Bozok
Journal of Engineering and Architecture

Araştırma Makalesi/Research Article

Parçacık Filtresinin Optimizasyonu için Genetik Algoritma Tabanlı Yeni Bir Yaklaşım

Fatma Selcen Turgun[1]*, Hasan Zorlu[2]

[1] Yozgat Bozok Üniversitesi, Mühendislik-Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Yozgat, Türkiye
[2] Erciyes Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Kayseri, Türkiye

MAKALE BİLGİSİ

Makale Tarihleri:

Geliş tarihi
28.04.2023
Kabul tarihi
12.06.2023
Yayın tarihi
21.06.2023

Anahtar Kelimeler:

Parçacık Filtresi
Durum Tahmini
Genetik Algoritma
Mutasyon
Çaprazlama

ÖZET

Parçacık filtresi algoritması, Bayes tahmin teorisi çerçevesinde Monte Carlo fikrini kullanan bir filtreleme yöntemidir. Parçacıkları ve ağırlıklarından oluşan ayrık rasgele ölçüyü kullanarak olasılık dağılımına yaklaşır, yeni ayrık rasgele ölçüyü algoritmaya göre yinelemeli olarak günceller. Parçacık filtresi (PF) algoritması, doğrusal olmayan Gauss olmayan herhangi bir sisteme durum tahmini için uygulanabilir ve son on yılda birçok mühendislik alanında büyük ilgi görmüştür. Ancak standart parçacık filtresi mevcut ölçülen değeri dikkate almaz, bu da bazı iterasyonlardan sonra birkaç parçacık dışında kalan parçacıkların ağırlığının neredeyse yok denecek kadar az olmasına sebep olur. Böylelikle parçacık bozulması sorunu ortaya çıkar. Bozulmayı önlemek için PF yeniden örnekleme tekniğini kullanır, fakat bu parçacık çeşitliliğini azaltır ve parçacık yoksullaşmasına neden olur. Bu makalede, filtrenin bu sorununun üstesinden gelebilmek için Parçacık filtresine Genetik algoritma (GA) dahil edilmiştir. Genetik algoritmadaki seçim, çaprazlama ve mutasyon operatörlerinin özellikleriyle birleştirilmiş geliştirilmiş bir parçacık filtresi önerilmiştir. PF algoritmasının yeniden örnekleme aşamasından önce, en yüksek ağırlığa sahip parçacıklar bir genetik algoritma kullanılarak evrimleştirilir. Önerilen algoritmada, GA tarafından optimize edilen parçacık kümesi hedefin gerçek durumunu daha iyi ifade eder ve anlamlı parçacıkların sayısı önemli ölçüde artmıştır. Son olarak, önerilen yöntemin etkinliğini göstermek için bir bilgisayar simülasyonu gerçekleştirilmiştir. Simülasyon sonuçları, önerilen GA tabanlı yeni algoritmanın tahmin doğruluğunu standart parçacık filtresine kıyasla önemli ölçüde iyileştirdiğini göstermektedir.

A New Approach Based on Genetic Algorithm for Optimization of Particle Filter

ARTICLE INFO

Article history:

Received
28.04.2023
Accepted
12.06.2023
Published
21.06.2023

Keywords:

Particle Filter
State Estimation
Genetic Algorithm
Mutation
Crossover

ABSTRACT

The particle filter algorithm is a filtering method that uses the idea of Monte Carlo within the framework of Bayesian estimation theory. It approximates the probability distribution using a discrete random measure consisting of particles and their weights, iteratively updating the new discrete random measure according to the algorithm. The particle filter (PF) algorithm can be applied to any nonlinear non-Gaussian system for state estimation and has attracted great interest in many engineering fields over the last decade. However, the standard particle filter does not take into account the current measured value, which leads to the fact that after some iterations the weight of all but a few particles is almost negligible. Thus the problem of particle distortion arises. To avoid distortion, the PF uses a resampling technique, but this reduces particle diversity and leads to particle impoverishment. In this paper, a Genetic algorithm (GA) is incorporated into the Particle filter to overcome this problem of the filter. An improved particle filter combined with the features of selection, crossover and mutation operators in the genetic algorithm is proposed. Before the resampling phase of the PF algorithm, the particles with the highest weight are evolved using a genetic algorithm. In the proposed algorithm, the set of particles optimized by the GA better expresses the true state of the target and the number of significant particles is significantly increased. Finally, a computer simulation is performed to demonstrate the effectiveness of the proposed method. The simulation results show that the new proposed GA-based algorithm significantly improves the prediction accuracy compared to the standard particle filter.

ORCID ID: Fatma Selcen TURGUN: 0000-0003-4372-7026; Hasan ZORLU: 0000-0001-8173-6228

***Sorumlu yazar(lar)/Corresponding author(s):** Yozgat Bozok Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Yozgat, Türkiye
Tel: +90 354 242 10 02-1920
Fax:
E-mail: f.selcen.bozkurt@bozok.edu.tr

Bu makaleye atıfta bulunmak için/To cite this article: F. S. Turgun, H. Zorlu, "Parçacık Filtresinin Optimizasyonu için Genetik Algoritma Tabanlı Yeni Bir Yaklaşım", Bozok Journal of Engineering and Architecture, vol. 2, no. 1, pp. 24-33, 2023

1. GİRİŞ

Sistem durumlarının gerçek zamanlı olarak elde edilmesi, endüstrideki uygulamalar için oldukça önemlidir. Fakat gizli durumlara (veya dahili durumlara) genellikle sensörler tarafından erişilemez. Durum tahmini, gürültüyle bozulmuş ölçümlerden gizli durumları çıkarmak için önerilen bir tekniktir. Yapılan çalışmalarda sistemlerin durumlarının tahmin edilmesi gerekmektedir ve bu sistemler çoğunlukla Gauss olmayan gürültüye sahip doğrusal olmayan sistemlerdir.

