

BİR SCARA ROBOTUN PD VE ÖĞRENME DENETLEYİCİLER İLE PERFORMANSININ KARŞILAŞTIRILMASI

Serhan YAMAÇLI¹

Hüseyin CANBOLAT²

¹Elektronik ve Bilgisayar Eğitimi Bölümü
Tarsus Teknik Eğitim Fakültesi Fakültesi
Mersin Üniversitesi, 33480, Tarsus, Mersin

²Elektrik-Elektronik Mühendisliği Bölümü
Mühendislik Fakültesi

Mersin Üniversitesi, 33343, Çiftlikköy, Mersin.

¹e-posta: syamacli@mersin.edu.tr

²e-posta: huseyinc@mersin.edu.tr

Anahtar sözcükler: Konum izleme, SCARA tipi robot, Simulink, öğrenme metotlu kontrolör

ABSTRACT

This paper presents a comparison of the control of SCARA robot manipulators utilizing a proportional-differential (PD) controller and a learning-based controller. Computer simulations of an experimental SCARA-type manipulator are carried in MATLAB-Simulink environment. Simulation results show that, learning controller reduces the position error effectively without adding so much complexity to the system.

1. GİRİŞ

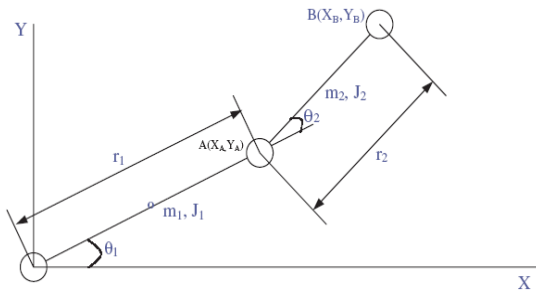
Robot manipülatörleri, dikdörtgenel, silindirik, küresel gibi birçok şekilde hareket edebilecek şekilde gerçekleştirilebilirler. Endüstriyel açıdan bakıldığında, üretim bandında yer alan tut-ve-koy, ergitme ve boyama gibi işlemler için, çokça kullanılan bir robot tipi Seçici Uyumlu Eklemler Robot Kolu (Selective Compliance Articulated Robotic Arm-SCARA) tipi robottur. SCARA tipi robotun, genellikle dört serbestlik derecesi vardır. Omuz, dirsek ve bilek eklemleri servo motorlar vasıtasıyla hareket ettirilirken, dördüncü hareket pnömatik olarak sağlanmaktadır.

Uygulamaya bağlı olarak çok değişik tipte robot manipülatörleri tasarlanmaktadır, ancak robotun kol şekli kadar önemli olan bir nokta da, robota istenen hareketi en az hata ile yaptırabilecek olan kontrolörün tasarlanmasıdır. Literatürde robot kontrolüne yönelik klasik oransal-integral-türevsel (PID) kontrolörlerin yanında [1], daha az hatayı sağlamak üzere adaptif [2, 3], öğrenme [4-9], yapay zeka algoritmalarını [10,11] ve bulanık mantık algoritmalarını [6] kullanan çeşitli kontrolörlerin tasarlanmasına ve bu algoritmaların mikroişlemci üzerinde gerçekleştirilmesine yönelik çalışmalar hız kazanmıştır.

Bu çalışmada, Serpent-1 modeli bir SCARA robot manipülatörünün, oransal-türevsel (PD) ve tekrarlı-öğrenme (repetitive-learning) algoritması kullanan denetleyiciler ile kontrol edilmesi sonucunda elde edilen performansların karşılaştırılması yapılarak, öğrenme algoritmasının getirmiş olduğu hata azalması ile sisteme eklemiş olduğu karmaşıklığın bir değerlendirilmesi yapılmıştır.

2. SCARA-TİPİ ROBOT KONFIGÜRASYONU

Bu çalışmada ele alınan SCARA-tipi robot, omuz ve dirsek eklemleri servo motorlarla kontrol edilen, üçüncü hareketi pnömatik olan deneysel bir robottur ve kartezyen koordinat sistemindeki şekli, Şekil-1'de gösterilmiştir [1, 11-20].



