Çok Katmanlı-Geri Beslemeli Yapay Sinir Ağı (MFLNN) ve Parçacık Sürü Optimizasyon Algoritması (PSO) Kullanarak Bir DC Motor Tanılaması

Identification of A DC Motor Using the Multifeedback-Layer Neural Network (MFLNN) and the Particle Swarm Optimization Algorithm (PSO)

İnayet Özge AKSU[1], *Ramazan ÇOBAN*[2]

[1]Bilgisayar Mühendisliği Bölümü
Adana Bilim ve Teknoloji Üniversitesi
oaksu@adanabtu.edu.tr

[2]Bilgisayar Mühendisliği Bölümü
Çukurova Üniversitesi
rcoban@cu.edu.tr

Özet

Bu çalışmada, bir DC motorun dinamik yapısının tanılanması için Çok Katmanlı-Geri Beslemeli Sinir Ağı (MFLNN) kullanılmıştır. MFLNN ağı, ileri yönlü ve geri yönlü bağlantılar içeren yinelemeli bir yapay sinir ağı türüdür. MFLNN ağı; kuş ve kuş sürülerinin doğadaki davranışlarından esinlenerek geliştirilen sezgisel bir yöntem olan Parçacık Sürü Optimizasyonu (PSO) algoritması ile eğitilerek bağlantı ağırlıkları ayarlanmıştır. Çalışmada maksimum 2400 rpm şaft hızıyla çalışan Digiac 1750 DC motor deney seti kullanılmıştır. MFLNN-PSO tanılama sistemi ilk olarak eğitim verileriyle eğitilmiştir. Daha sonra test aşamasına geçilmiştir. Bu aşamada sisteme farklı bir veri seti verilmiştir. Çalışma sonunda elde edilen sonuçlar, sayısal verilerle ve grafiklerle sunulmuştur. Deney sonuçları MFLNN-PSO tanılama sisteminin başarısını göstermektedir.

Abstract

In this study, the Multifeedback-Layer Neural Network (MFLNN) was employed for identification of a dc motor dynamics. The MFLNN is a type of recurrent neural networks, which includes feedforward and feedback connections. The connection weights of the MFLNN were adjusted by the PSO algorithm which is a intuitional method developed by inspiring the behavors of the birds and birds flocks in the nature. In the study, the Digiac 1750 DC motor setup, which operates at maximum 2400 rpm shaft speed was used. At first, the MFLNN-PSO identification system was trained by a training data set. Then, the test phase was carried out. At this stage, a different data set rather than the training data set was presented to the identification system. The results obtained at the end of the study were exhibited by means of some numerical and graphical data. Results of the experiments show performance of the MFLNN-PSO identification system.

1. Giriş

Son yıllarda yapay sinir ağı yapısı hemen hemen tüm alanlarda kullanılmaya başlanmıştır. İlk başlarda yalnızca doğrusal sistemler üzerinde kullanılsa da günümüzde doğrusal olmayan sistemlerin tanılanması ve kontrolü alanlarında sıklıkla karşımıza çıkmaktadır. [1] ve [2]'de doğrusal olmayan sistemlerin tanılanması için farklı yapılarda sinir ağları kullanılmıştır.

Elman [3] ve Jordan [4] tarafından geliştirilmiş yinelemeli (recurrent) yapay sinir ağları, nöronları kendi aralarında geri besleme olarak bilgi gönderebilen sinir ağı türüdür. Farklı yapılardaki yinelemeli sinir ağları; [5]'de dinamik sistemlerin kontrolü için kullanılırken, [6] ve [7]'de dinamik sistemlerin tanılanması ve kontrolünde kullanılmışlardır.

Bu çalışmada, bir DC Motorun dianamik davranışının tanılanmasında Savran [8] tarafından geliştirilmiş yeni bir yinelemeli sinir ağı olan MFLNN kullanılmıştır. MFLNN diğer yinelemeli sinir ağlarından farklı olarak; geri besleme aşamasında birim gecikmeli geri besleme yerine katman yapısı kullanmaktadır. Yapay sinir ağının ağırlık katsayılarının eğitimi için; hızlı bir optimizasyon algoritması olması ve eğitme algoritması olarak sadece uygunluk fonksiyonu kullanmasından dolayı PSO algoritması tercih edilmiştir.