Farklı türden dinamiklerin gizli değişkenlerini tahmin etmek amacıyla, literatürde birçok filtreleme yöntemi bulunmaktadır. Bunlardan doğrusal dinamik sistemler için Kalman filtresi [1] ve sonlu durum sistemleri için ızgara tabanlı filtre [2,3] kullanılır. Genişletilmiş Kalman filtresi (EKF), Kokusuz Kalman filtresi (UKF) ve yaklaşık ızgara tabanlı filtre doğrusal olmayan dinamikler için uygundur [2,3].

Parçacık filtresi (PF), durum tahmini için Bayes tahmin teorisi çerçevesinde ardışık Monte Carlo kullanan bir filtreleme yöntemidir ve son yıllarda artan bir ilgi görmüştür [4]. PF, doğrusal olmayan ve Gauss olmayan sistemler için geleneksel Kalman filtreleme yöntemlerine göre önemli avantajlar sağlar, bu nedenle birçok uygulama alanında yaygın olarak kullanılmaktadırlar [3,5]. PF'nin uygulanması sistem modelinden bağımsız olduğundan, farklı doğrusal olmayan sistemlerde kolayca uygulanabilir. EKF ve UKF ile karşılaştırıldığında PF, özellikle doğrusal olmayan sistemler için daha doğru durum tahmin sonuçları sağlayabilir. PF'nin bir başka üstün performansı, diğer filtreleme teknikleri için zor olabilecek Gauss olmayan sistemlerde durum tahminine uygulanabilmesidir. Bu çekici özellikler sayesinde, PF hem akademik hem de endüstriyel alanlardan büyük ilgi görmüştür [6–8].
Son yirmi yılda, PF'lerin uygulamaları, mobil robot lokalizasyonu ve haritalama, nesne izleme, doğrusal olmayan stokastik sistemlerde arıza teşhisi, kablosuz telekomünikasyonda kullanıcı tespiti, işitsel uygulamalarda konuşmacı takibi gibi çeşitli alanlara genişledi [9].

Parçacık filtresinin teorisi parçacıklardan ve ağırlıklarından oluşan ayrık rasgele ölçümü kullanarak ilişkili olasılık dağılımına yaklaşmak ve algoritmaya göre ayrık rasgele ölçümü yinelemeli olarak güncellemektir [10]. Parçacıklarla gizli durumların arka yoğunluklarına yaklaşırlar ve parçacıkları yerleştirerek, ağırlıklandırarak ve çoğaltarak zaman içinde ardışık olarak ayrık rasgele ölçümü günceller [11].

Bununla birlikte, geleneksel parçacık filtresi mevcut ölçümü hesaba katmaz, bu önem yoğunluk fonksiyonundan örneklenen parçacıkların, gerçek sonsal olasılık yoğunluk işlevinden örneklenen parçacıklardan oldukça farklı olmasına neden olur. Parçacık filtrelerinde parçacık bozulmasının üstesinden gelmek için daha küçük ağırlıktaki parçacıkları çıkaran yeniden örnekleme yöntemi kullanılır. Yeniden örnekleme, basitçe büyük önem ağırlıklarına sahip parçacıkları seçer ve küçük ağırlıklara sahip olanları eler ama bu da parçacık yoksullaşmasına sebep olur [12].

Son yıllarda, yeniden örnekleme aşamasının getirdiği parçacık yoksullaşması soruna karşı Zhao ve ark. [13], Genetik Yeniden Örneklemeye (GRPF) dayalı geliştirilmiş bir parçacık filtresi sunmuştur. Bu algoritmada, yeniden örnekleme stratejisinin yerine genetik algoritmanın çaprazlama ve mutasyon operatörlerin kullanılması önerilmiştir. Han ve ark. [14] parçacık yoksullaşma sorununu önlemek için Bağışıklık Genetik Algoritması (IGA-PF) ile parçacık filtresi kullanmışlardır. Walia ve Kapoor [15] tarafından parçacık yoksulluğunu önlemek için Gelişmiş Guguk Kuşu Araması (ICS) adı verilen bir meta-sezgisel optimizasyon tekniği bir Parçacık Filtresi algoritmasına uygulandı. Ayrıca, karınca kolonisi optimizasyonu (ACO), yapay bağışıklık sistemi (AIS) ve yapay balık sürüsü algoritmaları (AFSA) gibi diğer bazı evrimsel hesaplama yöntemleri de tahmin performansını iyileştirmek için parçacık filtresine uygulanmıştır [16,17,18].

Genetik algoritma (GA) ve PF arasında benzerlikler olduğu için, bazı araştırmacı ve akademisyenler GA'yı parçacık yoksulluğu sorununu hafifletmek için PF algoritmasına dahil etmeye çalışmıştır. Yeniden örnekleme aşamasından önce evrimsel algoritmaların kullanılması, parçacık filtresi algoritmalarındaki parçacıkları optimize eder [19]. Bununla birlikte hesaplama yükünü azaltır ve hızını artırır.

Bu makalede, durum tahmini için Genetik algoritmanın Parçacık filtresine dahil edilerek geliştirilmesi önerilmiştir. Makalede önerilen yaklaşımda yeniden örnekleme adımından önce Genetik algoritma operatörleri uygulanır, bu parçacıkların durum arama uzayını genişletir ve sonsal dağılıma daha yeterli bir şekilde yaklaşılabilir. Parçacık yoksulluğu problemini hafifletmek için tasarlanmış yeni bir değiştirilmiş PF önerilmiştir. Genel PF ile karşılaştırıldığında, daha doğru durum tahmin sonuçları sağlayabilir. Önerilen GA tabanlı PF, durum tahmini için doğrusal olmayan bir problem üzerinde uygulanmaktadır. Tahmin sonuçları tatmin edicidir.

