Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 318-324, Nisan 2020

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 318-324, April 2020

Research Article

Yapay Arı Koloni Algoritması ile Eğitilmiş Tekrarlayıcı Sinir Ağlarının Robot Navigasyonu İçin Kullanılması**

Ebru Yönem[1*], Rüştü Akay[2]

[1]Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Anabilim Dalı, Kayseri, Türkiye (ORCID: 0000-0003-3374-0593)
[2]Erciyes Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü, Kayseri, Türkiye (ORCID: 0000-0002-3585-3332)

(Konferans Tarihi: 5-7 Mart 2020)

(**DOI:**10.31590/ejosat.araconf41)

ATIF/REFERENCE: Yönem, E. & Akay, R. (2020). Yapay Arı Koloni Algoritması ile Eğitilmiş Tekrarlayıcı Sinir Ağlarının Robot Navigasyonu İçin Kullanılması. *Avrupa Bilim ve Teknoloji Dergisi*, (Özel Sayı), 318-324.

Öz

Örneklere bağlı olarak dinamik öğrenme yetenekleri sayesinde, doğrusal ve doğrusal olmayan ilişkileri çözümleyerek başarılı sonuçlar üreten yapay sinir ağları birçok alanda karşımıza çıkmaktadır. Yapay sinir ağlarında istenen düzeyde performansın sağlanması birçok parametreye bağlı olmakla birlikte, kullanılan ağ modeli ve bu ağın eğitiminde kullanılan algoritmalar üzerinde yapılan çalışmalar giderek artmaktadır. Bu çalışmada, arıların doğada yiyecek arama davranışlarından esinlenilerek geliştirilen yapay arı koloni (Artificial Bee Colony, ABC) algoritması ile eğitilmiş tekrarlayıcı sinir ağlarının (Recurrent Neural Network, RNN) robot navigasyonunda kullanımına yönelik yeni bir tasarım önerilmiştir. Robotun kontrol stratejisi için üzerine yerleştirilen 24 adet ultrasonik sensörden elde edilen veriler kullanılmıştır. Literatürdeki benzer çalışmalarla karşılaştırmak için ortalama karesel hatanın karekökü ve simetrik oransal ortalama mutlak hata ölçüm metrikleri kullanılmıştır. Elde edilen sonuçlar, önerilen tasarım modelinin robotun hareket yönünün tayininde etkin bir şekilde kullanılabileceğini göstermiştir. Özellikle çok sayıda sensör kullanıldığında önerilen modelin performansı diğer modellere nazaran çok daha iyi olmuştur.

Anahtar Kelimeler: robot navigasyonu; tekrarlayıcı sinir ağları; yapay arı koloni; akıllı kontrol.

Using Recurrent Neural Network Models Trained With Artificial Bee Colony Algorithm for Robot Navigation

Abstract

Artificial neural networks, which produce successful results by establishing linear and nonlinear relations by their dynamic learning abilities, are used in a wide range of fields. Even if the desired performance of networks depends on too many parameters, the number of researches on networks models and learning algorithms are gradually increasing. In this paper a new recurrent neural network (RNN) trained with artificial bee colony (ABC) algorithm was proposed for robot navigation problem. The RNN network trained with sample dataset, obtained from 24 sensors, was used for control strategy in robot movements. Root mean square error (RMS) and symmetric mean absolute percentage error (SMAPE) evalutian metrics were used to compare the proposed method against state of the art algorithms. The results showed that, the performance of the proposed method in determination of robots motion is good and especially, when a large number of sensors used, the proposed model as better performance than the other models.

Keywords: navigation; recurrent neural network; artificial bee colony; intelligent control.

*Sorumlu Yazar: Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Anabilim Dalı, Kayseri, Türkiye, ORCID: 0000-0003-3374-0593, yonemebru@gmail.com

Bu makale ***International Conference on Access to Recent Advances in Engineering and Digitalization (ARACONF 2020)* de sunulmuştur.

