

## **BULANIK KÜMELEMELİ YAPAY SİNİR AĞLARI İLE HAREKETLİ ROBOTLARDA YÖRÜNGE HESABI**

**Serkan AYDIN<sup>1\*</sup>**

<sup>1</sup> Celal Bayar Üniversitesi, TMYO, Endüstriyel Elektronik Programı, Turgutlu – Manisa, TÜRKİYE

**Özet:** Hareketli robotlarda, çevreden gelen kısıtlamalar ve robotun fiziksel kısıtlamaları dikkate alınarak optimal yörünge planlamaları yapılabilmektedir. Genellikle global optimum çözümlere hesaplama sürelerinin çok uzun olduğu nümerik metotlarla ulaşılabilmektedir. Uzun hesaplama sürelerini kısaltmak amacıyla da esnek hesaplama teknikleri ön plana çıkmaktadır. Bu çalışmada önceden görünebilirlik çizgesi (VÇM) metoduyla planlanmış yol üzerinde zaman-optimal yörüngeler bulunmuştur. Yörüngeler, düz kısımlardan ve eğri kısımlardan oluşturulmuşlardır. Eğri kısımları oluşturmak için sadece dönüş açısı ve eğri kısım öteleme hız parametrelerinin bilinmesi yeterli olmaktadır. Olası tüm eğri kısımlar kümesi içerisinde, engellerle çakışmayan zaman-optimal amaç ölçütünü sağlayan yörünge seçilmesi esnek hesaplama metodu diferansiyel evrim ile gerçekleştirilebilmektedir. Eğri kısımların oluşturulmasında yapay sinir ağları (YSA) ve bulanık kümelemeli (BS) YSA kullanılmıştır. BS-YSA modellerinin daha başarılı olduğu görülmüştür.

**Anahtar Kelimeler:** *Hareketli robot, optimizasyon, yörünge, yapay sinir ağları, bulanık kümeleme.*

## **A FUZZY CLUSTERING NEURAL NETWORKS TRAJECTORY PLANNING FOR MOBILE ROBOTS**

**Abstract:** Optimal trajectories of mobile robots can be planned under the consideration of the constraints from environment and physical constraints of the mobile robot. In general, global optimum solutions are obtained by using time consuming numerical methods. Artificial intelligence methods are used to obtain shorter calculation time. In this study, a method is proposed to find the time optimal trajectory on predefined path, which is found by using visibility graph method. The planned trajectories are composed of line segments and curves. The structure of the curves is determined by only two parameters (turn angle and translational velocity in the curve). A curves set is formed by all possible curves. The time optimal trajectories curves are found by using differential evolution method. Curves are calculated by using artificial neural networks (ANN) and Fuzzy Clustered (FCM) ANN models. It is seen that, FCM-ANN models are better than the classic ANN models.

