

가상링크 기반의 ROBOKER 머리의 실시간 대상체 추종 성능 향상을 위한 신경망 제어

Neural Network Compensation for Improvement of Real-Time Moving Object Tracking Performance of the ROBOKER Head with a Virtual Link

김 동 민, 최 호 진, 이 근 형, 정 슬*
(Dong Min Kim, Ho Jin Choi, Geun Hyung Lee, and Seul Jung)

Abstract: This paper presents the implementation of the real-time object tracking control of the ROBOKER head. The visual servoing technique is used to track the moving object, but suffers from ill-estimated Jacobian of the virtual link design. To improve the tracking performance, the RBF (Radial Basis Function) network is used to compensate for uncertainties in the kinematics of the robot head in on-line fashion. The reference compensation technique is employed as a neural network control scheme. Performances of three schemes, the kinematic based scheme, the Jacobian based scheme, and the neural network compensation scheme are verified by experimental studies. The neural compensation scheme performs best.

Keywords: object tracking, RBF network, visual servoing method, RCT scheme, virtual link

I. 서론

영상처리 기술의 응용은 모든 산업 분야에 사용할 정도로 보편화되었으며, 특히 로봇 분야에서의 영상은 빼놓을 수 없는 기술이 되었다. 로봇이 점차로 지능화하여 감에 따라 주변의 정보를 받아들이는 장치로 비전이 매우 중요한 센서로 대두하게 되었다. 또한, 최근 컴퓨터의 성능 향상으로 실시간 제어가 가능할 정도로 영상처리를 빠르게 수행할 수 있게 되었다. 미국 MIT의 COG라는 로봇은 실시간으로 움직이는 물체를 잘 추종하는 것을 보여준 바 있다.

이에 따라 vision과 robot manipulator의 조합을 고려한 비주얼 서보잉이라는 분야가 빠르게 발전하고 있다[1-8]. 비주얼 서보잉 기술은 인간이 물체를 보면서 팔로 작업하는 것을 모사한 것이다. 움직이는 물체의 경우에는 카메라를 통해 물체의 위치를 정확하게 알아내어 로봇 팔이 물체를 따라 움직이도록 위치 명령어를 만들어 내야 한다. 성공적인 비주얼 서보잉 작업을 위해서는 물체의 위치를 정확하게 알아내는 것이 무엇보다도 중요한데, 3차원 영상기술과 실시간 영상 처리 기술이 필요하다. 따라서, 움직이는 물체의 좌표를 알아내기 위한 방법으로 카메라의 위치와 개수에 따른 분류와 제어 방식 등 몇 가지 기준에 따라 분류할 수 있다[1,2].

본 논문에서는 스테레오 비전을 장착한 로봇머리가 시야에서 대상체를 놓치지 않기 위해 쫓아가는 시스템을 구현해 보았다. 로봇 머리는 가장 간단한 형태로 tilt, pan의 2개의 회전 조인트를 사용한다. 휴머노이드 로봇의 머리는 eye-in-hand 방식의 camera configuration을 가졌으며, 스테레오 카메라를

이용한 위치기반 비주얼 서보잉의 제어 방식을 사용한다. 특히, 움직이는 물체의 추종은 움직이는 물체의 속도 정보를 파악하는 것이 중요하다. 로봇 팔의 경우에는 로봇 팔이 움직일 수 있는 장점이 있지만 로봇 머리는 대상체와 닿을 수 없는 형태를 가지고 있어 로봇 팔과는 다른 방식의 접근 방법이 필요하다.

이에 대한 해법으로 본 논문에서는 새로운 가상의 팔 개념을 도입하였다. 물체와 카메라와의 거리를 가상의 팔로 간주하고 마치 로봇 팔 처럼 기구학을 해석하면 간단하게 카메라와 물체와의 관계를 구할 수 있다. 가상의 팔을 도입하여 변환을 해주면 마치 로봇 팔을 제어하는 것처럼 속도를 제어할 수 있다. 하지만, 거리 오차에 따른 가상 팔의 오차에 대한 보상이 필요하게 된다. 따라서, 가상 팔의 자코비안 정보에 대한 오류를 해결해야 한다. 본 논문에서는 가상의 자코비안 정보 보정을 위해 RBF 신경회로망을 사용하여 실시간으로 보상할 수 있는 입력 보상 방식의 신경회로망 제어 방식을 채택하였다[9]. 시뮬레이션과 실험을 통하여 제안한 방식의 성능을 검증하였다.

