

## KALMAN FİLTRESİNDE SENSÖR HATASI VE ÖRNEKLEME ETKİLERİNİN SİMÜLASYONU

Ahmet Eren KÜRPE

*Bursa Uludağ Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, Bursa, TÜRKİYE*

### Özet

Bu çalışma, Kalman filtresi performansını etkileyen iki temel faktör olan çoklu sensörlerin hata payı ve örnekleme sayısının etkilerini incelemek amacıyla bir simülasyon uygulaması sunmaktadır. Çalışmada, farklı sensör hata seviyeleri ve örnekleme frekansları altında filtre performansı analiz edilmiştir. Simülasyon, sensör füzyonunun doğruluğunu ve güvenilirliğini artıran faktörlerin nasıl optimize edilebileceğine dair önemli bilgiler sunmaktadır. Sonuçlar, sensörlerin hata paylarının ve örnekleme yoğunluğunun Kalman filtresi çıktılarına olan etkisini açıkça ortaya koymakta ve gerçek zamanlı sistem tasarımına yönelik önerilerde bulunmaktadır.

## SIMULATION OF SENSOR ERROR AND SAMPLING EFFECTS IN KALMAN FILTER

### Abstract

This study presents a simulation application to examine the effects of two critical factors influencing Kalman filter performance: the error margins of multiple sensors and the number of samples. The analysis investigates filter performance under varying sensor error levels and sampling frequencies. The simulation provides valuable insights into optimizing factors that enhance the accuracy and reliability of sensor fusion. The results clearly demonstrate the impact of sensor error margins and sampling density on Kalman filter outputs, offering recommendations for real-time system design.

## 1. Giriş

Kalman filtresi, dinamik sistemlerdeki belirsizlikleri azaltmaktadır. Ayrıca sistem durumlarını en doğru şekilde tahmin etmek amacıyla kullanılan etkili bir algoritmadır. Kalman filtresi, birden fazla sensörden gelen verilerin doğruluğunu artırmak için sensör füzyonunda yaygın olarak kullanılır. Ancak, çoklu sensör sistemlerinde sensörlerin hata payları ve veri örnekleme yoğunlukları, filtrenin genel performansını doğrudan etkileyen kritik parametrelerdir.

Bu çalışma, Kalman filtresi performansı üzerindeki bu iki önemli faktörü (sensör hata payları ve örnekleme sayısı) analiz etmeyi amaçlamaktadır. Geliştirilen simülasyon, farklı hata seviyeleri ve örnekleme frekansları altında sistemin davranışını modelleyerek, sensör füzyonunun etkinliğini değerlendirmektedir. Elde edilen bulgular, gerçek zamanlı sistemlerin tasarımında sensör seçimi ve veri toplama stratejilerine yönelik önemli öneriler sunmaktadır.

Bu makalede, öncelikle Kalman filtresinin teorik temelleri ve sensör füzyonundaki rolü ele alınacaktır. Daha sonrada geliştirilmiş olan simülasyon uygulamasının detayları ve sonuçları paylaşılacaktır. Çalışma, sensör verilerinin doğru ve güvenilir bir şekilde birleştirilmesinde uygulanacak Kalman filtresinin sağladığı avantajları daha iyi anlamayı hedeflemektedir.

## 2. Literatür Taraması

### 2.1 Kalman Filtresi

Kalman Filtresi (KF), sinyal iyileştirme, sensör verilerinin birleştirilmesi, konum belirleme sistemleri ve hedef takibi gibi çok çeşitli uygulama alanlarında yaygın olarak kullanılan bir algoritmadır. Matematiksel ve istatistiksel temel prensiplere dayanan bu yöntem, sistemin geçmiş durum verilerinden

yararlanarak gelecekteki durumunu tahmin eder. Oldukça etkili bir yöntem olmasına rağmen, uygulama ve kavrama aşamaları genellikle karmaşık ve zordur. Kalman Filtresi ismini, bu yenilikçi algoritmanın geliştiricisi Rudolph Kalman'dan almaktadır. Literatürde bazen "Doğrusal Kuadratik Tahmin" [1] ya da "Kalman-Bucy Filtresi" [2] olarak da anılır.

