

Pasif Sensör ile Gemi Takibi

MTH 407 - Otomatik Hedef İzleme Temelleri

Civan Rumet Arı

Yaren Sakarya

13 Haziran 2025

Özet

Bu rapor, çoklu sensör kullanarak sadece açı ölçümleri ile gemi takibi yapan bir Genişletilmiş Kalman Filtresi (EKF) uygulamasını sunmaktadır. Proje, gerçek zamanlı sensör simülasyonu, Doğrusal Olmayan En Küçük Kareler (NLS) ile başlangıç tahmin ve EKF ile konum takibi algoritmalarını içermektedir. Sistem C dilinde geliştirilmiş olup, bearing ölçümlerini dosyadan okuyarak gerçek zamanlı işleme yapmaktadır. Gauss-Newton optimizasyonu ile başlangıç konumu belirlenmekte, ardından EKF predict/update döngüsü ile hedefin hareketi takip edilmektedir. Sonuçlar MATLAB ile görselleştirilmekte ve sistem performansı analiz edilmektedir.

İçindekiler

1	Giriş	3
1.1	Proje Amacı	3
1.2	Problem Tanımı	3
1.3	Yaklaşım	3
2	Matematiksel Model	3
2.1	Durum Vektörü ve Hareket Modeli	3
2.1.1	Durum Uzayı Gösterimi	3
2.1.2	Sürekli Zaman Dinamik Modeli	3
2.1.3	Ayrık Zaman Durum Geçiş Matrisi	4
2.1.4	Süreç Gürültüsü Kovaryans Matrisi	4
2.2	Doğrusal Olmayan Ölçüm Modeli	4
2.2.1	Bearing Ölçüm Fonksiyonu	4
2.2.2	Ölçüm Jacobian Matrisi	4
3	Doğrusal Olmayan En Küçük Kareler ile Başlangıç Tahmini	4
3.1	Problem Formülasyonu	4
3.2	Gauss-Newton Algoritması	5
3.2.1	Residual Vektörü	5
3.2.2	Jacobian Matrisi	5
3.2.3	Gauss-Newton Güncelleme Adımı	5

4	Geniřletilmiş Kalman Filtresi Uygulaması	5
4.1	Predict Adımı	5
4.2	Update Adımı	5
4.3	Zaman Mantığı	5
5	Simölasyon Kurulumu	6
5.1	Gerçek Gemi Hareketi	6
5.2	Sensör Düzeni	6
5.3	Radar Tarama Açıklaması	6
6	Sonuçlar ve Görselleřtirme	7
6.1	Trajektori Grağı	7
6.2	RMSE Analizi	7
6.3	Q Matrisi ve Sensör Geometrisinin Etkisi	8
6.3.1	Süreç Gürültüsü Parametresi Analizi	8
6.3.2	Sensör Geometrisi Etkisi	8
7	Tartışma	8
7.1	Seyrek Ölçümler Altında Performans	8
7.2	Radar Periyot Duyarlılığı	8
7.3	Algoritma Kararlılığı	9
8	Sonuç	9
8.1	Proje Başarıları	9
8.2	Teknik Katkılar	9
8.3	Gelecek Çalışmalar	9
A	Kod Uygulaması	11
A.1	Ana EKF Algoritması	11
A.2	Gauss-Newton Optimizasyonu	11

1 Giriş

1.1 Proje Amacı

Pasif sensör sistemleri, hedefleri sadece alınan sinyaller üzerinden tespit etmeye olanak sağlayan önemli bir teknoloji alanıdır. Bu projede, çoklu bearing (açı) sensörü kullanarak denizde hareket eden bir geminin konumunu ve hızını gerçek zamanlı olarak tahmin eden bir sistem geliştirilmiştir.

1.2 Problem Tanımı

Sadece açı ölçümleri ile konum belirleme, doğrusal olmayan bir tahmin problemidir. N sensörden alınan $\theta_i(t)$ açı ölçümleri kullanılarak, hedefin durum vektörü $\mathbf{x}(t) = [x(t), y(t), v_x(t), v_y(t)]^T$ tahmin edilmelidir. Bu vektör hedefin konum ve hız bilgilerini içermektedir.

