T.C.
MİLLÎ SAVUNMA ÜNİVERSİTESİ
HEZÂRFEN HAVACILIK VE UZAY TEKNOLOJİLERİ ENSTİTÜSÜ
UZAY BİLİMLERİ ANABİLİM DALI
UYDU TEKNOLOJİLERİ PROGRAMI

GEO UYDUSU İÇİN KALMAN FİLTRESİ

HM641 İLERİ AVİYONİK SİSTEMLER

Alper ŞANLI
1192101

Prof.Dr. Cengiz HACIZADE

İçerik

Semboller

x, y, z = Uydunun kartezyen koordinatlarıdır.

r = Uydunun kütle merkezi ile dünyanın kütle merkezi arasındaki mesafedir

γ = Kepler sabitidir.

M = Dünyanın kütlesidir.

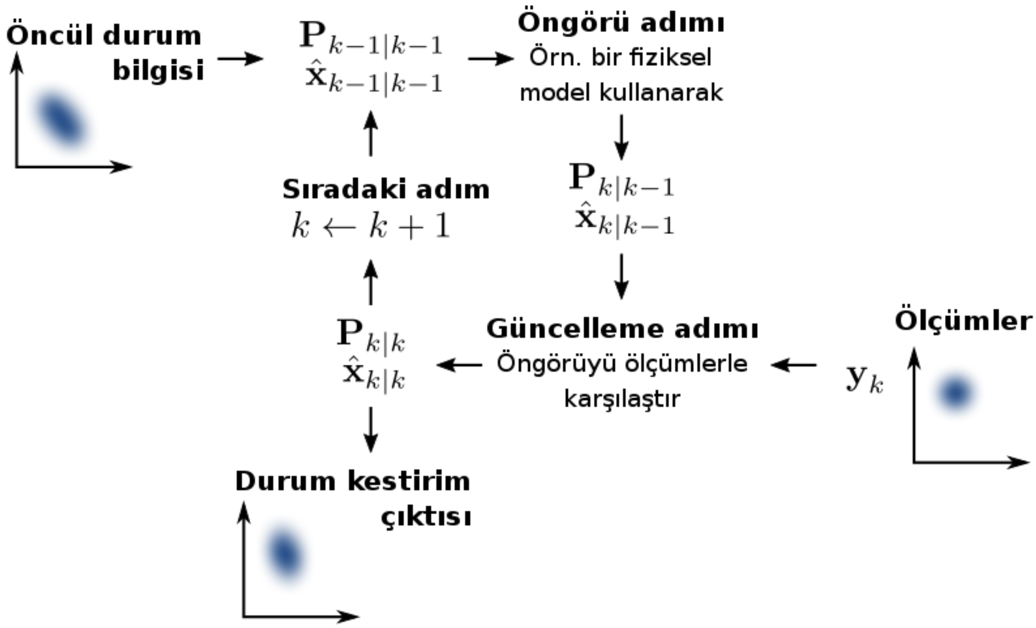
1 Giriş

Kalman Filtresi, durum uzayı modeli ile gösterilen bir dinamik sistemde, modelin önceki bilgileriyle birlikte giriş ve çıkış bilgilerinden sistemin durumlarını tahmin edilebilen filtredir. Macar asıllı Amerikan matematiksel sistem teoristi Rudolf Kalman tarafından bulunmuştur[1].

Kalman Filtresi, 1960'lardan sonra araç navigasyonu başta olmak üzere (havacılık uygulamaları tipik olmasına rağmen, başka uygulama alanlarında da) kullanılan ve sistemin durumu hakkında optimize edilmiş bir tahmin sağlayan bir algoritmadır. Algoritma, gürültülü bir gözlem veri akışı (tipik olarak, sensör ölçümleri) üzerinde gerçek-zamanlı, özyinelemeli çalışarak hatayı en aza indirecek şekilde filtreleme yapar ve sistemin fiziksel karakteristiklerinin modellenmesi ile üretilen gelecek durumun matematiksel tahminine göre optimize eder.

Model tahmini, gözlem ile karşılaştırılır ve bu fark, Kalman kazancı olarak bilinen bir çarpan ile ölçeklendirilir. Bu daha sonra sıradaki tahminleri iyileştirmek için modele bir girdi olarak geri beslenir. Kazanç performansı iyileştirmek için ayarlanabilir. Yüksek kazanç değerleri kullanılırsa, filtre çıkışı gözlemleri daha yakından takip eder. Düşük kazanç değeri kullanıldığında filtre model tahminlerini daha yakından takip eder. Yöntem, gerçek bilinmeyen değerlere, tek bir ölçüme veya sadece model tahminlerine dayanarak elde edilebilecek tahminlerden daha yakın tahminler üretmek için kullanılmaktadır.

Her bir zaman adımında, Kalman Filtresi, gerçek bilinmeyen değerlerin tahminlerini belirsizlikleriyle beraber üretir. Sıradaki ölçümün sonucu gözlendiğinde, bu tahminler, belirsizliği düşük tahminlere daha fazla ağırlık vererek, ağırlıklı ortalama ile güncellenir.



Şekil 1.1: Kalman Filtresinin Temeli [2]

Sistemlerin matematiksel modellemesi ile ilgili olarak pratikte karşılaşılan iki temel sorun vardır [3]:

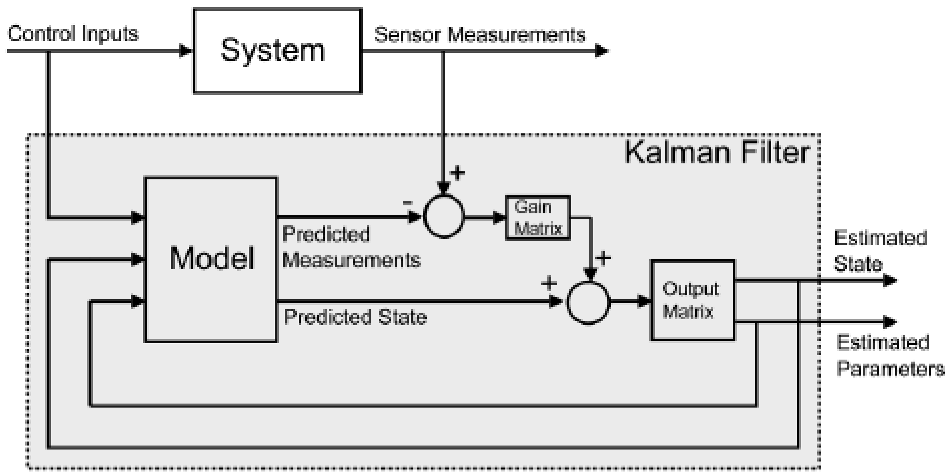
1) Sistem modelinin mükemmel olmaması, yani gerçek sistemi tam olarak temsil etmemesi. Örnekler:

- Sistemde gerçekte var olan bazı dinamiklerin modelde dikkate alınmamış olması.
- Bazı parametrelerin tam doğru seçilememiş olması.
- Sistem parametrelerini bir süre sonra değişmesi, böylece en başta modellemede kullanılmış olan değerlerinden artık farklılık göstermesi.

2) Sistem çıkışından alınana ölçümlerin mükemmel olmaması. Örnekler:

- Sensörlerin ölçümlerinde bir miktar hata payı olması.
- Ölçümlere gürültü karışması.

Bu durumlarda ölçülen çıkışları daha temiz bir hale getirmek ve sistemin dinamiklerini yöneten durum değişkenlerini tahmin etmek için Kalman filtresi kullanılabilir.



Şekil 1.2: Kalman Filtsi Model Örneği [4]

Bu çalışmada bir GEO uydusu için Kalman Filtresi uygulanmıştır.

2 Uydu Hareket Simülasyonu

Uyduların eliptik yörüngelerinin ifade edilebildiği sistemlerden biri Kepler denklemleridir. Bu denklem sistemi diferansiyel formda olan aşağıdaki 3 denklemden oluşmaktadır [5]:

$$\frac{d^2x}{dt^2} = -\gamma M \frac{x}{r^3}$$

$$\frac{d^2y}{dt^2} = -\gamma M \frac{y}{r^3} \qquad \text{Burada:}$$

$$\frac{d^2z}{dt^2} = -\gamma M \frac{z}{r^3}$$

$\gamma = 6.67 \times 10^{-11} \frac{m^3}{kg.m^2}$ Kepler sabitidir.

$M = 5.976 \times 10^{24} kg$ Dünyanın kütlesidir.

r Uydunun kütle merkezi ile dünyanın kütle merkezi arasındaki mesafedir

x, y, z uydunun kartezyen koordinatlarıdır.

Bunları 1. dereceden differansiyel denklemleri:

$$\left.\begin{aligned} \frac{dx}{dt} &= U \\ \frac{dy}{dt} &= V \\ \frac{dz}{dt} &= W \end{aligned}\right\}$$

$$\frac{dU}{dt} = -\gamma M \frac{x}{r^3} \qquad \frac{dx}{dt} = U \Rightarrow \frac{x_{i+1} - x_i}{\Delta t} \Rightarrow x_{i+1} = x_i + \Delta t.U$$

$$\frac{dV}{dt} = -\gamma M \frac{y}{r^3} \qquad \frac{dy}{dt} = V \Rightarrow \frac{y_{i+1} - y_i}{\Delta t} \Rightarrow y_{i+1} = y_i + \Delta t.V$$

$$\frac{dW}{dt} = -\gamma M \frac{z}{r^3} \qquad \frac{dz}{dt} = W \Rightarrow \frac{z_{i+1} - z_i}{\Delta t} \Rightarrow z_{i+1} = z_i + \Delta t.W$$

Toplam 6 denklem elde edilir:

$$\left.\begin{aligned} U_{i+1} &= U_i - \Delta t.\gamma.M\,\frac{x_i}{r_i^3} \\ V_{i+1} &= V_i - \Delta t.\gamma.M\,\frac{y_i}{r_i^3} \\ W_{i+1} &= W_i - \Delta t.\gamma.M\,\frac{z_i}{r_i^3} \end{aligned}\right\}$$

$$\left.\begin{aligned} x_{i+1} &= x_i + \Delta t.U_i \\ y_{i+1} &= y_i + \Delta t.V_i \\ z_{i+1} &= z_i + \Delta t.W_i \end{aligned}\right\} \qquad r_i^{\,2} = x_i^{\,2} + y_i^{\,2} + z_i^{\,2} \qquad r_i = \sqrt{x_i^{\,2} + y_i^{\,2} + z_i^{\,2}}$$

Bu denklemleri çözmek için başlangıç koşulları alınır, uydunun konumu ve hızları bulunmaktadır.