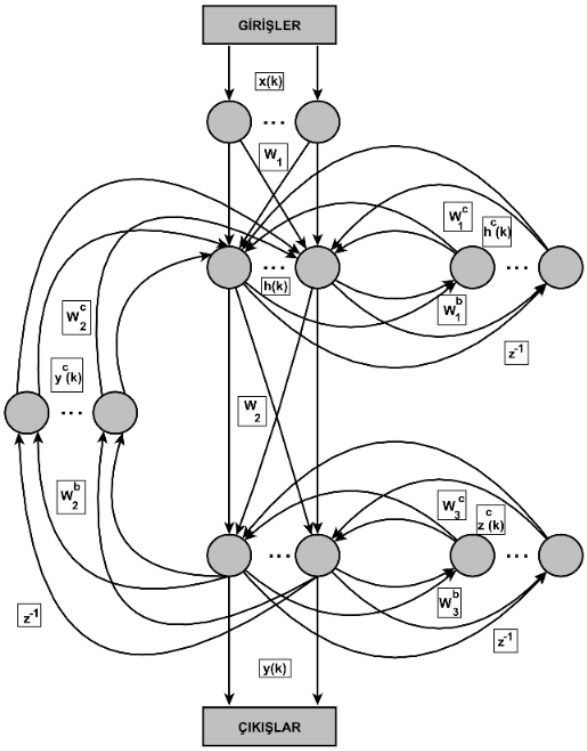
Çalışmanın ikinci bölümünde MFLNN yapısı, üçüncü bölümünde PSO yapısı ve dördüncü bölümünde DC Motor modeli ele alınmaktadır. Beşinci bölümünde eğitme yapısı incelenmiştir. Altıncı bölümünde benzetim sonuçları sunulmuştur. Yedinci bölümde yöntemin sistem üzerindeki başarısı vurgulanmıştır.

2. MFLNN

Bu çalışmada bir DC motorun dinamik davranışının tanılanmasında MFLNN kullanılmıştır. MFLNN, 2007

yılında A. Savran tarafından geliştirilmiş özyinelemeli bir yapay sinir ağı çeşididir [8]. Bir gizli katman, bir giriş katmanı ve bir çıkış katmanı tercih edilerek kullanılan MFLNN'nin genel yapısı Şekil 1'de gösterilmiştir.

Şekilde anlaşılabilirliğin artması için bias değerleri gösterilmemiştir. (k) ağın giriş değeri $y(k)$ ise çıkış değeridir. Katmanlar arasında iletişim ileri yönlü ve geri yönlü sağlanmaktadır. W_1 ve W_2 katsayıları ileri yönlü katman katsayıları olup giriş ve gizli katman arasındaki ve gizli katman ile çıkış katmanı arasındaki bilgi akışında kullanılmaktadırlar. W_1^b , W_2^b ve W_3^b geri besleme katmanlarının giriş değerlerinin katsayılarıdır. $h^c(k)$, $y^c(k)$ ve $z^c(k)$ geri besleme katmanlarının çıkış değerleridir ve aynı zamanda gizli katman ve çıkış katmanının giriş değerleridir. W_1^c , W_2^c ve W_3^c gizli katman ve çıkış katmanının giriş değerlerinin katsayılarıdır. $^{-1}$ birim gecikme zamanını ifade ederken k zaman indeksini ifade etmektedir.



Şekil 1: MFLNN yapısı.

Ağın eğitiminin sağlanabilmesi için tanılama sisteminde hata değerinin hesaplanması gerekmektedir. Eğitim $k=1$ den $k=T$ zaman adımına kadar yapılmaktadır. Her bir k zaman adımında hata değeri MFLNN'nin çıkışı ile istenen değer arasındaki fark alınarak hesaplanır:

$$e(k)=y(k)-\hat{y}(k) \tag{1}$$

Geri besleme nöronlarının girişleri şu şekilde hesaplanır:

$$net_h^c(k)=\left[W_1^b h(k-1)\right]+B_1^b \tag{2}$$

$$net_y^c(k)=\left[W_2^b y(k-1)\right]+B_2^b \tag{3}$$

$$net_z^c(k)=\left[W_3^b y(k-1)\right]+B_3^b \tag{4}$$

W_1^b , W_2^b , W_3^b geri besleme katmanının giriş ağırlıklarını; B_1^b , B_2^b , B_3^b geri besleme ağının bias değerlerini gösterir. Daha sonra bu değerler aktivasyon fonksiyonlarından geçerek geri besleme nöronlarının çıkış değerlerini oluştururlar.