1. Giriş

Teknolojideki gelişmelere paralel olarak robotların insanların günlük yaşamında kullanılmasına yönelik çalışmalar da giderek yaygınlaşmaktadır. Robotik uygulamalarda herhangi bir başlangıç noktasına konumlandırılmış bir robot verilen görevleri gerçekleştirmek için farklı hedeflere ulaşmak zorundadır. Bu ulaşma işleminin kabul edilebilir süre içerisinde ve doğru bir şekilde yapılması önemli bir araştırma konusu olmaya devam etmektedir. Özellikle engellerle ve hareketli nesnelerle dolu hareket ortamlarında gerçek zamanlı olarak robotun kontrolü en önemli zorluklardandır [1], [2].

Uygulamalarda robotların hareket yönlerini belirlemek için kızılötesi, lazer, ultrasonik gibi sensörler veya kameralar kullanılmaktadır. Bu donanımlar yalnız başlarına kullanılabildiği gibi bir arada da kullanılabilmektedir. Bunların seçimi tamamen gereksinimlere, uygulamaya ve kullanıcıya bağlıdır. Bazı araştırmacılar robot navigasyonu ile ilgili detaylı araştırma makaleleri sunmuşlardır [3]-[5] Robot hareket yönlerini kontrol etmek için önerilen klasik modellere istatistiksel modellemeyi içeren [6] ve Monte-Carlo örneklemesine dayalı [7] çalışmalar örnek verilebilir. Bu uygulamalar gelecekteki olası çevre hakkında bilgi önceden bilinmediği için dinamik model oluşturmanın zorluğunu göstermektedir. Bunun yanında bu tür bir problem, gerçek zamanlı sensör okumalarının öğrenilmiş bir sistem vasıtası ile gelecekteki robot hareket yönünün belirlenmesinin temel amaç olduğu bir optimizasyon problemi olarak görülebilir. Bu yaklaşımdan da hareketle dinamik öğrenme yetenekleri sayesinde başarılı sonuçlar çıkarabilen yapay sinir ağları (Artificial Neural Network, ANN) bu tür uygulamalar için de doğru bir yöntem olabilmektedir. Günümüzde yapay zekânın gelişimine paralel olarak en hızlı ve güçlü bilgi işleme aracı olduğu bilinen insan beyninin matematiksel olarak modellenmesiyle ortaya çıkan ANN hayatımızın birçok alanında yerini almıştır. ANN modelleri biyolojik ağların karmaşıklığını tam olarak yansıtmasa da temel yapısından esinlenerek yüksek performans sağlayacak şekilde yoğun bağlantılar içeren basit hesaplama elemanlarından meydana gelmiştir. ANN modellerinden istenen düzeyde performansın sağlanması birçok parametreye bağlı olmakla birlikte ağın eğitiminde kullanılan algoritmalar üzerinde en çok çalışılan konulardandır. Gradyan düşüm, eşlenik gradyan düşüm, esnek geri yayılım, Quasi Newton ve Levenberg Marquart algoritmaları sıklıkla kullanılan klasik eğitim algoritmalarıdır. Bu klasik algoritmaların yanında bazı sezgisel yöntemler ağın performansını artırmak için kullanılabilmektedir. Dash ve arkadaşları sezgisel yöntemlerden yerçekimsel arama yöntemi (Gravitational Search, GS) algoritması ile eğittikleri yapay sinir ağını ve GS ile parçacık sürü optimizasyon (Particle Swarm Optimization, PSO) algoritmasını içeren GSPSO hibrid algoritması ile eğittikleri tek gizli katmanlı ileri beslemeli sinir ağını (Feed Forward Neural Network, FFNN) robotların hareket yönlerini belirlemede kullanmışlardır [8], [9].