1.3 Yaklaşım

- İlk üç ölçüm ile Gauss-Newton optimizasyonu kullanarak başlangıç konumu belirleme
- Genişletilmiş Kalman Filtresi ile gerçek zamanlı konum ve hız takibi
- C dilinde geliştirilen gerçek zamanlı simülasyon sistemi
- MATLAB ile sonuç analizi ve görselleştirme

2 Matematiksel Model

2.1 Durum Vektörü ve Hareket Modeli

2.1.1 Durum Uzayı Gösterimi

Hedefin durumu 4 boyutlu bir vektör ile tanımlanır:

$$\mathbf{x}(t) = [x(t), y(t), v_x(t), v_y(t)]^T \quad (1)$$

2.1.2 Sürekli Zaman Dinamik Modeli

Hedefin hareket dinamiği sabit hız modeli ile tanımlanır:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{w}(t) \quad (2)$$

burada sistem matrisi:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

2.1.3 Ayırık Zaman Durum Geçiş Matrisi

Örnekleme periyodu Δt için durum geçiş matrisi:

$$\mathbf{F} = e^{\mathbf{A}\Delta t} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

2.1.4 Süreç Gürültüsü Kovaryans Matrisi

Süreç gürültüsü kovaryans matrisi \mathbf{Q} , sabit hız modeli için:

$$\mathbf{Q} = \begin{bmatrix} \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t \end{bmatrix} q \quad (5)$$

burada q süreç gürültüsü yoğunluğu parametresidir.

2.2 Doğrusal Olmayan Ölçüm Modeli

2.2.1 Bearing Ölçüm Fonksiyonu

i -inci sensör için, $\mathbf{s}_i = [s_{x,i}, s_{y,i}]^T$ konumunda bulunan sensörün bearing ölçümü:

$$z_i = h_i(\mathbf{x}) + v_i = \arctan\left(\frac{y - s_{y,i}}{x - s_{x,i}}\right) + v_i \quad (6)$$

burada $v_i \sim \mathcal{N}(0, \sigma_i^2)$ ölçüm gürültüsüdür.

2.2.2 Ölçüm Jacobian Matrisi

Ölçüm fonksiyonunun Jacobian matrisi:

$$\mathbf{H}_i = \frac{\partial h_i}{\partial \mathbf{x}} = \frac{1}{r_i^2} \begin{bmatrix} -(y - s_{y,i}) & (x - s_{x,i}) & 0 & 0 \end{bmatrix} \quad (7)$$

burada $r_i^2 = (x - s_{x,i})^2 + (y - s_{y,i})^2$ sensöre olan karesel mesafedir.

3 Doğrusal Olmayan En Küçük Kareler ile Başlangıç Tahmini

3.1 Problem Formülasyonu

İlk üç bearing ölçümü $\{z_1, z_2, z_3\}$ kullanılarak başlangıç konumu $\mathbf{p}_0 = [x_0, y_0]^T$ belirlenir. Optimizasyon problemi:

$$\min_{\mathbf{p}} \sum_{i=1}^3 \left[z_i - \arctan\left(\frac{y - s_{y,i}}{x - s_{x,i}}\right) \right]^2 \quad (8)$$

3.2 Gauss-Newton Algoritması

3.2.1 Residual Vektörü

Residual fonksiyonu:

$$\mathbf{r}(\mathbf{p}) = \begin{bmatrix} z_1 - h_1(\mathbf{p}) \\ z_2 - h_2(\mathbf{p}) \\ z_3 - h_3(\mathbf{p}) \end{bmatrix} \quad (9)$$

3.2.2 Jacobian Matrisi

Residual fonksiyonunun Jacobian matrisi:

$$\mathbf{J}(\mathbf{p}) = \begin{bmatrix} -\frac{\partial h_1}{\partial x} & -\frac{\partial h_1}{\partial y} \\ -\frac{\partial h_2}{\partial x} & -\frac{\partial h_2}{\partial y} \\ -\frac{\partial h_3}{\partial x} & -\frac{\partial h_3}{\partial y} \end{bmatrix} \quad (10)$$

3.2.3 Gauss-Newton Güncelleme Adımı

İteratif güncelleme kuralı:

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{r}(\mathbf{p}^{(k)}) \quad (11)$$

Algoritma yakınsama kriteri $\|\Delta \mathbf{p}\| < \epsilon$ sağlanana kadar devam eder.