GEO uydusu için başlangıç koşullarındaki değerler :

$x_0(m)$	$y_0(m)$	$z_0(m)$	$r(m)$
10000000	20000000	?	35000000

$U_0(m/s)$ Uzunlamasına Hız	$V_0(m/s)$ Yanlamasına Hız	$W_0(m/s)$ Dikey Hız	$S_0(m/s)$ Bileşke Hız
2000	2000	?	3070

Başlangıç koşulları Kepler Denklemleri'ne yazılarak $z_0(m)$ ve $W_0(m/s)$ değerleri bulunmuştur.

Sonuçlar :

$z_0(m) = 26925824.03567252$

$W_0(m/s) = 1193.6917525056458$

3 Uydu Hareket Ölçüm Simülasyonu

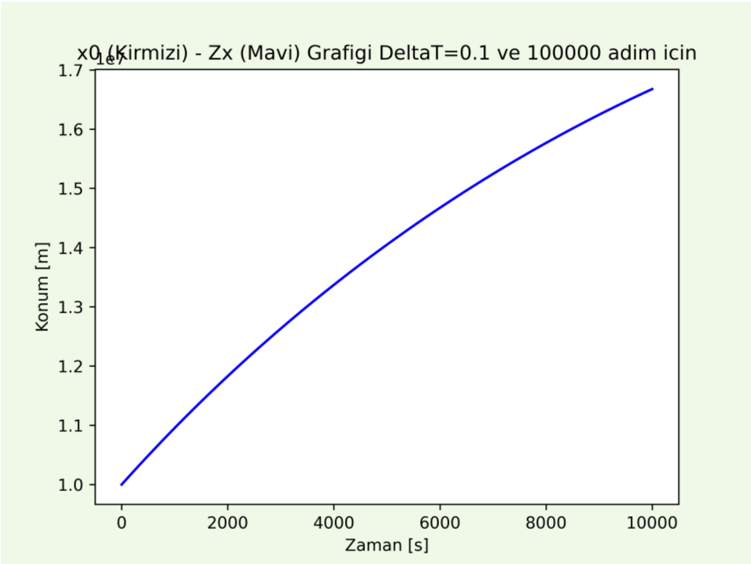
GEO uydusuna ölçüm değeleri aşağıda verilen bozucu değerler ile verilmiştir.

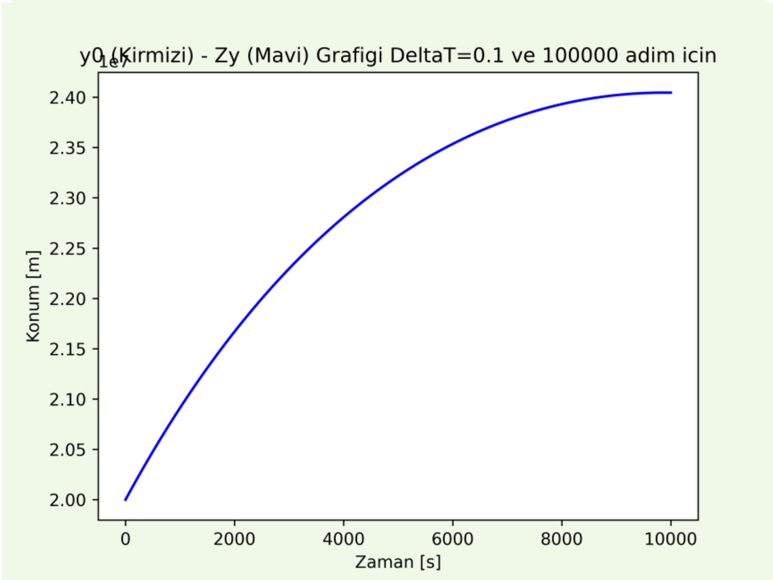
$$Z_x = x + \sigma_x * randn$$
$$Z_y = y + \sigma_y * randn$$
$$Z_z = z + \sigma_z * randn$$
$$Z_U = x + \sigma_U * randn$$
$$Z_V = x + \sigma_V * randn$$
$$Z_W = x + \sigma_W * randn$$

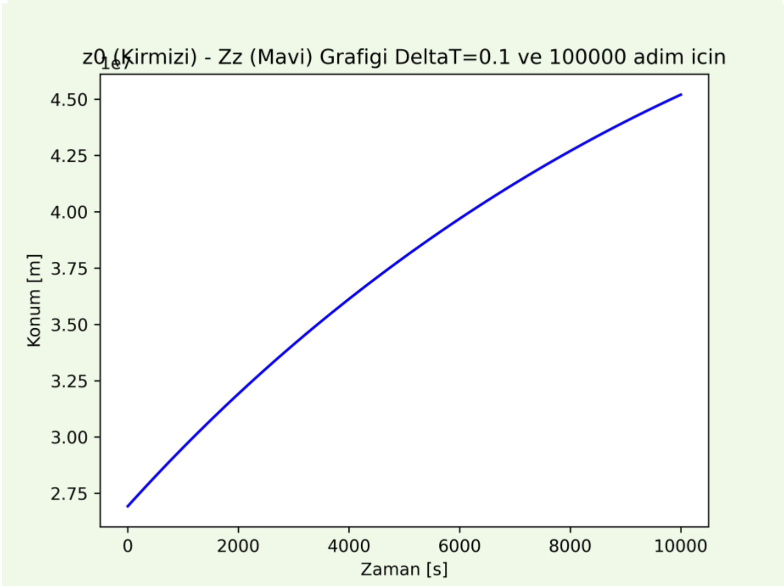
$$\sigma_x = 10m; \sigma_y = 10m; \sigma_z = 15m$$
$$\sigma_U = 0.02m/s; \sigma_V = 0.02m/s; \sigma_W = 0.02m/s$$

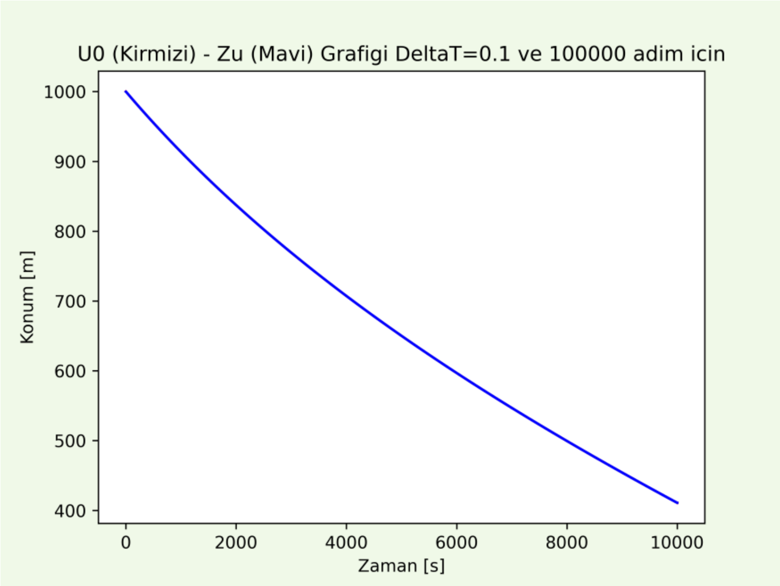
Kalman filtresi uygulamasında ölçüm değerleri ve Kepler denklemlerinden gelen teori değerleri kullanılacaktır.

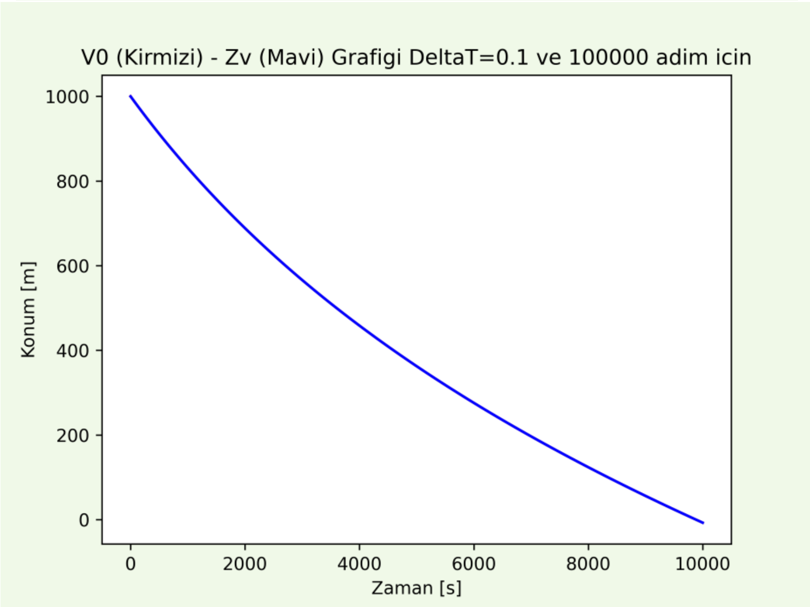
İstenilen adım sayılarına göre teori, ölçüm grafik sonuçları :