Bu makalenin organizasyonu şu şekildedir. Bölüm 2 de, Parçacık filtreleme algoritmasının temelleri, Genetik algoritmanın ilkeleri ve parçacık yoksullaşmasını gidermek için Parçacık filtresiyle Genetik algoritmanın kombinasyonu açıklanmaktadır. Gerçekleştirilen simülasyonlar ve deneysel sonuçlar Bölüm 3'te ele alınmıştır. Son olarak, sonuç açıklamaları Bölüm 4'te sunulmaktadır.

2. MATERYAL VE METOT

2.1. Genel Parçacık Filtresi

Filtreleme işlemi, geçmiş durum ve en son geçerli ölçüm değeri kullanılarak sistemin gizli olan en son durum tahmin değerinin elde edilmesi işlemidir [20]. Durum tahmini $p(x_t|x_{t-1}, y_{0:t})$ şeklinde olasılık yoğunluk fonksiyonu (PDF) olarak ifade edilir. Burada t anındaki x_t durum ve y_t gözlem verileridir. y_t gözlem verisi, gürültülü ölçümleri içeren sensör ve donanımlardan okunan verileri tanımlamaktadır. PF ile hedeflenen bir sistemin analizi için zamana bağlı durum geçiş ve ölçüm modeline gerek duyulmaktadır. Durum uzayı dinamik modeli, denklemler (1) ve (2) ile tanımlanır:

$$x_k = f(x_{k-1}, w_{k-1}) \tag{1}$$
$$y_k = h(x_k, v_k) \tag{2}$$

Burada x_k, k zaman anındaki durum vektörü, y_k ise ölçüm vektörüdür. w_{k-1} ve v_k sırasıyla sistem ve ölçüm gürültüleridir. Ayrıca f ve h zamana bağlı, doğrusal olmayan durum geçiş ve ölçüm fonksiyonlarıdır.
Amaçlanan sonsal dağılım $p(x_k|y_{1:k})$, Bayes yaklaşımıyla özyinelemeli olarak iki adımdan oluşan hesaplama adımıyla gerçekleştirilebilmektedir. İlk adım tahmin adımıdır ve $k-1$ anında amaçlanan $p(x_k|y_{1:k-1})$ dağılımını, (3) ile ifade edilen eşitlikte $p(x_{k-1}|y_{1:k-1})$ değeri kullanılarak elde etmeyi amaçlar.

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})d_{x_{k-1}} \tag{3}$$

$p(x_k|y_{1:k-1})$ değeri Bayes Kuralı'nın özyinelemeli yapısıyla, $p(x_k|x_{k-1})$ değeri ise (1) ile belirtilen eşitlik yardımıyla hesaplanan değerlerdir. 2. Adım olan güncelleme adımı ise bir önceki adım olan tahmin adımından elde edilen değer ve yeni ölçüm (y_k) değerleri kullanılarak (4) eşitliğiyle x_k üzerinde sonsal dağılımın hesaplanması olarak ifade edilmektedir.

$$p(x_k|y_{1:k}) \; \alpha \; p(y_k|x_k)p(x_k|y_{1:k-1})d_{x_{k-1}} \tag{4}$$

Bir N parçacığı $\{x_k(i),\ i=1,2,\ldots,N\}$ kümesi, hedef dağılımdan örnekleme yapmanın zor olduğu durumlarda taslak dağılımdan örnek seçilmesi yoluyla oluşturulur. Hedef dağılım p_x yerine, taslak dağılım $q(x_k, y_{1:k})$ örneklenir. Amaç, denklem (5) ile yaklaşık olarak sonsal PDF'yi $p(x_k|y_{1:k})$ tanımlamaktır. En son durumu ifade eden k anındaki sonsal dağılıma ait gösterim (5)'te sunulmuştur.

$$\hat{p}(x_k|y_{1:k}) = \sum_{i=1}^{N} w_k^{(i)}\, \delta\left(x_k - x_k^{(i)}\right) \tag{5}$$

Burada $\delta(.)$, Dirac delta işlevidir ve $w_k^{(i)}$, denklem (7) ile ifade edilen normalleştirilmiş önem ağırlıklarıdır.
Örneklerin $q(x_k|y_{1:k})$ ile ifade edilen önem yoğunluk fonksiyonundan seçildikleri düşünüldüğünde parçacık ağırlıkları (6) ile ifade edilen eşitlik yardımıyla hesaplanabilir.

$$w_k^{(i)} \alpha \; \frac{p\left(x_k^i|y_{1:k}\right)}{q\left(x_k^i|y_{1:k}\right)} \tag{6}$$

Elde edilen yeni ölçüm değerleri kullanılarak ağırlıklar (7) ile belirtilen denklem şeklinde ifade edilebilmektedir.

$$w_k^{(i)} = \; w_{k-1}^{(i)} \frac{p\left(y_k|x_k^i\right)\, p\left(x_k^{(i)}|x_{k-1}^{(i)}\right)}{q\left(x_k^{(i)}|x_{k-1}^{(i)}, y_k\right)} \tag{7}$$

Optimal önem yoğunluğu fonksiyonunun kullanılması durumunda, Parçacık filtresi algoritmasının gerçekleştirilmesi çok zordur. Basit ve kullanışlı bir alternatif yöntem, denklem (8) ile ifade edilen alt-optimal önem yoğunluğu fonksiyonu olarak $p(x_{k+1}|x_k)$ seçmektir.

$$q(x_k|x_{k-1}^{(i)}, y_k) = p(x_k|x_{k-1}^{(i)}) \tag{8}$$

Bu önem yoğunluk fonksiyonundan örnekleme yoluyla, $x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)})$ parçacıklarına sahip oluruz. Parçacıkların ağırlıkları $w_k^{(i)} = w_{k-1}^{(i)}\, p(y_k|x_k^{(i)})$ ile hesaplanır ve $\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{N} w_k^{(i)}}$ ile normalize edilir.

