



NARMA L-2 Denetleyici ile Top Çubuk Kontrolü

Ball and Beam Control based on NARMA L-2 Controller

Enes VARDAR¹, Kaplan KAPLAN^{1,*}, H. Metin ERTUNÇ¹

¹ Mekatronik Mühendisliği, Kocaeli Üniversitesi, Umuttepe Kampüsü, Kocaeli, 41380, Türkiye

Araştırma Makalesi

Gönderilme Tarihi : 03/12/2018

Kabul Tarihi : 31/12/2018

Anahtar Kelimeler

Top-çubuk,
Narma L-2,
YSA

Özet

Bu çalışmada, Yapay Sinir Ağları (YSA) kontrol algoritmalarından biri olan NARMA L-2 kontrol yapısı incelenerek top-çubuk sisteminde topun konum kontrolü gerçekleştirilmiştir. Top-çubuk sisteminin gerçekçi bir simülasyonu MATLAB ortamında oluşturularak topun konum kontrolü ilk önce manuel olarak el ile kontrol edilmiş ve YSA'nın NARMA L-2 kontrol yapısı eğitilmek için gerekli olan çıkış ve denetim sinyalleri bulunmuştur. Bulunan bu veriler ile tasarlanan YSA yapısı eğitilmiş ve NARMA L-2 kontrol yapısı MATLAB ortamında oluşturulmuştur. Elde edilen simülasyon programı ile farklı referans sinyalleri için top-çubuk konum kontrolü NARMA L-2 kontrol yapısı ile başarıyla gerçekleştirilmiştir.

Research Paper

Received Date : 03/12/2018

Accepted Date : 31/12/2018

Keywords

ANN,
Ball and beam,
Narma L-2.

Abstract

In this study, NARMA L-2 control structure, which is one of Artificial Neural Networks (ANN) control algorithms, was examined and ball position control was performed in the ball and beam system. A realistic simulation of the ball and beam system was created in the MATLAB environment and ball position control was first manually checked manually and the output and control signals required to train the NARMA L-2 control structure of the ANN were found. The ANN structure designed with these data was trained and the NARMA L-2 control structure was constructed in the MATLAB environment. Based on the simulation program developed, ball and beam position for different reference signals in this study was successfully achieved with the NARMA L-2 control structure.

1. Giriş

Günümüzde kontrol uygulamalarında YSA tabanlı denetim sistemleri, karmaşık ve doğrusal olmayan sistemlerin modellerin oluşturulmasında iyi performans gösterdiklerinden dolayı etkin bir araç olarak kullanılmaya başlanmıştır [1]. Literatürdeki çalışmalarda genellikle eğitim aşamasında geri yayılım algoritmasını kullanan MLP (Multi-Layered Perceptron) tipi YSA kullanıldığı göze çarpmaktadır [1]. YSA tabanlı kontrol yapısının temeli, sistem girişi ve çıkışının o andaki ve önceki değerleri ile sistemin ileri zamandaki çıkışını tahmin edebilecek bir model oluşturması esasına dayanmaktadır [2]. Bu tahminin doğruluğu için oluşturulan YSA'nın

öğrenme yeteneği çok önemlidir. YSA'da öğrenme yeteneği ve eğitim süresi, hem nöron sayısına hem de ara katman sayısına bağlı olduğundan bu parametrelerin seçimi oldukça kritiktir.

Literatürde YSA ile model oluşturularak yapılan birçok çalışma vardır. Örneğin [3] numaralı çalışmada araştırmacılar, pendulum sistemini YSA ile ters modelleme sayesinde pendulum sistemini başarıyla kontrol etmişlerdir. Pendulum sistemi ters modelleyerek sistemin belli bir andaki çıkışı ile referans sinyali arasındaki fark sinyalinden gerekli kontrol çıkışını elde etmişlerdir. Çalışmada uygulanan ters modelleme yöntemi, MATLAB ortamında *neural network toolbox* kullanılarak gerçekleştirilmiştir.