Şekil-1. SCARA-tipi robot konfigürasyonu [1]

Bu robotun elektriksel ve mekanik denklemleri

$$\begin{bmatrix} L_{a1} & 0 \\ 0 & L_{a2} \end{bmatrix} \begin{bmatrix} \dot{I}_{a1} \\ \dot{I}_{a2} \end{bmatrix} + \begin{bmatrix} R_{a1} & 0 \\ 0 & R_{a1} \end{bmatrix} \begin{bmatrix} I_{a1} \\ I_{a2} \end{bmatrix} + \begin{bmatrix} K_{e1} & 0 \\ 0 & K_{e2} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} V_{a1} \\ V_{a2} \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} K_{T1} & 0 \\ 0 & K_{T2} \end{bmatrix} \begin{bmatrix} I_{a1} \\ I_{a2} \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} A & B \\ E & D \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C \\ F \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \quad (3)$$

ile verilebilir. (3) denklemindeki matris elemanları ise,

$$A = J_{m1} + \frac{(J_1 + J_2)}{N_1^2} + \frac{m_1 r_1^2 + m_2 r_2^2 + 4m_2 r_1^2}{4N_1^2} + \frac{m_2 r_1 r_2}{N_1^2} \begin{pmatrix} \cos\left(\frac{\theta_{m1}}{N_1} + \frac{\theta_{m2}}{N_2}\right) \cdot \cos\left(\frac{\theta_{m1}}{N_1}\right) \\ + \sin\left(\frac{\theta_{m1}}{N_1} + \frac{\theta_{m2}}{N_2}\right) \cdot \sin\left(\frac{\theta_{m1}}{N_1}\right) \end{pmatrix} \quad (4.1)$$

$$B = \frac{J_2}{N_1 N_2} + \frac{m_2 r_2^2}{4N_1 N_2} + \frac{m_2 r_1 r_2}{2N_1 N_2} \begin{pmatrix} \cos\left(\frac{\theta_{m1}}{N_1} + \frac{\theta_{m2}}{N_2}\right) \cdot \cos\left(\frac{\theta_{m1}}{N_1}\right) \\ + \sin\left(\frac{\theta_{m1}}{N_1} + \frac{\theta_{m2}}{N_2}\right) \cdot \sin\left(\frac{\theta_{m1}}{N_1}\right) \end{pmatrix} \quad (4.2)$$

$$C = \frac{m_2 r_1 r_2}{N_1 N_2} \left(\frac{\dot{\theta}_{m1} + \dot{\theta}_{m2}}{N_1} + \frac{\dot{\theta}_{m2}^2}{2N_2} \right) \begin{pmatrix} \cos\left(\frac{\theta_{m1}}{N_1} + \frac{\theta_{m2}}{N_2}\right) \cdot \sin\left(\frac{\theta_{m1}}{N_1}\right) \\ - \sin\left(\frac{\theta_{m1}}{N_1} + \frac{\theta_{m2}}{N_2}\right) \cdot \cos\left(\frac{\theta_{m1}}{N_1}\right) \end{pmatrix} \quad (4.3)$$

$$D = J_{m1} + \frac{J_2}{N_2^2} + \frac{m_2 r_2^2}{4N_2^2} \quad (4.4)$$

$$E = \frac{J_2}{N_1 N_2} + \frac{m_2 r_2^2}{4N_1 N_2} + \frac{m_2 r_1 r_2}{2N_1 N_2} \begin{pmatrix} \cos\left(\frac{\theta_{m1}}{N_1} + \frac{\theta_{m2}}{N_2}\right) \cdot \cos\left(\frac{\theta_{m1}}{N_1}\right) \\ + \sin\left(\frac{\theta_{m1}}{N_1} + \frac{\theta_{m2}}{N_2}\right) \cdot \sin\left(\frac{\theta_{m1}}{N_1}\right) \end{pmatrix} \quad (4.5)$$

$$F = \frac{m_2 r_1 r_2}{2N_1 N_2} \left(\frac{\dot{\theta}_{m1}^2}{N_1} \right) \begin{pmatrix} \sin\left(\frac{\theta_{m1}}{N_1} + \frac{\theta_{m2}}{N_2}\right) \cdot \cos\left(\frac{\theta_{m1}}{N_1}\right) \\ - \cos\left(\frac{\theta_{m1}}{N_1} + \frac{\theta_{m2}}{N_2}\right) \cdot \sin\left(\frac{\theta_{m1}}{N_1}\right) \end{pmatrix} \quad (4.6)$$