**Keywords:** *Mobile robot, optimisation, trajectory, neural networks, FCM algorithm.*

---

**\*Sorumlu Yazar**

serkan.aydin@bayar.edu.tr

## 1. GİRİŞ

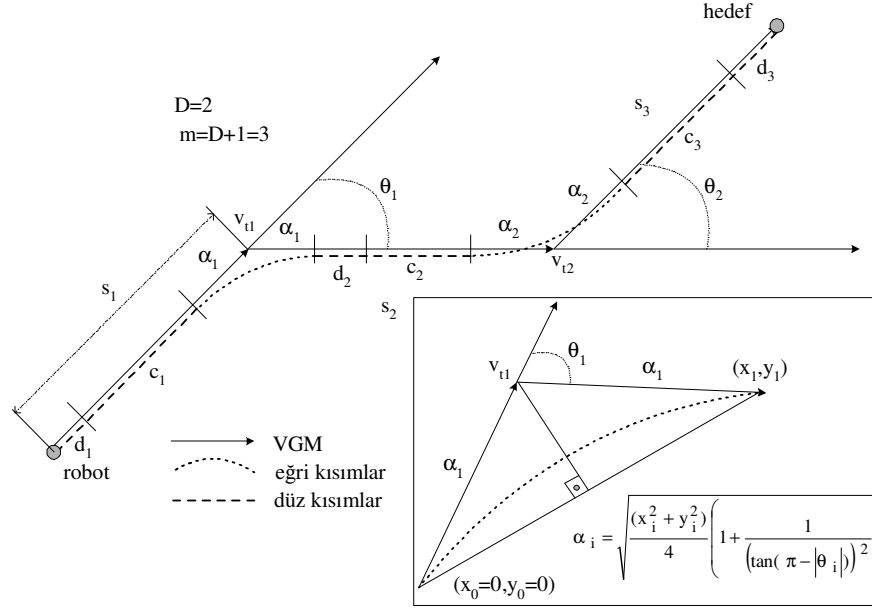
Hareketli robotlarda yörünge planlama üzerine son 20 yıldır oldukça yaygın bir şekilde çalışılmaktadır. Literatürdeki bu çalışmaları: 1) Kapalı planlamalar (gerçek zamanda, lokal) ve 2) Açık planlamalar (off-line, global) şeklinde kümelendirilebilir [1]. Planlama, hareketli robot hareket halindeyken yapılıyorsa buna kapalı planlama denmektedir. Bu tür planlamalarda amaç ölçütümüz sadece hedefe varmaktan ibarettir. Diğer bir deyişle, kapalı planlamalarda, planlanan yörünge istenilen özellikleri sağlama (minimum mesafe, minimum zaman vb.) her zaman mümkün değildir. Bu şartlar ancak açık planlamalarla yerine getirilebilir. Genellikle açık-planlamalarda yörünge: *yol planlama*, planlanan yolu *düzgünleştirme* ve *düzgünleştirilmiş yolu izleme* olmak üzere 3 aşamalı bir hiyerarşiye sahiptirler. Açık planlamaları da, açık-ayrık ve açık-sürekli olmak üzere iki kısma ayırmak mümkündür. Açık-ayrık planlamalarda bu 3 aşama sırasıyla gerçekleşmektedir. Önce ayrıklaştırılmış uzayda (hücrelere ayırma, görünebilirlik çizgesi metodu vb.) yol planlanır, daha sonra çeşitli şekillerde planlanan yol düzleştirilir. 2. aşamada literatürde, B-spline eğrileri, kübik spiraller, polinomal eğriler ve klotheidler [2, 3, 4, 5] gibi değişik eğri aileleri kullanılmıştır. Son aşamada da bu düzleştirilmiş yolu izleyecek bir denetçiye ihtiyaç duyulmaktadır. Bu çalışmalarda genellikle minimum mesafe olarak alınan amaç ölçütü 1.aşamada kullanıldığından dolayı sağlanamayabilmektedir. Diğer bir deyişle global bir amaç ölçütü olmamaktadır. Açık sürekli planlamalarda ise yörünge planlama 2 aşamada tamamlanabilmektedir. İlk aşamada çalışma uzayı ayrıklaştırılmadan planlama yapılıp ve sonraki aşamada da bu planlanan yörünge bir denetçi ile izlenir. Bu kısımdaki çalışmalarda da genellikle varyasyonel

metotlar kullanılmıştır. Lagrange çarpanları metodu ile oluşturulan denklem sistemi nümerik metotlarla çözülmüştür. Bu çalışmada kullanılan robot Nomad 200 e benzer özellikteki Rhino ismi verilen robotun eşitlikleri [14] yayınında çıkarılmıştır. Bu eşitlikler üzerine yazarlar Dinamik Pencere Yaklaşımı dedikleri bir yaklaşımla çözüme gitmeye çalışmışlardır.

Bir diğer çalışmada ise [6] yörünge düz kısımlar (dönme hızı  $v_s=0$ ) ve eğri kısımlardan (öteleme hızı  $v_t=sabit$ ) oluştuğu kabul edilmiştir. Düz kısımlarda öteleme hızının artıp azalması maksimum ivme ( $a_t$ ) ile gerçekleştirilmektedir. Eğri parçacıklarında dönme hızının artış ve azalışları da maksimum ivme ( $a_s$ ) ile gerçekleştirilirken öteleme hızı sabit alınmaktadır.

En son olarak *diferansiyel evrim* (DE) ile serbest konfigürasyon uzayında ( $C_{serbest}$ ), düz kısım ve eğrilerden oluşan, global zaman-optimum amaç ölçütünü sağlayan yörüngeler elde edilmiştir. Uzun olan diferansiyel evrim çözüm süresini kısaltmak amacıyla eğri kısımların hesaplanmasında deterministik model (Robotun denklemlerinin McLauren seri açılımıyla) yerine yapay sinir ağı (YSA) öğrenme yoluyla olmak üzere iki değişik şekilde gerçekleştirilmiştir [6, 7]. YSA modellerinin kullanımı ile optimizasyon süresinin 1/8'e kadar düştüğü gösterilmiştir.