YSA ile yapılmış model çıkarma çalışmalarından biride [4] numaralı çalışmada DC motor hız kontrol için

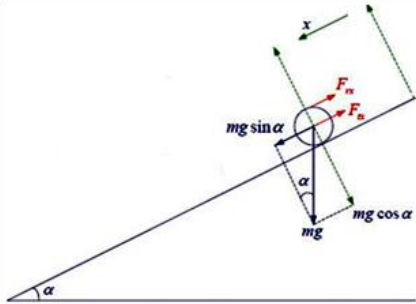
* Sorumlu Yazar (Corresponding Author): kaplan.kaplan@kocaeli.edu.tr

oluşturulan sistem modellemidir. Çalışmada kullanılan YSA; giriş katmanı 4, ara katmanı 20, çıkış katmanı ise 1 nörondan oluşmaktadır. YSA sistem çıkışı, denetim girişi ve önceki değerleri ile geri yayılım algoritması kullanılarak MATLAB ortamında eğitilmiştir. Çalışma sonunda sisteme ve oluşturulan modele aynı denetim girişi verilerek maksimum hata değeri yaklaşık 0.0002 bulunmuştur.

Literatürdeki YSA ile kontrol uygulamalarından bir başkası[5] numaralı çalışmadaki NARMA L-2 kontrol yapısı ile DC motor hız kontrol uygulamasıdır. Bu kontrol tipinin temel fikri, doğrusal olmayan sistem dinamiğini, doğrusal bir sistem dinamiği fonksiyonuna dönüştürmektir. Çalışma MATLAB ortamında NARMA L-2 *toolbox* kullanılarak gerçekleştirilmiştir. Yapılan uygulama PID kontrol ile karşılaştırılarak, NARMA L-2 denetleyici performansının PID Denetleyicisi üzerinde yükselme süresi, maksimum aşma ve sabit durum hatası kriterlerine göre daha iyi olduğu gözlenmiştir.

Bu çalışma, [5] numaralı çalışmadan esinlenerek yapılmıştır. Fakat bizim çalışmamızda 5 numaralı çalışmadan farklı olarak daha karmaşık bir sistem olan top çubuk sistemi üzerinde NARMA L-2 kontrol yapısı uygulanmıştır. Ayrıca NARMA L-2 kontrol yapısı MATLAB/GUI ortamında geliştirilmiştir.

2. Top Çubuk Matematiksel Modeli



Şekil 1. Top çubuk fiziksel diyagramı

Lisans düzeyinde birçok üniversite laboratuvarında bulunan top-çubuk deney düzeneği modelleme açısından oldukça ilginç ve nonlineer bir yapıya sahip bir problemdir. Şekil 1’de top- çubuk probleminin, çubuğun bir ucundan kontrol edildiği model gösterilmektedir. Bu modelde çubuğun yatayda yaptığı alfa (α) açısıyla topun çubuk üzerindeki konumu (x) kontrol edilmek istenmektedir.

Matematiksel modelleme için öncelikle topa etki eden kuvvetler incelenmelidir. Yer çekimi nedeniyle kütle çekim kuvveti ve dönme tarafından üretilen tork nedeniyle dönme kuvveti bulunmaktadır. Yerçekimi nedeniyle x yönünde oluşan kuvvet Denklem (1)’de gösterilmiştir.

Tablo 1. Sisteme ait parametreler

| Sembol | Açıklama-Birim |
|----------|--|
| J | Top Atalet Momenti [kgcm^2] |
| R | Top Yarıçapı [cm] |
| m | Top kütlesi [kg] |
| g | Yerçekimi sabiti [m/s^2] |
| F_{tx} | Kütle çekim kuvveti [N] |
| F_{rx} | Dönme kuvveti [N] |
| x | Topun konumu [cm] |
| α | Çubuk Açısı [rad] |

$$F_{tx}=mgsina \quad (1)$$

Topun dönme hareketi tarafından üretilen tork hem Denklem (2)’de gösterildiği gibi topun yarıçapı ve dönme kuvveti çarpımına hem de Denklem (3)’de gösterildiği gibi topun atalet momentiyile açısal ivmesi çarpımına eşittir. Topun açısal ivmesi ise topun konumunun ikinci dereceden türevi ile Denklem (4)’deki gibi elde edilir.

$$T_r=F_{rx} R \quad (2)$$

$$T_r=Ja \quad (3)$$

$$a=\frac{\ddot{x}}{R} \quad (4)$$

Yarıçapı R olan katı küre için topun atalet momenti Denklem (5) ile gösterilmiştir. Denklem (5) ve yukarıdaki denklemler kullanılarak topun dönme kuvveti Denklem (6)’da gösterildiği gibi elde edilir.