Yukarıda bahsedilen optimal olmayan yoğunluk fonksiyonundan örnekleme yapmak bir soruna yol açar. Bu ağırlıkların varyansını arttırmaktır [21]. Bu süreçte parçacıkların çoğu sıfıra yakın normalleştirilmiş ağırlıklara sahip olacak ve sadece bir parçacık bire yakın büyük bir ağırlığa sahip olacaktır. Bu sorun parçacıkların dejenerasyonu yani parçacık bozulması olarak adlandırılır. Yeniden örnekleme, ağırlıkların varyansını küçülterek bozulma problemini çözmek için kullanılan bir yöntemdir. Yeniden örnekleme adımında, yüksek ağırlığa sahip örnekler birkaç kez kopyalanır ve düşük ağırlığa sahip örnekler çıkarılır [22]. Böylelikle bozulma için çözüm bulunmuş gibi dursa da başka bir sorun haline gelen parçacık yoksullaşması sorunu ortaya çıkar. Yüksek ağırlıklı parçacıkların aşırı çoğaltılması nedeniyle, anlamlı parçacıkların sayısı azalır ve bu da yeni parçacıklar kümesinin bilgi kapasitesinin düşmesine neden olur [19]. Etkin parçacık sayısı $\widehat{N}_{eff}$, denklem (9) ile hesaplanır:

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} \left(\tilde{w}_k^{(i)}\right)^2} \tag{9}$$

2.2. Genetik Algoritma

Genetik algoritma (GA), 1970'lerde Michigan Üniversitesi'nde profesör olan Holland tarafından ileri sürüldü. Ana fikri, biyolojik organizmaların evrimsel sürecine dayanmaktadır. Seçim, çaprazlama ve mutasyon gibi doğal evrimden ilham alan teknikleri kullanarak çözümler üreten Darwin'in evrim teorisine dayanmaktadır [23]. Her yinelemede, her bireyin (veya çözümün) uygunluk derecesi verilen bir amaç fonksiyonuna göre değerlendirilir. Yüksek uyumlu bireyler hayatta kalmak için daha fazla fırsata sahipken, daha az uyumlu bireyler elenecektir. Seçimden sonra hayatta kalan bireyler (veya ebeveyn bireyler) üreme prosedürüne katılabilir. Çaprazlama operatörü çoğaltma işlemine dahil edilmiştir. Ebeveyn bireyler, yavru bireyleri bir çaprazlama oranına göre üretmek için bilgilerinin bir kısmını değiş tokuş ederler. Aynı zamanda yavru bireylerde de mutasyon meydana gelebilir. Yavru bireyler elde edildikten sonra bir iterasyon tamamlanır. Stokastik bir arama algoritması olarak GA, çeşitli doğrusal olmayan ve çok amaçlı optimizasyon problemlerinde yaygın olarak kullanılır.

Bilgisayar uygulamalarında GA, orijinal olarak bir bireydeki bir kromozomun genlerinin ikili dizi temsilinden türetilen Şema (Schema) Teoremi tarafından yönetilir [24]. Şema Teoremi şu şekilde ifade edilebilir:

$$m(\in, t+1) \geq m(\in, t)\frac{f(\in)}{\bar{f}}\left(1 - p_c\frac{\delta(\in)}{L-1}\right)(1-p_m)^{o(\in)} \tag{10}$$

Burada, $m(\in, t)$ t neslindeki şema ($\in$) sayısı, $f(\in)$ aynı şemaya sahip kromozomların ortalama uygunluğu, $\bar{f}$ tüm popülasyonun ortalama uygunluğu, p_c çaprazlama olasılığı, $\delta(\in)$ bir şemanın uzunluğu, L kromozom uzunluğu, p_m mutasyon olasılığı ve $o(\in)$ bir şemanın sırasıdır.

GA'nın genel prosedürü aşağıdaki gibidir:

1) Başlangıç Popülasyonunun Oluşturulması:

• Optimizasyon problemine uygun bir başlangıç popülasyonu oluşturulur. Başlangıç popülasyonu, potansiyel çözümlerin rastgele veya belirli bir stratejiye göre seçilerek oluşturulabilir.

2) Yineleme:

• Uygunluk Değerinin Hesaplanması: Her bireyin uygunluk değeri, optimizasyon probleminin amaç fonksiyonunu kullanarak hesaplanır.

• Seçim Operatörü ile Ebeveyn Seçimi: Uygunluk değerlerine göre popülasyon içinden ebeveyn bireyler seçilir. Yüksek uygunluk değerine sahip bireylerin seçilme olasılığı daha yüksektir, ancak düşük uygunluk değerine sahip bireylerin de seçilme şansı bulunur. Bu, doğal seleksiyon prensibine benzer bir yaklaşımdır.

• Çaprazlama Operatörü ile Çaprazlama: Seçilen ebeveyn bireyler, genetik materyallerini değiştirerek yeni bireyler oluşturmak için çaprazlama operatörü kullanılır. Çaprazlama, genetik materyallerin karıştırılması yoluyla çeşitliliği artırarak daha iyi çözümlere ulaşmayı hedefler.
• Mutasyon Operatörü ile Mutasyon: Oluşturulan yeni bireylerin genetik materyallerinde rastgele değişiklikler yapmak için mutasyon operatörü kullanılır. Mutasyon, genetik çeşitliliği artırarak popülasyonun çeşitliliğini korur ve potansiyel olarak daha iyi çözümler keşfedilmesini sağlar.
• Yeni Popülasyonun Oluşturulması: Çaprazlama ve mutasyon operatörleri kullanılarak oluşturulan yeni bireyler, eski popülasyon ile birleştirilerek yeni bir popülasyon oluşturulur.
3) Bitiş:
• Sonlandırma koşulu sağlanıyorsa yinelemeyi durdurun, aksi takdirde 2. adıma geçin.