3. PD VE ÖĞRENME DENETLEYİCİLER

Kontrol sistemlerinde sıklıkla kullanılan denetleyici yapılarından biri oransal-türevsel (PD) kontrolördür. PD kontrolörün avantajı, denetleyici sisteminin kolaylıkla gerçekleştirilebilmesidir. Özellikle düşük işlem hızı olan mikrodenetleyici sistemlerinde PD kontrolör kullanılabilir. Diğer taraftan

şeklinde. Bu denklemlerde; I_{a1} ve I_{a2} motor akımlarını, V_{a1} ve V_{a2} motor gerilimlerini, T_1 ve T_2 motorların ürettiği torkları, θ_1 ve θ_2 kol açılarını, θ_{m1} ve θ_{m2} ise motor açılarını göstermektedir. Denklemlerdeki fiziksel parametreler ve Serpent-1 model robot için değerleri ise Tablo-1'de verilmektedir [1].

Tablo-1: Robot parametreleri ve değerleri

Parametre	Anlamı	Değeri
L_{a1}	1. motorun sargı endüktansı	1.3mH
L_{a2}	2. motorun sargı endüktansı	1.3mH
R_{a1}	1. motorun sargı rezistansı	3.5 Ω
R_{a2}	2. motorun sargı rezistansı	3.5 Ω
K_{e1}	1. motorun ters e.m.k. sabiti	0.047V.s/rad
K_{e2}	2. motorun ters e.m.k. sabiti	0.047V.s/rad
K_{T1}	1. motorun tork sabiti	0.047Nm/A
K_{T2}	2. motorun tork sabiti	0.047Nm/A
J_1	1. kol ataleti	0.0980kgm ²
J_2	2. kol ataleti	0.0980kgm ²
J_{m1}	1. motor ataleti	3.3.10 ⁻⁶ kgm ²
J_{m2}	2. motor ataleti	3.3.10 ⁻⁶ kgm ²
m_1	1. kol kütlesi	1.90 kg
m_2	2. kol kütlesi	0.93 kg
r_1	1. kol uzunluğu	250 mm
r_2	2. kol uzunluğu	150 mm
N_1	1. dişli kutusu oranı	90
N_2	2. dişli kutusu oranı	220

kullanımının basitliğine rağmen, sağladığı performans gelişmiş denetleyici sistemlerine göre düşüktür.

Günümüzde öne çıkan yüksek performanslı denetleyicilerden üçü; öğrenme metodu kullanan algoritmalar, yapay zeka uygulaması ve bulanık mantık kullanan denetleyicilerdir. Yapay zeka ve bulanık mantık kullanan denetleyicilerde yüksek performans elde edilebilmekte ancak özellikle yapay

zeka ve bulanık mantık uygulamalarında, önceden bir eğitim aşamasının gerekmesi ve her yeni konum fonksiyonuna göre sistem parametrelerinin değiştirilmesinin gerekliliği bu sistemler açısından bir dezavantaj oluşturmaktadır [10]. Bunun yanında, öğrenme algoritması kullanan denetleyiciler diğer bir seçenek olarak ortaya çıkmaktadır ve literatürde öğrenme metoduna yönelik çalışmalar artmaktadır [4-9].

PD kontrolörün temel denklemi

$$V_k(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (5)$$

şeklinde ifade edilebilir. Burada $e(t)$ hata işaretini, K_p , oransal kontrol sabitini, K_d ise türevsel kontrol sabitini göstermektedir.

Diğer taraftan öğrenme algoritmalarının robot kontrolünde kullanımına yönelik çalışmalardan biri, Messner ve Horowitz'in yapmış olduğu çalışmadır [5]. Yine melez adaptif/öğrenme metodunu kullanan çalışmalar ise Dawson ve Canbolat tarafından 1995 yılında irdelenmiş ve bu yöntemlerin kullanımının sağlayacağı iyileştirmeler, Uğuz ve Canbolat tarafından literatüre sunulmuştur [7, 9].