$$J=\frac{2}{5} mR^2 \quad (5)$$

$$F_{rx}=\frac{2}{5} m\ddot{x} \quad (6)$$

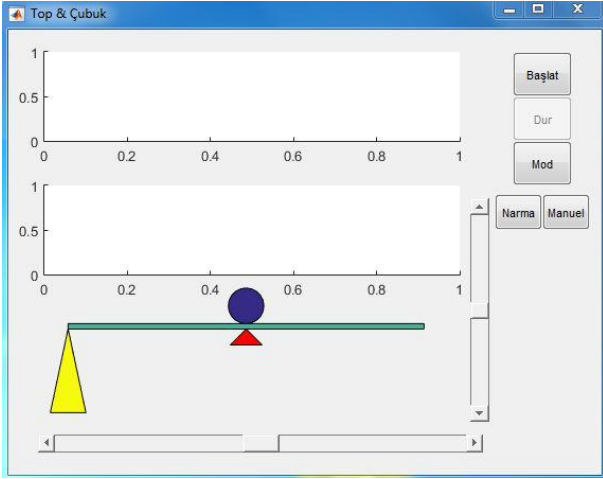
Yuvarlanan cisme etki eden net kuvvet cismin ivmesi ve kütlesi çarpımına eşittir.

$$m\ddot{x}=F_{tx}-F_{rx}=mgsina-\frac{2}{5} m\ddot{x} \quad (7)$$

Denklem (7) kullanılarak çubuk açısı ile topun konumu arasındaki ilişkiyi veren transfer fonksiyonu aşağıdaki gibi bulunmuştur. Çubuk açısı ile topun konumu arasındaki ilişkiyi veren transfer fonksiyonu Denklem 8’deki gibi bulunmuştur.

$$\frac{x(s)}{a(s)}=\frac{5g}{7s^2} \quad (8)$$

3. Top Çubuk Simülasyon Programı



Şekil 2. MATLAB/Gui Top Çubuk Simülasyonu

Çalışmada NARMA L-2 kontrol uygulaması gerçekçi bir sisteme uygulanmak istendiğinden, bir top çubuk simülasyon programına ihtiyaç duyulmuştur. Bu neden ile MATLAB'in sitesinde bulunan Ball and Beam adlı demo [6] kullanılmış ve hazırda bulunan çeşitli kontrol algoritmaları çıkartılarak bu çalışmada kullanılmak istenilen NARMA L-2 kontrol yapısı için simülasyon üzerinde çalışmalarda bulunulmuştur. MATLAB/GUI Şekil 2'deki gibi son halini almıştır.

Simülasyonda çubuğun uzunluğu 20 cm topun yarıçapının 1 cm olarak alınmıştır. Ayrıca topun konumu 0 ile 20 cm arasından çıktığı anda top düştü olarak kabul edilmiştir. Simülasyonda topun konumu değiştiren açı manüel ya da NARMA L-2 kontrol ile otomatik olarak 0.25 ile -0.25 rad değerleri arası değişmektedir. Zaman değişkeni (t) ise program başlat butonuna basıldıktan sonra dur butonuna kadar her döngüde 0.5 adım aralığıyla artan bir değişken olarak seçilmiştir. Bu zaman değişkeni ile topun birim zaman değişkeninde aldığı yoldan hız bilgisine ulaşılmıştır.

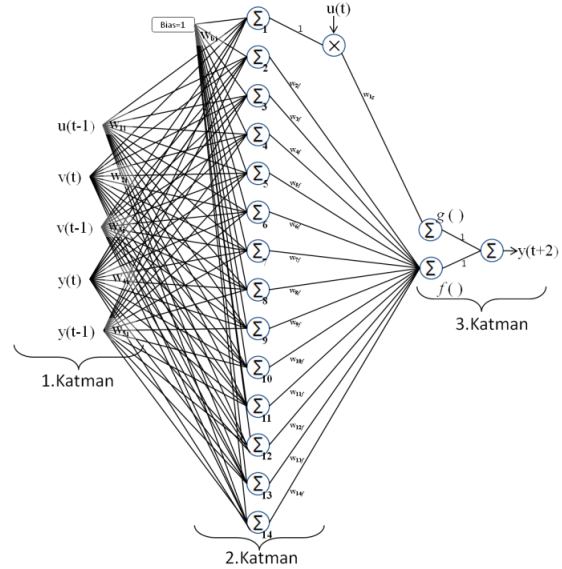
4. NARMA L2 Kontrol

Bu tip kontrolör yapısında temel amaç, doğrusal olmayan dinamik sistemi, doğrusal dinamik sisteme dönüştürmektir. NARMA L-2 kontroldeki ilk basamak kontrol edilecek olan sistemin YSA modelinin oluşturulmasıdır. İkinci basamak ise YSA eğitim aşamasıdır.