YSA modellerinin daha başarılı olabilmesi için eğitimde kullanılacak setlerin oluşturulması/seçilmesi büyük önem arz etmektedir. Bu amaçla değişik çalışmalar yapılmıştır [8, 9, 10]. Bu çalışmada da, yapay sinir ağlarında kullanılacak eğitim setleri bulanık kümeleme ile seçilmiştir [8, 9]. Bu şekilde elde edilen sonuçların önceki çalışmamız ile karşılaştırması yapılmıştır. Test aşaması hata değerlerini oldukça düşüğü gözlenmiştir.



**Şekil 1.**  $s_1$ ,  $s_2$  ve  $s_3$  doğrusal yol parçacıkları;  $\alpha_1$   $\alpha_2$  dönüşleri  $v_{t1}$   $v_{t2}$  öteleme hızlarıyla yapabilmek için gereken mesafe;  $d_1$  ise 0' dan  $v_{t1}$ 'e,  $d_2$  ise  $v_{t1}$ 'den  $v_{t2}$ 'ye hız değişim mesafeleri.

## 2. MATERYAL ve METOT

### 2.1 Materyal

Materyalimiz Hareketli robot Nomad 200 Şekil 2a' da görüldüğü gibidir.

### 2.2 Metot

Hareketli robot Nomad 200 kullanılarak öteleme ivmesi  $a_t=0$  alınarak, öteleme hızı  $v_t \in [0, 1.016]$  m/sn ve  $\theta \in (0, \pi]$  parametreleri için belirtilen aralıklarda değişik değerler alınarak robotun çizdiği yörüngeler kaydedilmiştir.

#### 2.2.1 Yörünge planlama

Hareketli robotlar için önceden görünebilirlik çizgesi metodu (VÇM) ile planlanmış yörünge üzerinde çoğul lineer/lineer olmayan kısıtlar altında oluşturulacak izlenebilir yörünge planı bir optimizasyon problemi olarak ele alınabilmektedir [6]. VÇM ile üretilen yol, düz ve eğri kısımlardan oluşmaktadır (Şekil 1).

#### 2.2.2 Eğri ve düz kısım denklemleri

Tekerlek yapısının Şekil 2b' de verildiği, hareketli robot (HR) Nomad 200'ün denklemlerinde düz ve eğri kısımların elde edilişi burada incelenmiştir. Bu çalışmada kullanılan robot Nomad 200 e benzer özellikteki Rhino ismi verilen robotun eşitlikleri [14] yayınında çıkarılmıştır. Bu eşitliklerin yaklaşık çözümlerini gerçekleştirerek kendi geliştirdikleri dinamik pencere metodunda kullanmışlardır. Burada ise robotun denklemleri Şekil 1' de aktarılan yollar için uygun şekilde çözülme işleminde YSA modeli kullanılmıştır.

Robotun hareket esnasında kaymaması:

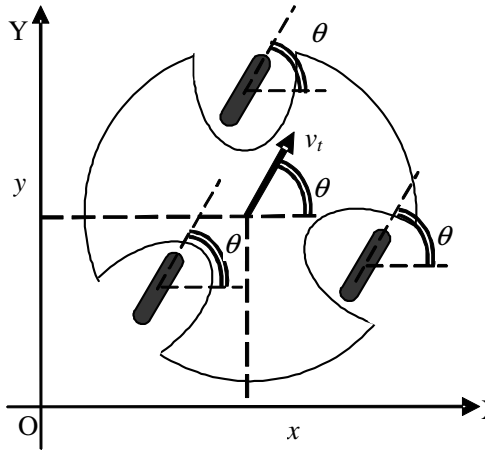
$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \quad (1)$$

nonholonomik kısıtlamasının sağlanmasını gerektirir. Yani verilen bir geometrik konfigürasyon  $(x, y, \theta)$  altında anlık öteleme hız vektörünün yönü her zaman tekerlek yönüne paraleldir.