[5]'de önerilen tekrarlamalı-öğrenme denetleyicisinin temel denklemi

$$T(t) = \hat{w}_r(t) + F_v e_v(t) + F_p e(t) + d_m(x_p, \dot{x}_p) + q_n(e_v) \quad (6)$$

şeklinde. Bu denklemde; $T(t)$ üretilen tork, $\hat{w}_r(t)$ mekanik belirsizlikleri içeren fonksiyon, F_v ve F_p , PD kazançlar, $e_v(t)$ referans hız hata fonksiyonu, $e(t)$ konum hata fonksiyonu, $d_m(x_p, \dot{x}_p)$ sürtünme kompanzasyon fonksiyonu, $q_n(e_v)$ ise, eğrisel bir kompanzasyon fonksiyonudur. Bu denklemde de görülebileceği gibi, tekrarlı-öğrenme algoritmasının içinde de PD mekanizması vardır, ama buna ek olarak getirmiş olduğu fark, bilinmeyen periyodik etki fonksiyonu, $\hat{w}_r(t)$, ile öğrenmenin elde edilmesidir. (6)'daki bileşenlerden konum hatası

$$e_p(t) = x_d(t) - x(t) \quad (7)$$

ile tanımlanır. Burada $x(t)$ manipülatörün gerçekleşen konumu, $x_d(t)$ ise manipülatörün istenen konumudur.

Referans hız hata fonksiyonu ise,

$$e_v(t) = \dot{e}_p(t) + \lambda e_p(t), \quad \lambda > 0 \quad (8)$$

olarak verilir. Eğrisel kompanzasyon fonksiyonu ise,

$$q_n(e_v) = \sigma |e_p|^2 |e_v| \quad (9)$$

ile verilmektedir. Öğrenmeyi gerçekleştiren periyodik etki fonksiyonu $\hat{w}_r(t)$ ise,

$$\hat{w}_r(t) = \int_0^T K(t, \tau) \hat{c}(t, \tau) d\tau \quad (10)$$

$$\frac{\partial \hat{c}(t, \tau)}{\partial t} = K(t, \tau) K_L R^T e(t) \quad (11)$$

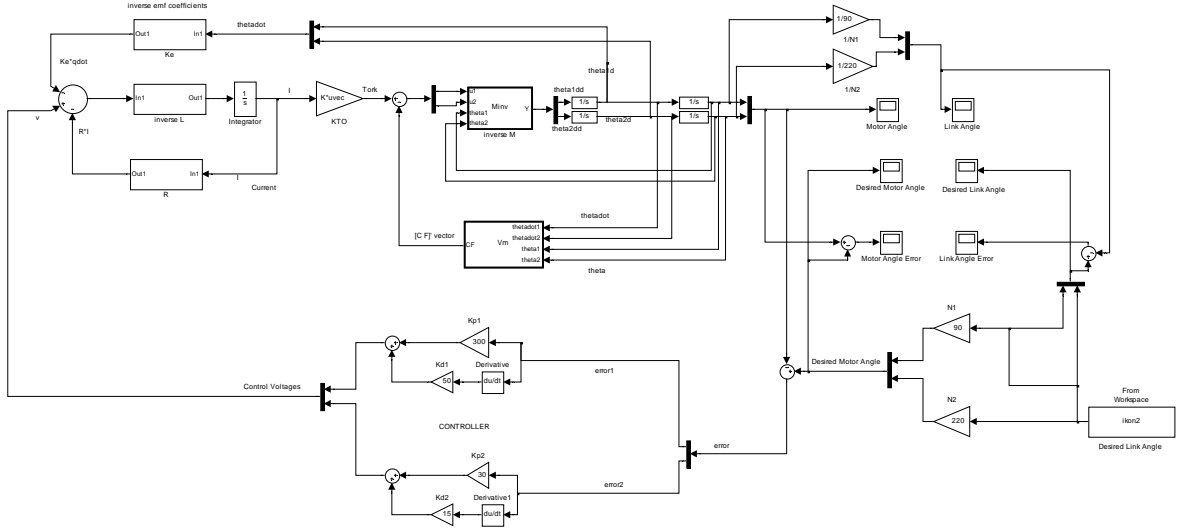
ile verilir. Burada, $K(t, \tau)$ tasarımcının seçebileceği bilinen bir fonksiyon, K_L ve R ise, tasarım sabitleridir.