4.1. Sistem Modeli Oluşturma

NARMA L-2 kontrolör algoritmasının en önemli adımlarından birisi, sistem tanımlama aşamasıdır. Sistem tanımlamada ise YSA eğitiminin iyi yapılmış olması

oldukça önemlidir. Bu nedenle, sunulan çalışmada YSA eğitiminde sıklıkla kullanılan ve başarılı sonuçlar veren geri yayılım algoritması tercih edilmiştir [7].



Şekil 3. Sistem için tasarlanan YSA modeli

Genel ayrık zamanlı doğrusal olmayan sistemleri temsil eden bir standart model Denklem (9) ile ifade edilen NARMA modelidir [1].

$$y(t+d)=f[y(t),y(t-1),\dots,y(t-n+1),u(t-1),\dots,u(t+1)] +g[y(t),y(t-1),\dots,y(t-n+1),u(t-1),\dots,u(t-m+1)]*u(t) \quad (9)$$

Burada $u(t)$ denetim girişini, $y(t)$ ise sistem çıkışını $y(t-1),\dots,y(t-n+1)$ ve $u(t-1),\dots,u(t-m+1)$ önceki çıkışları ve denetim girişlerini göstermektedir. d ise sistem çıkışının tahmin edildiği zaman uzayıdır. NARMA modeli Denklem (9)'da verildiği gibi $f(t)$ ve $g(t)$ fonksiyonlarından oluşmaktadır. Bu çalışmada tasarlanan, NARMA modeli oluşturmak için eğitilen YSA Şekil 3'de gösterilmiştir. Çalışmada genel NARMA modelinden farklı olarak sınır ağına topun zaman içindeki hız bilgisi (v) de verilmiştir.

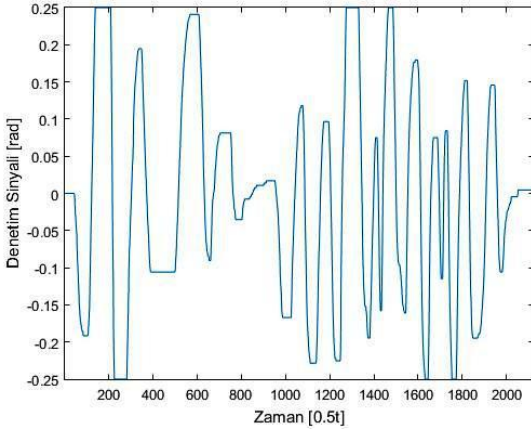
YSA 1.katman (giriş katmanı), 2.katman (ara katman) ve 3.katman (çıkış katmanı) olmak üzere üç katmandan oluşmaktadır. Giriş katmanı $u(t)$ ve $u(t)$ 'nin $u(t-1),y(t)$ ve $y(t)$ 'nin $y(t-1),v(t)$ ve $v(t)$ 'nin $v(t-1)$ zaman gecikmeli sistem çıkışı, denetim girişi ve hız değerlerini ara katmana ulaştırdığından 5 nörondan oluşmaktadır. Ara katman 14, çıkış katmanı ise 3 nörondan oluşmaktadır. YSA aktivasyon fonksiyonu lineer seçilmiş ve nöron sadece toplam fonksiyonlarından oluşmaktadır. Bu YSA ile oluşturulan NARMA modelde $g(t)$ fonksiyonu ara katmandaki 1.nörondan gelen değer ile o anki sistem giriş sinyali $u(t)$ 'nin çarpımından, $f(t)$ fonksiyonu ise ara katmandaki 2. ve 14'e kadar olan nöronlardan gelen

değerlerin toplamından oluşmaktadır. Sonuç olarak oluşturulan NARMA modeli Denklem (10) ile ifade edilebilir.