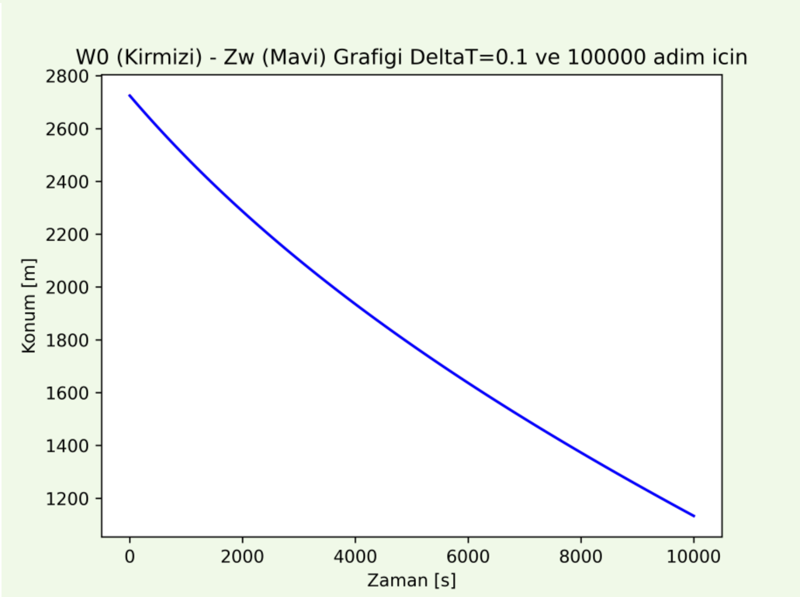


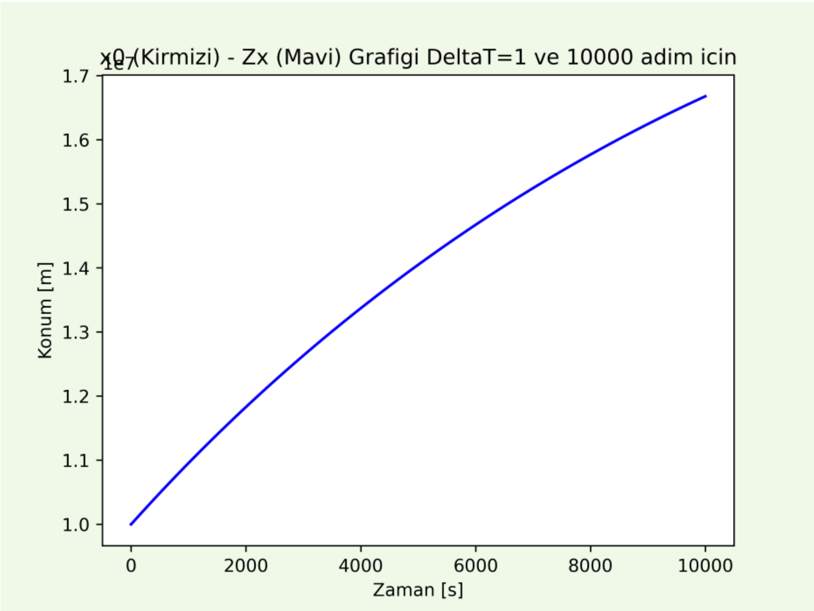


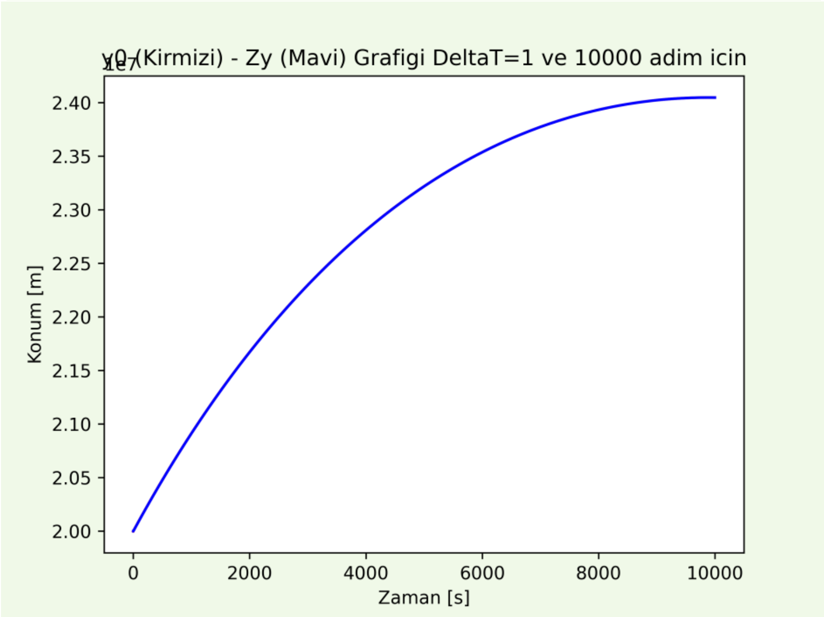


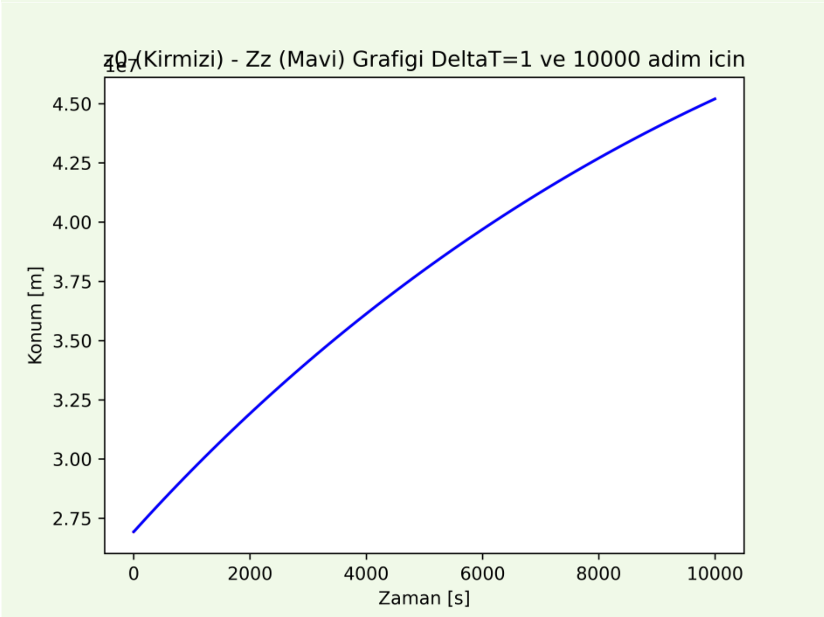

U0 (Kirmizi) - Zu (Mavi) Grafigi DeltaT=0.1 ve 100000 adim icin
Konum [m]
Zaman [s]
1000
900
800
700
600
500
400
0
2000
4000
6000
8000
10000

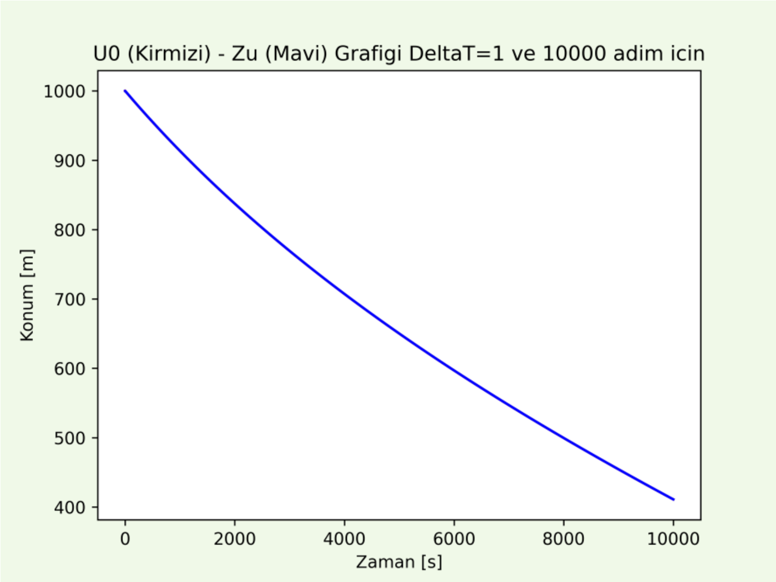

V0 (Kirmizi) - Zv (Mavi) Grafigi DeltaT=0.1 ve 100000 adim icin
Konum [m]
Zaman [s]
1000
800
600
400
200
0
0
2000
4000
6000
8000
10000

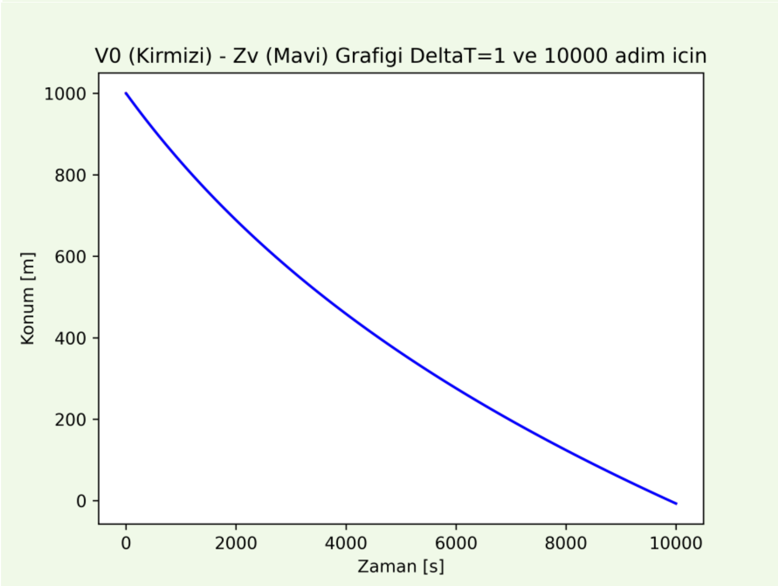

W0 (Kirmizi) - Zw (Mavi) Grafigi DeltaT=0.1 ve 100000 adim icin
Konum [m]
Zaman [s]
2800
2600
2400
2200
2000
1800
1600
1400
1200
0
2000
4000
6000
8000
10000

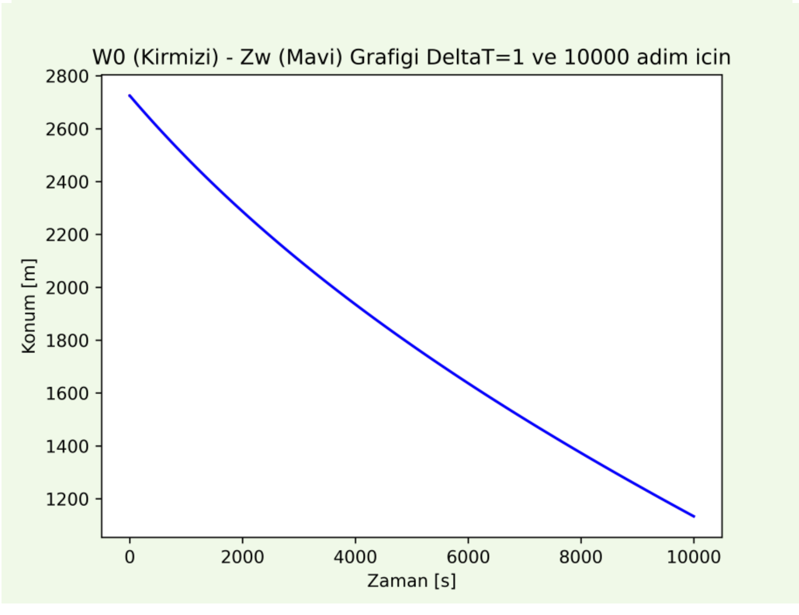

x0 (Kirmizi) - Zx (Mavi) Grafigi DeltaT=1 ve 10000 adim icin
1e7
Konum [m]
Zaman [s]


y0 (Kirmizi) - Zy (Mavi) Grafigi DeltaT=1 ve 10000 adim icin
1e7
Konum [m]
Zaman [s]


z0 (Kirmizi) - Zz (Mavi) Grafigi DeltaT=1 ve 10000 adim icin
1e7
Konum [m]
Zaman [s]


U0 (Kirmizi) - Zu (Mavi) Grafigi DeltaT=1 ve 10000 adim icin
Konum [m]
Zaman [s]
1000
900
800
700
600
500
400
0
2000
4000
6000
8000
10000