KF, geleneksel filtreleme yaklaşımlarından farklı olarak yalnızca bir filtreleme mekanizması sunmakla kalmaz; aynı zamanda gelecekteki durumların tahmini için etkili bir aracı düzenek görevini üstlenir. Geleneksel filtreler, verilerdeki gürültü, hata ya da istenmeyen unsurları ortadan kaldırmayı hedeflerken, Kalman Filtresi bu hataları sistem modeli ve ölçüm verilerini kullanarak özyinelemeli bir şekilde minimize eder [3].

1960'lı yıllarda ortaya çıkan bu algoritma, çeşitli problem türlerine uygun çözümler sunmak üzere çok sayıda varyasyon geliştirilmesine olanak sağlamıştır. Genişletilmiş Kalman Filtresi (EKF), Unscented Kalman Filtresi (UKF) ve Adaptif Kalman Filtresi (AKF) gibi türler, sistemin yapısına, çözülmesi gereken problemin özelliklerine ve hesaplama maliyetlerine göre seçilir.

Kalman Filtresi; sistem durumlarının tahmini, ölçülemeyen değişkenlerin belirlenmesi, hata analizi ve hata tespiti gibi çeşitli görevlerde çok sık kullanılmaktadır. Hava ve kara taşıtlarının takibinde özellikle tercih edilen bu yöntem, tarihsel olarak NASA'nın Apollo uzay programlarında kritik bir rol oynamıştır [4].

Günümüzde gelişen teknolojilere paralel olarak, Kalman Filtresi yeni ve daha karmaşık sistemlere adapte edilebilecek şekilde geliştirilmeye devam etmektedir. Bu çeşitlilik ve gelişme potansiyeli,

algoritmanın güncel bilimsel çalışmalarda önemli bir yer edinmesini sağlamaktadır.

## 2.2 Kalman Filtresi Matematiksel Modeli

Kalman filtresi dinamik bir sistemin durumunu gürültülü ölçümler altındandan tahmin etmek ve bu tahmini ölçümlerle sürekli olarak güncellemek üzerinde tasarlanmıştır. Temelde iki aşamadan oluşur: **tahmin aşaması** ve **güncelleme aşaması**.

### 2.2.1 Kalman Filtresi Parametreleri

#### *Durum Uzayı Modeli Parametreleri*

##### *A Durum Geçiş Matrisi*

Sistemin bir durumdan diğerine nasıl geçtiğini belirlemektedir. Örnek olarak konum-ivme sisteminde, konum, hız ve ivme arasındaki ilişkiyi modellemektedir.

##### *B Kontrol Girdisi Matrisi*

Dış kontrol girişlerinin sistem durumuna olan etkisini modellemektedir. Örnek olarak bir motorun hızını artıran bir kuvveti modellemek için kullanılır.

##### *H Ölçüm Matrisi*

Ölçüm vektörü ile durum vektörü arasındaki ilişkiyi tanımlar. Her sensörün, hangi durumları ölçtüğünü göstermektedir.

#### *Gürültü Parametreleri*

##### *Q Süreç Gürültüsü Kovaryans Matrisi*

Sistemin iç dinamiklerinden kaynaklanan belirsizlikleri modellemektedir. Daha yüksek  $Q$  değerleri, modelin daha belirsiz olduğunu belirtmektedir.

#### *R Ölçüm Gürültüsü Kovaryans Matrisi*

Sensörlerin ölçüm hatalarını ve belirsizliklerini tanımlamaktadır. Güvenilir sensörler için  $R$  düşük olurken, güvenilirmez sensörler için  $R$  daha yüksek olmaktadır.

#### *Durum ve Ölçüm Değerleri*

##### *$x_k$ Durum Vektörü*

Sistemin mevcut durumunu (örneğin, hız, konum, ivme) temsil etmektedir. Tahmin edilmesi gereken büyüklüktür.

##### *$z_k$ Ölçüm Vektörü*

Sensörlerden gelen ölçüm verileridir. Gürültü ve durum vektörüyle ilişkili olan büyüklükleri içermektedir.

#### *Kovaryans Matrisleri*

##### *$P_{k|k}$ Durum Kovaryansı*

Durum tahmin belirsizliğini göstermektedir. Düşük  $P_{k|k}$ , daha güvenilir tahminler anlamına gelir.

#### *Kalman Kazancı*

##### *$K_k$ Kalman Kazancı*

Ölçümlerin tahmin üzerindeki ağırlığını belirlemektedir. Düşük ölçüm gürültüsü  $R$  veya yüksek tahmin belirsizliği  $P_{k|k-1}$  durumunda  $K_k$  artar.