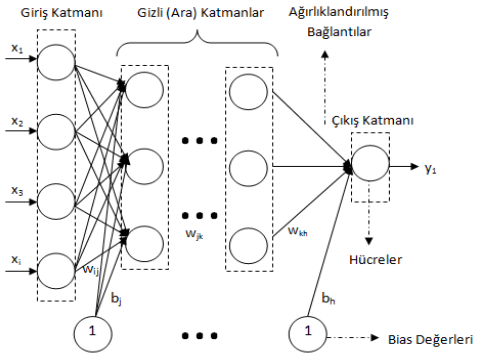
Bu algoritmalardan farklı olarak sezgisel yöntemlerden biri olan yapay arı koloni algoritmasının (Artificial Bee Colony, ABC) birçok mühendislik probleminin çözümünde kullanılmasının yanında [10] ANN eğitiminde de oldukça başarılı sonuçlar verdiği bilinmektedir [11], [12]. 2005 yılında Karaboğa tarafından önerilen bu algoritma arıların doğada yiyecek arama davranışlarından esinlenilerek geliştirilmiştir [13].Tekrarlayıcı ya da yinelemeli ağ yapıları (Recurrent Neural Network, RNN) olarak adlandırılan ve geri besleme bağlantıları içeren ağların özellikle anlık alınan veri setleri üzerinde diğer ağ modellerine nazaran daha başarılı olduğu söylenebilir [14], [15], [16]. Bu çalışmada literatürdeki çalışmalardan farklı olarak robot hareket kontrolünde ABC algoritması ile eğitilmiş RNN ağ modelinin kullanılması önerilmiştir. Önerilen modelden elde edilen sonuçlar literatürdeki farklı çalışmalarla kıyaslanmıştır.

Bu amaçlar doğrultusunda ANN, RNN ve ABC algoritması sırası ile Bölüm-2 ve Bölüm-3'de verilmiştir. Önerilen modelin göstermiş olduğu performans sonuçları Bölüm-4'de verilmiştir. Bölüm-5'de edilen sonuçlar yorumlanmıştır.

2. Materyal ve Metot

2.1. Yapay Sinir Ağları

Yapay sinir ağları yapay sinir hücrelerinin birbirine bağlanmasıyla oluşan yapılardır. Bu ağlar giriş katmanı, ara katmanlar ve çıkış katmanı olarak üç ana katmanda incelenmektedir. Bir ANN'nin temel yapısı ve bulundurduğu birimler Şekil 1'de görülmektedir.

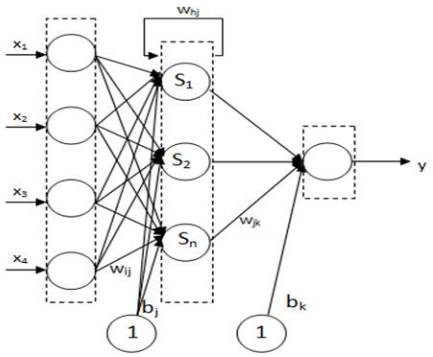


Şekil 1. Yapay Sinir Ağı

Giriş Katmanı, ANN'ye dış dünyadan girdilerin geldiği katmandır. Bu katmanda giriş sayısı kadar hücre bulunmaktadır ve genellikle girdiler herhangi bir işleme uğramadan diğer katmanlara iletilmektedir. *Gizli(Ara) Katman*, giriş katmanından çıkan bilgilerin geldiği katmandır. Ara katman sayısı ağdan ağa değişebilir. Bazı yapay sinir ağlarında ara katman bulunmadığı gibi bazı yapay sinir ağlarında ise çok sayıda ara katman bulunabilir. Ara katmanlardaki nöron sayıları giriş ve çıkış sayısından bağımsızdır. Birden fazla ara katman olan ağlarda ara katmanların kendi aralarındaki hücre sayıları da farklı olabilir. Ara katmanların ve bu katmanlardaki nöronların sayısının artması hesaplama karmaşıklığını ve süresini arttırmasına rağmen yapay sinir ağının daha karmaşık problemlerin çözümünde de kullanılabilmesini sağlar. Bu katman sayısının belirlenmesinde bazı varsayımlar olmakla birlikte henüz kesin bir bilgi bulunmamaktadır. Problemin amacına ve uygulama alanına göre en uygun katman ve nöron sayısı seçimi deneme yoluyla belirlenmektedir. *Çıkış Katmanı,* ara katmanlardan gelen bilgileri işleyerek ağın çıktılarını üreten katmandır. Bu katmanda üretilen çıktılar dış dünyaya gönderilir.

Bir ANN tasarımında öncelikle ağ mimarisinin seçilmesi, katman sayısı ve nöron sayısı gibi yapısal özelliklerinin belirlenmesi gerekir; sonraki aşamalarda ise işlemci elemanların kullandığı fonksiyonların belirlenmesi; öğrenme algoritması ve gerekli parametrelerin belirlenmesi gerekmektedir.

