

Darbe-Doppler Hava Radarı için Parazit Yankı Tespit Algoritmaları

Clutter Detection Algorithms for Airborne Pulse-Doppler Radar

Ahmet Güngör, Sinan Gezici

Elektrik ve Elektronik Mühendisliği Bölümü
Bilkent Üniversitesi
{agungor, gezici}@ee.bilkent.edu.tr

Özetçe

Parazit yankı tespiti, hedef tespiti için çok önemli bir ön aşamadır. Gerçekçi arazi modelleri ve hareketli platformlar göz önüne alındığında, parazit yankı sıfır Doppler frekansı etrafında bulunmayabilmektedir. Bu bildiri, mesafe-Doppler matrisi elemanlarını kullanarak, parazit yankı tespit edilmesine yönelik iki algoritma ve performans analizleri sunulmuştur. Algoritmalarından birincisi daha yüksek hata oranına ancak daha düşük işlem karmaşıklığına, ikincisi ise daha düşük hata oranına ancak daha yüksek işlem karmaşıklığına sahiptir. Algoritmalar mesafe-Doppler matrisi elemanlarını doğrusal olmayan süzgeçlerden geçirerek parazit yankının konumunu tespit etmektedir.

Abstract

Clutter detection is an important stage of target detection. Clutter may not always appear around zero Doppler frequency when realistic terrain models and moving platforms are considered. Two algorithms developed for clutter detection using range-Doppler matrix elements and their performance analysis are presented in this paper. The first algorithm has higher error rates but lower computational complexity whereas the second one has lower error rates but higher computational complexity. The algorithms detect clutter position by filtering range-Doppler matrix elements via non-linear filters.

1. Giriş

Günümüzde hedef olarak tanımlanan çeşitli nesnelerin konumunu ve hızını tespit edilebilmek için karmaşık radar sistemleri kullanılmaktadır. Yaygın olarak kullanılmakta olan bir radar tipi de darbe-Doppler hava radarıdır. Gönderilen radar sinyalinin hedeften yansması hedef sinyalini, hedef haricindeki nesnelere yansması ise parazit yankı sinyalini oluşturmaktadır. Bu sinyaller işlenerek, sinyallerin yansıdığı nesnelerin konumunu ve hızını içeren mesafe-Doppler matrisi verileri elde edilmektedir. Parazit yankı, hedef tespitini zorlaştıran bir unsurdur. Hava radarlarında, parazit yankıya en fazla arazi şekilleri sebep olmaktadır [1-6].

Bazı hedef tespit algoritmaları, parazit yankı konumu bilgisine ihtiyaç duymaktadır. Parazit yankı konumu tespit edildiğinde, parazit yankıya uzak bölgelerde ve parazit yankıya komşu bölgelerde hareketli hedef tespit algoritmaları,

parazit yankı içerisinde ise durağan hedef tespit algoritmaları kullanılabilir.

Literatürde mesafe-Doppler matrisi üzerinde parazit yankı tespiti yapılmasında kullanılan bir algortmada, mesafe-Doppler matrisi elde edildikten sonra arkaplan gürültüsünün ortalama değeri kullanılarak bir eşik değeri seçilmektedir. Mesafe-Doppler matrisindeki her bir indisin değeri, seçilen eşik değeri ile karşılaştırılmaktadır. Eğer mesafe-Doppler matrisinin herhangi bir indisindeki değer seçilen eşik değerinden büyükse, o indis işaretlenmektedir. Bütün mesafe-Doppler matrisi bu şekilde taranmaktadır. Birkaç komşu indisten fazla komşu indis işaretlenmiş ise, o bölge parazit yankı olarak tanımlanmaktadır [7]. Diğer bir algortmada ise, mesafe hücrelerindeki yüksek değerlere sahip indisler parazit yankı olarak tanımlanmaktadır [8].

Bu çalışmada, birincisi daha yüksek hata oranına ancak daha düşük işlem karmaşıklığına, ikincisi ise daha düşük hata oranına ancak daha yüksek işlem karmaşıklığına sahip, en büyük ve ortanca değerlerin seçildiği doğrusal olmayan süzgeçler kullanan iki algoritma önerilmiştir. Kullanılan ortanca süzgecinin boyutu esnek tutulmaktadır.

2. Parazit Yankı Tespit Algoritmaları

Parazit yankı tespiti, mesafe-Doppler matrisi A 'nın elemanları kullanılarak yapılmaktadır. Bu çalışmada önerilen iki algoritma aşağıda açıklanmaktadır.