II. 가상팔을 도입한 로봇 머리 기구학

로봇 기구학의 경우 X_w, Y_w, Z_w 의 원점을 기준으로 팔 끝 좌표계의 원점의 자세를 구해준다. Tilt와 pan이 포함되어 있으므로 tilt와 pan에 관해서 행렬 미분을 해주면 Base 좌표계 기준으로 한 팔 끝 좌표계의 원점의 속도 \dot{p} 과 tilt, pan의 속도 간의 관계인 로봇 자코비안을 얻을 수 있다. Base 좌표계는 이후 휴머노이드 로봇의 팔과의 결합을 위해 그림 1과 같이 설정하였다. 로봇 D-H 변수를 바탕으로 순 기구학인 $p(t) = F(q(t))$ 를 먼저 구하고 행렬미분을 통해 자코비안을 구하게 된다[10].

물체의 좌표를 구하기 위해 D-H 변수를 이용하여 $p(t) = F(q(t))$ 를 구하면 식 (1)과 같다.

* 책임저자(Corresponding Author)

논문접수: 2008. 9. 8., 채택확정: 2009. 5. 11.

김동민, 최호진, 이근형, 정슬: 충남대학교 지능로봇시스템 전공
(ellegan@cnu.ac.kr/qorwlcjja@naver.com/sadthink@paran.com/jungs@cnu.ac.kr)

※ 본 연구는 교육과학기술부의 특정기초사업(R01-2008-000-10992-0)과 한국산업기술재단 지역혁신 인력양성 사업으로 수행된 연구결과로 지원에 감사드립니다.

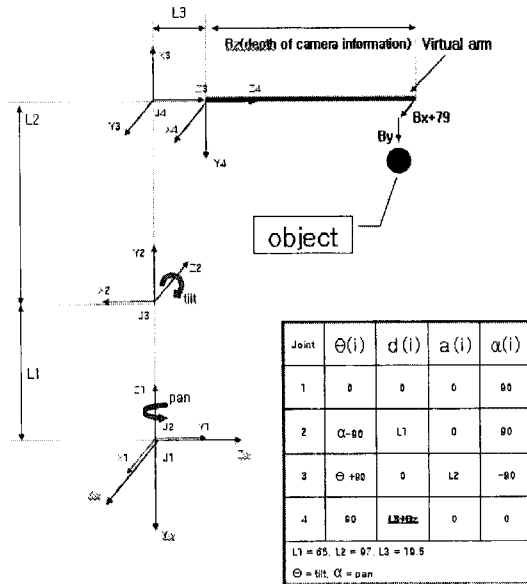


그림 1. 가상의 링크를 고려한 robot(head)의 D-H 변수.
 Fig. 1. D-H parameter considering virtual link.

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} L_2 \sin \theta \sin \alpha + (L_3 + B_z) \cos \theta \sin \alpha \\ -L_1 - L_2 \cos \theta + (L_3 + B_z) \sin \theta \\ L_2 \sin \theta \cos \alpha + (L_3 + B_z) \cos \theta \cos \alpha \end{bmatrix} \quad (1)$$

로봇 자코비안은 회전 조인트의 속도와 Base(robot) 좌표계를 기준으로 한 로봇팔 끝의 속도 간의 관계를 나타낸다. L_1, L_2, L_3 는 각 링크의 길이로서 상수이며, θ 는 tilt 모터의 각도, α 는 pan 모터의 각도이다.

아래 (2)는 순기구학을 통해 얻은 기준 좌표계를 기준으로 하는 로봇팔 끝의 position 관계식을 이용하여 구한 로봇의 역자코비안이다. 가상의 팔을 B_z 의 길이로 만들어 주었을 경우의 역자코비안은 다음과 같다.

$$J^\# = \begin{bmatrix} \frac{2 \sin \alpha (39 \sin \theta + 2 P_z \sin \theta - 194 \cos \theta)}{39157 + 156 P_z + 4 P_z^2} \\ \frac{-2 \cos \alpha}{39 \cos \theta + 2 P_z \cos \theta + 194 \sin \theta} \\ \frac{2(39 \cos \theta + 2 P_z \cos \theta + 194 \sin \theta)}{39157 + 156 P_z + 4 P_z^2} \\ 0 \\ \frac{-2 \cos \alpha (39 \sin \theta + 2 P_z \sin \theta - 194 \cos \theta)}{39157 + 156 P_z + 4 P_z^2} \\ \frac{-2 \sin \alpha}{39 \cos \theta + 2 P_z \cos \theta + 194 \sin \theta} \end{bmatrix} \quad (2)$$

III. 비주얼 서보잉 제어

대상체가 매우 느리게 움직인다고 가정하고 $v_o = 0$, 이 순방향 전체 관계식을 구하면 식 (3)과 같다[1].