#### *Başlangıç Parametreleri*

##### *$X_0$ Başlangıç Durumu*

Sistem durumunun başlangıçtaki tahminidir. Eğer doğru bir tahmin yapılamıyorsa, genellikle sıfır veya bir ortalama değerle başlatılmaktadır.

### *P<sub>0</sub> Başlangıç Kovaryansı*

Başlangıç durumunun belirsizliğini temsil etmektedir. İlk tahminler genellikle daha belirsiz olduğundan  $P_0$  yüksek bir değerle başlatılmaktadır.

### *Zaman Adımları*

#### *Δt Zaman Adımı*

Kalman filtresinin hızını belirlemektedir. Daha küçük zaman adımları, daha sık tahmin ve güncelleme anlamına gelmektedir.

### *2.2.2 Kalman Filtresi Matematiksel Modeli*

#### *Durum Denklemleri*

Durum geçiş denklemi:

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (1)$$

Ölçüm denklemi:

$$z_k = Hx_k + v_k \quad (2)$$

#### *Gürültü Kovaryansları*

Süreç gürültüsü ( $w_k$ ):

$$w_k \sim \mathcal{N}(0, Q) \quad (3)$$

Ölçüm gürültüsü ( $v_k$ ):

$$v_k \sim \mathcal{N}(0, R) \quad (4)$$

#### *Tahmin Aşaması*

Durum tahmini:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \quad (5)$$

Kovaryans tahmini:

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (6)$$

### *Güncelleme Aşaması*

Kalman kazancı:

$$K_k = P_{k|k-1}H^T (HP_{k|k-1}H^T + R)^{-1} \quad (7)$$

Durum güncellemesi:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H\hat{x}_{k|k-1}) \quad (8)$$

Kovaryans güncellemesi:

$$P_{k|k} = (I - K_kH)P_{k|k-1} \quad (9)$$

### *2.2.3 Kalman Filtresi Çalışması*

Kalman Filtresi Tahmin ve Güncelleme Aşaması şeklinde sürekli olarak iki aşamada çalışmaktadır.



Şekil 1. Kalman Filtresi çalışma diyagramı

#### *Tahmin Aşaması*

Bu aşama, sistemin bir sonraki zaman adımındaki durumunu ve belirsizliğini (kovaryansı) mevcut bilgiye dayanarak hesaplamaktadır.

#### *Durum Tahmini*

Sistemin mevcut durumu, Denklem (5)'te görüldüğü üzere sistemin dinamik modeli, kontrol girdisi, ve gürültü kullanılarak tahmin edilmektedir.

### *Kovaryans Tahmini*

Sistemin durum tahminindeki belirsizlik, Denklem (6)'da görüldüğü üzere süreç gürültüsü ve sistem dinamikleri dikkate alınarak hesaplanmaktadır.

### *Güncelleme Aşaması*

Bu aşamada, yapılan tahminler ölçümlerle  $z_k$  karşılaştırılmaktadır ve tahminler bu ölçümler aracılığı ile güncellenmektedir.

### *Kalman Kazancının Hesaplanması*

Kalman kazancı, ölçümden ne kadar bilgi alınacağını belirlemektedir. Denklem (7)'de görüldüğü gibi ölçüm belirsizliği ve tahmin belirsizliği kullanılarak hesaplanmaktadır.

### *Durum Güncellemesi*

Denklem (8)'de yer alan tahmin edilen durum, ölçümle karşılaştırılmaktadır ve hesaplanan Kalman kazancı kullanılarak güncellenmektedir.

### *Kovaryans Güncellemesi:*

Durum güncellemesinden sonra Denklem (9) kullanılarak sistemin belirsizliği tekrar hesaplanır.

## **3. Yöntem**

### **3.1 Simülasyonun Genel Çerçevesi**

Bu çalışmada geliştirilen simülasyonun amacı, Kalman filtresinin çoklu sensörlerin hata paylarına ve örnekleme sayısına olan duyarlılığını analiz etmektir. Çalışmada, gerçek zamanlı bir simülasyon ortamı sağlanarak, sensör füzyonu sürecinin canlı olarak gözlemlenmesi hedeflenmiştir. Geliştirilen simülasyonda C# programlama dili kullanılmıştır. Kullanıcı arayüzü, simülasyon sonuçlarının görselleştirilmesi ve kullanıcı girdilerinin alınabilmesi

amacıyla tasarlanmıştır. Kullanıcı arayüzü tasarımı sırasında iki boyutlu bir çizim alanı oluşturulmuş ve bu alan, simülasyonda sistem girdisi ve sensör verilerinin görselleştirilmesi için kullanılmıştır.