2.3. Genetik Algoritma Tabanlı Optimize Edilmiş Parçacık Filtresi Algoritması

Parçacık filtresini iyileştirmek için genetik algoritmaların kullanılmasının nedeni, genetik algoritmanın, parçacıkların kullanım verimliliğini artırabilen, sonsal olasılık dağılımına yaklaşmak için gereken parçacıkları daha az hale getirebilen ve yeniden örneklemeni getirdiği sorunu önleyebilen benzersiz optimizasyon yeteneğine sahip olmasıdır.
GA hesaplamayı bir dereceye kadar azaltır, algoritmanın gerçek zamanını etkili bir şekilde iyileştirebilir. Ayrıca, genetik operatörler parçacıkların çeşitliliğini etkili bir şekilde artırabildiğinden, parçacık bozulmasını çözebildiğinden algoritmanın doğruluğunu artırabilir, filtre sapması olgusunu etkili bir şekilde önleyebilir ve durum tahmin doğruluğunu geliştirebilir.
Bu makale de önerilen geliştirilmiş algoritmanın belirli adımları aşağıdaki gibi ifade edilebilir:
Adım 1, ölçümün değerini elde ederiz ve uygunluk fonksiyonunu tanımlarız. Geleneksel parçacık filtresi optimalin altında bir önem fonksiyonu kullanır, bu nedenle parçacık örnekleme süreci optimalin altındadır, parçacık filtresi örnekleme sürecini optimize etmek için en son ölçülen değerler örnekleme sürecine dahil edilmiştir, uygunluk fonksiyonunu (11) eşitliğiyle tanımlarız:

$$y_k\,(fitness) = \exp\left[-\frac{1}{2R_k}\left(y_k - \widehat{y^i}_{(k|k-1)}\right)^2\right] \tag{11}$$

Burada R_k, ölçüm gürültü varyansıdır; y_k en son ölçülen değerdir; $\widehat{y^i}_{(k|k-1)}$ ise tahmin değeridir.
Adım 2, ilklendirme: k = 0 anında önem yoğunluk fonksiyonundan N adet parçacık alınır. Bu parçacıkları belirtmek için $\{x^i{}_{0:k}, w^i{}_k\}_{i=1}^N$ ifadesini kullanıyoruz. Her örneğin başlangıç ağırlıklarını $\left\{w_k^i = \frac{1}{N},\ i = 1,2,\dots,N\right\}$ olarak ve önem yoğunluk fonksiyonunun transferinin önsel olasılığı $x_k^i \in q\left(x_k^i \middle| x_{k-1}^i, y_k\right) = p\left(x_k^i \middle| x_{k-1}^i\right)$ şeklinde tanımlarız.
Adım 3, her bir parçacık durum modeline dahil edilir.
Adım 4, ağırlıkların güncellemesi: En son ölçülen değerlere göre, mevcut parçacık ağırlıklarını (12) eşitliğiyle güncelleriz.

$$w_k^i = w_{k-1}^i \exp\left[-\frac{1}{2R_k}\left(y_k - \widehat{y^i}_{(k|k-1)}\right)^2\right] \tag{12}$$

Adım 5, ağırlıkları normalleştirme (13) ile gerçekleştirilir. Ardından (14) eşitliğine bakılır ve $N_{eff} < N_{eşik}$ ise Adım 6'ya geçilir, değilse Adım 7'den devam edilir.

$$\bar{w}_k^i = {w_k^i}\Big/{\sum_{i=1}^N w_k^i} \tag{13}$$

$$N_{eff} = {1}\Big/{\sum_{i=1}^N \left(w_k^{(i)}\right)^2} < N_{eşik} \tag{14}$$

Adım 6, genetik işlemlerin uygulanmasıdır.
1) Çaprazlama (Crossover)
Parçacık kümesinden rastgele iki parçacık $(x_k^m, x_k^n)_{m,n=1}^{N_s}$ seçilir. Aşağıdaki iki denkleme (15) ve (16) göre çaprazlama işlemi gerçekleşir.

$$\tilde{x}_k^m = \alpha x_k^m + (1-\beta) x_k^n \tag{15}$$

$$\tilde{x}_k^n = \beta x_k^m + (1-\alpha) x_k^n \tag{16}$$

Eğer $\tilde{x}_k^m$, çaprazlama ilkesi $p(y_k|\tilde{x}_k^m) > \max\{p(y_k|\tilde{x}_k^m), p(y_k|\tilde{x}_k^n)\}$ şartını sağlıyorsa kabul edilir, aksi takdirde olasılığı $p(y_k|\tilde{x}_k^m)/\max\{p(y_k|\tilde{x}_k^m), p(y_k|\tilde{x}_k^n)\}$ olan parçacıklar kabul edilir. $\tilde{x}_k^n$ parçacığını kabul etme veya etmeme durumu $\tilde{x}_k^m$ ile aynı işlemlerin ardından belirlenir.

2) Mutasyon (Mutation)

Parçacık kümesinden rastgele bir parçacık $(x_k^j)_{j=1}^{N_s}$ seçilir. Daha sonra aşağıdaki formüle (17) göre mutasyon işlemi yapılır.

$$\tilde{x}_k^j = x_k^j + \eta \tag{17}$$

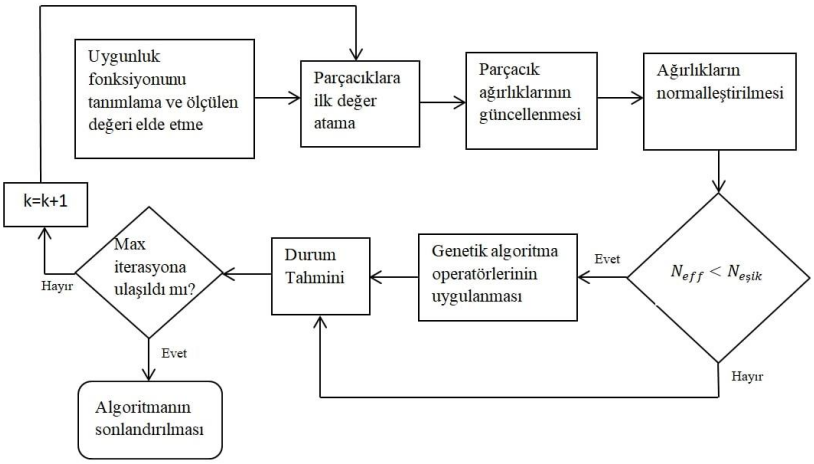
Varyans kriteri $p(y_k|\tilde{x}_k^i) > p(y_k|\tilde{x}_k^j)$ ise, $\tilde{x}_k^j$ parçacığını kabul edilir. Aksi takdirde, olasılığı $p(y_k|\tilde{x}_k^i)/p(y_k|\tilde{x}_k^j)$ olan parçacıkları kabul edilir.