$$\begin{aligned} \dot{s} &= \frac{\partial s}{\partial c} \frac{\partial c}{\partial p} \frac{\partial p}{\partial q} \frac{\partial q}{\partial t} + \frac{\partial s}{\partial t} \\ &= L_s^c v_n^n J_n \dot{q} + \frac{\partial s}{\partial t} \end{aligned} \quad (3)$$

여기서, s : 영상의 전체 위치(scene space)

$v_i = v_c - v_o$: 영상의 전체 속도(camera space)

v_o : velocity of a feature considering only about a target (camera space)

v_c : 영상의 속도(camera space)

c : 영상의 위치(camera space)

p : 팔 끝의 위치(Base space)

$v_p = \dot{p}$: 팔 끝의 속도(Base space)

순방향 전체 관계식에서 대상체가 느리게 움직인다고 가정하고 식 (3)에서 $\frac{\partial s}{\partial t}$ 를 0이라 두었다. 그리고 나머지 모두를 순서대로 역을 취하면 \dot{q} 을 구할 수 있다.

$$\dot{q} = {}^n J_n^{\#c} v_n^n L_s^\# \dot{s} \quad (4)$$

여기서 #는 의사 역행렬을 나타낸다. 위의 역방향 관계식을 살펴보면 영상 평면 공간의 feature(s)가 입력으로 interaction matrix L_s 를 거쳐 카메라 공간으로 변환되고 다시 v-matrix를 거쳐 카메라에서 로봇팔 끝 공간으로 변환된다. 로봇 자코비안은 조인트 속도와 로봇팔 끝의 속도(로봇팔 끝 좌표계의 원점의 속도)간의 관계이므로 \dot{q} 을 구할 수 있게 된다.

본 논문에서는 영상 평면과 카메라 간의 관계는 고려하지 않는다. Bumblebee의 오른쪽 카메라를 기준으로 한 feature의 3D 좌표를 PGR사의 API함수를 통해서 얻을 수 있기 때문에 그것을 직접 사용하였다. 따라서 위의 역방향 식 (4)에서 맨 뒤에 있는 $L_s^\# \dot{s}$ 은 그림 2의 Bumblebee사의 position $B = [B_x, B_y, B_z]^T$ 으로 통합된다.

비주얼 서보잉은 두 개의 제어 루프가 존재한다. 하나는 비주얼 루프이며, 다른 하나는 조인트 컨트롤 루프이다. 비주얼 루프를 위해 존재하는 task function은 위의 control design 상에서 로봇의 역 자코비안 전까지의 오차 입력을 정의한 관계식이다.

$$\dot{s} = L_s v \quad (5)$$

여기서 v 는 식 (6)과 같이 표현된다.

$$v = L_s^\# \dot{s} \quad (6)$$

$$\text{Task function: } v = -\lambda L_s^\# (s(t) - s^*) \quad (7)$$

Feature의 위치어러에 비례 제어 이득 값을 곱하여 속도를

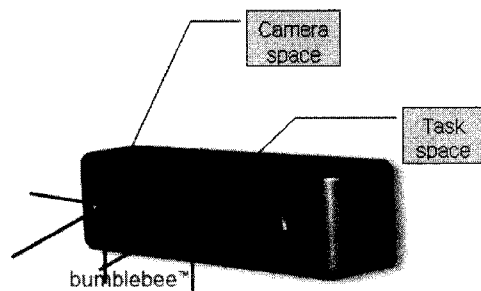


그림 2. Camera space와 End-effector space.
 Fig. 2. Camera space and End-effector space.

구현하였다. 더욱 정밀한 시스템 구현을 위해서는 샘플링 타임을 고려한 속도를 생각해야 한다. 하지만 동역학을 고려하지 않은 단순 자코비안을 이용한 방식에서는 위치오차 (S_error)가 속도(S_error/Ts)와 정비례 관계에 있으므로 비례 제어의 P-gain에 의해 통합되어질 수 있다. Bumblebee가 3D feature 정보를 내어주므로 Pose estimation 과정이 필요가 없게 되며, $L_s^{\#}$ 는 I-matrix가 된다. 또한 $s(t) - s^*$ 는 scene 공간을 기준으로 한 feature의 위치 오차이며, $L_s^{\#}$ 을 거친 후에는 카메라 공간에서 로봇팔 끝 공간으로의 변환관계가 된다. 카메라 공간에서 로봇팔 끝 공간으로의 변환관계를 고려하면 Bumblebee의 B_x, B_y, B_z 의 X성분에 +79를 해주어야 로봇팔 끝 공간의 위치 P_x, P_y, P_z 를 구할 수 있게 된다. 또한 s^* 는 $P_x=0, P_y=0$ 이고 P_z 도 0으로 놓게 되면 식 (8)과 같다.