Simülasyonda kullanılan sistem, aşağıdaki bileşenlerden oluşmaktadır:

### *Fare Konumu (Kalman Girdisi):*

Kullanıcının bilgisayarındaki fare hareketi, Kalman filtresinin birincil girdisi olarak tanımlanmıştır. Farenin konum bilgisi, sistemin gerçek durumunu temsil etmek üzere ele alınmıştır. Fare hareketleri, sürekli olarak canvas üzerinde takip edilmiştir.

### *Sanal Sensörler:*

Simülasyonda kullanılmak üzere iki adet sanal sensör tanımlanmıştır. Her iki sensör, fare konum bilgisini temel alarak normal dağılım (Gaussian noise) ile bir hata olarak eklenmiş şekilde konum bilgisini tekrardan üretmektedir. Bu sensörler, sistemin hata paylarını test etmek ve Kalman filtresi performansını analiz etmek amacıyla kullanılmıştır.

### *Kalman Filtresi ve Sensör Füzyonu:*

Simülasyonda, Fare konumu ve iki adet sanal sensör olmak üzere toplamda üç girdi bulunmaktadır. Eğer tüm sensörler aktif hale getirilirse, bu girdilerden gelen veriler sensör füzyonu yöntemiyle birleştirilerek Kalman filtresi ile işlenmektedir. Sensör füzyonu sayesinde, her bir sensörün hatası minimize edilmiş ve daha doğru bir tahmin elde edilmiştir.

### *Kullanıcı Arayüzü:*

Kullanıcı arayüzünde, kullanıcı kullanılmak girdileri seçebilmektedir. Çizim alanında girdilerin ve çıktıların görünürlüğünü değiştirip, bu alanı temizleyebilmektedir.

Ayrıca sensörlerin veri üretme sıklıklarını ve hata varyanslarını değiştirebilmektedir.

### Simülasyonun İşleyişi

Çizim alanında kullanıcı, fare hareket ettikçe gerçek zamanlı olarak farenin konum bilgileri alınmaktadır. Sensörler, gerçek fare konumuna dayalı olarak rastgele hata eklenmiş konum verileri üretmektedir. Üretilen tüm sensör verileri, eşzamanlı olarak Kalman filtresine girdi olarak sağlanmıştır. Kalman filtresi, bu girdilerden elde edilen tahmini sonuçları gerçek zamanlı olarak görselleştirmiştir. Bu simülasyon, kullanıcıların Kalman filtresi sürecini ve sensör füzyonunun etkilerini anlamalarına olanak sağlamış, farklı sensör hata seviyelerinde sistemin performansını test etmeyi mümkün kılmıştır.

### 3.2 Durum Uzayı Modeli

Bu modelde, fare ve iki sensörden gelen pozisyon ölçümlerine göre Kalman filtresini genişletilmiştir. Her ölçüm, sistemin gerçek pozisyon bilgisine katkı yapmaktadır.

#### Durum Vektörü

Durum vektörü, sistemin tahmin edilmek istenen değişkenlerini içermektedir. Burada, pozisyonlar ve hızlar modellenmektedir.

$$x = \begin{bmatrix} x_{\text{pozisyon}} \\ y_{\text{pozisyon}} \\ x_{\text{hız}} \\ y_{\text{hız}} \end{bmatrix} \quad (10)$$

#### Durum Geçiş Matrisi

Durum geçiş matrisinde, pozisyon ve hız arasındaki ilişki tanımlanmaktadır.

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

#### Ölçüm Matrisi

Ölçüm matrisi, fare, sensör 1 ve sensör 2'nin pozisyon ölçümlerinin gerçek durumu nasıl etkilediğini tanımlamaktadır. Kullanıcının kullanmak istediği girdilere göre bu matris yeniden tanımlanmaktadır.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (12)$$

#### Ölçüm Gürültüsü Kovaryansı

Ölçüm gürültüsü matrisi, girdilerin ölçüm hatalarını temsil eder. Ölçüm hataları bağımsız ise diyagonal bir matris olur. Kovaryans matrisinin değerlerinin büyük olması girdinin güvenilirliğinin düşük olmasını ifade ederken, küçük değerler girdinin güvenilirliğinin yüksek olduğunu ifade etmektedir.