Geometrik konfigürasyona öteleme ve dönme hızlarını ve bunların türevlerini katarak ( $x$ ,  $y$ ,  $\theta$ ,  $v_t$ ,  $v_s$ ) *dinamik konfigürasyon* olarak



a)



b)

Şekil 2. Nomad 200'ün a) resmi b) tekerlek yapısı

isimlendirilmiştir [5]. Burada  $v_t$  öteleme hızı,  $v_s$  ise dönme (steering velocity) hızıdır. Denklem 1 bu uzayda

$$\begin{aligned}\dot{x} &= v_t \cos(\theta) = v_x \\ \dot{y} &= v_t \sin(\theta) = v_y \\ \dot{\theta} &= v_s \\ \dot{v}_t &= a_t \\ \dot{v}_s &= a_s\end{aligned}\quad (2)$$

şeklinde gösterilebilir (parametrelerin üstündeki nokta zamana göre türev anlamındadır). Burada  $a_t$  ve  $a_s$  sırasıyla öteleme ve dönme ivmeleridir. Robotun dinamik özelliklerinden dolayı hız ve ivmelerin üzerine:

$$\begin{aligned}|v_t| &\leq v_{t \max} \\ |v_s| &\leq v_{s \max} \\ |a_t| &\leq a_{t \max} \\ |a_s| &\leq a_{s \max}\end{aligned}\quad (3)$$

kısıtlamaları gelmektedir. Uygulamalarda denetleme vektörü  $u=[v_t \ v_s]^T$  olarak alınmıştır. Denklem 2'nin her iki tarafının entegralini alırsak

$$\begin{aligned}x &= x_0 + \int_{t_0}^t v_t \cos(\theta) dt \\ y &= y_0 + \int_{t_0}^t v_t \sin(\theta) dt \\ \theta &= \theta_0 + \int_{t_0}^t v_s dt \\ v_t &= v_{t0} + \int_{t_0}^t a_t dt \\ v_s &= v_{s0} + \int_{t_0}^t a_s dt\end{aligned}\quad (4)$$

elde edilir. Giriş kısmındaki problem tanımımızdan dolayı Denklem 4'ün son iki eşitliğindeki  $a_t=a_{t \max}$  ve  $a_s=a_{s \max}$  olarak sabit alındığında:

$$\begin{aligned} v_t &= \left\{ v_{t0} + a_t \Big|_{t_0}^t = v_{t0} + a_t (t - t_0) \right\}_{a_t = a_{t \max}} \\ v_s &= \left\{ v_{s0} + a_s \Big|_{t_0}^t = v_{s0} + a_s (t - t_0) \right\}_{a_s = a_{s \max}} \end{aligned} \quad (5)$$

şeklini alır. Denklem 5’i Denklem 4’ün üçüncü eşitliğinde kullanırsak

$$\begin{aligned} \theta &= \theta_0 + \int_{t_0}^t \left\{ (v_{s0} + a_s (t - t_0)) \right\}_{a_s = a_{s \max}} dt \\ &= \theta_0 + \left\{ (v_{s0} - a_s t_0)(t - t_0) + \frac{a_s (t - t_0)^2}{2} \right\}_{a_s = a_{s \max}} \end{aligned} \quad (6)$$

Denklem 5–6 Denklem 4’ün ilk iki eşitliğinde yerlerine konursa

$$\begin{aligned} x &= x_0 + \\ &\int_{t_0}^t \left\{ \cos \left( \theta_0 + (v_{s0} - a_s t_0)(t - t_0) + \frac{a_s (t - t_0)^2}{2} \right) \right\}_{a_s = a_{s \max}, a_t = a_{t \max}} dt \end{aligned} \quad (7)$$

$$y = y_0 +$$

$$\int_{t_0}^t \left\{ \sin \left( \theta_0 + (v_{s0} - a_s t_0)(t - t_0) + \frac{a_s (t - t_0)^2}{2} \right) \right\}_{a_s = a_{s \max}, a_t = a_{t \max}} dt$$

denklemini elde edilir. Denklem 7’deki integrand terimlerinin içerisindeki trigonometrik ifadeler  $t^2$  li ifadeler içerdiğinden dolayı bu entegrallere *fresnel* entegralleri denir [11]. Belli bir çalışma noktası etrafında trigonometrik terimler

$$\begin{aligned} \cos(\theta) &= \sum_{n=0}^{\infty} (-1)^n \frac{\theta^{2n}}{(2n)!} = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} \dots \\ \sin(\theta) &= \sum_{n=0}^{\infty} (-1)^n \frac{\theta^{2n+1}}{(2n+1)!} = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} \dots \end{aligned} \quad (8)$$

şeklinde McLaurin serilerine açılarak yaklaşık sonuçlar elde edilebilir. Giriş

kısımındaki çözüm yaklaşımımızdan hatırlanacağı gibi, yörüngeler düz ve eğri kısımlardan oluşmaktadır. İşlemlerin daha iyi görülebilmesi için, düz kısımlar ve dönüş eğrilerinde,  $t_0=0$  alınarak bu kısımlar için Denklem 7 yeniden yapılandırılacaktır.