2.2. Tekrarlayıcı Sinir Ağları



Şekil 2. Tekrarlayıcı Ağ Yapısı

Şekil 2'de gösterilen yapıda (t-1) anındaki çıkış katmana giriş olarak verilerek t anındaki çıkışı doğrudan etkiler. Yani her düğüm için o andaki (t) ve yakın geçmişteki olarak adlandırabileceğimiz (t-1) olarak iki giriş mevcuttur. Şekil 2'de görülen yapı için çıktı Denklem (1)'de, gizli katman çıktısı Denklem (2)'de, çıktı katmanı net girdisi Denklem (3)'de ve ağın net çıktısı Denklem (4)'te verilmiştir.

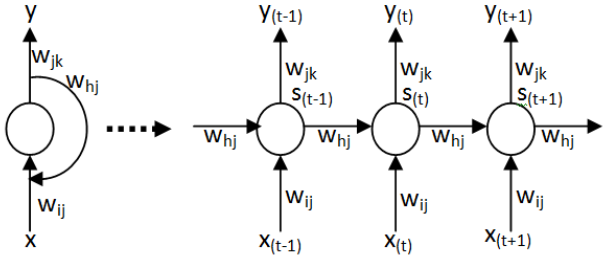
$$s_j = \sum_{i=1}^{N_i}(w_{ij}x_i + b_j) + \sum_{h=1}^{N_h}(w_{hj}h_j^{-1}) \quad (1)$$

$$h_j = f(s_j) \quad (2)$$

$$s_k = \sum_{j=1}^{N_h}(w_{jk}h_j + b_k) \quad (3)$$

$$y_k = f(s_k) \quad (4)$$

Şekil 3'de RNN ağının geri besleme bağlantıları açılarak ileri beslemeli sinir ağına dönüştürülmüş şekli görülmektedir.



Şekil 3. RNN Ağ Yapısının Açık Şekli

2.3. Yapay Arı Koloni Algoritması

Arıların akıllı yiyecek arama davranışı modelinden esinlenerek Karaboğa tarafından önerilen yapay arı kolonisi algoritması optimizasyon problemlerinde kullanılmaktadır [13]. Algoritmanın temel adımları Şekil 4'de verilmiştir.

1: Başlangıç popülasyonunun oluşturulması
2: Uygunluk değerlerinin hesaplanması
3: Repeat
4: İşçi arı safhası
5: Gözcü arı safhası
6: Kâşif arı safhası
7: Seçme ve güncelleme
8: Until

Şekil 4. ABC algoritmasının temel adımları

2.3.1. İşçi Arı Safhası

İşçi arı aşaması, çözümlerin daha iyi arama alanı vaat eden bölgelere doğru hareket etmesini sağlar. Bu aşamada her bir çözüm vektörünün komşulukları Denklem (5) ile tanımlanan yeni çözüm üretme mekanizması kullanılarak aranır.

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (5)$$

Burada v_{ij}yeni aday çözüm vektörü, φ, [-1,1] aralığında üretilen rastgele bir değer, *k* rastgele seçilmiş komşu bir çözüm vektörüdür. Üretilen aday çözüm sonrası açgözlü seçim metodu kullanılarak, yeni çözüm ve mevcut çözüm karşılaştırılır. Yeni çözüm daha iyi ise mevcut çözümün yerini alır.

2.3.2. Gözcü Arı Safhası

Gözcü arı aşamasında da yeni çözüm üretmek için Denklem (1) kullanılır. İyileştirilme yapılacak çözümün belirlenmesinde iyi çözümlerin seçilme şansının daha fazla olduğu olasılıksal seçim metodu kullanılır. Böylelikle iyi bireylerin seçilme şansı artarak onların etrafında daha fazla yerel arama yapılabilmektedir. Temel ABC algoritmasında olasılık seçimi için Denklem (6) ile Hesaplanan olasılık değerleri kullanılır.