4. SİMÜLASYON SONUÇLARI VE KARŞILAŞTIRMALAR

Bu çalışmada, parametreleri 2. Bölüm'de verilen Serpent-1 modeli SCARA robot manipülatörünün PD ve tekrarlı-öğrenme tipi denetleyicilerle kontrolü sonucunda elde edilen konum izleme performansları, MATLAB Simulink ortamındaki simülasyonlarla incelenmiştir.

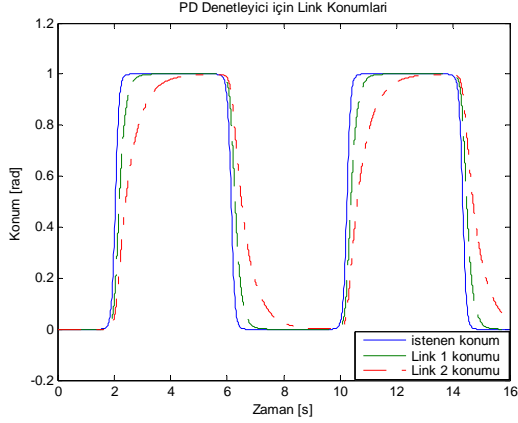
Simulink'de oluşturulan model dosyasının görünümü Şekil-2'de verilmiştir.

Robotun izlemesi istenen konum, Simulink tarafından MATLAB workspace'den alınan tut-ve-koy tarzı bir harekettir ve Şekil-3'de gösterilmiştir. Bu fonksiyon, bir robot ve denetleyicisi için zor hareketlerden birisidir çünkü çok yüksek hız ve ivme değerlerine sahiptir. Testlerde özellikle bu tip bir istenilen konum seçilmiştir, bunun amacı robot için anlamlı bir test ortamı oluşturmaktır.

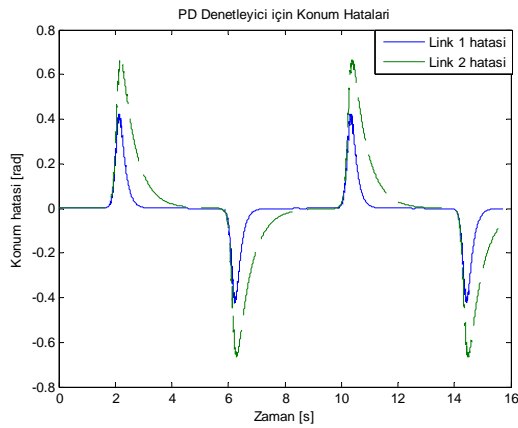


Şekil-2. SCARA robot ve denetleyicisinin Simulink modeli

İlk simülasyonda denetleyici olarak denklemleri (5)'de gösterilen PD denetleyici kullanılmıştır. Birinci link için oransal kontrol sabiti $K_p=300$, türevsel kontrol sabiti $K_d=50$; ikinci link için de $K_p=30$, $K_d=15$ alınarak simülasyonlar gerçekleştirilmiştir [1]. PD denetleyici için elde edilen link konumları ile konum izleme hataları sırasıyla Şekil-3 ve Şekil-4'de gösterilmiştir.



Şekil-3. İstenen link konumları ile PD denetleyici kullanıldığında elde edilen link konumları



Şekil-4. PD denetleyici kullanıldığında link konum izleme hataları

Şekil-4'den de görüldüğü gibi link 1 hatasının maksimum değeri 0.4 radyan iken link 2 hatasının maksimum değeri 0.65 radyandır.

İkinci simülasyonda ise, Şekil-2'deki PD denetleyici kaldırılarak yerine Şekil-5'de gösterilen öğrenme algoritmasını gerçekleştiren Simulink blokları kullanılmıştır. Tam bir karşılaştırma yapmak için robotun fiziksel parametreleri değiştirilmeden simülasyon yapılmıştır.