$$R = \begin{bmatrix} \sigma_{\text{fare}}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\text{fare}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\text{sensör1}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\text{sensör1}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\text{sensör2}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\text{sensör2}}^2 \end{bmatrix} \quad (13)$$

### **Durum Gürültüsü Kovaryansı**

Durum gürültüsü, sistem dinamiklerindeki belirsizliği temsil etmektedir.

$$Q = \sigma_Q^2 \cdot I \quad (14)$$

### **Başlangıç Durumu ve Kovaryansı**

Durum vektörü, sistemin tahmin edilmek istenen değişkenlerini içermektedir. Burada, pozisyon ve hızları modellenmektedir.

$$x_0 = \begin{bmatrix} x_0 \text{ pozisyon} \\ y_0 \text{ pozisyon} \\ x_0 \text{ hız} \\ y_0 \text{ hız} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (15)$$

### **3.3 Sensör Modeli**

#### **Fare Girdisini Sensör Ölçümü Olarak Tanımlama:**

Fare girdisi, belirli bir pozisyon veya hız ölçümünü temsil etmektedir. Ancak, bu girdilere normal dağılım hatası (gürültü) eklenerek gerçek bir sensör gibi modellenir.

#### **Matematiksel Model:**

$$z_{\text{sensör}} = x_{\text{gerçek}} + v_{\text{gürültü}} \quad (16)$$

Sanal sensör ölçümü Denklem (16)'daki gibi ifade edilmektedir.  $x_{\text{gerçek}}$  gerçek fare bilgisidir ve  $v_{\text{gürültü}}$  değeri belirlenen gürültü modeli ile toplanıp sensör bilgisi olarak modellenmektedir.

#### **Gürültü Modeli:**

$$v_{\text{gürültü}} \sim \mathcal{N}(\mu, \sigma^2) \quad (17)$$

Gürültü ortalama değeri sıfır kabul edilmektedir. Gürültü varyansı değeri sensörün hata miktarını temsil etmektedir. Kullanıcı arayüzünde değiştirilebilmektedir.

### **3.4 Simülasyon Parametreleri**

#### **Simülasyon Döngü Süresi**

Simülasyonun her döngüde ne kadar süreceğini kontrol eder. Bu süre milisaniye birimi olarak girilmektedir.

#### **Sensör Yenileme Sıklığı**

Her sensörün ölçüm gönderme sıklığı saniyedeki örnek sayısı SPS (Sample per second) olarak girilmektedir.

#### **Dinamik Hata (R Matrisi) Güncellemesi**

Kullanıcı arayüzünden bu özellik aktifleştirildiğinde sensörlerin doğruluğuna göre R matrisi tekrardan hesaplanıp güncellenmektedir. Hatası yüksek olan sensörlerin kovaryans değerleri otomatik olarak artmaktadır.

#### **Sensör Hata Payı Güncellemeleri**

Her sensör için gürültü varyansı kullanıcı arayüzü üzerinden simülasyon parametresi olarak güncellenebilmektedir.

#### **Giriş Kontrolü**

Kullanıcı, simülasyon sırasında gerçek zamanlı olarak farenin çizim alanındaki pozisyonlarını değiştirebilmektedir.

### 3.6 Dinamik R Matrisi Güncelleme Özelliği ve Çalışma Mekanizması

Kalman filtresinde ölçüm gürültüsü kovaryans matrisi (R), sensörlerin hata oranlarını temsil etmektedir ve ölçümlerin güvenilirliğini belirlemektedir. Dinamik R matrisi güncelleme özelliği, sensörlerin performansının zaman içinde değişiklik gösterebileceği durumlarda, bu matrisi gerçek zamanlı olarak yeniden hesaplayarak filtreleme doğruluğunu artırmaktadır.

#### Dinamik Güncellenmenin Amacı

Sensörlerin ölçüm doğruluğu, çevresel faktörler, donanım hataları veya sistemin çalıştığı koşullara bağlı olarak değişebilmektedir. Sabit bir R matrisi, bu değişikliklere uyum sağlayamazken, dinamik bir güncelleme, anlık sensör performansını dikkate alarak Ölçüm güvenilirliğine göre sensörlerin etkisini dengelemektedir. Yüksek hata oranına sahip sensörlerin tahmin üzerindeki etkisini azaltmaktadır. Filtrenin, anlık durumlardaki değişikliklerine daha hızlı uyum sağlamasını sağlamaktadır. Bu işlem için gerçek kabul edilecek sonuçlara da ihtiyaç duyulmaktadır. Bu durumun simülasyon da test edilebilmesi simülasyonun çeşitliliğini arttırmaktadır.