2.1. Algoritma 1

Bu algortmada her mesafe hücresindeki maksimum büyüklükteki Doppler elemanın indisinin, bahsedilen mesafe hücresindeki parazit yankı hücresinin indisini verdiği kabul edilmektedir.

$$C_r = \arg \max_v |A(r, v)|, r = 1, 2, \dots \quad (1)$$

Burada C_r , r 'ninci mesafe hücresindeki parazit yankının indisini belirtmektedir.

2.2. Algoritma 2

Bu algortmada parazit yankının tespit edileceği mesafe hücresindeki ve bu mesafe hücresine komşu birkaç mesafe hücresindeki maksimum büyüklükteki Doppler elemanlarının indisleri bulunup, ortanca süzgeçinden geçirilmektedir. Süzme

sonucunda elde edilen indis, bahsedilen mesafe hücresindeki parazit yankının bulunduğu indis olarak kabul edilmektedir.

$$t(r, i) = \arg \max_v |A(r+i, v)| \quad (2)$$

$$C_r = \text{ortanca_süzgeci} \{t(r,-i), \dots, t(r,0), \dots, t(r,i)\}, r = 1, 2, \dots \quad (3)$$

3. Algoritmaların Performans Analizleri

Algoritmaların performansları hesaplanırken, A matrisinin sadece gürültü ve parazit yankı değerlerini içerdiği ve parazit yankının her mesafe hücresinde sadece bir Doppler indisinde bulunduğu kabul edilmektedir. Algoritmaların bir mesafe hücresinde bulunan parazit yankının indisini yanlış bulma olasılıkları aşağıda verilmektedir.

$P_N(z)$ gürültünün olasılık yoğunluk fonksiyonunu, $F_C(z)$ ve $F_N(z)$ ise sırasıyla parazit yankının ve gürültünün birikimli olasılık yoğunluk fonksiyonlarını vermektedir. Her mesafe hücresindeki gürültünün n tane Doppler indisinde bulunduğu ve ortanca süzgecinin uzunluğunun $M=2K+1$ olduğu kabul edilmektedir. ε , parazit yankının yerini belirlemektedir. Eğer parazit yankının mesafe hücresinin tam ortasında yer aldığı biliniyorsa ε 'un değeri 0.5 alınmalıdır.

Genel çözümler elde edildikten sonra numerik hesaplamalar yapabilmek için parazit yankı dağılımı Rice dağılımı olarak, gürültü dağılımı ise Rayleigh dağılımı olarak kabul edilmektedir. Grafiklerde $n=10$ iken ve gürültünün gücü sabitken hata yapma olasılığının parazit yankı gücü $E[|C|^2]$ değişirkenki değerleri verilmektedir.

3.1. Algoritma 1

Bu algoritmanın hata yapma olasılığını veren denklemler aşağıdaki şekilde hesaplanmaktadır.

$$P_e = 1 - P(C > \max(N_i, i = 1, \dots, n)) \quad (4)$$

$$\tilde{N} = \max(N_1, \dots, N_n) \quad (5)$$

$$P_e = 1 - \int P(C > z) P_{\tilde{N}}(z) dz \quad (6)$$

N_i 'ler özdeş ve bağımsız dağılıma sahip olduğunda,

$$P_{\tilde{N}}(z) = n[F_N(z)]^{n-1} P_N(z) \quad (7)$$

ve

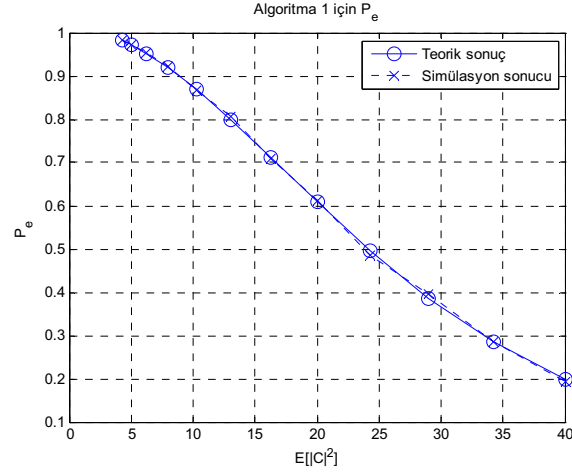
$$P_e = 1 - \int (1 - F_C(z)) n[F_N(z)]^{n-1} P_N(z) dz \quad (8)$$

ifadeleri elde edilmektedir.