Yukarıdaki yöntemler yeni bir $\{\tilde{x}_k^i, \overline{w}_k^i\}_{i=1}^{N_s}$ parçacık kümesi elde etmek için çaprazlama ve mutasyon işlemlerini gerçekleştirir.

Adım 7'de durum tahminleri (18)'de verilen denklem doğrultusunda gerçekleştirilir.

$$x_i = \sum_{i=1}^{N} w_k^i x_k^i \tag{18}$$

Adım 8, algoritmanın devam edip etmeyeceğinin kontrolünü yapar. k anının hedefin son iterasyonu olup olmadığına karar verir ve son iterasyon ise algoritma sonlandırılır; eğer değilse, k = k +1 şeklinde artırılır, sonra yinelemeli olarak Adım 2'ye döner.



Şekil 1. Önerilen GA tabanlı PF algoritmasının akış diyagramı

3. BULGULAR VE TARTIŞMA

Bu makalede dinamik (statik olmayan) büyüme modelini ve modeli simüle etmek için MATLAB (R2015b) yazılımı kullanılmıştır. PF'yi GA tabanlı PF ile karşılaştırmak için tek boyutlu doğrusal olmayan referans modeli [25] ele alınmıştır. Sistem süreç modeli (19) ve durum modeli (20) denklemleri aşağıdaki gibidir:

$$x(t) = 0.5x(x-1) + \frac{25x(t-1)}{1+[x(t-1)]^2} + 8\cos[1.2(t-1)] + w(t) \tag{19}$$

$$y(t) = \frac{x(t)^2}{20} + v(t) \tag{20}$$

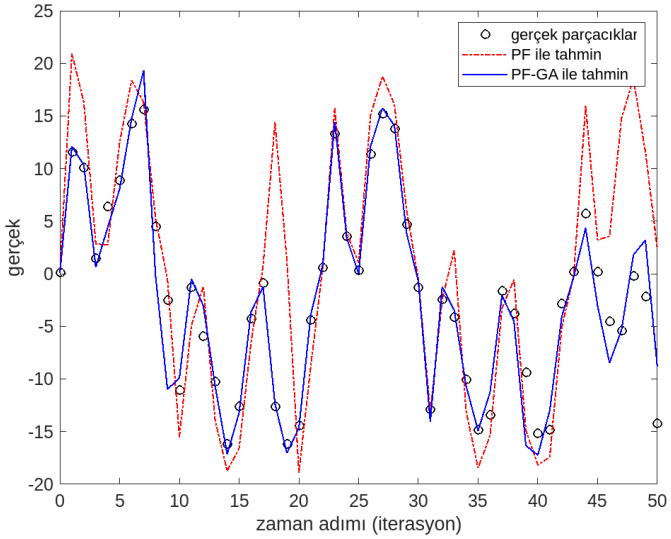
Ölçüm gürültü varyansı R = 1, sistem gürültü varyansı ise sırasıyla Q = 1, 3, 9 seçilmiştir. Genetik operatörlerin çaprazlama olasılığı p_c = 0.7, mutasyon olasılığı p_m = 0.3 olarak belirlenmiştir. İterasyon sayısı sırasıyla 10, 20, 50 ve simüle etmek için bu makalede önerilen GA tabanlı PF algoritmasında parçacık sayısını N = 10 seçilerek sonuçlar karşılaştırılmıştır. Yapılan karşılaştırmalarda Ortalama Kare Hata (MSE) (21) ve Kök Ortalama Karesel Hata (RMSE) (22) ölçütleri performans değerlendirmesi için kullanılmıştır.

Ortalama Karesel Hata (MSE): Öngörülen ve gerçek sonuçlar arasındaki kare farkının ortalamasını ölçer. x_n beklenen değerleri, $\tilde{x}_n$ tahmin edilen değerleri ve N örnek sayısını göstermektedir. MSE'deki kare alma, pozitif sonuçlar verir [26].
Kök Ortalama Karesel Hata (RMSE), MSE'nin bir uzantısıdır. Karekök hatası, gözlemlenen verilerin orijinal birim ölçeğini korur. Formüldeki x_n beklenen değerleri, $\tilde{x}_n$ tahmin edilen değerleri ifade eder. RMSE, bir modelin uyum kalitesini değerlendirmek için de yararlıdır [25].

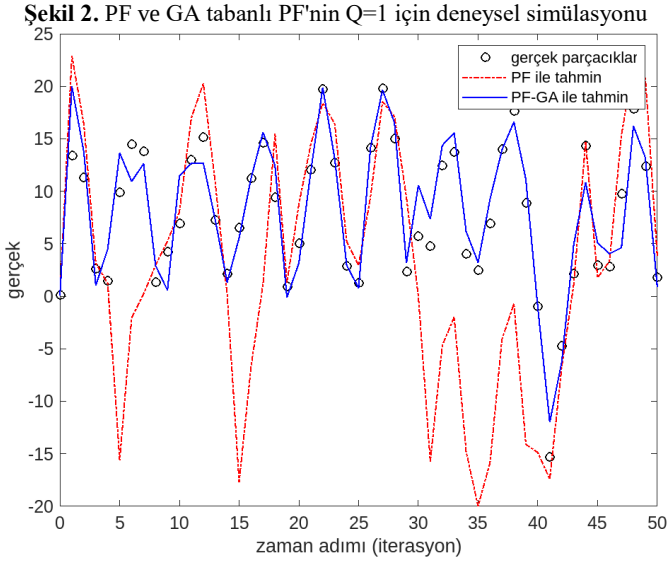
$$MSE = \frac{1}{N}\sum_{i=1}^{n}(x_n - \tilde{x}_n)^2 \tag{21}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(x_n - \tilde{x}_n)^2} \tag{22}$$