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} B_x + 79 - 0 \\ B_y - 0 \\ B_z - 0 \end{bmatrix} = \begin{bmatrix} B_x + 79 \\ B_y \\ B_z \end{bmatrix} \quad (8)$$

위의 task function 식 (7)은 식 (9)와 같다.

$$v = \lambda \begin{bmatrix} B_x + 79 \\ B_y \\ B_z \end{bmatrix} \quad (9)$$

여기서 v 는 로봇팔 끝 의 속도를 의미한다. 따라서 위의 task function과 로봇 자코비안을 고려하면 전체 시스템의 입력과 출력의 관계식 식 (10)과 같이 간략화 할 수 있다.

$$\dot{q} = \lambda J(q)^{\#} \begin{bmatrix} B_x + 79 \\ B_y \\ B_z \end{bmatrix} \quad (10)$$

여기서 자코비안은 다음과 같이 구해진다.

$$J^{\#} = \begin{bmatrix} \frac{2}{39157} \sin \alpha (39 \sin \theta - 194 \cos \theta) & \frac{78}{39157} \cos \theta + \frac{388}{39157} \sin \theta & 0 \\ \frac{-2 \cos \alpha}{39 \cos \theta + 194 \sin \theta} & 0 & 0 \\ -\frac{2}{39157} \cos \alpha (39 \sin \theta - 194 \cos \theta) & \frac{-2 \sin \alpha}{39 \cos \theta + 194 \sin \theta} & 0 \end{bmatrix} \quad (11)$$

그림 3은 기구학 기반의 로봇 머리 비주얼 서보잉 제어 블록도를 나타낸다.

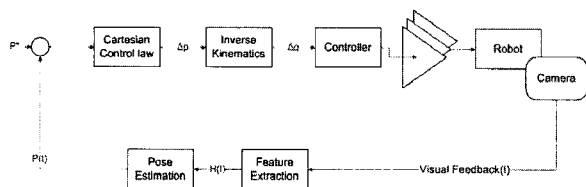


그림 3. 기구학 기반 제어 방식.
Fig. 3. Kinematics based control scheme.

IV. 신경회로망 보상 방식

물체와의 거리 문제를 해결하기 위해 가상의 팔을 도입하였으며, 가상의 팔을 도입함으로 인해 잘못된 수 밖에 없는 자코비안을 입력보상 방식의 RBF 신경망을 이용하여 보정해주었다. 그림 4는 신경회로망을 사용하는 비주얼 서보잉 제어 블록도이다.

RBF신경회로망은 비선형 함수를 가진 한 개의 은닉층을 가지고 있다. 은닉층에서 사용하는 비선형 함수는 가오시안 함수를 사용하고 있다. 이러한 이유로 다층 퍼셉트론 넷에 비해서 빠른 연산이 가능하여 실시간 시스템에 적용하기 적당하다. 그림 5에는 RBF 신경망의 구조가 나타나 있다.

입력 보상 방식은 기존의 제어기에 수정을 가하지 않은 상태에서 시스템의 불안정성을 보상해주는 방식으로, 현재 시스템에서는 조인트 제어기를 수정하지 않고 비주얼 연산을 하는 호스트 시스템에서 자코비안 오차를 보상하는 신경회로망을 동작시키는 형태로 구현되어 있다[9].

신경회로망에 $P_x(t), P_x(t-1), P_x(t-2)$,과 $P_y(t), P_y(t-1), P_y(t-2)$ 으로 각 P_x 와 P_y 에 대하여 두 개의 RBF network를 사용하였다. 목적함수는 각각 P_x, P_y 를 설정하였다. 신경회로망의 출력 P_x, P_y 는 현재 비주얼 루프의 오차 P_x, P_y 를 보상한다.

V. 시뮬레이션

그림 6은 그림 4에 나타난 자코비안을 사용한 속도제어를 시뮬레이션한 모델이다. 그림 7과 8에서 보면 신경망을 사용할 경우의 오차가 사용하지 않은 경우보다 줄어들음을 알 수 있다.

VI. 실험결과

1. 실험 환경

스테레오 카메라로는 point grey research 사의 bumblebee를

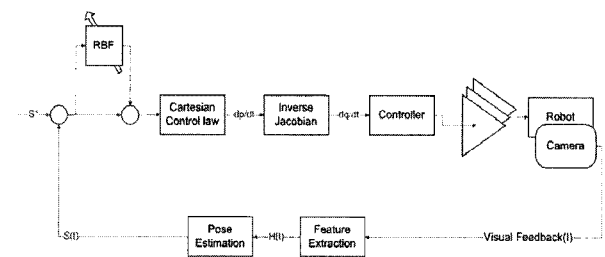


그림 4. Jacobian 기반 신경회로망 보상 방식.
Fig. 4. Jacobian based neural compensation scheme.

