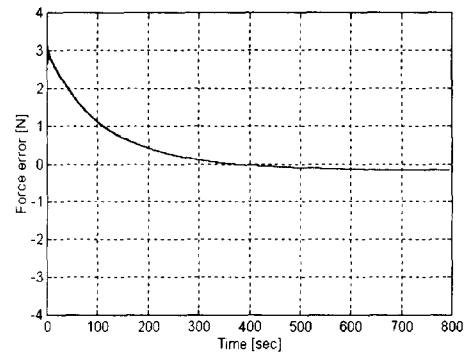


그림 7. 로봇 Arm과 접촉점간의 오차 수정.
Fig. 7. Modified error between robot arm and contact point.

6. 결론

본 연구에서는 로봇이 주어진 작업환경에 접촉하여 작업경로를 따라 디버링 작업을 수행할 때, 로봇과 작업환경간의 접촉력을 일정하게 유지하면서 작업할 수 있음을 보이고 있다. 시뮬레이션 결과에서 본 것처럼 로봇과 작업환경간의 접촉력이 요구값보다 작은 경우 신경망은 속도를 줄여 접촉력을 요구값에 도달하게 한다. 이때 접촉력이 요구값에 도달하는 응답속도가 약 400 [Sec]로 안정된 값을 유지하는데에는 일정한 시간이 요구됨을 알 수 있다. 이 응답속도는 적절한 알고리즘을 적용하므로써 개선할 수 있을것으로 생각한다. 이 응답속도의 개선은 차후의 연구 과제로 한다.

7. 참고 문헌

- [1] Sukhan Lee and Jang M. Lee "The Control of Surface Contact and Slide Using Wrist Force/Torque Sensor," 9th IASTED Int. Symp. on Robotics and Automation, Santa Barbara, pp.49-54, 1987.
- [2] 한 명 철 "불확실한 로봇 시스템의 건설 하이브리드 제어기 설계," 한국자동제어학술회의 논문집(1), pp.418 - 421, 1995.
- [3] D.E. Rumelhart, G.E. Hilton and R. J. Williams "Learning Internal Representations by Error Propagation," in D.E. Rumelhart and J.L. McClelland(Eds), Parallel of Cognition, Vol. 1 : Foundations, MIT Press, pp.318-362, 1986.
- [4] Prasad N.Akella and Mark R. Cutkosky "Contact Transition Control With Semiactive Soft Fingertips," IEEE Transaction on Robotics and Automation, Vol. 11, No.6, December , pp.859-867,1995.
- [5] Sunil K.Singh and Dan O.Popa "An Analysis of Some Fundamental Problems in Adaptive Control of Force and Impedance Behavior: Theory and Experiments," IEEE Transaction on Robotics and Automation, Vol. 11, No.6, December, pp.912-921,1995.