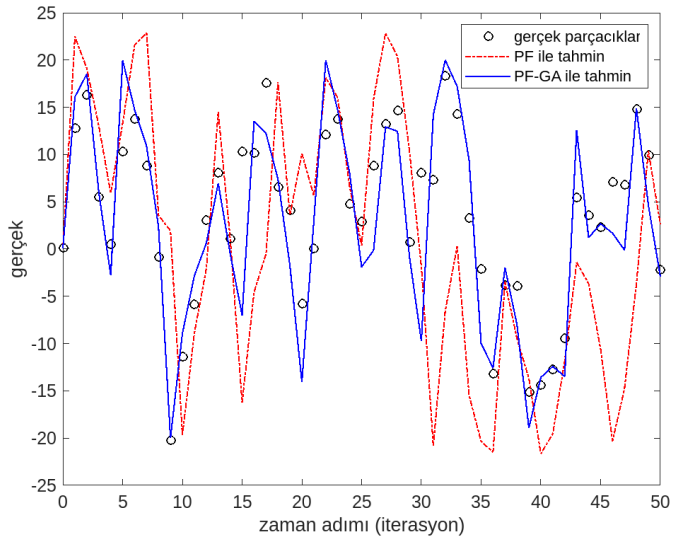
Sistemin ürettiği parçacıkların sayısı ne kadar büyükse, tahmin o kadar iyi olmaktadır, ancak o kadar fazla hesaplama karmaşası ortaya çıkar. Şekil 2, PF ve GA tabanlı PF'nin sistem gürültü varyansı 1 ve iterasyon sayısının 50 olduğu deneysel simülasyonudur. Şekil 3, sistem gürültü varyansı 3 ve iterasyon sayısının 50 olduğu, Şekil 4 ise sistem gürültü varyansı 9 ve iterasyon sayısının 50 olduğu simülasyon çıktısıdır.



Şekil 2. PF ve GA tabanlı PF'nin Q=1 için deneysel simülasyonu



Şekil 3. PF ve GA tabanlı PF'nin Q=3 için deneysel simülasyonu



Şekil 4. PF ve GA tabanlı PF'nin Q=9 için deneysel simülasyonu

50, 20 ve 10 iterasyon değerleri ile Q=1, 3 ve 9 için yapılan deneyin MSE ve RMSE verileri aşağıdaki gibidir:

Tablo 1. 50 İterasyonlu Filtreleme Performans İstatistikleri

İterasyon sayısı 50		
Sistem gürültü varyansı (Q)	**MSE**	**RMSE**
1	55.507649	7.450346
3	141.250509	11.884886
9	151.019695	12.289007

Tablo 2. 20 İterasyonlu Filtreleme Performans İstatistikleri

İterasyon sayısı 20		
Sistem gürültü varyansı (Q)	**MSE**	**RMSE**
1	59.345054	7.703574
3	47.839122	6.916583
9	164.223002	12.814952

Tablo 3. 10 İterasyonlu Filtreleme Performans İstatistikleri

İterasyon sayısı 10		
Sistem gürültü varyansı (Q)	**MSE**	**RMSE**
1	166.534842	12.904838
3	90.719492	9.524678
9	374.569374	19.353795

Şekil 2, 3 ve 4 PF ve GA tabanlı optimize edilmiş PF'nin simülasyon grafikleridir, GA tabanlı PF'nin geleneksel PF'den daha doğru tahmin sonuçları elde ettiği açıktır. Tablo 1, 2 ve 3'de gösterilen deneysel verileri farklı gürültülü ortamda, GA tabanlı PF'nin parçacık bozulmasını engellediğini ve parçacık çeşitliliğini artırmada etkili olduğunu gösterir. Tablolar karşılaştırıldığında gürültüde artış olması durumunda parçacık bozulmasını hala engellediği ve parçacık çeşitliliğini artırdığı görülmektedir. Minimum RMSE değerleri, tahmininin en yüksek doğrulukta olduğunu göstermektedir. Sonuç tablolarına bakıldığında minimum RMSE değerinin sistem gürültü varyans değerinin 3 olduğu ve iterasyon sayısının 20 olduğu filtreleme sonucunda elde edildiği gözlenmektedir.

4. SONUÇLAR

Bu makalede, Parçacık filtresini iyileştirmek için Parçacık filtresine Genetik algoritma dahil edilmiştir. Geleneksel yeniden örnekleme yöntemlerinin getirdiği parçacık yoksullaşma sorununu en aza indirmek için genetik algoritmanın çaprazlama ve mutasyon işlemlerini kullanıldı. Böylece parçacık yoksullaşması sorunundan kaçınılmış, etkili parçacıkların sayısı artırılmış ve parçacık kullanımını iyileştirmiş olduk. Algoritmanın kesinliğini sağlamak için Genetik algoritma optimizasyonu ile geliştirilmiş Parçacık filtresi sürecinde etkili bir parçacık eşiği belirledik. Aynı zamanda, algoritmanın verimliliği ve gerçek zamanlılığı iyileştirildi. MATLAB (R2015b) programında gerçekleştirilen simülasyon sonuçları, bu makalede önerilen GA tabanlı optimize edilmiş Parçacık filtresi algoritmasının performansının, geleneksel Parçacık filtresi algoritmasından çok daha iyi olduğunu göstermektedir. Makalede önerilen algoritmanın literatürde var olan Genetik Yeniden Örneklemeye dayalı geliştirilmiş Parçacık filtresi [13] algoritmasıyla karşılaştırıldığında daha verimli tahmin sonuçları elde ettiği gözlenmiştir.