#### Dinamik Güncelleme Nasıl Çalışır?

Dinamik R matrisi güncelleme özelliği, her simülasyon döngüsünde R matrisini yeniden tanımlayarak çalışmaktadır.

**Varyans Hesaplama:** Her sensör için ölçüm varyansları hesaplanmaktadır. Varyans, sensörlerdeki ölçüm değerlerinin gerçek değerlerden sapmalarının bir göstergesi olmaktadır.

Bu denklemde:

- $z_i$  , sensör ölçüm değerini,
- $z_{gercek}$  , sistemin gerçek değerini,
- $N$  , ölçüm sayısını ifade etmektedir.

**R Matrisinin Güncellenmesi:** Her sensörün ölçüm varyansı, R matrisindeki uygun köşegen elemanlarına atanır. Örnek olarak, birinci sensörün X ve Y eksenlerindeki varyansları R[2,2] ve R[3,3] elemanlarına yerleştirilmektedir.

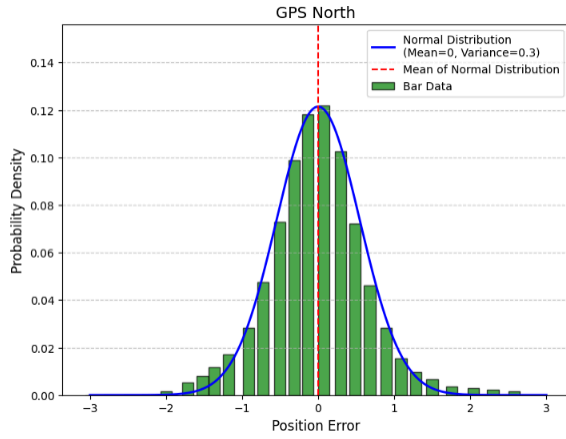
**Anlık Hesaplama ve Uyarılama:** Dinamik güncelleme özelliği aktif hale getirildiğinde, her simülasyon döngüsünde varyans yeniden hesaplanmaktadır ve R matrisi buna göre güncellenmektedir. Daha yüksek varyans değerleri, sensörün hata oranının arttığını gösterir ve filtre bu sensörün ölçümlerine daha az güvenmeye başlamaktadır. Daha düşük varyans değerleri ise sensörün güvenilirliğinin arttığını ifade eder ve tahminlerde bu ölçümlerin etkisi artırılmaktadır.

**Kullanıcı Arayüzü ile Entegrasyon:** Dinamik güncelleme özelliği, aktif veya pasif hale getirilebilmektedir. Aktif durumda olduğunda sensörlerin ölçüm varyansları ve R matrisi değerleri sürekli izlenir ve simülasyon tarafından güncellemelerle çalışmaktadır.

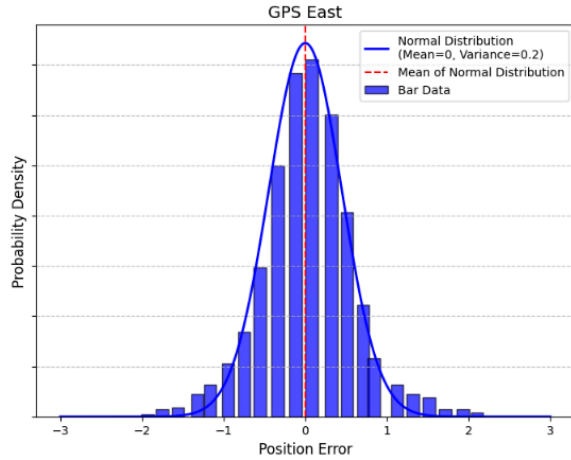
$$\sigma^2 = \frac{\sum_{i=1}^N (z_i - z_{gercek})^2}{N} \quad (18)$$