V0 (Kirmizi) - Zv (Mavi) Grafigi DeltaT=1 ve 10000 adim icin
Konum [m]
Zaman [s]
1000
800
600
400
200
0
0
2000
4000
6000
8000
10000


W0 (Kirmizi) - Zw (Mavi) Grafigi DeltaT=1 ve 10000 adim icin
Konum [m]
Zaman [s]
2800
2600
2400
2200
2000
1800
1600
1400
1200
0
2000
4000
6000
8000
10000

4 Kalman Filtresi

Navigasyon ölçümleri işleniş süresinde Kalman Filtreleri geniş olarak uygulanmaktadır. Kalman Filtreleri havacılıkta aşağıdaki amaçlarla kullanılmaktadır [5]:

1) Ölçüm hatalarının küçültülmesi ve ölçülen parametrenin daha doğru değerinin bulunması (Filtrasyon Problemi)
2) Durum vektörünün ölçülmeyen koordinatlarının kestirimi (estimate) (kestrim problemi – estimation problem)
3) Cismin matematiksel modelinin bilinmeyen parametrelerinin kestirimi (Identification – Tanılama)
4) Çeşitli bilgi kaynaklarının tümleştirilmesi (sensor fusion) veya (Integrated Navigation) da denir.
5) Uçakta meydana gelen bozulmaların teşhisi (Fault detection)

Kalman Filtresinin Çalışma Prensibi:
Durum Denklemi: $x(k+1) = \phi(k+1,k)x(k) + G(k+1,k)w(k)$
Gözlem Denklemi: $y(k) = H(k)x(k) + v(k)$
$x(k)$: Sistemin n boyutlu durum vektörüdür.
$\phi(k+1,k)$: Sistemin nxn boyutlu geçiş matrisidir. (transition matrix)
Sistemi k durumundan k+1 durumuna geçiren bir matrisdir.
$w(k)$: Sistemin r boyutlu bozuntu vektörüdür. (Sistem bozuntusudur – System Noise)
$G(k+1,k)$: Sistemin bozuntusunun nxr boyutlu geçiş matrisidir.
Sistem bozuntusunun ortalamasının sıfır, korelasyon matrisinin ise,

$$E\left[w(k)w^T(j)\right] = Q(k)\delta(kj)$$

Burada E statistik ortalama operatörüdür.

$$\delta(kj) = \begin{cases} 1, k = j \\ 0, k \neq j \end{cases}$$ Kroneker sembolüdür.

r vektörünün rastgele Gauss bozuntusu olduğu varsayılır.
$y(k)$: s boyutlu gözlem vektörüdür.
$H(k)$: sistemin sxn boyutlu gözlem matrisidir.
$v(k)$: ölçümlerin s boyutlu bozuntu vektörüdür. (measurement noise) onun matematiksel beklentisi (ortalaması) sıfır.
Korelasyon matrisi ise,

$$E\left[v(k)v^T(j)\right] = R(k)\delta(kj)$$

Başlangıç x_0 durumu, ortalaması $\overline{x_0}$ ve korelasyon matrisi $P(0)$ olan Gauss vektörüdür.
Sistem bozuntusu $w(k)$ ve ölçüm bozuntusu $v(k)$ arasında korelasyon ilişkisi yoktur.

$$E\left[w(k)v^T(j)\right] = 0, \forall_{k,j}$$

Durum vektörünün $y(k)$ gözlem vektörüne ardışıklığına göre değerinin bulunması istendiğinde Kalman Filtresine dayanan lineer filtreler kullanılması gerekir. Lineer ayrık dinamik sistemin durum vektörünün optimal (değer hatasının standart sapmasının minimum olması koşuluna göre) değerlendirme algoritması aşağıdaki denklemler sistemiyle yazılmaktadır:
Kestirilen (estimated) değerin Denklemi – Kestirim Denklemi (Estimation Equation):

$$\hat{x}(k/k) = \phi(k,k-1)\hat{x}(k-1,k-1) + K(k)\left[y(k) - H(k)\phi(k,k-1)\hat{x}(k-1/k-1)\right] \quad (1)$$
$$= \hat{x}(k/k-1) + K(k)\tilde{z}(k/k-1)$$

Extrapolation Equation (Tahmin):

$\hat{x}(k/k-1) = \phi(k,k-1)\hat{x}(k-1,k-1)$ (2)

Innovation Sequence (process) (İnovasyon süreç):

$\tilde{z}(k/k-1) = y(k) - \hat{x}(k,k-1)$ (3)

$K(k)$: Optimal Filtrenin matris kazanç katsayısıdır.

$$K(k) = P(k/k)H^T(k)R^{-1}(k) = P(k/k-1)H^T(k)\left[H(k)P(k/k-1)H^T(k) + R(k)\right]^{-1} \quad (4)$$

H: Ölçme Matrisi

R: Ölçmenin Korelasyon Matrisi

$P(k/k)$: Kestirim hatasının korelasyon matrisidir. (Correlation Matrix of estimation error)

$e(k) = \hat{x}(k/k) - x(k)$: Hata

$P(k/k-1)$: Tahmin hatasının korelasyon matrisi (Correlation Matrix of extrapolation error)

$$P(k/k) = P(k/k-1) - P(k/k-1)H^T(k)\left[H(k)P(k/k-1)H^T + R(k)\right]^{-1}H(k)P(k/k-1) \quad (5)$$
$$= P(k/k-1) - K(k)H(k)P(k/k-1) = (I - K(k)H(k))P(k/k-1) \quad (6)$$

Extrapolasyon Hatasının Korelasyon Matrisi:

$$P(k/k-1) = \phi(k,k-1)P(k-1/k-1)\phi^T(k,k-1) + G(k,k-1)Q(k-1)G^T(k,k-1) \quad (7)$$

Başlangıç Koşulları:

$\hat{x}(0/0) = x(0)$; $P(0/0) = P(0)$

$P(0)$ 6x6 ’ luk bir matris, diagonal 10’dur.

$$P(k/k-1) = \underbrace{\phi(k,k-1)}_{A}\underbrace{P(k-1/k-1)}_{P(0)}\underbrace{\phi^T(k,k-1)}_{A^T} + G(k,k-1)Q(k-1)G^T(k,k-1)$$

G matrisi I birim matrisi (6x6)

Q matrisi ise 6x6’luk ve diagonal değerleri 0.001

H ölçüm birim matrisi 6x6

R Ölçme simülasyonundaki σ_{randu}

$P(k/k)$ hesaplanır.

$\hat{x}(k/k-1)$ hesaplanır.

Innovation süreci hesaplanır.

Kalman Filtresi hesaplanır.

Uydu matrisleri P, R, Q, G, H 6x6 ‘dır.

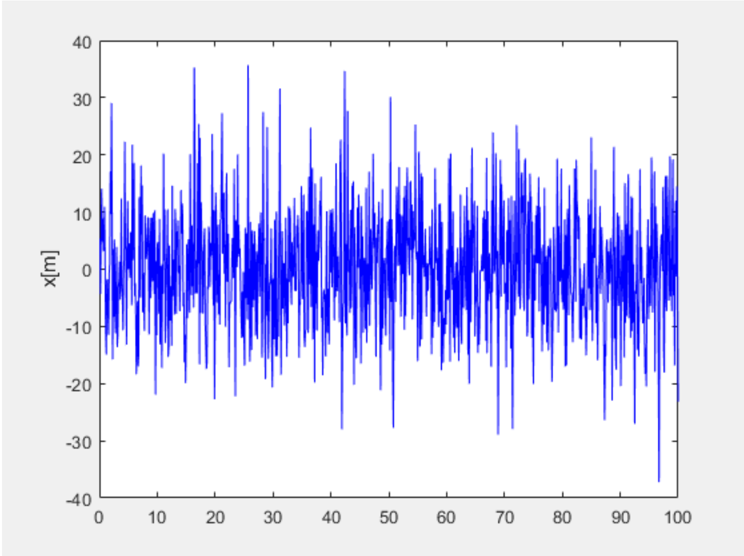
Uydu sistemi nonlineer bir sistem olduğu için Jacobian Matrisi hesaplanır.