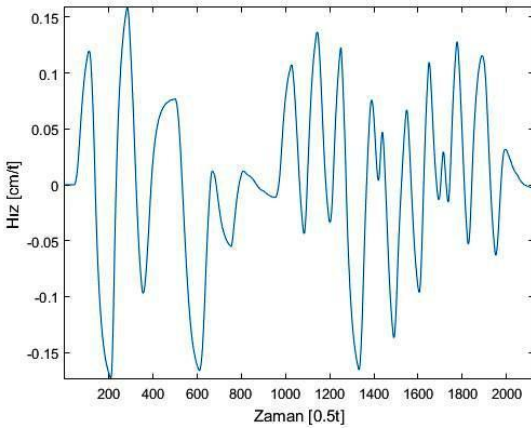
$$y(t+2) = f[y(t), y(t-1), v(t), v(t-1), u(t-1)] + g[y(t), y(t-1), v(t), v(t-1), u(t-1)] \cdot u(t) \quad (10)$$

4.2. YSA Eğitimi

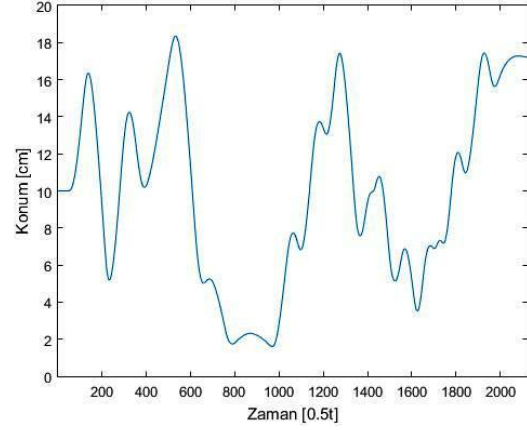
YSA'nın eğitim aşaması iki aşamadan oluşmaktadır. İlk aşama YSA'nın sistem modeli çıkartması için gerek olan eğitim verilerinin toplanması işlemidir. Bu veriler Şekil 2'de gösterilen simülasyon programında çubuk manuel kontrol edilerek bulunmuştur. Çubuk yatayda bulunan slider ile kullanıcı tarafından kontrol edilerek topun konumu rastgele şekilde değiştirilmiştir. Topun zaman içindeki konum, hız ve çubuk açı değerleri eğitimde kullanılmak üzere kaydedilmiştir.



Şekil 4. Denetim sinyali-zaman grafiği



Şekil 5. Hız-zaman grafiği



Şekil 6. Konum-zaman grafiği

Şekil 2'de gösterilen simülasyon programı kullanılarak YSA uygulaması için Şekil 4, Şekil 5 ve Şekil 6'da verilen açı, hız ve konum bilgileri elde edilmiştir. Simülasyon programı arkasında çalışan kod verileri kaydedilerek zaman değişkeni 0.5 kod adım aralığında arttığından her bir veriden 4000 tane olmak üzere eğitim seti elde edilmiştir. Bu veriler simülasyon programında topun çubuğun ortasında tutulmaya çalışılarak elde edildiğinden bu veriler ile eğitim yapılması uygun görülmüştür. Bu set geri yayılım algoritması kullanılarak YSA modeli eğitilmiştir. Eğitim YSA'nın düğüm ağırlık değerlerine başlangıçta rastgele değerler verilerek başlanır. Her zaman sabiti için YSA 1.katmanına, giriş değerleri verilerek çıkış kümesi oluştururlar. Her zaman sabitinin iki adım ötesindeki konum değeri de istenen çıkış kümesini oluşturur. Oluşturulan çıkış kümesi ile istenen çıkış kümesi karşılaştırılarak hata değerleri bulunur. Bu hata değeri belirlenen hata kriterine eşit olana kadar ya da belirlenen sabit bir iterasyon sayısı kadar döngüde tutularak ağırlık değerleri istenen çıkış kümesine göre ayarlanır. Bu çalışmada 100 iterasyon sayısı uygun görülmüştür. Algoritma, belirlenen iterasyon sayısına kadar uygulanır; ağırlıklar güncellenir ve istenen çıkış ile oluşan çıkış değerlerindeki toplam hata azaltılır. Geri yayılım algoritması aşağıdaki gibi 7 adımdan oluşmaktadır [7].