Düz kısımlarda  $a_s=0$  ve  $v_{s0}=0$  olduğundan dolayı

$$\begin{aligned} x &= x_0 + \int_0^t \left\{ (v_{t0} + a_t t) \cos(\theta_0) \right\}_{a_t = a_{t \max}} dt \\ &= \left\{ x_0 + \left( v_{t0} t + a_t \frac{t^2}{2} \right) \cos(\theta_0) \right\}_{a_t = a_{t \max}} \end{aligned} \quad (9)$$

$$\begin{aligned} y &= y_0 + \int_0^t \left\{ (v_{t0} + a_t t) \sin(\theta_0) \right\}_{a_t = a_{t \max}} dt \\ &= \left\{ y_0 + \left( v_{t0} t + a_t \frac{t^2}{2} \right) \sin(\theta_0) \right\}_{a_t = a_{t \max}} \end{aligned}$$

denklemini Denklem 7’den yazılabilir.

Eğri kısımlarda öteleme hızı  $v_t=v_{t0}$  olarak sabit olduğundan  $a_t=0$  alınınca:

$$\begin{aligned} x &= x_0 + \int_0^t \left\{ \cos \left( \theta_0 + v_{s0} t + \frac{a_s t^2}{2} \right) \right\}_{a_s = a_{s \max}} dt \\ y &= y_0 + \int_0^t \left\{ \sin \left( \theta_0 + v_{s0} t + \frac{a_s t^2}{2} \right) \right\}_{a_s = a_{s \max}} dt \end{aligned} \quad (10)$$

şeklini alır. Denklem 10, Denklem 8’deki gibi seriye açılarak çözülür. Eğri kısımlar için  $a_s$  anahtarlama süreleri hesaplanır.  $a_s$  anahtarlama sürelerine göre Denklem 10’daki  $\theta_0$ ,  $v_{s0}$  ve  $a_{s \max}$  yerlerine konularak  $x$  ve  $y$  değerleri elde edilmektedir [7].

### 2.2.3 Bulanık kümeleme metodu (BS)

Nesnelerin aralarındaki benzerliklere göre kümeleme ve gruplar halinde organize etmeye kümeleme analizleri denmektedir. Birbirine daha benzer nesnelere tarafından kümeler oluşturulur. Metrik uzayda benzerlik, her nesnenin kümesi temsil eden prototip nesnenin vektörüne olan uzaklığı ile tanımlanmaktadır. Prototip nesnelere genellikle başlangıçta bilinmemektedir. Kümeleme işlemi esasında oluşturulmaktadır. Bundan dolayı kümeleme metodları gözetimsiz metodlardandır. Prototipler kümedeki nesnelere aynı boyutta bir vektör olabileceği gibi lineer /lineer olmayan fonksiyonlu yüksek seviyeli geometrik nesnelere de olabilirler. Kümeler veri kümesinin bir alt kümesi olarak görülebilir. Bu alt kümelerin bulanık veya klasik olmaları durumuna göre kümeleme metodu belirlenebilmektedir. Klasik kümeleme metodları klasik küme teorisine göre yapılandırıldığından dolayı her nesne sadece tek bir alt kümenin elemanıdır. Bulanık kümeleme metodlarında ise her nesne değişik seviyelerdeki üyelikleriyle birden fazla kümeye ait olabilir. Böylece nesnelere bulanık alt kümeler ayrılmaktadır. Birçok uygulamada bulanık kümeleme klasik kümelemelere göre daha uygun olmaktadır. Bulanık kümeleme metodu birçok çalışmada kullanılmaktadır [12].

Bu çalışmada klasik çözüm metodu ile (Eşitlik 8) YSA' nın eğitme ve test aşamalarında kullanılmak üzere 23779 vektörden oluşan bir çalışma uzayı oluşturulmuştur. Her vektörün ve oluşturulan uzayın tanımlamaları:

$$z_k = [v_{ik}, \theta_k]^T, z_k \in \mathfrak{R}^2 \text{ ve}$$

$$Z = \{z_k | k = 1, 2, \dots, N\} \quad (11)$$

$$Z = \begin{bmatrix} v_1 & v_2 & \dots & v_N \\ \theta_1 & \theta_2 & \dots & \theta_N \end{bmatrix}$$

şeklinde. Burada N=23779 adet olarak alınmıştır.  $\theta = [0, 0.5^\circ, 1^\circ, 1.5^\circ, \dots, 149.5^\circ, 150^\circ]$  ve  $v_i = [0.0254, 0.0381, 0.0508, 0.0635, \dots, 1.0033, 1.016]$  m/s olarak alındığından  $\theta$

kümesinde 301 eleman ve  $v_t$  kümesinde ise 79 eleman bulunmaktadır. Test için kullanılan toplam set sayısı  $N=301 \times 79=23779$  şeklinde hesaplanmıştır.