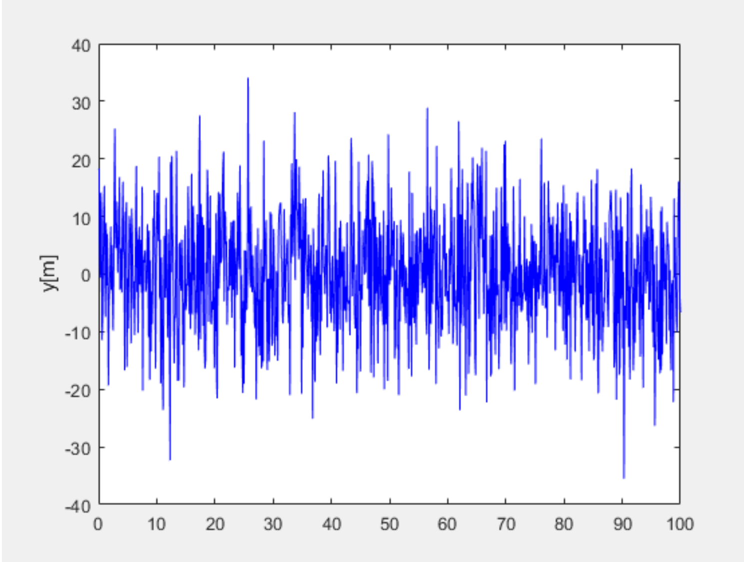
Matlab formunda Jacobian Matris =

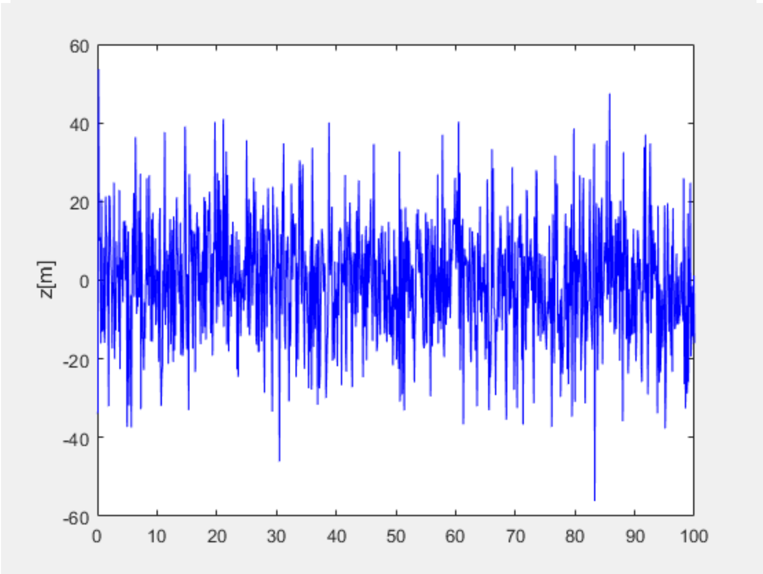
```
J(1,1)=1; J(1,2)=0; J(1,3)=0; J(1,4)=deltaT; J(1,5)=0; J(1,6)=0;
J(2,1)=0; J(2,2)=1; J(2,3)=0; J(2,4)=0; J(2,5)=deltaT; J(2,6)=0;
J(3,1)=0; J(3,2)=0; J(3,3)=1; J(3,4)=0; J(3,5)=0; J(3,6)=deltaT;
J(4,1)=-(deltaT*m_u*(-2*kalman_x(i)^2+kalman_y(i)^2+ kalman_z(i)^2))/(kalman_x(i)^2 + kalman_y(i)^2 + kalman_z(i)^2)^(5/2);
J(4,2)=(3*deltaT*m_u*kalman_x(i)*kalman_y(i))/(kalman_x(i)^2+kalman_y(i)^2+kalman_z(i)^2)^(5/2);
J(4,3)=(3*deltaT*m_u*kalman_x(i)*kalman_z(i))/(kalman_x(i)^2+kalman_y(i)^2+kalman_z(i)^2)^(5/2);
J(4,4)=1; J(4,5)=0; J(4,6)=0;
J(5,1)=(3*deltaT*m_u*kalman_x(i)*kalman_y(i))/(kalman_x(i)^2+kalman_y(i)^2 + kalman_z(i)^2)^(5/2);
J(5,2)=-(deltaT*m_u*(kalman_x(i)^2 - 2*kalman_y(i)^2 + kalman_z(i)^2))/(kalman_x(i)^2 + kalman_y(i)^2 + kalman_z(i)^2)^(5/2);
J(5,3)=(3*deltaT*m_u*kalman_y(i)*kalman_z(i))/(kalman_x(i)^2 + kalman_y(i)^2 + kalman_z(i)^2)^(5/2);
J(5,4)=0; J(5,5)=1; J(5,6)=0;
J(6,1)=(3*deltaT*m_u*kalman_x(i)*kalman_z(i))/(kalman_x(i)^2 + kalman_y(i)^2 + kalman_z(i)^2)^(5/2);
J(6,2)=(3*deltaT*m_u*kalman_y(i)*kalman_z(i))/(kalman_x(i)^2 + kalman_y(i)^2 + kalman_z(i)^2)^(5/2);
J(6,3)=-(deltaT*m_u*(kalman_x(i)^2 + kalman_y(i)^2 - 2*kalman_z(i)^2))/(kalman_x(i)^2 + kalman_y(i)^2 + kalman_z(i)^2)^(5/2);
J(6,4)=0; J(6,5)=0; J(6,6)=1;
```

5 Kalman Sonuçları

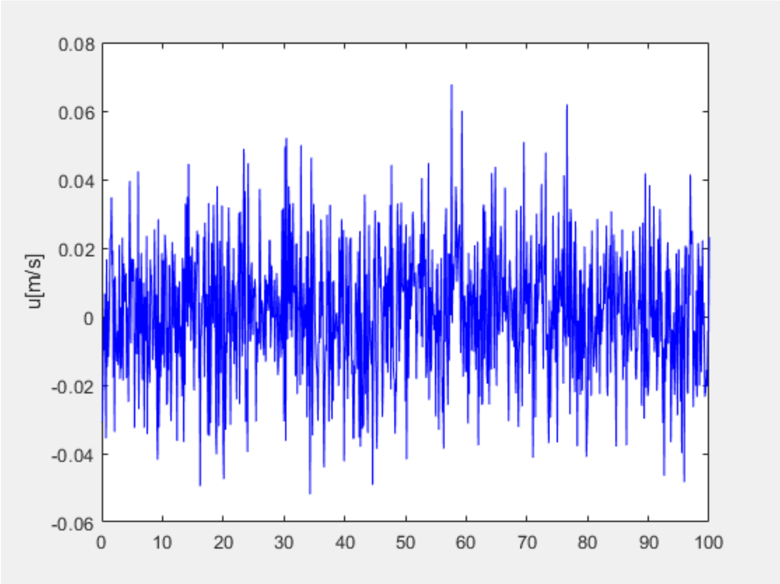
- Ölçüm, Konum

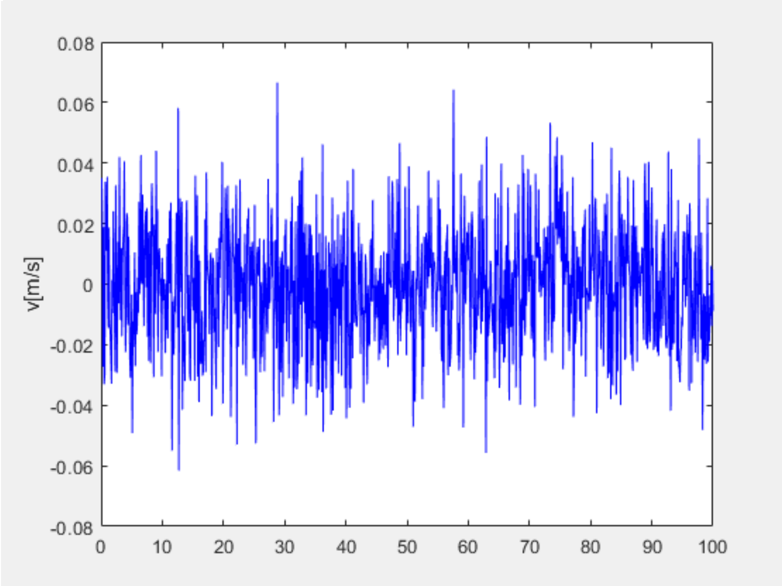


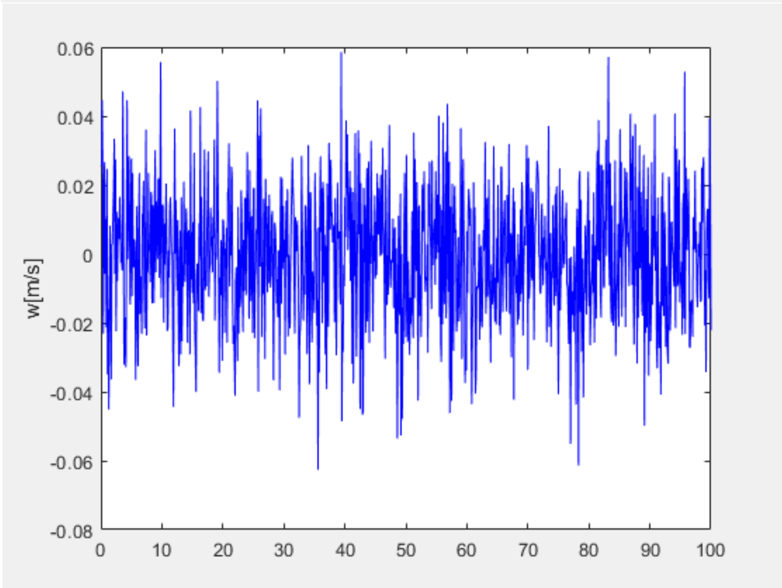




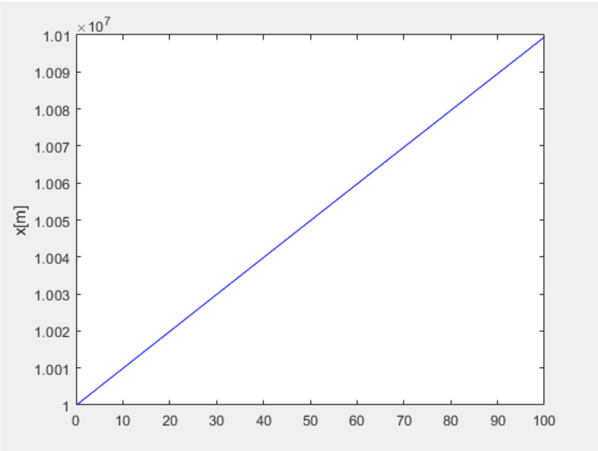
- Ölçüm, Hız

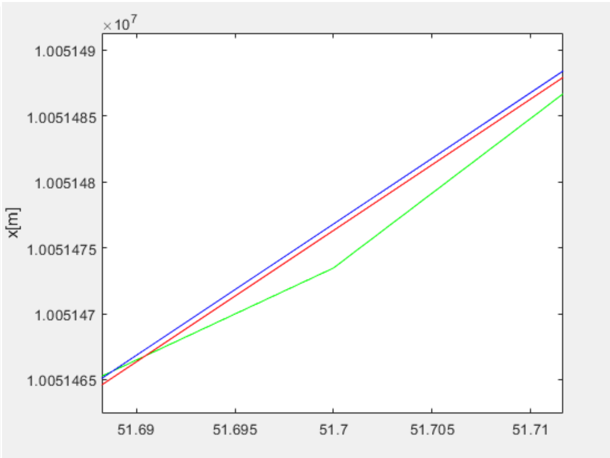


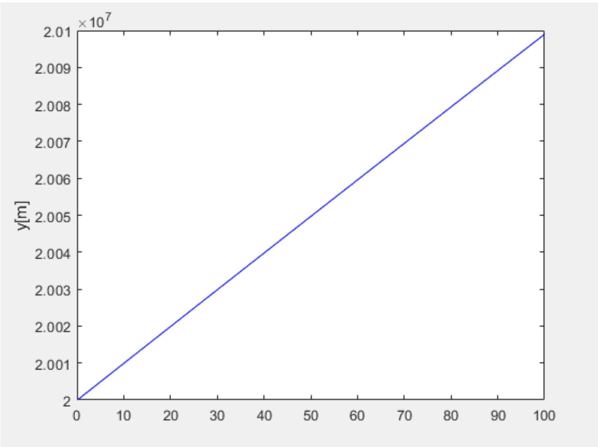


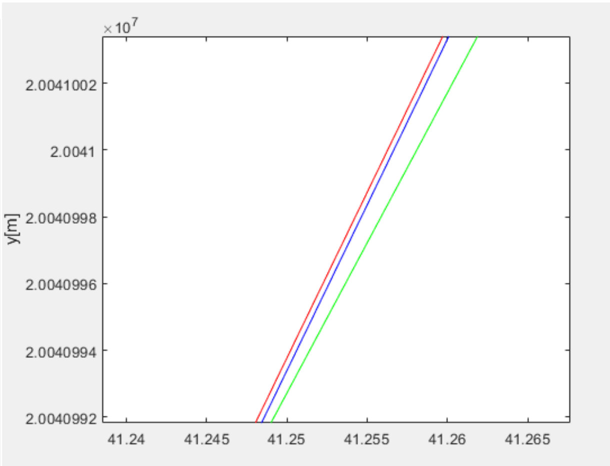


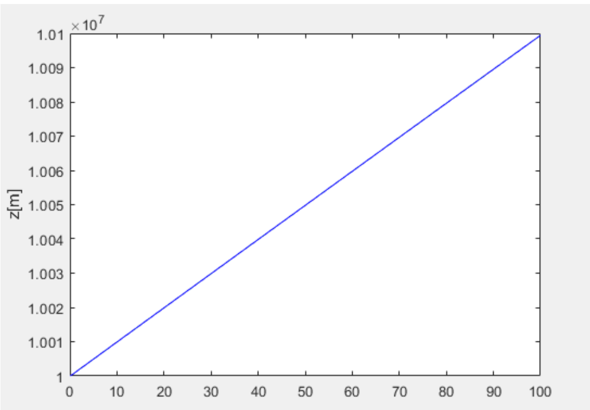
- Konum, Teori (Mavi), Ölçüm (Yeşil), Kalman (Kırmızı)