$$h^c(k)=\varphi_h^c(net_h^c(k)) \tag{5}$$

$$y^c(k)=\varphi_y^c(net_y^c(k)) \tag{6}$$

$$z^c(k)=\varphi_z^c(net_z^c(k)) \tag{7}$$

h^c , y^c , z^c geri besleme nöronlarının çıkışını; φ_h^c , φ_y^c , ve φ_z^c geri besleme nöronlarının aktivasyon fonksiyonlarını göstermektedir. Gizli katman nöronlarının çıkışının hesabı şu şekilde yapılmaktadır:

$$net_h(k)=\left[W_1 x(k)\right]+\left[W_1^c h^c(k)\right]+\left[W_2^c y^c(k)\right]+B_1 \tag{8}$$

$$h(k)=\varphi_h\left(net_h(k)\right) \tag{9}$$

B_1 gizli katman nöronunun bias değerini, φ_h gizli katmanının aktivasyon fonksiyonunu gösterir. W_1^c ve W_2^c geri besleme katmanının çıkış ağırlığını gösterir. Çıkış katmanı nöronlarının çıkışı şu şekilde hesaplanmaktadır:

$$net_y(k)=\left[W_2 h(k)\right]+\left[W_3^c z^c(k)\right]+B_2 \tag{10}$$

$$y(k)=\varphi_y\left(net_y(k)\right) \tag{11}$$

B_2 çıkış katman nöronunun bias değerini, φ_y çıkış katmanının aktivasyon fonksiyonunu gösterir. W_3^c .geri besleme katmanının çıkış ağırlığını gösterir.

3. PSO

PSO algoritması, kuşların doğal davranışından esinlenerek Kennedy ve Eberhart [9] tarafından geliştirilmiş bir optimizasyon tekniğidir. Global arama özelliği sayesinde doğrusal olmayan problemlerin çözümünde kullanılmaktadır. Algoritma rastgele çözümler içeren bir çözüm havuzu ile çalışmaya başlar. Her bir iterasyon adımında nesilleri güncelleyerek optimal sonucu arar. Çözüm uzayındaki parçacıkların konumları *pbest* (o parçacığın o ana kadar ki en iyi uygunluk değerine sahip pozisyonu) ve *gbest* (tüm parçacıklar içinde o ana kadar ki en iyi uygunluk değerine sahip parçacığın pozisyonu) değerlerine göre yapılmaktadır.

Sürüdeki parçacıkların hız ve konum güncellemesi şu denklemlerle yapılmaktadır:

$$v_{id} = w \times v_{id} + c_1 \times r_1 \times \left(p_{id} - x_{id} \right) + c_2 \times r_2 \times \left(p_{gd} - x_{id} \right) \quad (12)$$

$$x_{id} = x_{id} + v_{id} \quad (13)$$

x_{id} i. parçacığın konumunu, v_{id} i. parçacığın hızını göstermektedir. c_1 ve c_2 öğrenme faktörleridir. r_1 ve r_2 değerleri [0,1] arasında üretilen rastgele sayılardır. p_{id} i. parçacığın pbest değerini, p_{gd} gbest değerini göstermektedir. w eylemsizlik ağırlığıdır.

c_1, c_2, ve w değerleri PSO algoritmasının performansı üzerinde önemli rol oynamaktadırlar. c_1 değeri parçacıkları kendi lokal en iyilerine çekerken, c_2 değeri global en iyiye doğru yönlendirmektedir. w değeri ise lokal ve global arama arasında denge sağlamaktadır. PSO algoritmasının adımları Şekil 2'de gösterilmiştir.