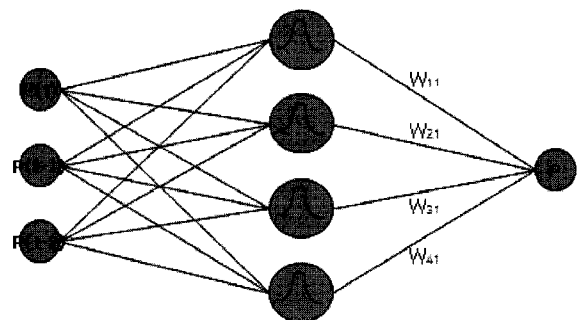


그림 5. RBF 신경회로망 구조.
Fig. 5. RBF neural network structure.

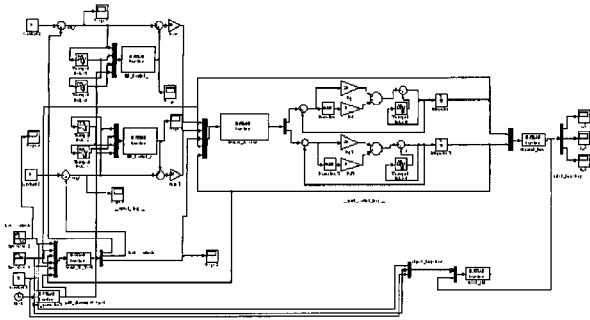


그림 6. Simulink 제어 모델.
Fig. 6. Simulink control model.

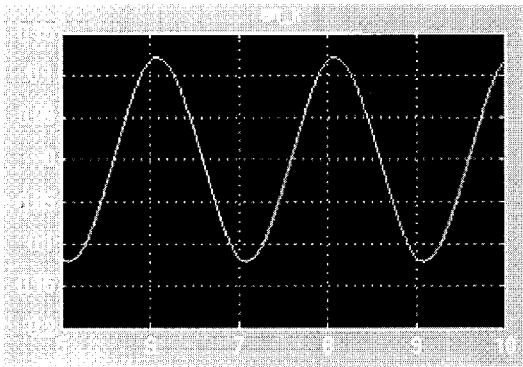


그림 7. 신경망을 사용하지 않은 visual loop error.
Fig. 7. Error of kinematic based control.

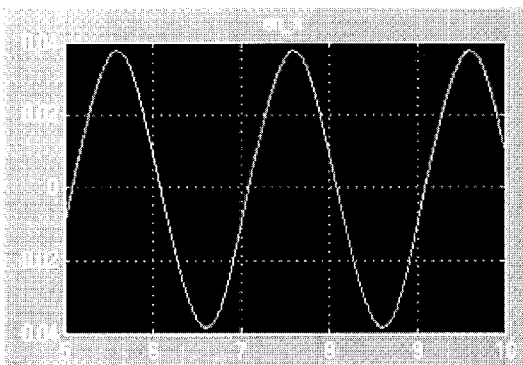


그림 8. 신경망을 사용한 visual loop error.
Fig. 8. Error of RBF network based control.

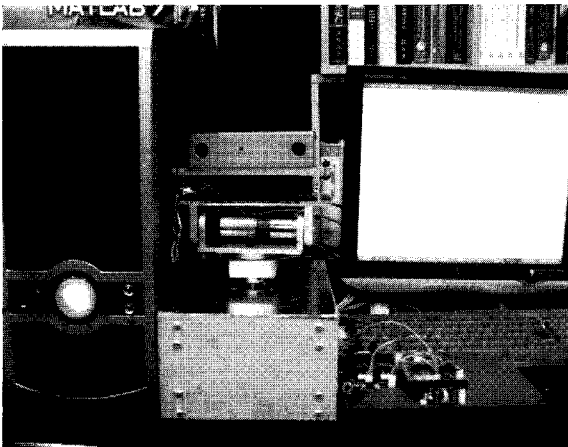


그림 9. 실험구조.
Fig. 9. Experimental setup.



그림 10. GUI 프로그램.
Fig. 10. GUI Program.