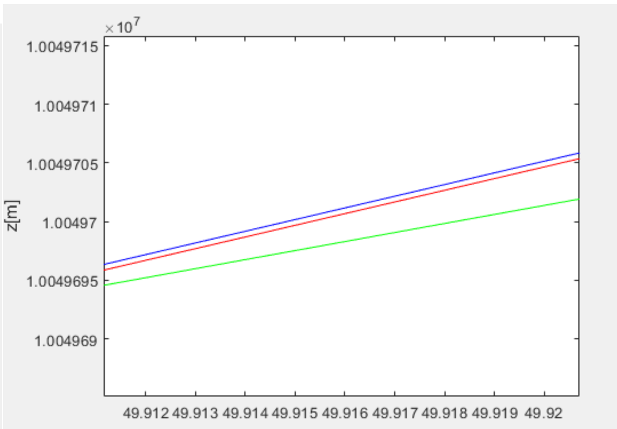




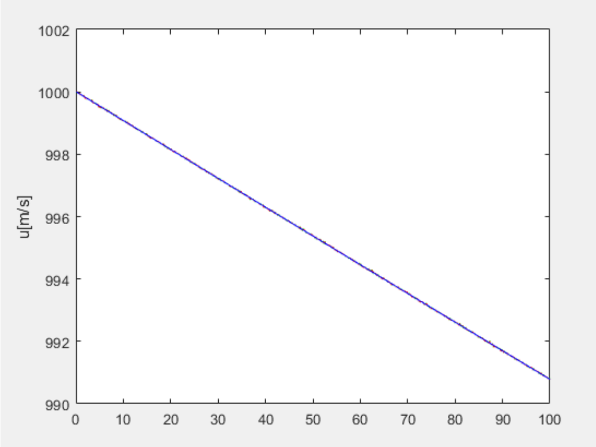


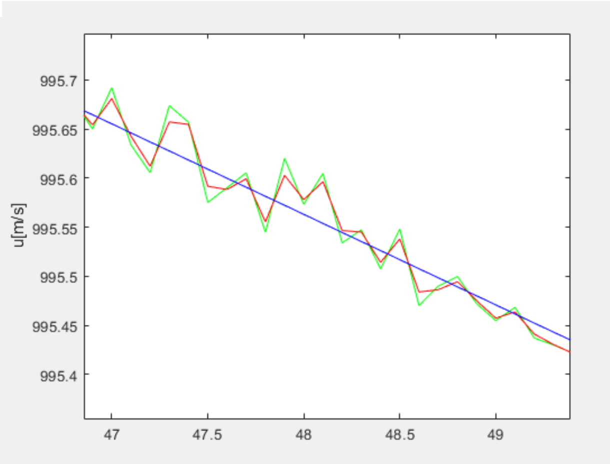


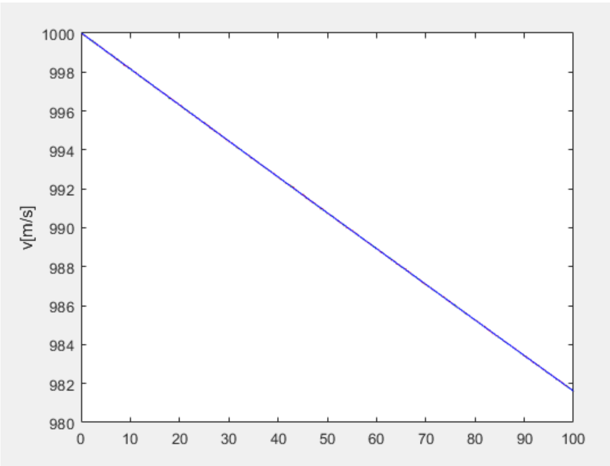


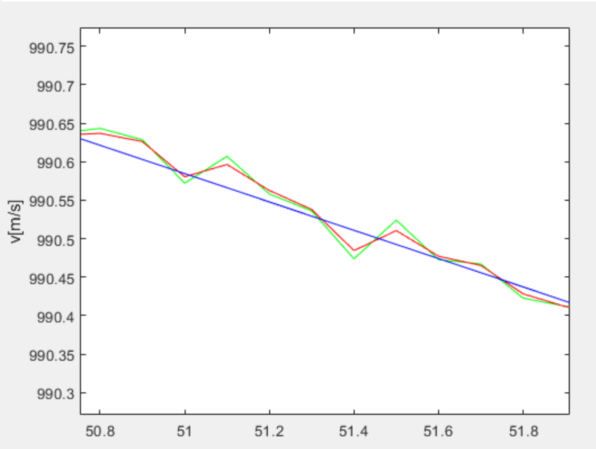


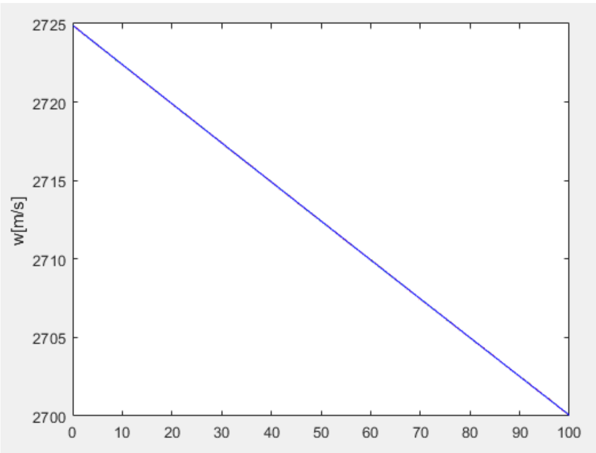
- Hız, Teori (Mavi), Ölçüm (Yeşil), Kalman (Kırmızı)