Tablo 2: Geri yayılım algoritmasında kullanılan değişkenler

| Sembol | Açıklama |
|----------|---|
| i | Yapay sinir ağlarının girişi ($i=1, 2, \dots, 5$) |
| $y(t+2)$ | Yapay sinir ağlarının çıkışındaki istenilen değeri |
| y' | Yapay sinir ağ çıkışı |
| q_h | Ara katmandaki nöronlar ile bias ağırlık değerleri |
| w_{ih} | Giriş katmanı ile ara katman arasındaki ağırlık değerleri |
| w_{ch} | Çıkış katmanı ile ara katman arasındaki ağırlık değerleri |
| t | Sistemin incelediği zaman ($t=0, 0.5, 1, 1.5, \dots, 2000$) |

Adım1: Öğrenme katsayısı (η) belirlenir. Büyük veri setlerinde η küçük seçilmesi gerektiğinden 0.0001 öğrenme katsayısı seçilmiştir [1].

Adım2: Tüm düğüm ağırlıklarına rastgele sıfır ile 1 arasında küçük değerler atanır.

Adım3: MATLAB’de çıkarılan her $u(t)$, $u(t-1)$, $u(t-2)$, $y(t)$, $y(t-1)$ ve $y(t-2)$ verileri YSA’ya uygulanır ve YSA çıkışı Denklem 13 ile bulunur.

$$g(t) = u(t-1) * w_{11} + v(t) * w_{21} + v(t-1) * w_{31} \dots \dots \dots + y(t) * w_{41} + y(t-1) * w_{51} \quad (11)$$

$$f(t) = \sum_{h=2}^{14} (u(t-1) * w_{1h} + v(t) * w_{2h} + v(t-1) * w_{3h} \dots \dots \dots + y(t) * w_{4h} + y(t-1) * w_{5h}) \quad (12)$$

$$c = f(t) + g(t) * u(t) \quad (13)$$

Adım4: Aşağıdaki gibi çıkış birimi için hata terimini (δ_c) hesaplanır.

$$\delta_c = y(t+2) - y' \quad (14)$$

Adım5: Denklem (15) ile ara katmandaki her nöron için hata terimi hesaplanır. (δ_h)

$$\delta_h = \sum_{h=1}^{14} w_{ch} * \delta_c \quad (15)$$

Adım6: Her ağ bağlantısı için ağırlık değerleri güncellenir.

$$w_{1h} = w_{1h} + \eta * \delta_h * u(t-1) \quad (16)$$

$$w_{2h} = w_{2h} + \eta * \delta_h * u(t-2) \quad (17)$$

$$w_{3h} = w_{3h} + \eta * \delta_h * u(t-2) \quad (18)$$

$$w_{4h} = w_{4h} + \eta * \delta_h * y(t-1) \quad (19)$$

$$w_{5h} = w_{5h} + \eta * \delta_h * y(t-2) \quad (20)$$

$$w_{c1} = w_{c1} + \eta * \delta_h * g(t) * u(t) \quad (21)$$

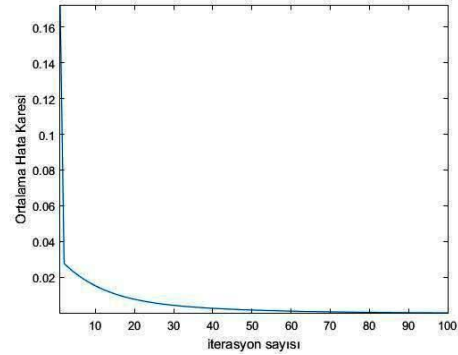
$$w_{ch} = w_{ch} + \eta * \delta_h * f(h) \quad (22)$$

$$q_h = q_h + \eta * \delta_h \quad (23)$$

Adım7: İterasyon sayısı kontrol edilir. Eğer iterasyon sayısı 100 değerine eşit ise eğitim tamamlanır. Değil ise Adım 3’e dönülür ve iterasyon sayısı 100koşulunu sağlayana kadar bu döngüler tekrarlanır.

YSA eğitiminde eğitimin doğruluğunu test etmek için her bir iterasyon da eğitimin ortalama hata karesi (OHK) Denklem (24) ile hesaplanmıştır.

$$OHK = \frac{1}{N} \sqrt{\sum_{t=0}^N (y(t+2) - cks)^2} \quad (24)$$



Şekil 7: Ortalama Hata karesi grafiği

Elde edilen eğitim verileri kullanılarak eğitilen YSA’nın her bir iterasyon da hesaplan OHK Şekil 7’de gösterilmiştir. Sonuç olarak, OHK 100. iterasyonda kabul edilebilir bir eğitim hatası olan 0.00015564 hata değeri bulunmuştur.