사용하였다. 로봇은 휴머노이드 로봇을 사용하였다. 그림 9에서 로봇의 머리는 목 부위에 tilt, pan 두 개의 모터가 있다. 휴머노이드 로봇의 머리부만을 실험하기 위하여 베이스를 만들어 분리된 상태로 실험하였다. CPU는 3Ghz dual core 프레스러 925와 DDR2(PC6400) 2G 램, Geforce nVidia 6800GS 512MB RAM의 그래픽카드의 PC를 사용하였으며, Interboard에서 구매한 OMNI-DSP2812 모듈을 모터제어에 사용하였다. OS는 윈도우즈를 사용하여 MFC로 프로그래밍 하였다. 통신은 115200bps의 RS-232 통신을 사용하였으며 영상처리에는 주로 인텔사의 공개용 영상처리 라이브러리인 openCV를 사용하였다. 대상체는 얼굴로 선정하였으며 얼굴을 찾기 위한 알고리즘으로 openCV에 포함된 CBCH (Cascade of Boosted Classifier working with Haar-like feature) 소스를 사용하였다.

그림 10은 MFC Multiple Document Interface로 작성한 GUI가 나타나 있다. 현재 기본적인 영상처리 알고리즘을 적용해볼 수 있고, USB카메라와 bumblebee로 영상을 받아올 수 있으며 버퍼 및 영상 저장, 읽기, 시리얼 통신 등을 구현하였다.

2. 실험결과

동적 물체 추종 오차를 계산하기 위해 일정한 간격으로 움직이는 물체를 고려하였다. 그림 11과 같이 사람의 얼굴을 인식하여 모터로 구동하여 로봇이 계속적인 원 경로를 추종하도록 하고 실험하였다.

기구학기반 제어 방식은 역 기구학을 이용하여 원하는 조

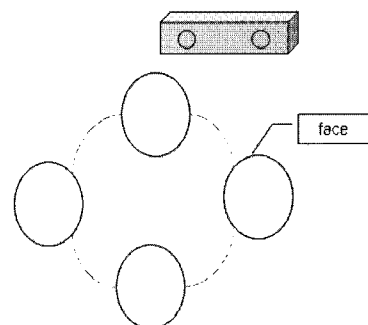


그림 11. 실험 환경: 반복적인 대상체의 움직임.
Fig. 11. Experimental environment: repeated movement of object.

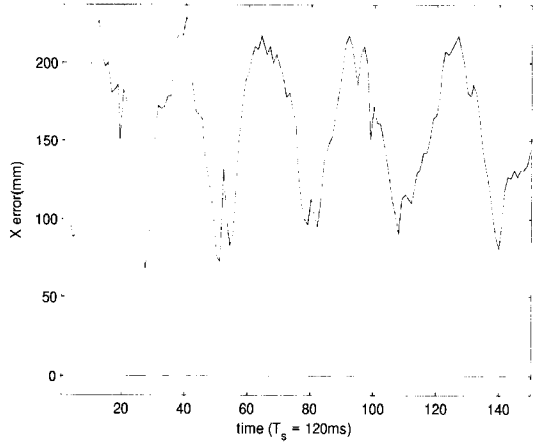


그림 12. 기구학을 이용한 위치 제어의 결과(X_error).
Fig. 12. X axis error of kinematics based control.

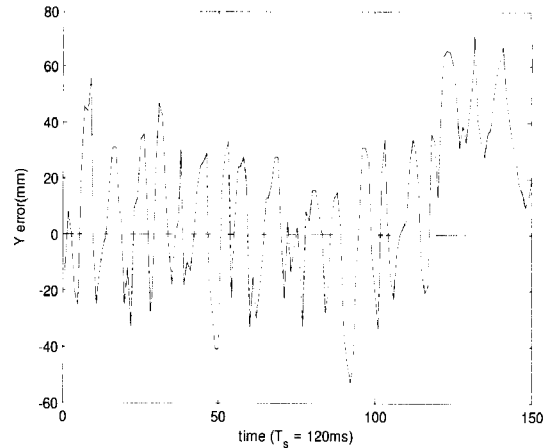


그림 15. 자코비안을 이용한 속도 제어의 결과(Y_error).
Fig. 15. Y axis error of Jacobian based control.

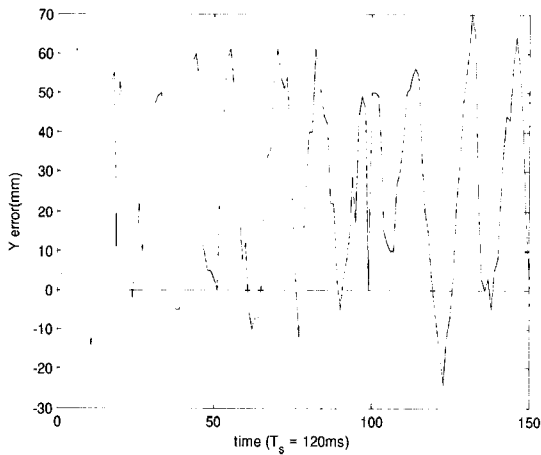


그림 13. 기구학을 이용한 위치 제어의 결과(Y_error).
Fig. 13. Y axis error of kinematics based control.