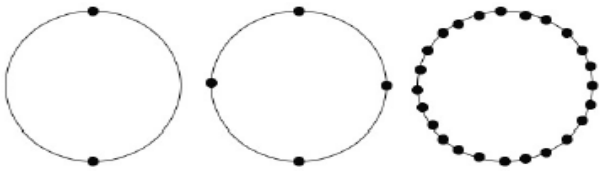
$$p_i = \frac{uygunluk_i}{\sum_{n=1}^{SN} uygunluk_n} \quad (6)$$

2.3.3. Kaşif Arı Safhası

Doğada çalışma süreçleri nedeni ile işçi arılar ve gözcü arılar bazı gıda kaynaklarını tüketebilmektedir. Bu ABC algoritması açısından bir çözümün komşuluklarının yeterince aranması ve çözümün artık geliştirilemiyor olması anlamına gelir. Bu nedenle bu çözümün iyileştirilmeye çalışılmasına artık gerek yoktur, onun yerine rastgele başka bir çözüm üretilebilir. Kaynağınyeterince aranıp aranmadığı *limit* adı verilen algoritmaya özel bir kontrol parametresi ile belirlenir. x_i konumundaki çözüm vektörü *limit* parametresi sayısınca gelişememiş ise $x_i$çözüm vektörü terk edilir ve o kaynağın arısı kaşif arı haline gelerek rastgele araştırma yapar.

3. Araştırma Sonuçları ve Tartışma

Gerçekleştirilen model Matlab 2016 programlama dilinde kodlanmış, 3 GB RAM'e sahip, INTEL Core i3 işlemcili bilgisayarda Windows 7 işletim sistemi üzerinde sonuçlar alınmıştır. Kullanılan veri setinde robotun bir odada ilerlemesi için üzerine yerleştirilen 24 adet ultrasoniksensörden yararlanılmaktadır. Birinci senaryoda (PR1) önüne ve arkasına yerleştirilmiş iki adet, ikinci senaryoda (PR2) 90º açılarla yerleştirilmiş önünde, arkasında, sağında ve solunda olan dört adet ve üçüncü senaryoda (PR3) 15º likaçılarla yerleştirilmiş olan 24 adet ultrasoniksensörün tamamı kullanılmıştır. Ağın çıktısı ise düz ilerle, keskin sağa dön, hafif sağa dön ve hafif sola dön şeklinde dört farklı yönlendirme komutundan oluşan tek bir çıktıdır. Robot üzerine yerleştirilen sensörlerin şematik gösterimi Şekil 5'de,genel veri kümesindeki sınıf dağılımının özeti Tablo 1'de verilmiştir [6].



Şekil 5. İki, Dört Ve Yirmi dört Sensörlü Robotun Şematik Gösterimi

Tablo 1. Veri Setindeki Sınıf Dağılımı

Yön	**Örnek Sayısı**	**Oran %**
Düz İlerle	2205	% 40.41
Keskin Sağa Dönüş	2097	% 38.43
Hafif Sağa Dönüş	826	% 15.14
Hafif Sola Dönüş	328	% 6.01

Her veri setinde toplam 5456 sensör örneğinin %70'i eğitim ve %30'u test için kullanılmıştır. Veriler kullanılmadan önce aşağıda verilen Denklem (7) ile normalize edilmiştir. Burada y_i, i. veri elemanının normalleştirilmiş değeridir. ABC algoritması için gerekli parametrelerden iterasyon sayısı 1000 ve limit değeri 100 alınırken popülasyon büyüklüğü performans değerleri 20, 30 ve 50 değerleri için ayrı ayrı alınmıştır. Eğitilen RNN ağında giriş sayıları her bir senaryodaki sensör sayısı kadar alınırken, tek bir gizli katman kullanılmıştır. Gizli katmandaki nöron sayıları, giriş sayılarına eşit, iki ve üç katı kadar alınarak ayrı ayrı incelenmiştir. Sonuçların güvenilirliği açısından her bir senaryo 30'ar kez çözülmüş ve sonuç tablolarında bu çözümlerden elde edilen en küçük, ortalama, en büyük ve standart sapma istatistik değerleri verilmiştir.

$$y_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{7}$$

Önerilen modelin değerlendirilmesi amacıyla performans metriklerinden, mutlak hata fonksiyonlarından ortalama karesel hatanın karekökü olarak ifade edilen RMS (Root Mean Square) ve çok büyük ya da çok küçük hata değerlerine daha duyarlı simetrik hata fonksiyonlarından SMAPE (Symmetric Mean Absolute Percentage Error) kullanılmıştır. RMS ve SMAPE hata fonksiyonlarının formülleri sırası ile Denklem (8) ve (9)'da verilmiştir.