### 3.5 Simülasyonun Doğrulama ve Test Süreci

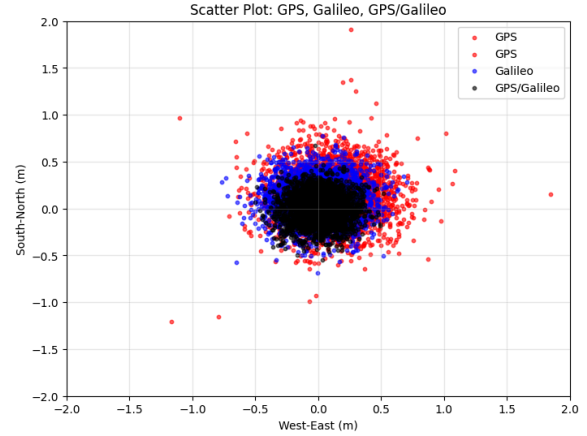


Şekil 1. Pozisyon hatalarının Normal Dağılım ile modellenmesi (Kuzey)

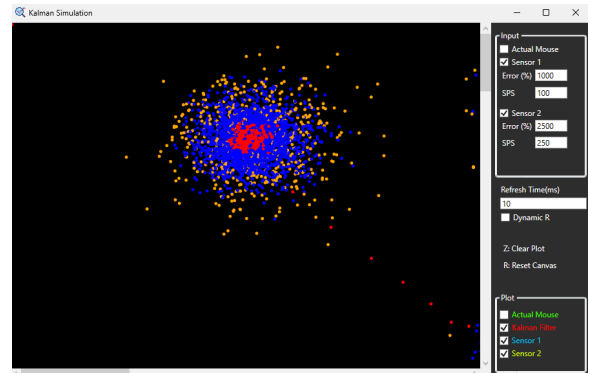


Şekil 2. Pozisyon hatalarının Normal Dağılım ile modellenmesi (Doğu)

Şekil 1 ve Şekil 2'deki grafikte görüldüğü üzere GPS modülünün hata bilgisi normal dağılım ile modellenebilmektedir [5]. Gerçek sonuçlara yakın değerler ile simülasyonda hatanın varyans bilgisi girilerek gerçeğe yakın sensörlerin simüle edilmesi mümkün olabilmektedir.

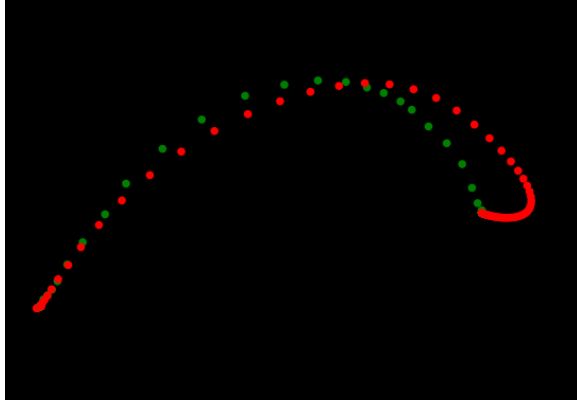


Şekil 3. GPS ve Galileo pozisyon dağılımları



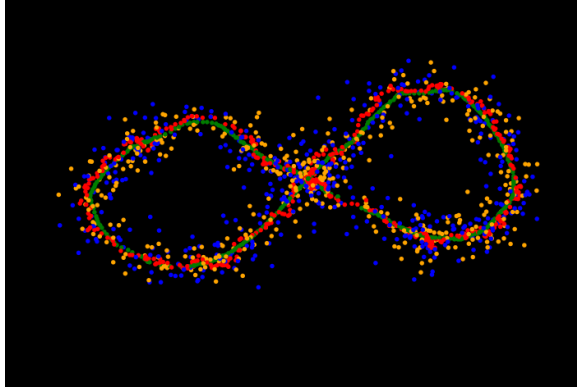
Şekil 4. GPS ve Galileo pozisyon dağılımlarının simülasyonda modellenmesi

Şekil 3 ve Şekil 4 incelendiğinde sabit konumda üretilen hata değerleri benzerlik göstermektedir. Sensör 1, Galileo uyduları ile oluşan pozisyon hatalarını uygun konum verileri üretirken; Sensör 2, GPS uydularının oluşturduğu pozisyon hatalarına örnek olmaktadır kırmızı ise iki konum bilgisinin aynı kalman filtresinde uygulanması sonucu ortaya çıkan pozisyonu göstermektedir. Gürültünün önemli bir derecede filtrelendiği söylenebilmektedir [5].