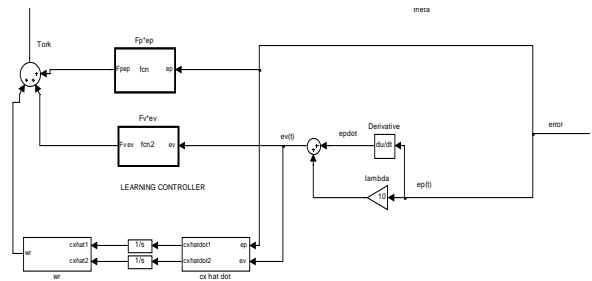
Öğrenme algoritması kullanan denetleyici parametreleri ise

$$F_p = \begin{bmatrix} 2000 & 0 \\ 0 & 160 \end{bmatrix} \quad (12.1)$$

$$F_v = \begin{bmatrix} 200 & 0 \\ 0 & 4 \end{bmatrix} \quad (12.2)$$

$$K_L = \begin{bmatrix} 2000 & 0 \\ 0 & 175 \end{bmatrix} \quad (12.3)$$

ve $\lambda_p = 10$, $\sigma_n = 0$, $d_m(x_p) = 0$ alınmıştır [5].



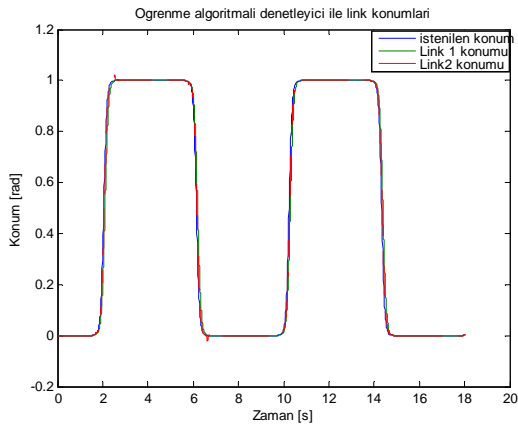
Şekil-5. Öğrenme algoritmasını simülasyon ortamına taşıyan Simulink blokları

Öğrenme algoritması kullanan denetleyicide $\hat{w}_r(t)$ bilinmeyen periyodik etki fonksiyonunu hesaplamak için (10) ve (11)'de görüldüğü gibi $K(t, \tau)$ fonksiyonu da seçilmelidir. Bu fonksiyon

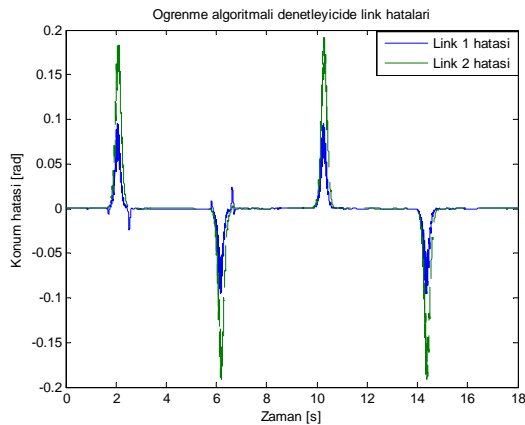
$$K(t, \tau) = f(t - \tau - kT), \quad |t - \tau| \leq kT - \frac{T}{2} \quad (13)$$

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-0,5\left(\frac{t}{\sigma}\right)^2\right) \quad (14)$$

olarak seçilmiştir. (14)'deki fonksiyonda, $\sigma = 0.04$ ve $T = 8.192s$ alınmıştır. Bu şartlar altında elde edilen link konumları ile konum izleme hataları sırasıyla Şekil-6 ve Şekil-7'de gösterilmiştir. (10) ve (11)'deki integralleri simülasyonda hesaplamak için nümerik integrasyon kullanılmıştır. Nümerik integrasyon MATLAB kodları cxhdatot ve wr gömülü MATLAB bloku içindedir.



Şekil-6. İstenen link konumları ile öğrenme algoritmalı denetleyici kullanıldığında elde edilen link konumları



Şekil-7. Öğrenme algoritmasını kullanan denetleyici kullanıldığında link konum izleme hataları

Şekil-7'den de görüldüğü gibi öğrenme algoritmasını kullanan denetleyici kullanıldığında, link 1 hatasının maksimum değeri 0.1 radyan iken link 2 hatasının maksimum değeri 0.18 radyandır. PD denetleyici ile elde edilen değerlerle karşılaştırıldığında, öğrenme

algoritması ile hatanın PD denetleyiciye göre %75 oranında azaldığı görülmüştür.