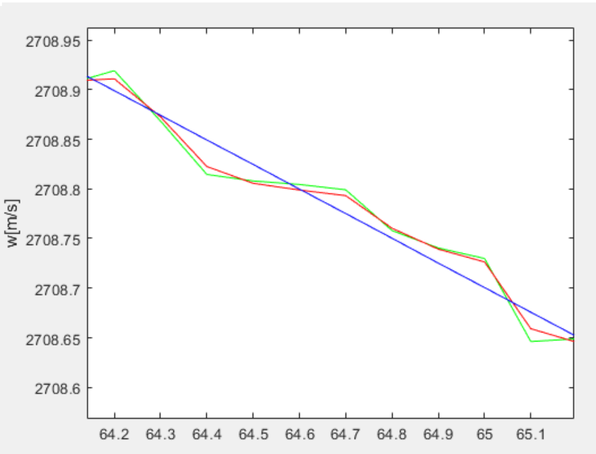




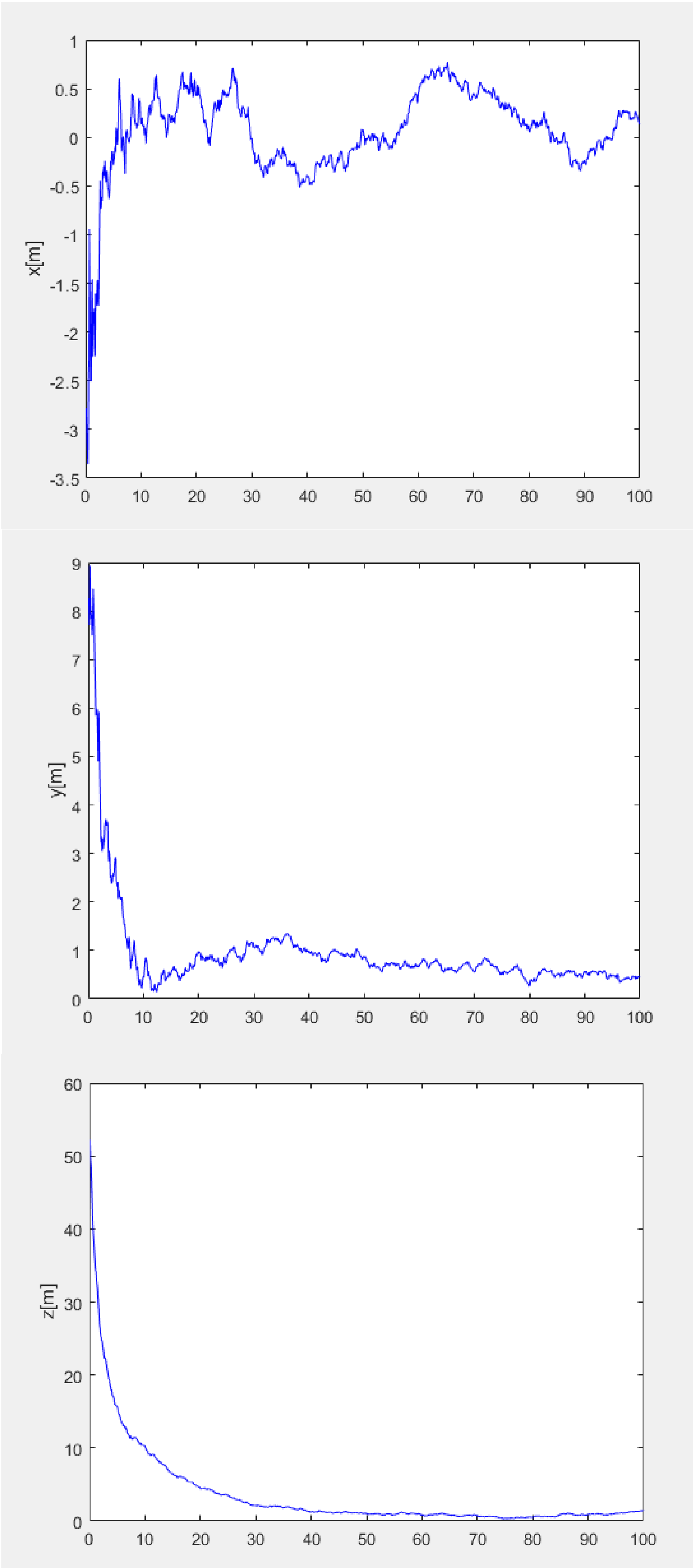




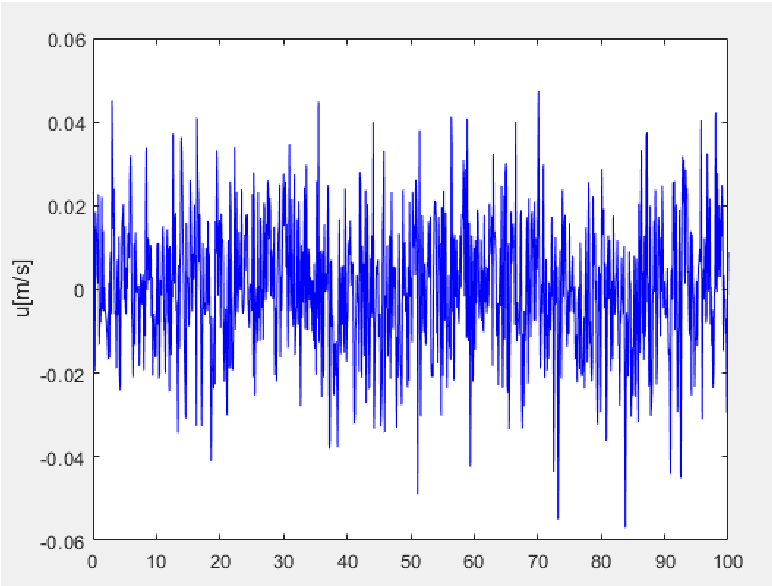


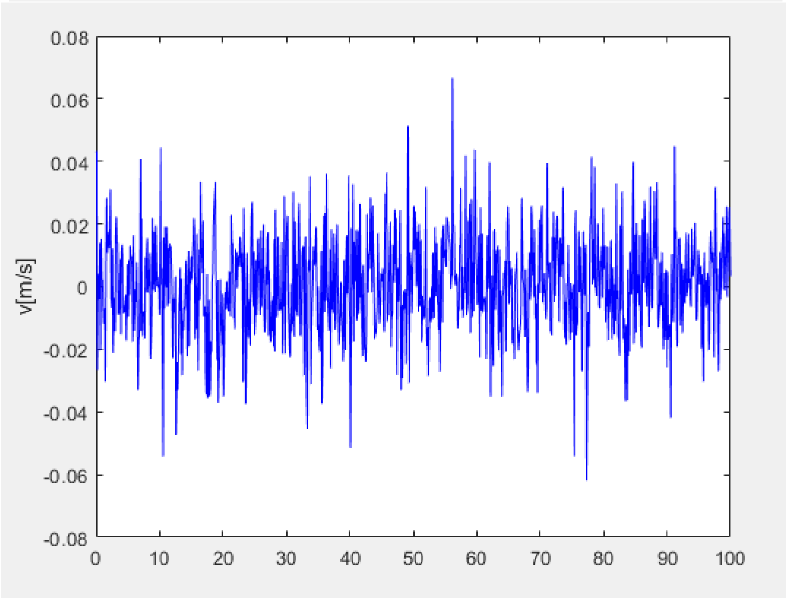


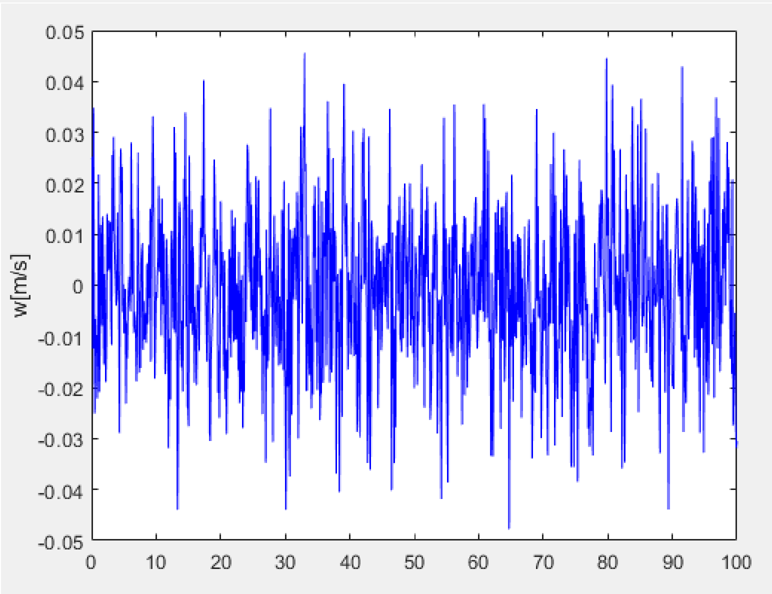
- Kalman Filtresi, Konum, Hata



- Kalman Filtresi, Hız, Hata

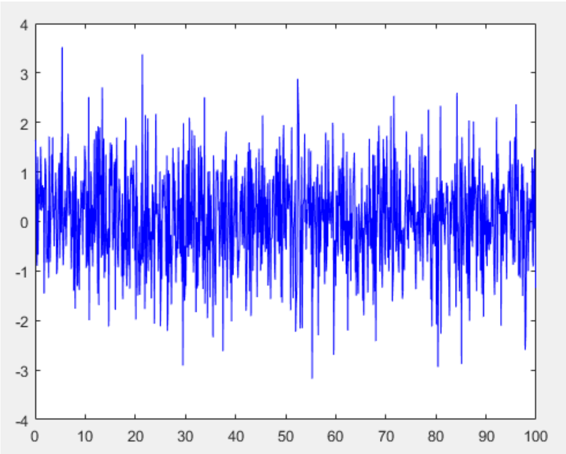




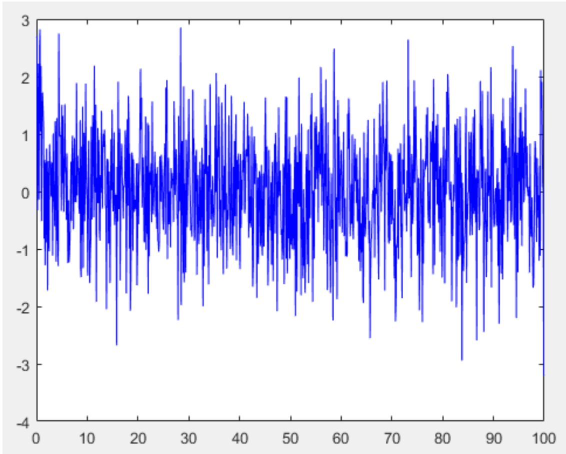


- Normalize inovasyon

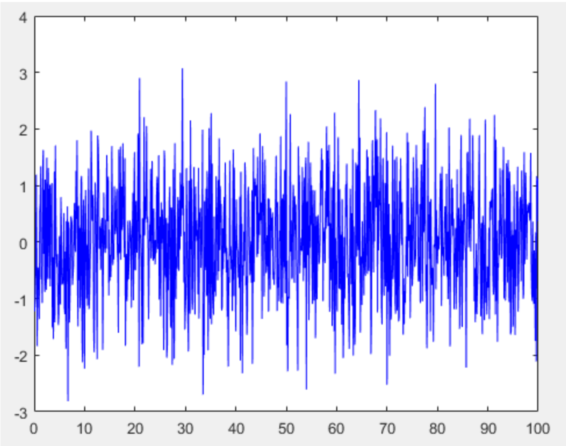
1



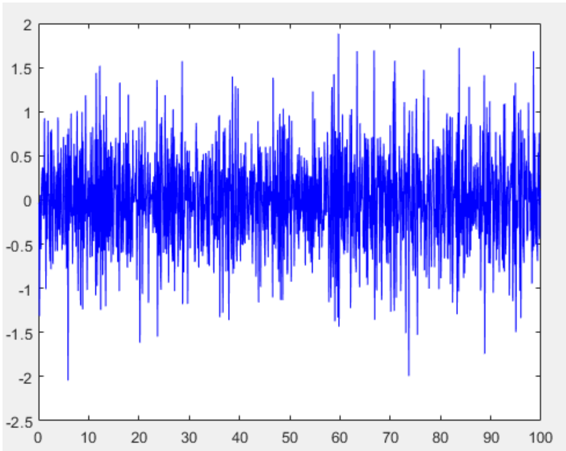
2



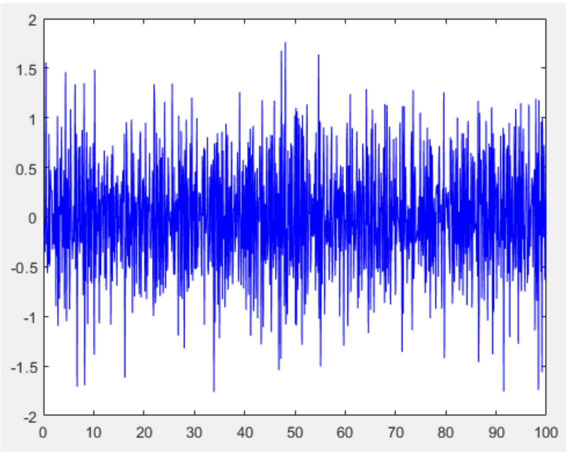
3



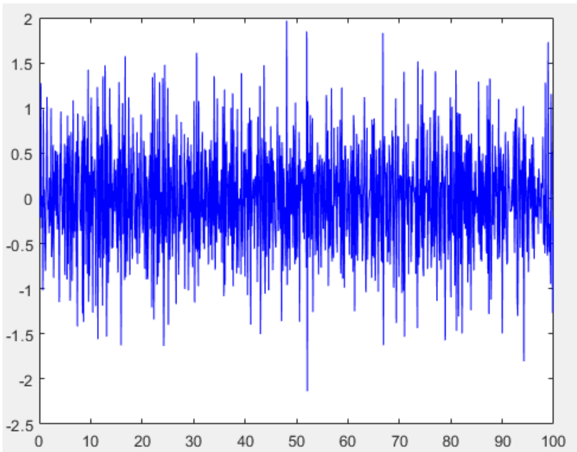
4



5

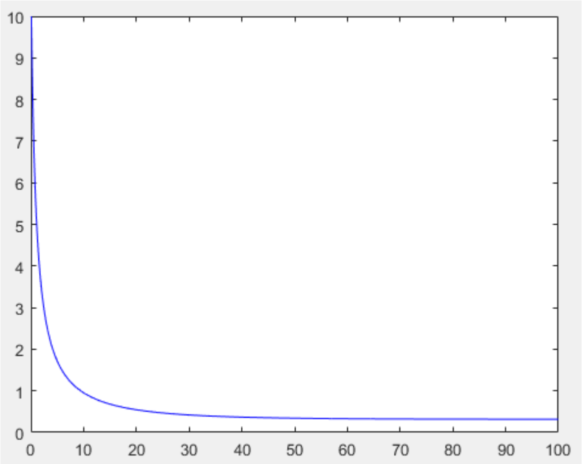


6

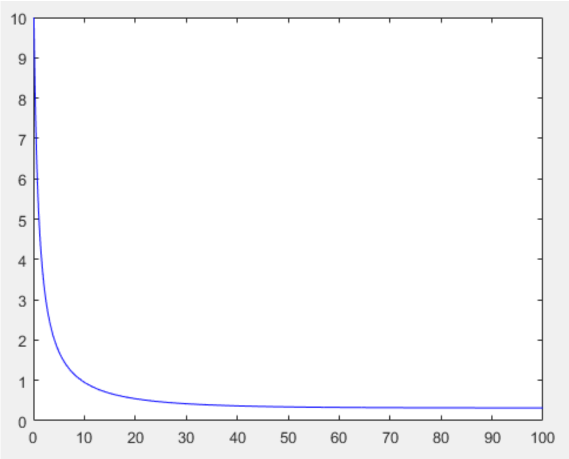


- P Diagonal

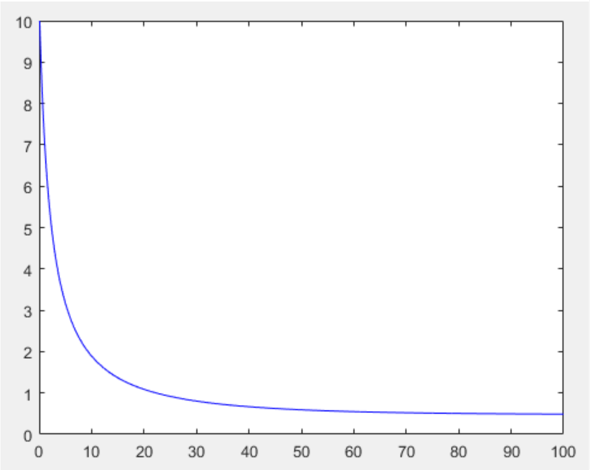
1



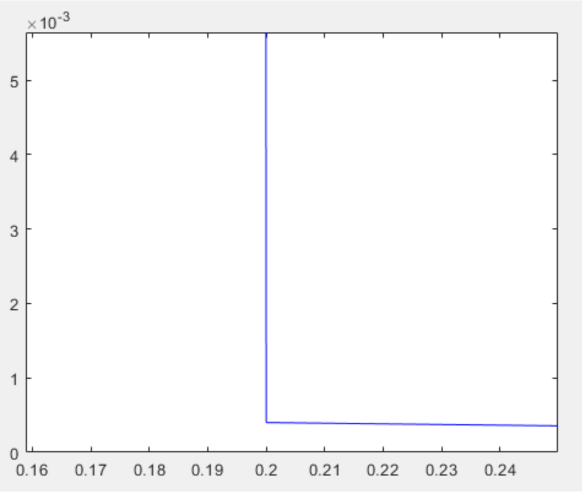
2



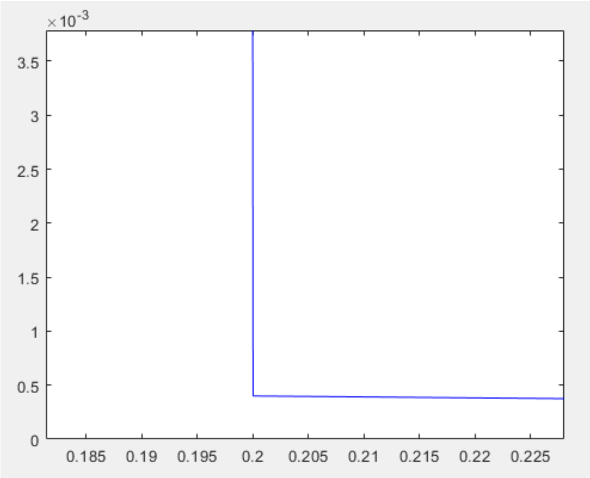
3



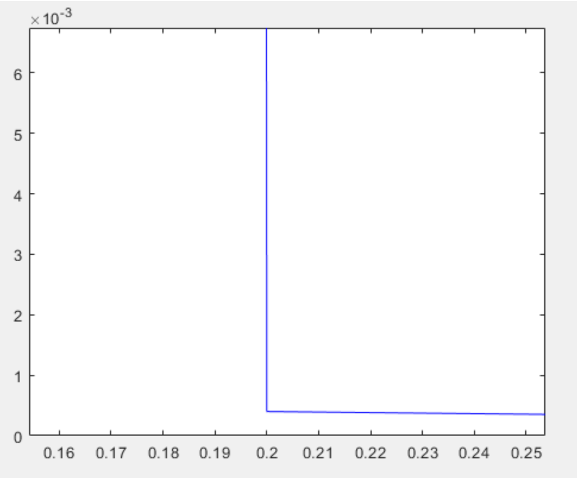
4



5



6



	İlk	Son	Kat
σ_x	10	0.5634	17.7494
σ_y	10	0.5634	17.7494
σ_z	15	0.6984	14.3184
σ_u	0.02	0.0173	1.1561
σ_v	0.02	0.0173	1.1561
σ_w	0.02	0.0173	1.1561

(Son, P diagonalin karekökü)

6 Ek

- Teori, Ölçüm, Kalman Filtresi Kodları

```
x(1) = 10000000;  y(1) = 20000000;  z(1) = 26925824.03567252;   u(1) = 1000; v(1) = 1000; w(1)
= 2724.866969229874;  r_v(:,1)=[x(1);y(1);z(1)]; r_b(1)=norm(r_v(:,1));
V_v(:,1)=[u(1);v(1);w(1)]; V_b(1)=norm(V_v(:,1));
yy = 6.6742 * (10 ^ (-11)); M = 5.97214 * (10 ^ (24)); m_u = yy * M;
sure = 100; deltaT = 0.1; n = sure/deltaT; aralik=deltaT:deltaT:sure;
Gamax = 10;Gamay = 10; Gamaz = 15; Gamau = 0.02; Gamav = 0.02; Gamaw = 0.02;
for i = 1: (n-1)
    x(i+1) = x(i) + deltaT * u(i); y(i+1) = y(i) + deltaT * v(i); z(i+1) = z(i) + deltaT *
w(i); r_v(:,(i+1))=[x((i+1));y((i+1));z((i+1))]; r_b((i+1))=norm(r_v(:,(i+1)));
    u(i+1) = u(i) + deltaT * (((-m_u)*x(i))/((r_b(i))^3)); v(i+1) = v(i) + deltaT * (((-
m_u)*y(i))/((r_b(i))^3));
    w(i+1) = w(i) + deltaT * (((-m_u)*z(i))/((r_b(i))^3));