For Sürüdeki her bir parçacık için
  Parçacıkların pozisyonları için rastgele atama yap
  Parçacıkların hızları için rastgele atama yap
End
Do
  For Sürüdeki her bir parçacık için
    Uygunluk değeri hesapla
    Uygunluk değeri *pbest* değerinden daha iyiyse
      pbest değerini yeni uygunluk değeri ile güncelle
  End
  En iyi uygunluk değerine sahip parçacığı *gbest* olarak belirle
  For Sürüdeki her bir parçacık için
    Parçacığın hızını güncelle (Denklem 12)
    Parçacığın konumunu güncelle (Denklem 13)
  End
While İterasyon sayısı tamamlanana kadar

Şekil *2: PSO algoritmasının adımları.*

4. DC Motor Modeli

Bu çalışmada maksimum 2400 rpm şaft hızıyla çalışan Digiac 1750 DC deney seti kullanılmıştır. Sistemin giriş sinyali motor armatür gerilimi, çıkış sinyali ise volt(gerilim) cinsinden şaft hızıdır [10]. Bu çalışmada örnekleme aralığı 30 ms alınarak giriş-çıkış verileri elde edilmiştir. Motor-elektromekanik sisteminin denklemi aşağıdaki gibidir:

$$v_a(t) = L_a \frac{di_a(t)}{dt} + R_a i_a(t) + K_m \omega_m(t) \quad (14)$$

$$J_m \left(\frac{d\omega_m(t)}{dt} \right) = T_m(t) - T_s(t) - R_m \omega_m(t) - T_f(\omega_m) \quad (15)$$

$$j_L \left(\frac{d\omega_L(t)}{dt} \right) = T_s(t) - R_L \omega_L(t) - T_d(t) - T_f(\omega_L) \quad (16)$$

$$T_s(t) = k_s(\theta_m(t) - \theta_L(t)) + B_s(\omega_m(t) - \omega_L(t)) \quad (17)$$

$$\frac{d\theta_m(t)}{dt} = \omega_m(t), \qquad \frac{d\theta_L(t)}{dt} = \omega_L(t) \quad (18)$$

Burada v_a armatür gerilimini, R_a armatür bobin direncini ve L_a indüktansı göstermektedir. i_a armatür akımı, K_m tork sabiti ve T_m üretilen motor torkudur. ω_m ve ω_L motorun açısal hızını, J_m ve J_L eylemsizlik momentini göstermektedir. R_m R_L ve B_s viskos sürtünme katsayılarıdır. T_d harici yük momenti, T_f lineer olmayan sürtünme ve T_s iletilen şaft torkudur. k_s yay sabitini ve t .zamanı göstermektedir.

Bu çalışmada kullanılan motor değerleri: armatür terminal gerilimi = 12 V, boşta akım = 120 mA, aşırı yük akımı (stall current) = 1.93 A, başlangıç torku = 7 Ncm/A, armatür endüktansı = 5 mH, armatür direnci= 6.2Ω, zaman sabiti = 19.6 ms, and tork sabiti= 3.5 Ncm/A.

5. Eğitme Yapısı

Bu çalışmada bir DC motorun dinamik davranışının tanılamasında MFLNN ağının katsayıları PSO algoritması ile eğitilmiştir. Başlangıçta ağ katsayıları rastgele atanmıştır. Tüm veriler ağda işlenerek RMSE hata değeri elde edilmiştir. PSO algoritması için gerekli olan uygunluk değeri için bu RMSE değeri kullanılmıştır [11,12]. RMSE değerinin hesaplanması şu şekildedir:

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left(y(k) = \hat{y}(k) \right)^2} \quad (19)$$

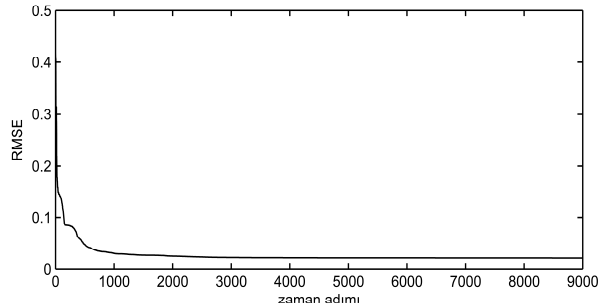
Burada N değeri veri sayısını, $\hat{y}(k)$ referans değerini $y(k)$ ağdan elde edilen çıkış değerini göstermektedir.