Kümelemenin amacı Z matrisinden c adet küme oluşturmaktır. Bu çalışmada önceki çalışmamızla eşit olarak  $c=1064$  alınmıştır.  $\theta = [0, 2^\circ, 4^\circ, 6^\circ, \dots, 148^\circ, 150^\circ]$  ve  $v_i = [0.0254, 0.0762, 0.1524, 0.2286, \dots, 0.9398, 1.016]$  m/s olarak alındığından  $\theta$  kümesinde 76 eleman ve  $v_t$  kümesinde ise 14 eleman bulunmaktadır. Test için kullanılan toplam set sayısı  $c=76 \times 14=1064$  şeklinde hesaplanmıştır.  $c \times N$  boyutlu  $U = [\mu_{ik}]$  matrisi bulanık kümeleri oluşturur eğer aşağıdaki şartları sağlarsa:

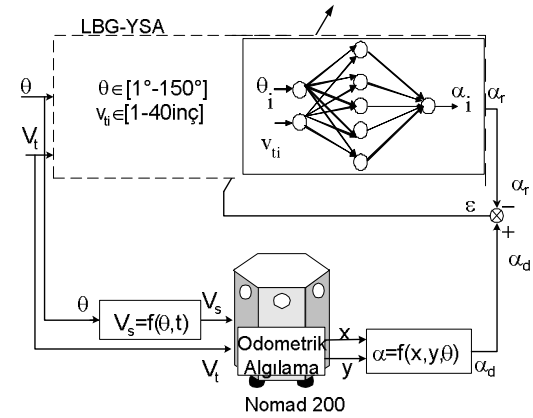
$$\mu_{ik} \in [0,1], 1 \leq i \leq c, 1 \leq k \leq N \quad (12)$$

$$\sum_{i=c}^c \mu_{ik} = 1, 1 \leq k \leq N \quad (13)$$

$$0 < \sum_{k=1}^L \mu_{ik} < N, 1 \leq i \leq c. \quad (14)$$

Burada c bulanık küme sayısı iken  $\mu_{ik}$  ise üyelik derecesini göstermektedir.

$z_k = [v_{ik}, \theta_k]^T$  nıncı vektör  $1 \leq i \leq c$  nınci



Şekil 3. YSA ile eğri kısımların modellenmesi

kümeye aittir. Bulanık kümeleme modelinin amacı [10]  $m_k$  ve küme merkezleri  $v_i$  arasındaki ağırlıklı karesel mesafelerin

toplamını minimum yapan  $v_i$  leri bulmaktır. Bundan dolayı amaç fonksiyonu

$$J(Z, U, V) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{ik}^2 \quad (15)$$

burada  $U=[\mu_{ik}]$ ,  $Z'$  nin bulanık kümeleri,  $V=[v_1, v_2, \dots, v_c]$  küme prototipleri (merkezler), ve  $m \in (1, \infty)$  kümenin bulanıklığını belirleyen üs ağırlığıdır ve genellikle  $m=2$  alınır.  $D_{ik}^2$  uygun bir norm olarak belirlenir. Mesela

$$D_{ik}^2 = \|z_k - v_i\| = \sqrt{(z_k - v_i)^T A (z_k - v_i)} \quad (16)$$

şeklindeki bir A-norm olabilir. [9] yayınında belirtildiği gibi eşitlik (15) değişik metotlarla çözülebilen lineer olmayan optimizasyon problemidir. Çözümle ilgili olarak [13] gibi değişik kaynaklara bakılabilir.