4.3. NARMA L-2 Kontrolün Uygulanması

Sistem tanımlamak için kullanılan NARMA modeli Denklem 21’deki gibi yazılabilir.

$$u(t) = \{y(t+2) - f[y(t), y(t-1), v(t), v(t-1), u(t-1)]\} / g[y(t), y(t-1), v(t), v(t-1), u(t-1)] \quad (25)$$

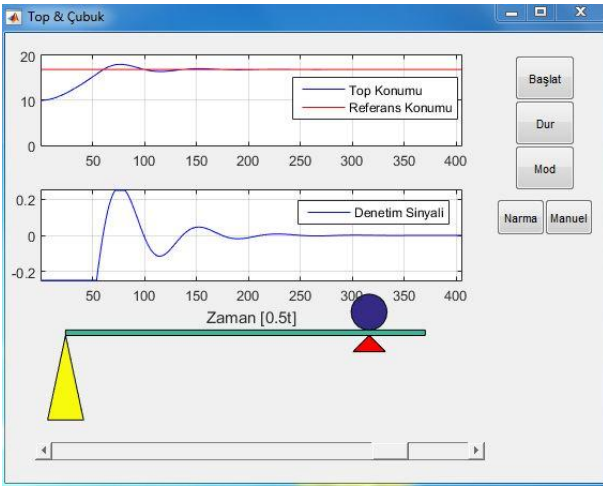
Böylelikle $y(t+2)$ sistem çıkışını referans sinyali $R(t+2)$ ’i tam olarak takip ettirmek mümkündür. $f(t)$ ve $g(t)$ bilinen fonksiyonlar olduğu için sistemin o andaki ve

önceki çıkış ve denetim girişi değerleri YSA'ya giriş olarak verilir ve denetim sinyali Denklem (26) ile elde edilir.

$$u(t) = \{R(t+2) - f[y(t), y(t-1), v(t), v(t-1), u(t-1)]\} / g[y(t), y(t-1), v(t), v(t-1), u(t-1)] \quad (26)$$

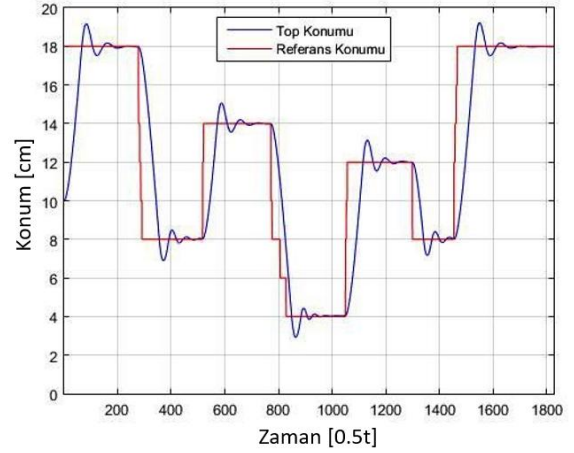
Denklem (26)'de verildiği gibi referans sinyali ile $f(t)$ fonksiyonu farkı $g(t)$ fonksiyonu çıkışına bölünerek denetim sinyali elde edilmiştir. Bu kontrol sinyali ile birlikte denetim sinyalinin sistem girişine verilerek elde edilen sistem çıkışı da YSA'ya tekrar giriş olarak verilir.

5. Simülasyon Sonuçları



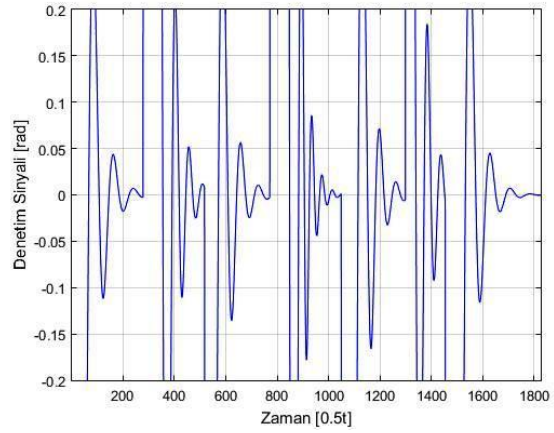
Şekil 8: MATLAB/GUI Top Çubuk Simülasyonu