Algoritma 1 için bulunan teorik formülün simülasyon sonuçları ile uyum içerisinde olduğu Şekil-1'de görülmektedir.

3.2. Algoritma 2

Bu algoritmanın hata yapma olasılığını veren denklemler aşağıdaki şekilde elde edilmektedir.



Şekil-1: Algoritma 1 için hata yapma olasılığının teorik ve simülasyon sonuçları.

$$c = \int (1 - F_C(z)) n[F_N(z)]^{n-1} P_N(z) dz \quad (9)$$

$$x = (1 - c)\varepsilon \quad (10)$$

$$y = (1 - c)(1 - \varepsilon) \quad (11)$$

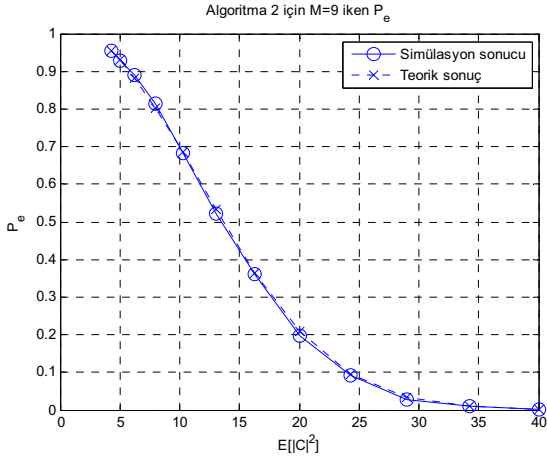
$$P_e = P_{ex} + P_{ey} \quad (12)$$

$$P_{ex} = \frac{(2K+1)!}{(K+1)!} x^{K+1} \sum_{a=0}^K \frac{c^a y^{(K-a)}}{a!(K-a)!} + \frac{(2K+1)!}{(K+2)!} x^{K+2} \sum_{a=0}^{K-1} \frac{c^a y^{(K-1-a)}}{a!(K-1-a)!} + \dots + \frac{(2K+1)!}{(2K+1)!} x^{2K+1} \sum_{a=0}^0 \frac{c^a y^{(0-a)}}{a!(0-a)!} \quad (13)$$

$$P_{ey} = \frac{(2K+1)!}{(K+1)!} y^{K+1} \sum_{a=0}^K \frac{c^a x^{(K-a)}}{a!(K-a)!} + \frac{(2K+1)!}{(K+2)!} y^{K+2} \sum_{a=0}^{K-1} \frac{c^a x^{(K-1-a)}}{a!(K-1-a)!} + \dots + \frac{(2K+1)!}{(2K+1)!} y^{2K+1} \sum_{a=0}^0 \frac{c^a x^{(0-a)}}{a!(0-a)!} \quad (14)$$

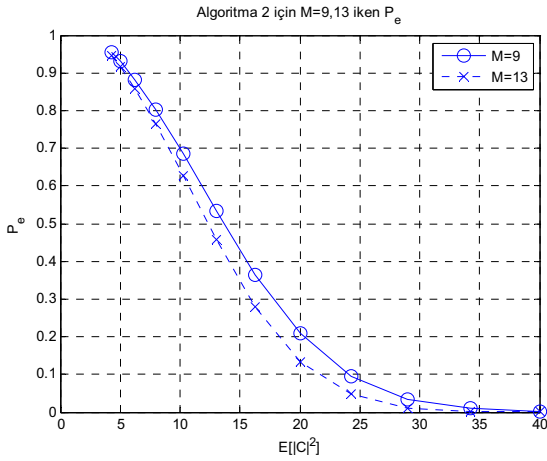
Yukarıdaki denklemler düzenlendiğinde, aşağıdaki ifadeye ulaşılmaktadır.

$$P_e = (2K+1)! \left[\sum_{a=K+1}^{2K+1} \frac{x^a}{a!} \left(\sum_{b=0}^{2K+1-a} \frac{c^b y^{(2K+1-a-b)}}{b!(2K+1-a-b)!} \right) \right] + (2K+1)! \left[\sum_{a=K+1}^{2K+1} \frac{y^a}{a!} \left(\sum_{b=0}^{2K+1-a} \frac{c^b x^{(2K+1-a-b)}}{b!(2K+1-a-b)!} \right) \right] \quad (15)$$