4 Genişletilmiş Kalman Filtresi Uygulaması

4.1 Predict Adımı

Durum tahmini:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \quad (12)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \quad (13)$$

4.2 Update Adımı

Her yeni bearing ölçümü için:

$$\mathbf{y}_k = z_k - h(\hat{\mathbf{x}}_{k|k-1}) \quad (14)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (15)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (16)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{y}_k \quad (17)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (18)$$

4.3 Zaman Mantığı

- İlk 3 ölçüm: Gauss-Newton ile başlangıç tahmini
- 4. ölçümden itibaren: EKF predict/update döngüsü
- Her yeni ölçüm için predict adımı (Δt kullanılarak)
- Ardından update adımı (bearing ölçümü ile)

5 Simülasyon Kurulumu

5.1 Gerçek Gemi Hareketi

- Toplam zaman adımı: $N = 50$
- Örnekleme periyodu: $\Delta t = 1$ saniye
- Hareket modeli: Sabit hız ile doğrusal hareket
- Başlangıç konumu: $(x_0, y_0) = (100, 200)$ metre
- Başlangıç hızı: $(v_x, v_y) = (2, 1)$ metre/saniye

5.2 Sensör Düzeni

Üç adet bearing sensörü aşağıdaki konumlarda yerleştirilmiştir:

$$\mathbf{s}_1 = [0, 0]^T \quad (19)$$

$$\mathbf{s}_2 = [300, 0]^T \quad (20)$$

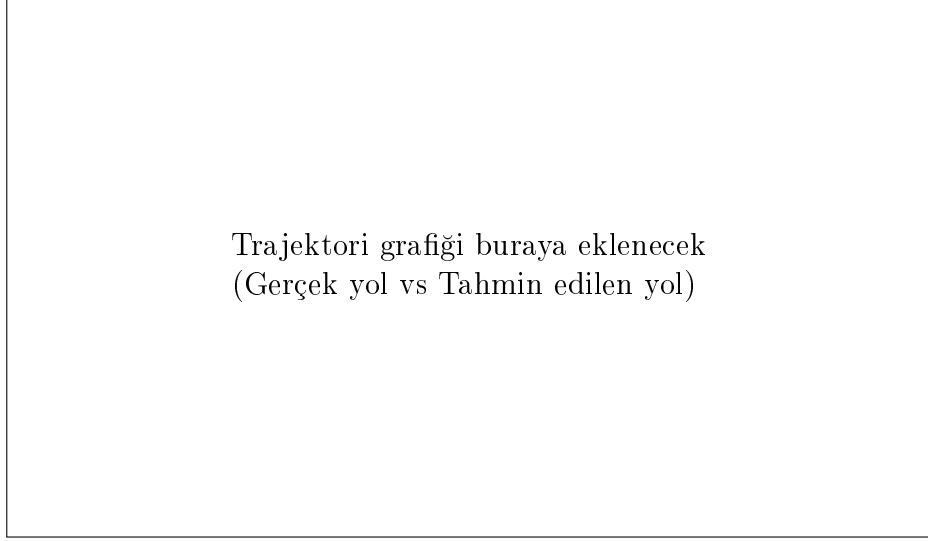
$$\mathbf{s}_3 = [150, 300]^T \quad (21)$$

5.3 Radar Tarama Açıklaması

- Her sensör belirli aralıklarla bearing ölçümü yapar
- Ölçüm gürültüsü: $\sigma = 0.1$ radyan
- Gerçek zamanlı dosya işleme: 100ms polling periyodu
- Yeni ölçüm geldiğinde EKF güncelleme