6. Benzetim

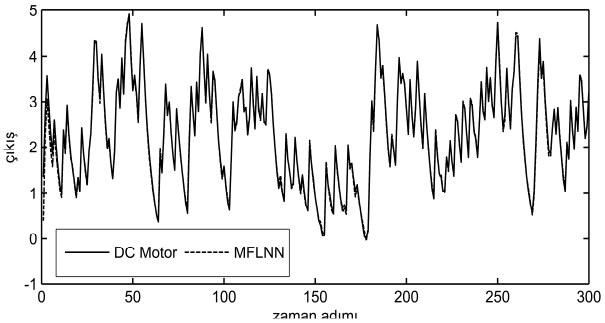
DC motorun dinamik yapısının tanılanması için MFLNN-PSO algoritması kullanılmıştır. MFLNN yapısında 1 giriş katman nöronu, 3 gizli katman nöronu ve 1 çıkış katman nöronu kullanılmıştır. MFLNN ağının katsayıları PSO algoritması ile eğitilmiştir. PSO algoritmasının çalışması için gerekli olan öğrenme kriteri olarak RMSE hata değeri kullanılmıştır.

Tanılama sürecinde ilk olarak eğitim aşaması yürütülmektedir. Eğitim aşamasında 300 veri kullanılmıştır ve 39 parametre eğitilmiştir. Eğitim aşamasında yapay sinir ağının yakınsamasını gösteren grafik Şekil 4'de verilmiştir. 9000 zaman adımında tamamlanan bu süreç sonundaki RMSE değeri 0.02'dir.

Sistemin test aşamasında da 300 veri kullanılmıştır. Test aşaması sonunda elde edilen RMSE değeri 0.09'dur. Test verisi için sistemin gösterdiği tanılama performansı Şekil 5'de gösterilmiştir.



Şekil *3: MFLNN için RMSE eğrisi.*



Şekil *4: MFLNN ve modelin çıkışı.*

7. Sonuç

Bu çalışmada bir DC motorun dinamik davranışının tanılanmasında yinelemili bir sinir ağı olan MFLNN yapısı kullanılmıştır. MFLNN ağının eğitilmesi için PSO algoritması önerilmiştir. Sonuçlar incelendiğinde, dinamik sistem davranışının başarıyla tasvir edildiği gözlemlenmektedir.

8. Kaynaklar

[1] Fu, Z.-J., Xie, W.-F., Han, X. ve Luo, W.-D., "Nonlinear Systems Identification and Control Via Dynamic Multitime Scales Neural Networks", IEEE Trans. Neural Netw. Learning Syst., 24(11), 1814-1823, 2013.

[2] Zhao H., Gao S., He Z., Zeng X., Jin W. ve Li T., "Identification of Nonlinear Dynamic System Using a Novel Recurrent Wavelet Neural Network Based on the Pipelined Architecture", IEEE Transactions on Industrial Electronics ,61(8),4171-4182, 2014.

[3] Elman J. L., "Finding Structures in Time", Cogn. Sci., 14, 179–211, 1990.

[4] Jordan M. I., "Supervised Learning and Systems with Excess Degrees of Freedom", COINS, Mass. Inst. Technol., Cambridge, MA, Tech.Rep. 88-27, 1988.

[5] Ku C.-C. ve Lee K. Y., "Diagonal Recurrent Neural Networks for Dynamic Systems Control", IEEE Transactions on Neural Networks, 6(1), 144-156, 1995.

[6] Lee C. H. ve Teng C. C., "Identification and control of dynamic systems using recurrent fuzzy neural networks", IEEE Trans. Fuzzy Syst., 8, 349-366, 2000.

[7] Wang, J.-S. ve Chen, Y.-P, "A Fully Automated Recurrent Neural Network for Unknown Dynamic System Identification and Control", IEEE Trans. Circuits and Systems- I, 53 (6), 1363-1372, 2006.

[8] A.Savran, "Multifeedback-Layer Neural Network",IEEE Transactions on Neural Networks,18(2),373-384, 2007.

[9] Kennedy, J. ve Eberhart. R., "Particle Swarm Optimization", IEEE International Conference on Neural Networks, 1942-1948, 1995.

[10] Eker, I., "Second-order sliding mode control with experimental application", ISA Trans., 49 (3), 394-405, 2010.

[11] Aksu, I. O. ve Coban, R., "Identification of Disk Drive Systems using the Multifeedback-Layer Neural Network and the Particle Swarm Optimization Algorithm", The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE2013), 2013,231-235.

[12] Aksu, I. O. ve Coban, R., "Training The Multifeedback-Layer Neural Network Using The Particle Swarm Optimization Algorithm", 2013 International Conference on Electronics, Computer and Computation (ICECCO 2013), 2013, 172-175.