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i^2)} \tag{8}$$

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n} 200.|s_i| \tag{9}$$

ABC algoritması ile eğitilen RNN ağ modelinin performans metrikleri eğitim verileri için Tablo 2'de, test verileri için Tablo 3'de ayrı ayrı verilmiştir.

Tablo 2.RNN Ağının Eğitim Performans Metrikleri

		PR1	**PR2**	**PR3**
RMS	Min	0,218	0,207	0,205
	Ort.	0,226	0,213	0,211
	Max	0,231	0,217	0,217
	Std.	0,003	0,003	0,003
SMAPE	Min	4.63.10-5	4.55.10-5	5.47.10-5
	Ort.	4.81.10-5	4.98.10-5	6.02.10-5
	Max	4.99.10-5	5.35.10-5	6.41.10-5
	Std.	9.53.10-7	1.45.10-6	2.49.10-6

Tablo 3. RNN Ağının Test Performans Metrikleri

		PR1	PR2	PR3
RMS	Min	0,224	0,201	0,208
	Ort.	0,235	0,217	0,221
	Max	0,273	0,233	0,234
	Std.	0,011	0,006	0,007
SMAPE	Min	1.11.10-4	1.09.10-4	1.29.10-4
	Ort.	1.17.10-4	1.22.10-4	1.39.10-4
	Max	1.22.10-4	1.34.10-4	1.58.10-4
	Std.	2.59.10-6	5.69.10-6	2.49.10-6

Tablo 2 ve Tablo 3'deki RMS ve SMAPE hata fonksiyon değerleri incelendiğinde önerilen modelin kabul edilebilir oranda başarı sağladığı gözlemlenmiştir. Elde edilen sonuçları literatürdeki farklı çalışmalarla karşılaştırmak amacıyla aynı veri setini kullanan Dash ve arkadaşlarının GS-FFNN ve GSPSO-ANN modelleri ile yaptıkları çalışmalar kullanılmıştır. Tablo 4'de PR1, PR2 ve PR3 olmak üzere üç farklı senaryo için kıyaslanan çalışmalar ve önerilen RNN-ABC yapısına ait başarı yüzdeleri verilmiştir.

Tablo 4.Aynı Veri Kümesi için Farklı Metotların Başarı Yüzdelerinin Karşılaştırılması

Yöntem	Pop.	PR1	PR2	PR3
RNN-ABC	20	77.228	76.277	75.521
RNN-ABC	30	76.847	73.854	**80.268**
RNN-ABC	50	69.701	**76.847**	79.352
GS-FFNN [8]	20	86.382	70.088	69.721
GS-FFNN [8]	30	85.686	67.705	69.721
GSPSO-ANN [9]	30	79.670	73.663	71.886
GSPSO-ANN [9]	50	79.341	73.150	73.992
GD-ANN [17]	-	**89.560**	71.590	46.500

Tablo 4 incelendiğinde 2 girişli olan PR1 için ağın giriş nöron sayısı azalmakta ve matematiksel işlem yükü azaldığı için GD algoritması daha başarılı sonuçlar üretebilmektedir; ancak iki sensörden alınan verileri gerçek ortamda tüm engellerden kaçmak çok olası görülmemektedir. Giriş sayısı arttıkça ağın karmaşıklığı artmakta ve bu noktada önerilen yöntem daha iyi sonuçlar üretmektedir. 4 sensörlü PR2 için en başarılı performans popülasyon büyüklüğü 50 olan RNN-ABC algoritmasından alınırken, PR3 için en başarılı sonuç popülasyon büyüklüğü 30 olan ABC algoritmasından alınmıştır.

Ayrıca genel olarak kullanılan sensör sayısı ile başarı oranının doğru orantılı olduğu, daha fazla sensör kullanılan senaryolarda başarı oranının daha yüksek olduğu söylenebilir.