Bu bölümde oluşturulan NARMA L-2 kontrol sisteminin farklı referans sinyallerindeki sonuçları gözlenmiştir. Şekil 8'de ise referans noktası belirlenmiş ve NARMA L-2 kontrol algoritması uygulanmış simülasyonun son hali gözükmetedir. Topun başlangıç konumu çubuğun ortasından olduğundan Şekil 8'deki top konum grafiğinde topun konumu 10 cm'den başlamış ve slider ile belirlenen kırmızı referans noktasına başarıyla ulaşılmış ve kalıcı durum hatasına (ess) rastlanılmamıştır. Şekil 8'deki denetim sinyali grafiği de beklenen gibi çıkmış; çubuk açısı 0.2 ile -0.2 radyan arası salınıp yapıp bu salınımın genliği zamanla azalmış ve top istenen referans konumuna ulaştığında denetim sinyali sıfıra oturmuştur. Programda kullanılan zaman değişkeni her bir kod döngüsünde 0.5'lik adımlarla artan bir değişken olduğundan sistem yaklaşık 200. program adımında istenen referans sinyaline ulaştığı gözlemlenmiştir.



Şekil 9: Referans ve Top Konum Grafiği

Şekil 9'da ise referans konumu zamanla değiştirilmiş ve topun konumu nerede olursa olsun referans konumunu takip ettiği gözlemlenmiştir.



Şekil 10: Denetim Sinyali Grafiği

Şekil 10'da ise Şekil 9'da verilen grafiğin denetim sinyali gösterilmiştir. Şekil 10'da görüldüğü gibi topun konumu veya referans konumu nerede olursa olsun denetim sinyali yüksek genlikli bir salınım ile başlayıp top istenen referans konumuna ulaşıncaya salınım şiddeti sıfırlanmaktadır. Referans konumu değiştirilince salınım tekrar başlamakta ve sonra tekrar sıfır konumuna erişmektedir.

6. Sonuçlar ve Tartışma

Bu çalışmada, NARMA L-2 kontrol modeli kullanılarak top çubuk mekanizmasının kontrolü farklı referans sinyalleri verilerek test edilmiştir. Sistem çalışma halindeyken referans noktası değiştirilerek topun konumunun referans noktasına ulaşması sağlanmıştır. Kontrol modeli basit olarak bir NARMA L-2 kontrol

yapısına YSA modelinin entegre edilmesi ile geliştirilmiştir. YSA modeli sayesinde hiçbir kontrol parametre ayarı yapmadan kontrol algoritmasının sistemi kendi başına kontrol edebileceği görülmüştür. Sonuç olarak özgün bir kontrol algoritması geliştirilerek top-çubuk mekanizması başarılı bir şekilde kontrol edilmiştir. İleriki çalışmalarda bu çalışmanın farklı kontrol sistemleri üzerinde denenerek etkinliğinin tespit edilmesine yönelik çalışmalar gerçekleştirilecektir.

Kaynaklar

- [1] Elmas Ç., 2016. Yapay Zekâ Uygulamaları, Seçkin Yayıncılık, İstanbul.
- [2] Edited by Suzuki K., 2011. Artificial Neural Networks Industrial and Control Engineering Applications, India March,
- [3] Sharma S., Kumar V., Kumar R., 2012. Supervised Online Adaptive Control of Inverted Pendulum System Using ADALINE Artificial Neural Network with Varying System Parameters and External Disturbance. I.J. Intelligent Systems and Applications, **8**, 53-61.
- [4] Mute L., Chaudhari K., Khamari R., Singare A., 2015. System Identification Using Neural Network Model for Speed Control of DC Motor Dinesh. International Research Journal of Engineering and Technology, **2**(3).
- [5] Valluru S. K., Singh M., Kumar N., 2008. Implementation of NARMA-L2 Neuro controller for Speed Regulation of Series Connected DC Motor. Proceedings of the International Conference on Computer and Communication Engineering.
- [6] <https://www.mathworks.com/matlabcentral/fileexchange/151-ball---beam-demo>.
- [7] Cilimkovic M. Neural Networks and Back Propagation Algorithm Institute of Technology Blanchardstown Road. North Dublin 15 Ireland.