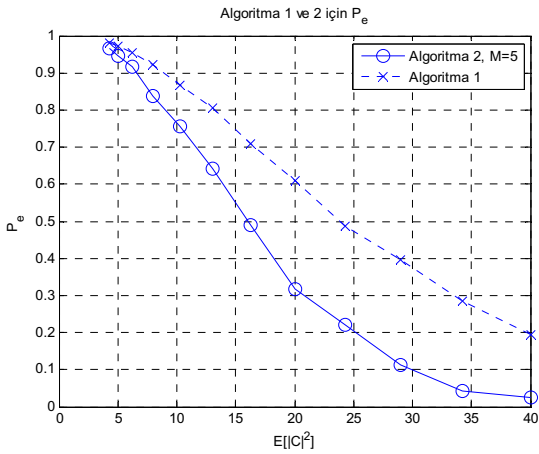
Şekil-2: Algorithm 2 için hata yapma olasılığının teorik ve simülasyon sonuçları.

Algorithm 2 için bulunan teorik formülün simülasyon sonuçları ile uyum içerisinde olduğu Şekil-2'de görülmektedir.



Şekil-3: Algorithm 2'nin ortanca süzgecinin değişik uzunlukları için performansı.

Algorithm 2'de kullanılan ortanca süzgecinin uzunluğunun artırılmasının, algoritma performansını artırdığı Şekil-3'te görülmektedir.



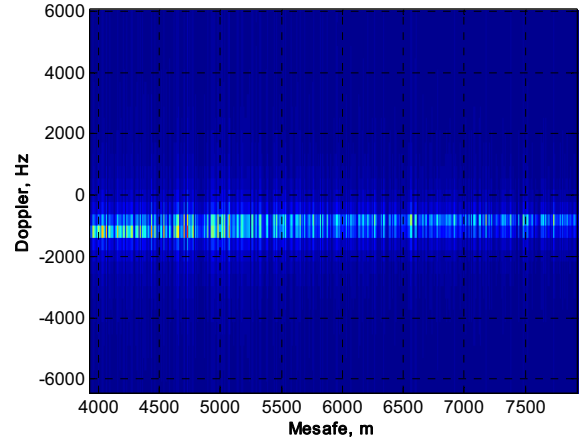
Şekil-4: Algorithm 1 ve 2'nin karşılaştırmalı performansları.

Şekil-4'te işlem karmaşıklığı daha yüksek olan algoritma 2'nin performansının, işlem karmaşıklığı daha düşük olan algoritma 1'e göre daha iyi olduğu görülmektedir.

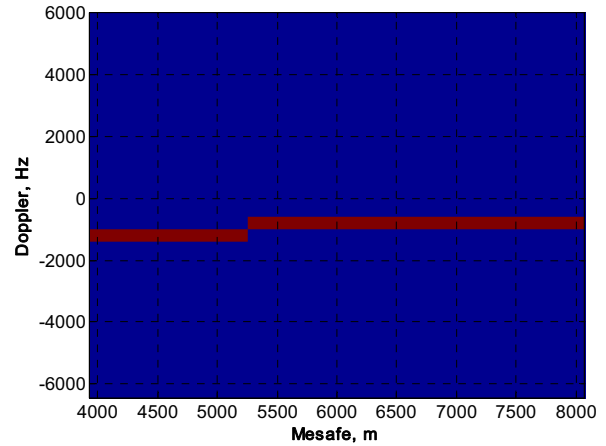
4. Gerçekçi Veriler Üzerindeki Performans

Gerçekçi arazi modelleri ve hareketli platformlar göz önüne alınarak elde edilen bir mesafe-Doppler matrisi örneği Şekil-5'te görülmektedir [9]. Parazit yankı -1000 Hz civarında bulunmaktadır.

Gerçekçi veriler üzerine, performansının daha iyi olduğu görülen algoritma 2 uygulanmaktadır. Aşağıdaki örneklerden Şekil-5'te, parazit yankı sinyalinin mesafe-Doppler düzlemindeki konumu görülmektedir. Şekil-6'da ise tespit edilen parazit yankının konumu görülmektedir. İki şekil arasında göz ile görülür bir uyumluluk bulunmaktadır.



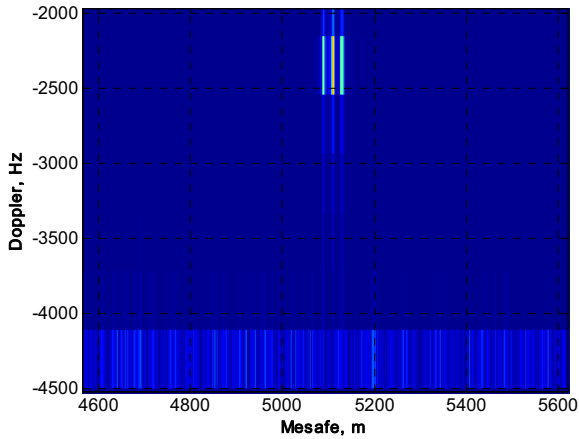
Şekil-5: Gerçekçi parazit yankının mesafe-Doppler düzlemindeki görünümü.