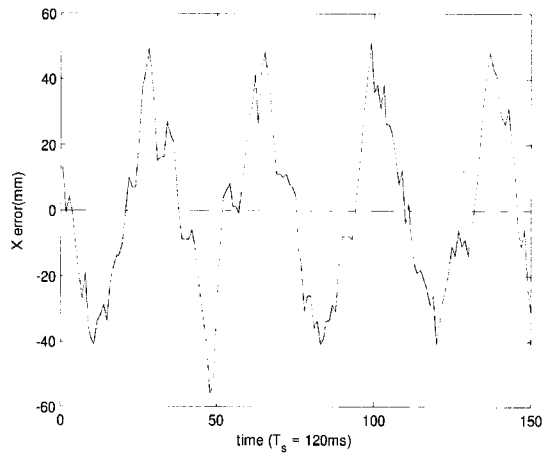


그림 16. 자코비안과 RBF Neural Network 제어(X_error).
Fig. 16. X axis error of Jacobian+RBF based control.

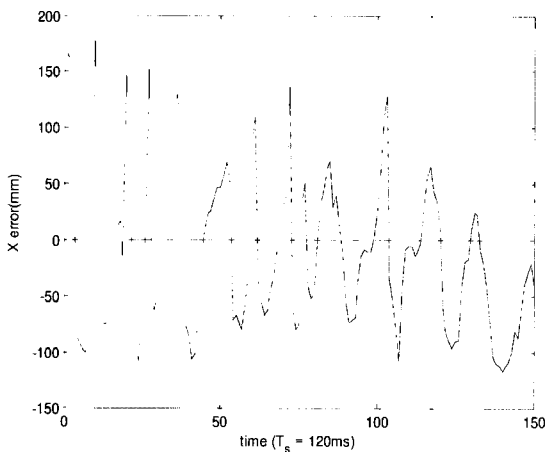


그림 14. 자코비안을 이용한 속도 제어의 결과(X_error).
Fig. 14. X axis error of Jacobian based control.

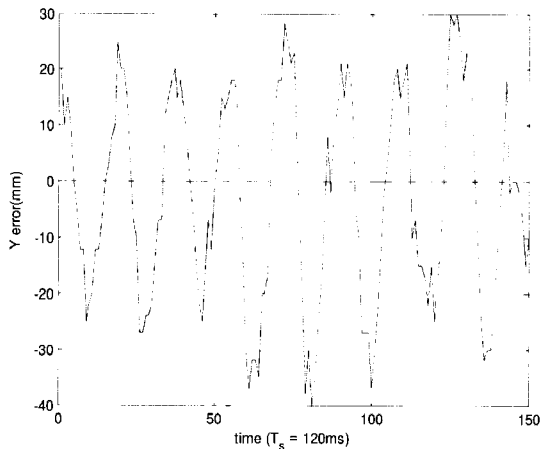


그림 17. 자코비안과 RBF Neural Network 제어(Y_error).
Fig. 17. Y axis error of Jacobian+RBF based control.

인트 각도를 입력으로 넣어주는 방식을 사용하였다. 이 방식은 속도를 고려하지 않았기 때문에 움직임이 끊어짐을 볼 수 있었다. 그림 12와 13의 결과는 기구학의 불확실성을 보여준다. 0을 중심으로 오차가 존재해야 하지만 다른 위치를 기준으로 오차가 존재함을 알 수 있다. 기구학이 잘못되어 그곳이 참값으로 여겨지기 때문에 나타나는 현상이다. 그림 14와

15는 자코비안에 기반한 제어 방식이다. 신경회로망이 보상된 방식의 결과는 그림 16과 17에 나타나 있다. 이전의 제어 방식보다 추종 오차가 작은 것을 알 수 있다. 입력보상 방식은 잘못된 자코비안을 보상해 준다.

위의 결과를 수치적으로 분석하기 위해 각 결과들에 대한 RMS 값, 평균값 그리고 표준편차를 구해보았다. 표 1에 나

표 1. 실험 결과의 RMS 오차.

Table 1. RMS error of experiments.

Schemes	Root Mean Squared Error	
	X axis	Y axis
Kinematics	1985.6	429.979
Jacobian	854.347	361.152
Jacobian+RCT	313.035	227.657

표 2. 결과의 평균과 표준편차.

Table 2. Error of average and standard deviation.