Şekil 5. Filtrenin beklenmedik durum tepkisi

Şekil 5'te görülen yeşil noktalar gerçek fare girdileri iken kırmızı olan noktalar kalman filtresi çıktılarıdır. Şekil 5'te de görüldüğü üzere filtre girdilerindeki beklenmedik hızlanma veya yavaşlama gibi sistem dinamiklerinde olan değişiklikler, Kalman filtresinin tahmin adımında hatalara neden olabilmektedir. Durum gürültüsü kovaryansı (Q) bu belirsizlikleri temsil etmektedir ve yüksek Q değerleri, ani değişimlere karşı sistemin daha esnek hale gelmesini sağlamaktadır.



Şekil 6. Sadece Sensör1 ve Sensör2 ile kalman filtresi

Şekil 6'da, yalnızca Sensör1 ve Sensör2 ölçümleri kullanılarak oluşturulan Kalman filtresinin sonuçları yer almaktadır. Sensör1, Galileo tabanlı konum verilerini sağlarken; Sensör2, GPS tabanlı ölçümler sunmaktadır. Bu iki sensörden gelen ölçümler, Kalman filtresi kullanılarak birleştirilmiştir ve filtrelenmiştir.

Filtreleme sonucunda:

**Konum Doğruluğu:** Galileo ve GPS verilerinin birleştirilmesi, tek bir sensöre kıyasla daha doğru bir konum tahmini sağlamaktadır.

**Gürültünün Azaltılması:** Şekil 6'da görüldüğü üzere, kırmızı çizgiyle temsil edilen Kalman filtresi sonucu, sensörlerden gelen bireysel ölçümlerden daha az sapma göstermektedir. Bu, filtreleme sürecinde hataların önemli ölçüde bastırıldığını ve sistemin stabil hale geldiğini kanıtlar.

### Sonuç

Bu çalışmada, GPS ve Galileo sistemlerinden elde edilen pozisyon verileri, normal dağılım modeli ile simüle edilerek gerçek dünyaya yakın sensör hatalarının modellenmesi sağlanmıştır. Simülasyon parametrelerinin kullanıcı tarafından dinamik olarak değiştirilmesine olanak tanıyan bir kullanıcı arayüzü geliştirilmiştir.

Simülasyon sonuçları, sensör hatalarının varyans bilgilerinin doğru bir şekilde belirlenmesi durumunda gerçeğe yakın pozisyon verilerinin elde edilebileceğini göstermiştir. Kalman filtresi kullanılarak, birden fazla sensörden alınan veriler birleştirilmiş ve gürültü önemli ölçüde filtrelenmiştir. Özellikle, simülasyonda farklı sensörlerin hata varyanslarının dinamik olarak güncellenmesi ve R matrisi hesaplamasının sensör doğruluğuna göre optimize edilmesi, hataların etkili bir şekilde azaltılmasını sağlamıştır.

Şekil 3 ve Şekil 4'te gösterilen sonuçlar, Galileo ve GPS uydu sistemlerinden elde edilen hataların benzer karakteristik özellikler taşıdığını ve her iki sistemin de normal dağılım ile modellenilebildiğini doğrulamaktadır. Özellikle, aynı Kalman filtresi içerisinde her iki sensörün

birleştirilmesi sonucunda, daha doğru pozisyon verilerine ulaşıldığı görülmüştür.

Bu çalışma, farklı sistemlerin doğruluğunu artırmaya yönelik simülasyon ortamlarının geliştirilmesi ve sensörlerin hata modellerinin analizi konusunda önemli bir temel sunmaktadır. Gelecekte, daha karmaşık dinamik modeller ve gerçek zamanlı uygulamalarla bu yöntemin genişletilmesi planlanmaktadır. Ayrıca, farklı sensör kombinasyonlarının performansını değerlendirmek için daha kapsamlı test süreçleri uygulanabilir.

### **Kaynaklar**

- [1] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35-45.
- [2] Bucy, R. S., & Joseph, P. D. (1968). *Filtering for stochastic processes with applications to guidance*. Wiley.
- [3] Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. Wiley.
- [4] Apollo Guidance Computer. (1969). *NASA Technical Reports*. Available at: <https://ntrs.nasa.gov/>.
- [5] Real-time single-frequency precise positioning with Galileo satellites. (2021). Available at: [https://www.researchgate.net/publication/355754403\\_Real-time\\_single-frequency\\_precise\\_positioning\\_with\\_Galileo\\_satellites/figures?lo=1](https://www.researchgate.net/publication/355754403_Real-time_single-frequency_precise_positioning_with_Galileo_satellites/figures?lo=1).