#### 2.2.4 YSA ile eğri kısımları modelleme

Evrimsel DE yöntemi ile yapılan optimizasyon işlem süresini azaltmak için yörüngedeki eğri kısımların hesaplanmasında YSA kullanılmış ve sonuçları incelenmiştir. Yörüngelerdeki eğri kısımların modellenmesinde önerilen YSA uygulaması Şekil 3' te verilmiştir Bu önerilen YSA yapısında, 2 giriş, 5 sinir hücreli 1 ve 2 gizli katman ve 1 çıkıştan (2:5:1 ve 2:5:5:1) oluşmaktadır. Bu iki giriş: dönüş açısı ( $\theta \in [1^\circ - 150^\circ]$ ) ve dönüşlerdeki öteleme hızı ( $v_{it} \in [1 - 1.016]$  m/sn ) şeklinde ifade edilir. Çıkış ise, robot doğrusal yol parçacığı üzerinde dönme

hareketinin başladığı noktadan itibaren kalan uzunluk miktarı olan  $\alpha_i$ 'dir (Şekil 1). Çok katmanlı ileri beslemeli ve hatanın geriye yayılımının kullanıldığı bir ağ mimarisi ile öğrenme işlemi gerçekleştirilmiş [6].

### 3. BULGULAR ve TARTIŞMA

Bu YSA yapılarının hata değerleri Çizelge 1'de verilmektedir. Çizelge 1'deki karşılaştırmalarda kullanılan hata değerlerini ifade etmek için:

$$E = \{\epsilon_i \mid \epsilon_i = (\alpha_{di} - \alpha_{ri}) / \alpha_{ri} \quad i \in [1, n]\} \quad (17)$$

bağlı hata kümesini tanımlayalım. Bu hata kümesindeki elemanların ortalaması :

$$E_O = \left( \sum_{i=1}^n 100|\epsilon_i| \right) / n \quad \% \quad (18)$$

$$E_M = \max(100E) \quad \% \quad (19)$$

ise hata kümesin maksimumunu vermektedir. Bağlı hatası %5 ten yüksek olan eleman sayısını  $E_5$  verir:

$$E_5 = \{\epsilon_i \mid |100\epsilon_i| > 5 \quad i \in [1, n]\} \quad (20).$$

Elde edilen sonuçlar Çizelge 1' de verilmiştir. Çizelgede, başında BS bulunan modellerde Bulanık Kümeleme algoritması kullanılarak eğitme verileri oluşturulurken diğer modellerde tüm veri uzayından lineer olarak eğitme verileri seçilmiştir. BS-YSA modellerinin hepsinde de ortalama hata değerinin ( $E_O$ ) herhangi bir kümelemenin kullanılmadığı YSA modellerine göre daha düşük olduğu gözlenmektedir.

**Çizelge 1.** 30000 iterasyon sonrasında farklı ağ yapılarının test fazı hata değerleri.

	2:5:1 Ağ Mimarisi			2:5:5:1 Ağ Mimarisi		
	E <sub>0</sub> %	E <sub>M</sub> %	E <sub>5</sub>	E <sub>0</sub> %	E <sub>M</sub> %	E <sub>5</sub>
YSA Levenberg-Marquardt	0.3517	12.145 3	98	0,0306	0,7532	0
<b>BK-YSA Levenberg-Marquardt</b>	<b>0.1346</b>	<b>7.9507</b>	<b>29</b>	<b>0.0282</b>	<b>2,5065</b>	<b>0</b>
YSA Quasi-Newton	0.9871	20.860 9	346	0.9817	21.3661	656
<b>BK-YSA Quasi-Newton</b>	<b>0.4787</b>	<b>13.985</b>	<b>154</b>	<b>0.1540</b>	<b>7.6604</b>	<b>31</b>
YSA Resilient	1.9	31.1	1116	1.6	26.1	2062
<b>BK-YSA Resilient</b>	<b>1.9</b>	<b>32.4</b>	<b>1505</b>	<b>0.8500</b>	<b>22.0227</b>	<b>412</b>
YSA Ölçekli eşlenik gradiyent (Scaled conjugate gradient)	1.0246	21.685 4	606	0.3591	11.3151	90
<b>BK-YSA Ölçekli eşlenik gradiyent (Scaled conjugate gradient)</b>	<b>0.9670</b>	<b>22.996</b> 7	<b>419</b>	<b>0,2245</b>	<b>6,9177</b>	<b>58</b>

#### 4. SONUÇLAR

Bu yayında, hareketli robot Nomad 200 için görünebilirlik çizgesi metodu ile planlanan yol üzerinde zaman-optimal yörünge için robotun denklemlerinin McLauren serisine açılmasıyla yapılması gereken hesaplamalar için değişik eğitime yöntemlerine sahip BK kümelemeli YSA modeli kullanılmıştır. Bu çalışmada BK-YSA modellerinde ortalama hata değerlerinin (E<sub>0</sub>) normal YSA modellerine göre daha düşük olduğu gözlenmektedir. 2:5:5:1 LBG-YSA Levenberg-Marquardt mimarisinde ise ortalama hata en düşük çıkmıştır.