4. Sonuç

Çalışmada yapay sinir ağ modellerinden biri olan tekrarlayıcı sinir ağlarının eğitimiminde, arıların doğada yiyecek arama davranışlarından esinlenilerek geliştirilen yapay arı koloni algoritmasının kullanılması önerilmiştir. Önerilen model ile robotun haraket kontrol stratejisi belirlenmiştir. Sonuçlar değerlendirildiğinde gerçekleştirilen tasarım modelinin robotun hareket yönünün tayininde başarılı olduğu görülmüştür. Çalışmanın farklı yapay sinir ağ modelleriyle genişletilmesi, ABC algoritmasının yakınsama hızını arttırmak için farklı hibrid optimizasyon algoritmalarının geliştirilmesi, hazır veri setleri yerine hazırlanacak olan donanımsal yapılardan elde edilecek kendi verilerimizin kullanılması ve aynı anda çok sayıda robotun hareket stratejisinin belirlenerek yönlendirilmesi gelecekte yapılacak çalışmalar olarak planlanmaktadır.

Teşekkür

Bu çalışmayı FYL-2017-7662 proje kodu ile destekleyen, Erciyes Üniversitesi BAP Koordinasyon Birimine teşekkür ederiz.

Kaynakça

[1] Kruse, T., Pandey, A.K., Alami, R.ve Kirsch, A. (2013). Human-aware robot navigation: A survey, Robotics and Autonomous Systems 61(12), 1726–1743.

[2] Katsev, M., Yershova, A., Tovar, B., Ghrist, R. Ve La Valle, S.M. (2011). Mapping and pursuit-evasion strategies for a simple wall-following robot, IEEE Transactions on Robotics 27(1), 113–128.

[3] Hoy, M., Matveev, A.S. ve Savkin, A.V. (2015). Algorithms for collision free navigation of mobile robots in complex cluttered environments: a survey. Robotica, 33(3), 463-497.

[4] Yang L, Qi J, Song D, Xiao J, Han J ve Xia Y. (2016). Survey of robot 3D path planning algorithms. Journal of Control Science and Engineering, 5,76-82.

[5] Patle, B.K., Ganesh B.L., A. Pandey, D.R.K. Parhi, A., Jagadeesh. (2019). A review: On path planning strategies for navigation of mobile robot. Defence Technology, InPress.

[6] Trautman, P, Ma, J., Murray, R.M.ve Krause, A. (2015). Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation, The International Journal of Robotics Research 34(3), 335–356.

[7] Li, T., Sun, Z., Xu Y. Ve Zhang, B. (2015). Robot navigation based on visual feature perception and Monte Carlo sampling, in: Control and Decision Conference (CCDC), 27th Chinese, IEEE, 3237–3242.

[8] Dash, T., Nayak, T. ve Swain, R.R. (2015). Controlling Wall Following Robot Navigation Based on Gravitational Search and Feed Forward Neural Network, Proceedings of the 2nd International Conference on Perception and Machine Intelligence, 196-200.

[9] Dash, T., Swain, R.R. ve Nayak, T. (2017). Automaticnavigation of wall-following mobile robot using a hybrid metaheuristic assisted neural network, Data Science, 1-17.

[10] Karaboğa D. ve Akay, B. (2009). A comparative study of Artificial Bee Colony algorithm, Applied Mathematics and Computation, 214(1), 108-132.

[11] Karaboğa, D. ve Akay, B. (2007). Artificial Bee Colony (ABC) Algorithm on Training Artificial Neural Networks. Eskişehir, IEEE 15th Signal Processing and Communications Applications.

[12] Karaboğa D., Akay B. ve Öztürk C. (2008). Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks, 4th International Conference on Modeling Decisions for Artificial Intelligence, Kitakyushu, Japonya, (4617), 318-321.

[13] Karaboga, D. (2005). An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report. Erciyes University. Kayseri.

[14] Catalbas, B. (2015). Recurrent Neural Network Learning With An Application To The Control Of Legged, Bilkent University.

[15] Shimodaira, H. (2002). Time-Series Prediction. Cornelius T. Leondes. Expert Systems The Technology of Knowledge Management and Decision Making for the 21st Century, Academic Press, 4, 1295-1311.

[16] Zhang, Y. et al. (2017). A recurrent neural network approach for visual servoing of manipulators. Macau, China, IEEE International Conference on Information andAutomation (ICIA), 614-619.

[17] Dash, T., Ranjan, S. ve Mishra, G. (2014). Neural network approach to control Wall following robot navigation,In Advanced Communication Control and Computing Technologies (ICACCCT), International Conference, India, 1072-1076