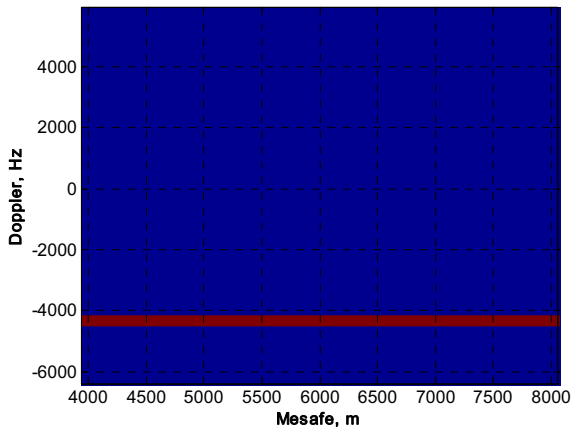
Şekil-6: Tespit edilen parazit yankının mesafe-Doppler düzlemindeki görünümü.

Bu algoritma için parazit yankı tespitinde en zor durumu, birden fazla ve birbirlerine yakın mesafedeki hareketli hedeflerin bulunduğu durum oluşturmaktadır. Bu durumda

hedeflerin bulunduğu mesafe hücresindeki maksimum değerler hedef kaynaklı olacağı için algoritmanın hedefleri parazit yankı olarak algılama riski bulunmaktadır. Bu sebepten dolayı algortima, her mesafe hücresi kullanılarak değil, her N_p mesafe hücresinden biri kullanılarak uygulanacak şekilde değiştirilmiştir. Yapılan bu değişiklik sayesinde, bahsedilen risk büyük ölçüde giderilmektedir. Şekil-7 ve 8'de de görüldüğü üzere, aralıklı olarak hareket eden üç hedefin bulunduğu durumda da parazit yankı tespit algoritması hatasız bir şekilde çalışmaktadır.



Şekil-7: Aralıklı olarak hareket eden hedeflerin ve parazit yankının bulunduğu mesafe-Doppler düzleminin görünümü.



Şekil-8: Aralıklı olarak hareket eden hedeflerin ve parazit yankının bulunduğu ortamda tespit edilen parazit yankının mesafe-Doppler düzlemindeki görünümü.

5. Sonuçlar

Bu bildiride, mesafe-Doppler matrisi elemanlarını kullanarak, parazit yankı tespit edilmesine yönelik iki algoritma ve performans analizleri sunulmuştur. Algoritmalarından ilki bir tane doğrusal olmayan süzgeç kullanırken, ikincisi iki tane doğrusal olmayan süzgeç kullanılmaktadır. İşlem karmaşıklığı daha yüksek olan ikinci algoritmanın hatalı tespit oranını, işlem karmaşıklığı daha düşük olan birinci algoritmaya göre daha düşük tuttuğu ve

gerçekçi veriler üzerine uygulandığında da parazit yankıyı başarıyla tespit edebildiği gözlemlenmiştir.

6. Kaynakça

- [1] C. A. Wiley. "Pulsed Doppler radar methods and apparatus." U.S. Patent 3 196 436, July 20, 1965.
- [2] G.W. Stimson. *Introduction to Airborne Radar*. SciTech Publishing, 1998.
- [3] M. A. Richards. *Fundamentals of Radar Signal Processing*. McGraw-Hill, 2005.
- [4] C. E. Cook and M. Bernfeld. *Radar Signals An Introduction to Theory and Application*. London: Artech House, 1993.
- [5] N. Levanon and E. Mozeson. *Radar Signals*. New York: Wiley, 2004.
- [6] D. A. Shnidman. "Radar Detection in Clutter." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 1056-1067, July 2005.
- [7] E. C. Pearson, Jr. and W. Nelson. "Method of tracking target in presence of clutter." U.S. Patent 4 559 537, Dec. 17, 1985.
- [8] S. J. Clancy and R. A. Phillips. "Method and system for tracking targets in a pulse Doppler radar system." U.S. Patent 4 450 446, May 22, 1984.
- [9] P. Martz. "Generating random fractal terrain." Internet: <http://www.gameprogrammer.com/fractal.html>, 1996.