5. SONUÇLAR VE GELECEK ÇALIŞMALAR

Bu çalışmada,

- SCARA tipi bir deneysel robotun fiziksel parametreleri baz alınarak Simulink modeli elde edilmiştir.
- Bu robotun tut-ve-koy tarzı bir konum fonksiyonunu gerçekleştirilmesi için; klasik PD denetleyici ve öğrenme algoritması ile kontrollerinin benzetimi Simulink ortamında yapılmıştır.
- Simulink benzetimlerinden elde edilen sonuçlara göre PD ile kontrolde elde edilen konum izleme hatalarının maksimum değeri, öğrenme algoritması ile elde edilenlerden 4 kat daha fazladır. Bu da öğrenme algoritması ile kontrolün, PD kontrole göre, sisteme çok fazla komplekslik eklemekten, hata oranını %75 düşürdüğü anlamına gelmektedir. Daha fazla eklemli robotlarda benzer sonuçların alınması beklenmektedir, çünkü teori genel bir robota uygulanabilir. Bunun için bu tip bir robotun modellenip simülasyon ve deneylerinin yapılması gereklidir.

Gelecek çalışmalarda ise, SCARA robotun deneysel olarak gösterdiği konum hataları elde edilecektir. Bu amaçla, Simulink modeli'nden Real Time Workshop vasıtasıyla DSP kodu elde edilecek ve deneysel SCARA robotun kontrol işlemcisine yüklenecek, sonuçlar ölçülecektir.

KAYNAKLAR

- [1] Das T., Dülger C, Mathematical Modeling, Simulation and Experimental Verification of a SCARA Robot, SIMULATION MODELLING PRACTICE AND THEORY, Vol: 13, pp. 257-271, 2005.
- [2] Kaneko K., Horowitz R., Repetitive and Adaptive Control of Robot Manipulators with Velocity Estimation, IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 13, NO. 2, pp. 204-217, 1997.
- [3] de Queiroz M.S., Dawson D.M., Canbolat H., Adaptive position/force control of BDC-RLED robots without velocity measurements, PROCEEDINGS OF IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, Vol:1, pp. 525-530, 1997.
- [4] Canbolat H., Hu J., Dawson D.M, A Hybrid Learning/Adaptive Partial State Feedback Controller for RLED Robot Manipulators, PROCEEDINGS OF THE 34TH IEEE CONFERENCE ON DECISION AND CONTROL, Vol: 1, pp. 39-44, 1995.
- [5] Messner W., Horowitz R., Kao W.W., Boals M, A new adaptive learning rule, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, Vol:36, pp.188-197, 1991.