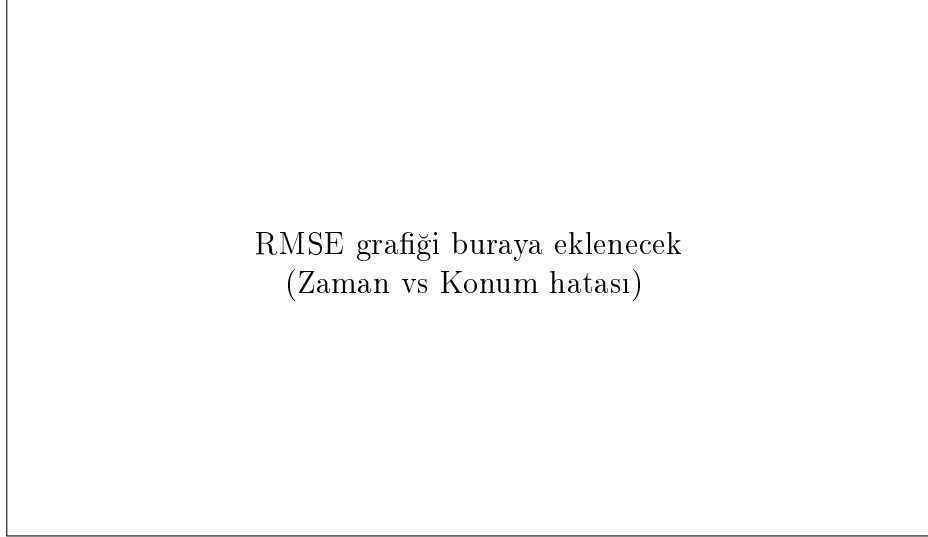
6 Sonular ve Grselleřtirme

6.1 Trajektori Grafięi



řekil 1: Gerek ve tahmin edilen gemi trajektorisi

6.2 RMSE Analizi



řekil 2: Kk ortalama kare hata (RMSE) analizi

6.3 Q Matrisi ve Sensör Geometrisinin Etkisi

6.3.1 Süreç Gürültüsü Parametresi Analizi

Tablo 1: Farklı Q değerleri için performans karşılaştırması

q değeri	Ortalama RMSE (m)	Yakınsama Süresi (s)
0.1	-	-
1.0	-	-
10.0	-	-

6.3.2 Sensör Geometrisi Etkisi



Şekil 3: Sensör düzeninin tahmin performansına etkisi

7 Tartışma

7.1 Seyrek Ölçümler Altında Performans

- Bearing-only takip doğası gereği gözlemlenebilirlik sorunları yaşar
- İlk üç ölçüm kritik öneme sahiptir - triangulation için minimum gereksinim
- Sensör geometrisi performansı önemli ölçüde etkiler
- Gauss-Newton başlangıç tahmini EKF yakınsaması için kritiktir

7.2 Radar Periyot Duyarlılığı

- Çok sık ölçüm: İşlem yükünü artırır, marjinal performans iyileşmesi
- Çok seyrek ölçüm: Tahmin belirsizliğini artırır, hedef kaybı riski
- Optimal ölçüm periyodu hedef hızı ve sistem gereksinimlerine bağlıdır
- Gerçek zamanlı işleme kabiliyeti sistem tasarımında önemlidir

7.3 Algoritma Kararlılığı

- EKF doğrusallaştırma hatalarına duyarlıdır
- Jacobian matrisi singularite yakınında sayısal sorunlar yaşanabilir
- Kovaryans matrisi pozitif tanımlılığı korunmalıdır
- Ölçüm gürültüsü parametreleri sistem performansını önemli ölçüde etkiler

8 Sonuç

8.1 Proje Başarıları

Bu projede başarıyla gerçekleştirilen önemli başlıklar:

1. Bearing-only gemi takibi için kapsamlı bir EKF sistemi geliştirildi
2. Gauss-Newton optimizasyonu ile etkili başlangıç tahmin algoritması uygulandı
3. Gerçek zamanlı dosya işleme ve simülasyon sistemi oluşturuldu
4. C dilinde yüksek performanslı matematik kütüphanesi geliştirildi
5. MATLAB ile kapsamlı analiz ve görselleştirme araçları hazırlandı