Schemes	Mean		Standard deviation	
	X (mm)	Y(mm)	X(mm)	Y(mm)
Kinematics	156.580	26.507	42.0267	23.020
Jacobian	-13.253	10.700	68.487	27.478
Jacobian +RCT	-0.1800	-2.787	25.559	18.378

타난 RMS는 0점을 기준으로 벗어난 값이므로 포괄적으로는 의미가 있다. 각 결과들의 평균을 구해보면 0이 아닌 것을 알 수 있다. 이는 잘못된 기구학, 자코비안의 정도를 의미한다. 표 2의 표준편차는 이들이 얼마나 오차를 적게 잘 수행하는가를 나타낸다.

VII. 결론

본 논문에서는 신경망을 이용하여 비주얼 서보잉에서의 성능을 개선하였다. 새롭게 거리대신 가상의 링크를 추가하여 로봇 기구학과 자코비안을 유도하였으며, 그에 따른 오차를 보정하기 위해 신경회로망으로 보상하였다. 시뮬레이션 및 실험결과를 종합해볼 때 정적인 특성을 갖는 기구학기반의 제어방식의 경우 신경망에 의해 효과적으로 보상됨을 알 수 있었다. 추후에 로봇 팔과의 연동을 통한 비주얼 서보잉 제어를 구현하는 것이다.

참고문헌

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automations*, vol. 12, no. 5, pp. 651-670, 1996.

[2] F. Chaumette and S. Hutchinson, "Visual servo control part I: Basic approaches," *IEEE Robotics and Automations Magazine*, pp. 82-90, Dec. 2006.

[3] Y. Shen, D. Sun, Y. H. Liu, and K. Li, "Asymptotic trajectory tracking of manipulators using uncalibrated visual feedback," *IEEE Trans. on Mechatronics*, vol. 8, no. 1, pp. 87-98, 2003.

[4] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," *IEEE Conf. On Robotics and Automations.*, pp. 2267-2272. 1991.

[5] S. H. Han, W. H. Seo, J. Lee, M. H. Lee, and H. Hashimoto, "Image-based visual servoing control of a SCARA type dual-arm robot," *International Symposium on Industrial Electronics*, pp. 517-522, 2002.

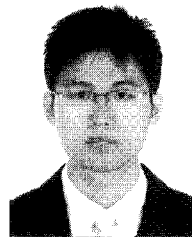
[6] P. J. Goncalves, L. F. Mendonca, J. M. Sousa, and J. R. Pinto, "Improving visual servoing using fuzzy filters," *FUZZ-IEEE*, pp. 1185-1190, 2004.

[7] Q. M. Wu and K. Staley, "Modular neural-visual servoing using a neural-fuzzy decision network," *IEEE Conf. on Robotics and Automations*, pp. 3238-3243, 1997.

[8] D. M. Kim, H. J. Choi, J. S. Kim, W. K. Lee, D. H. Song, and S. Jung, "Tracking control of a moving object for Robokers with stereo visual feedback," *IEEE ICIT*, pp. 52-57, 2007.

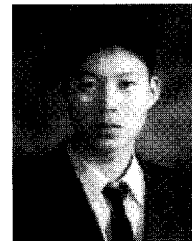
[9] S. Jung and T. C. Hsia, "Neural network compensation technique for Cartesian position control of robot manipulators with vision feedback," *International Conference on Advanced Robotics*, pp. 993-998, 2007.

[10] 정 슬, 로봇공학, 충남대 출판부, 2007.



김 동 민

2007년 충남대학교 메카트로닉스공학과 졸업. 충남대학교 메카트로닉스공학과 대학원 과정. 관심분야는 영상처리, 비주얼 서보잉 제어.



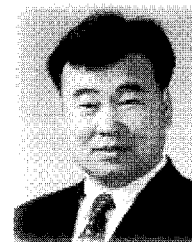
최 호 진

2007년 충남대학교 메카트로닉스공학과 졸업. 현재 충남대학교 메카트로닉스공학과 대학원 과정. 관심분야는 원격제어, 시간 지연 시스템 보상, 지능 시스템.



이 근 형

2006년 경일대학교 제어계측공학과 졸업. 2009년 충남대학교 메카트로닉스공학과 대학원 졸업. 관심분야는 지능제어, 지능 로봇 시스템.



정 슬

1988년 미국 웨인 주립대 전기 및 컴퓨터 공학과 졸업. 1991년 미국 캘리포니아대 데이비스 전기 및 컴퓨터 공학과 석사. 동 대학 박사. 1997년~현재 충남대학교 메카트로닉스공학과 교수. 관심 분야는 지능 제어 및 지능 로봇 시스템,

임베디드 제어기 설계, 로봇과 인간의 상호작용, 무인 로봇의 원격제어.