- [6] Horowitz R., Messner W., Moore J.B., Exponential convergence of a learning controller for robot manipulators, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, Vol: 36, pp. 890-894, 1991.
- [7] Uğuz H., Canbolat H., Simulation of a Hybrid Adaptive-Control Law, for a Rigid Link Electrically Driven Robot Manipulator, ROBOTICA, Vol: 24, pp. 349-354, 2006.
- [8] Canbolat H., Hu J., Dawson D. M., A hybrid learning/adaptive partial state feedback controller for RLED robot manipulators, INTERNATIONAL JOURNAL OF SYSTEMS SCIENCE, Vol: 27, pp. 1123-1132, 1996.
- [9] Uguz H., Elektriksel Tahrikli Robot Manipulatorlerinin Öğrenme Kontrol Metoduyla Kontrol ve Simulasyonu, M.Sc. Thesis, Mersin Üniversitesi Fen Bilimleri Enstitüsü, Mersin, 2003.
- [10] Golnazarian W., Time-Varying Neural Networks for Robot Trajectory Control, Ph.D. Thesis, University of Cincinnati, U.S.A., 1995.
- [11] Jungbeck, M., Madrid, M.K., Optimal neural network output feedback control for robot manipulators, PROCEEDINGS OF THE SECOND INTERNATIONAL WORKSHOP ON ROBOT MOTION CONTROL, pp.85-90, 2001.
- [12] Lewis F. L., Abdallah C. T., Dawson D. M., Control of Robot Manipulators, Macmillan, New York, USA 1993.
- [13] Chan S.P., Chen H., An Efficient Algorithm For Identification of Scara Robot Parameters Including Drive Characteristics, PROC OF THE 25th ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONICS SOCIETY, Vol: 2, pp. 1004-1019, 1999.
- [14] Harlow L., Design of a Disturbance Observer Applied to a SCARA Robotic Arm, M.Sc. Thesis, San Jose State University, U.S.A., 2005.
- [15] Villa-Rosade D.N., Dominiguez-Lopez J.A., A MATLAB Toolbox for Robotic Manipulators, Proceedings of the Sixth Mexican International Conference on Computer Science (ENC'05), pp. 256-263, 2005, Mexico.
- [16] Er M.J., Lim M.T., Lim H.S., A Real Time Hybrid Adaptive Fuzzy Control of a SCARA Robot, MICROPROCESSORS AND MICROSYSTEMS, Vol: 25, pp. 369-378, 2001.
- [17] Er M.J., Low C.B., Nah K.H., Lim M.H., Ng S.Y., Real-Time Implementation of a Dynamic Fuzzy Neural Network Controller for a SCARA Robot, MICROPROCESSORS AND MICROSYSTEMS, Vol: 26, pp. 449-461, 2002.
- [18] Bhatia P., Thirunarayanan J., Dave N., An Expert-System Based Design of a SCARA Robot, EXPERT SYSTEMS WITH APPLICATIONS, Vol:15, pp. 99-109, 1999.
- [19] Milutinovic D., Universal Compliant Device Based on a SCARA Concept, ROBOTICS AND COMPUTER INTEGRATED MANUFACTURING, Vol: 13, pp. 321-319, 1997.
- [20] Simpson J.W.L., Cook C.D., Li Z., Sensorless Force Estimation for Robots With Friction, PROC. OF AUSTRALASIAN CONFERENCE ON ROBOTICS AND AUTOMATION, pp. 94-99, Australia, 2002.

ÖZGEÇMİŞLER

Serhan YAMAÇLI: 10 Kasım 1981 yılında Tarsus'ta doğdu. 1999 yılında Tarsus A.K.B.A.L.'i bitirerek aynı yıl İ.T.Ü. Elektrik-Elektronik Fakültesi Elektronik ve Haberleşme Mühendisliği'nde lisans öğrenimine başladı. 2003 yılında lisans diplomasını aldı ve 2004 yılında Mersin Üniversitesi Tarsus Teknik Eğitim Fakültesi'nde öğretim görevlisi olarak çalışmaya başladı. 2005 yılından beri Mersin Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü'nde yüksek lisansına devam etmektedir. Serhan YAMAÇLI'nın 11 adet uluslararası, 11 adet ulusal yayını, araştırmacı olarak katıldığı ve hala yürüten 3 adet bilimsel araştırma projesi, 1 adet tamamlanmış AB Leonardo da Vinci projesi (yürütücü olarak) vardır. Serhan YAMAÇLI, Marquis Who'sWho in Science and Engineering (Bilim ve Mühendislikte Kim Kimdir) ansiklopedisi'ne 2006 yılı itibarıyla girmiştir. Serhan YAMAÇLI'nın şu anki çalışma konuları arasında CMOS analog entegre devre tasarımı, elektronik elemanların modellenmesi, mikrodalga devre tasarımı ve robotik vardır. İleride kuantum bilgisayarlar konusunda çalışmayı düşünmektedir.

Hüseyin CANBOLAT: 12 Aralık 1966'da Adana'da doğdu. 1985 yılında Adana Anadolu Lisesi'ni, 1989 ve 1993'te ODTÜ Elektrik Müh. Lisans ve yüksek lisans eğitimini bitirdi. 1997'de Clemson ABD'de doktora derecesini aldıktan sonra Mersin Üniv. Elektrik-Elektronik Müh. Bölümünde öğretim üyesi olarak göreve başladı. Halen aynı görevine devam etmektedir. Çalışma alanları kontrol ve kumanda sistemleri, robotik, mekatronik, elektromekanik sistem kontrolü ve MEMS'tir.