#### Kaynaklar

- [1] Zefran M., “Continuous Methods for Motion Planning”, PhD Thesis, University of Pennsylvania, Philadelphia, 1-13, (1996).
- [2] Zhang J., and Knoll A., “An Enhanced Optimization Approach for Generating Smooth Robot Trajectories in the Presence of Obstacles”, Proc. Of the European Chinese Automation Conference, London, 263-268, (1995).
- [3] Weber H., “A motion planning and execution system for mobile robots driven by stepping motors”, Robotics and Autonomous Systems, 33: 207–221, (2000).
- [4] Reuter J., “Mobile robots trajectories with continuously differentiable curvature: An optimal control approach”, IEEE/RSJ Proc. Of Int. Conf on Intelligent Rob. and Sys., Victoria B.C., Canada, 38-43, (1998).
- [5] Scheuer A., Xie M., “Continuous-Curvature Trajectory Planning for Manoeuvrable non-Holonomic Robots”, IEEE/RSJ Proc. Of Int. Conf on Intelligent Robots and System, Kyongju, 3: 1675-1680, (1999).
- [6] Aydın S., “Mobil Robotlarda Evrimsel Metotlar ile Hareket Planlama”, Doktora Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 43-56, (2003).
- [7] Aydın S., Temeltaş H., “Planning Optimal Trajectories for Mobile Robots Using an Evolutionary Method with Fuzzy Components”, Advances in Natural Computation”, Eds by L. Wang, K. Chen and Y.S. Ong, Springer, LNCS 3612, 703-712, (2005).
- [8] Bezdek J. C., Coray C., Gunderson R., and Watson J. “Detection and Characterization of Cluster Substructure” SIAM Journal of Applied Mathematics, 40(2): 339-372, (1981).



- [9] Babuska R., “Fuzzy Modeling and Identification”, Ph.D. Thesis, Delft University of Technology, Delft, 16-23, (1997).
- [10] Bezdek J. C., “Pattern Recognition with Fuzzy Objective Function Algorithms”, Plenum Press, New York, (1981).
- [11] Kreyszig E., “Advanced Engineering Mathematics”, John Wiley & Sons, inc., Singapore, (1993).
- [12] Bulut M., ve İstanbullu A., “Bulanık C-Ortalama (Fcm) Algoritmasına Dayalı Yeni Görüntü Bölütleme Sisteminin Geliştirilmesi” Teknoloji, 7(3): 361-367, (2004).
- [13] Barkó G., Abonyi J., and Hlavay J., “Application of Fuzzy Clustering and Piezoelectric Chemical Sensor Array for Investigation on Organic Compounds”, Analytica Chimica Acta, 398 (2-3), 219-22, (1999).
- [14] Fox D., Burgard W., Thrun S., “The Dynamic Window Approach To Collision Avoidance” IEEE Robotics & Automation Magazine, 4 (1), 23-33, (1997).

### **Semboller Ve Kısaltmalar Listesi**

<b><math>a_s</math></b>	: dönme ivmesi
<b><math>v_s</math></b>	: dönme hızı
<b><math>a_t</math></b>	: öteleme ivmesi
<b><math>v_t</math></b>	: öteleme hızı
<b><math>\theta</math></b>	: dönüş açısı
<b><math>\alpha</math></b>	: dönüş mesafesi
<b><math>u</math></b>	: denetleme vektörü
<b><math>C_{serbest}</math></b>	: serbest konfigürasyon uzayı
<b><math>\epsilon</math></b>	: hata
<b><math>E</math></b>	: bağıl hata
<b><math>E_0</math></b>	: ortalama bağıl hata
<b><math>E_M</math></b>	: maksimum hata
<b><math>E_5</math></b>	: % 5' ten büyük hata
<b>VÇM</b>	: görünebilirlik çizgesi metodu
<b>YSA</b>	: yapay sinir ağları
<b>BK</b>	: Bulanık Kümeleme
<b>HR</b>	: Hareketli robot
<b>DE</b>	: diferansiyel evrim

*Geliş Tarihi: 29/02/2008*

*Kabul Tarihi: 22/07/2008*