ÇIKAR ÇATIŞMASI

Herhangi bir çıkar çatışması bulunmamaktadır.

ETİK

Bu makalenin yayınlanmasında herhangi bir etik sorun bulunmamaktadır.

KAYNAKLAR

[1] Kalman, R. E. "A new approach to linear filtering and prediction problems". 1960.

[2] Ristic, B., Arulampalam, S., & Gordon, N. "*Beyond the Kalman filter: Particle filters for tracking applications*". Artech house. 2003.

[3] Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. "A tutorial on particle filters for online nonlinear/nongaussian bayesian tracking". Bayesian Bounds Param. *Estim. Nonlinear Filter. Track*, *50*, 723-737. 2007.

[4] Gordon, N. J., Salmond, D. J., & Smith, A. F. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In *IEE proceedings F (radar and signal processing)* (Vol. 140, No. 2, pp. 107-113). IET Digital Library. 1993, April

[5] Yang, J., Cui, X., Li, J., Li, S., Liu, J., & Chen, H., "Particle filter algorithm optimized by genetic algorithm combined with particle swarm optimization". Procedia Computer Science, 187, 206-211. 2021.

[6] Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., & Nordlund, P. J. "Particle filters for positioning, navigation, and tracking". *IEEE Transactions on signal processing*, *50*(2), 425-437. 2002.

[7] Shenoy, A. V., Prakash, J., Prasad, V., Shah, S. L., & McAuley, K. B. "Practical issues in state estimation using particle filters: Case studies with polymer reactors". *Journal of Process Control*, *23*(2), 120-131. 2013.

[8] Das, S., Kale, A., & Vaswani, N. "Particle filter with a mode tracker for visual tracking across illumination changes". *IEEE Transactions on Image Processing*, *21*(4), 2340-2346. 2011.

[9] Zhang, Q. B., Wang, P., & Chen, Z. H., "An improved particle filter for mobile robot localization based on particle swarm optimization. Expert Systems with Applications", 135, 181-193. 2019.

[10] Li, M., Pang, B., He, Y., & Nian, F. "Particle Filter Improved by Genetic Algorithm and Particle Swarm Optimization Algorithm". *J. Softw.*, *8*(3), 666-672. 2013.

[11] Park, S., Hwang, J. P., Kim, E., & Kang, H. J. "A new evolutionary particle filter for the prevention of sample impoverishment". *IEEE Transactions on Evolutionary Computation*, *13*(4), 801-809. 2009.

[12] Wang, W., Tan, Q. K., Chen, J., & Ren, Z. "Particle filter based on improved genetic algorithm resampling". In *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)* (pp. 346-350). IEEE. August, 2016.

[13] Zhao, B., Hu, J. W., & Ji, B. "An improved particle filter based on genetic resampling". In *2015 International Conference on Automation, Mechanical Control and Computational Engineering* (pp. 1353-1358). Atlantis Press. April, 2015.

[14] Han, H., Ding, Y. S., Hao, K. R., & Liang, X. "An evolutionary particle filter with the immune genetic algorithm for intelligent video target tracking". *Computers & Mathematics with Applications*, *62*(7), 2685-2695. 2011.

[15] Walia, G. S., & Kapoor, R. "Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search". *Expert Systems with Applications*, *41*(14), 6315-6326. 2014.

[16] J. P. Zhong, Y. F. Fung, "A biological inspired improvement strategy for particle filters," Industrial Technology, 2009, ICIT 2009, pp. 1-6, 10-13, Feb. 2009.

[17] Y. Qiao, Q. Zhang and J. Zhang, "A fault predication algorithm based on artificial immune particle filter," Applied Mechanics and Materials, vol. 44-47, pp. 3459-3463, 2010.

[18] X. Liang, J. Feng, Q. Li, T. Lu and B. Li, "A swarm intelligence optimization for particle filter," Intelligent Control and Automation, 2008. WCICA 2008., pp. 1986-1991, 25-27, June 2008.

[19] Gao, M. L., Li, L. L., Sun, X. M., Yin, L. J., Li, H. T., & Luo, D. S. "Firefly algorithm (FA) based particle filter method for visual tracking". *Optik*, *126*(18), 1705-1711. 2015

[20] Dilmen, H. & Talu, M. F. "Yapısal Özellikleri Kullanan Parçacık Filtresi İle Uzun Süreli Nesne Takibi". Gazi University Journal of Science Part C: Design and Technology, 5 (1), 107-118. Retrieved from https://dergipark.org.tr/tr/pub/gujsc/issue/28467/303419, 2017.

[21] Sadegh Moghadasi, S., & Faraji, N. "An efficient target tracking algorithm based on particle filter and genetic algorithm". *International Journal of Engineering*, *32*(7), 915-923. 2019.

[22] Gao, M., & Zhang, H. "Sequential Monte Carlo methods for parameter estimation in nonlinear state-space models". *Computers & Geosciences*, *44*, 70-77. 2012.

[23] Zhang, J., Pan, T. S., & Pan, J. S. "A parallel hybrid evolutionary particle filter for nonlinear state estimation". In *2011 First International Conference on Robot, Vision and Signal Processing* (pp. 308-312). IEEE. November, 2011.

[24] Kwok, Ngai Ming, Gu Fang, and Weizhen Zhou. "Evolutionary particle filter: re-sampling from the genetic algorithm perspective ". 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2005.

[25] Doucet, A., Andrieu, C., & Fitzgerald, W. (1998, September). Bayesian filtering for hidden Markov models via Monte Carlo methods. In *Neural Networks for Signal Processing VIII. Proceedings of the 1998 IEEE Signal Processing Society Workshop (Cat. No. 98TH8378)* (pp. 194-203). IEEE.

[26] Hodson, T. O. "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not". *Geoscientific Model Development*, *15*(14), 5481-5487. 2022.