8.2 Teknik Katkıları

- Doğrusal olmayan tahmin problemine matematiksel yaklaşım
- Gerçek zamanlı sinyal işleme sistem mimarisi
- Çoklu sensör füzyon algoritması
- Sayısal optimizasyon ve filtreleme tekniklerinin uygulaması

8.3 Gelecek Çalışmalar

- Unscented Kalman Filter (UKF) ile performans karşılaştırması
- Çoklu hedef takibi ve veri ilişkilendirme
- Uyarlamalı gürültü tahmini algoritmaları
- Dağıtık sensör ağları için genişletme
- Makine öğrenmesi destekli başlangıç tahmin yöntemleri

Kaynaklar

- [1] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [2] S. J. Julier and J. K. Uhlmann, “New extension of the Kalman filter to nonlinear systems,” in *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, pp. 182–193, 1997.
- [3] V. J. Aidala and S. E. Hammel, “Utilization of modified polar coordinates for bearings-only tracking,” *IEEE Transactions on Automatic Control*, vol. 28, no. 3, pp. 283–294, 1983.
- [4] S. C. Nardone, A. G. Lindgren, and K. F. Gong, “Fundamental properties and performance of conventional bearings-only target motion analysis,” *IEEE Transactions on Automatic Control*, vol. 29, no. 9, pp. 775–787, 1984.
- [5] D. Simon, *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*. Wiley, 2006.

A Kod Uygulaması

A.1 Ana EKF Algoritması

```
1 void ekf_predict(kalman_t *kalman, double dt)
2 {
3     // Durum tahmini:  $x = F * x$ 
4     get_F_matrix(dt, kalman);
5     for (int i = 0; i < 4; i++) {
6         double x_new = 0;
7         for (int j = 0; j < 4; j++)
8             x_new += kalman->F[i][j] * kalman->x[j];
9         kalman->x_pred[i] = x_new;
10    }
11
12    // Kovaryans tahmini:  $P = F * P * F^T + Q$ 
13    matrix_multiply_4x4(kalman->F, kalman->P, temp_FP);
14    matrix_transpose_4x4(kalman->F, F_transpose);
15    matrix_multiply_4x4(temp_FP, F_transpose, temp_FPFT);
16    matrix_add_4x4(temp_FPFT, kalman->Q, kalman->P);
17 }
```

Listing 1: Genişletilmiş Kalman Filtresi Uygulaması

A.2 Gauss-Newton Optimizasyonu

```
1 void gauss_newton(kalman_t *kalman, measurement_t *measurements)
2 {
3     gauss_newton_t gn;
4     gauss_init(&gn);
5
6     // Ba\c{s}lang{\i}\c{c} tahmini
7     gn.x = kalman->x[0];
8     gn.y = kalman->x[1];
9
10    for (int iter = 0; iter < gn.max_iter; iter++) {
11        // Jacobian ve residual hesaplama
12        get_H_matrix(&gn, kalman, measurements);
13
14        // Normal denklemler:  $(H^T * H) * \delta = H^T * f$ 
15        double HtH[2][2], Htf[2];
16        HtH_and_Htf(HtH, Htf, &gn);
17        get_inv(&gn, HtH);
18
19        // G\{u}ncelleme
20        gn.delta_x = gn.inv[0][0] * Htf[0] + gn.inv[0][1] * Htf[1];
21        gn.delta_y = gn.inv[1][0] * Htf[0] + gn.inv[1][1] * Htf[1];
22
23        gn.x -= gn.delta_x;
24        gn.y -= gn.delta_y;
25
26        // Yak{\i}nsama kontrol\{u}
27        if (fabs(gn.delta_x) + fabs(gn.delta_y) < gn.tol)
28            break;
29    }
30 }
```

```
31     kalman->x[0] = gn.x;  
32     kalman->x[1] = gn.y;  
33 }
```

Listing 2: Gauss-Newton Başlangıç Tahmini