    V_v(:,(i+1))=[u((i+1));v((i+1));w((i+1))]; V_b((i+1))=norm(V_v(:,(i+1)));
end
for k = 1 : n
    x1(k) = x(k) + Gamax * randn; y1(k) = y(k) + Gamay * randn; z1(k) = z(k) + Gamaz * randn;
    u1(k) = u(k) + Gamau * randn; v1(k) = v(k) + Gamav * randn; w1(k) = w(k) + Gamaw * randn;
end
olcum_v=[x1;y1;z1;u1;v1;w1];
olcum_r_v=[x1;y1;z1]; olcum_V_v=[u1;v1;w1];
P=10*eye(6); G=eye(6); Q=0.001*eye(6); H=eye(6);
P_diag(1,:)=[P(1,1) P(2,2) P(3,3) P(4,4) P(5,5) P(6,6)];
R(1,1)=Gamax^2; R(2,2)=Gamay^2; R(3,3)=Gamaz^2; R(4,4)=Gamau^2; R(5,5)=Gamav^2;
R(6,6)=Gamaw^2;
kalman_x(1)=x1(1); kalman_y(1)=y1(1); kalman_z(1)=z1(1);
kalman_u(1)=u1(1); kalman_v(1)=v1(1); kalman_w(1)=w1(1);
kalman_r_v=r_v(:,1); kalman_r_b=r_b(1); kalman_V_v=V_v(:,1); kalman_V_b=V_b(1);
for i=1:(n-1)
    J(1,1)=1; J(1,2)=0; J(1,3)=0; J(1,4)=deltaT; J(1,5)=0; J(1,6)=0;
    J(2,1)=0; J(2,2)=1; J(2,3)=0; J(2,4)=0; J(2,5)=deltaT; J(2,6)=0;
    J(3,1)=0; J(3,2)=0; J(3,3)=1; J(3,4)=0; J(3,5)=0; J(3,6)=deltaT;
    J(4,1)=-(deltaT*m_u*(- 2*kalman_x(i)^2 + kalman_y(i)^2 + kalman_z(i)^2))/(kalman_x(i)^2 +
kalman_y(i)^2 + kalman_z(i)^2)^(5/2);
    J(4,2)=(3*deltaT*m_u*kalman_x(i)*kalman_y(i))/(kalman_x(i)^2 + kalman_y(i)^2 +
kalman_z(i)^2)^(5/2);
    J(4,3)=(3*deltaT*m_u*kalman_x(i)*kalman_z(i))/(kalman_x(i)^2 + kalman_y(i)^2 +
kalman_z(i)^2)^(5/2);
    J(4,4)=1; J(4,5)=0; J(4,6)=0;
    J(5,1)=(3*deltaT*m_u*kalman_x(i)*kalman_y(i))/(kalman_x(i)^2 + kalman_y(i)^2 +
kalman_z(i)^2)^(5/2);
    J(5,2)=-(deltaT*m_u*(kalman_x(i)^2 - 2*kalman_y(i)^2 + kalman_z(i)^2))/(kalman_x(i)^2 +
kalman_y(i)^2 + kalman_z(i)^2)^(5/2);
    J(5,3)=(3*deltaT*m_u*kalman_y(i)*kalman_z(i))/(kalman_x(i)^2 + kalman_y(i)^2 +
kalman_z(i)^2)^(5/2);
    J(5,4)=0; J(5,5)=1; J(5,6)=0;
    J(6,1)=(3*deltaT*m_u*kalman_x(i)*kalman_z(i))/(kalman_x(i)^2 + kalman_y(i)^2 +
kalman_z(i)^2)^(5/2);
    J(6,2)=(3*deltaT*m_u*kalman_y(i)*kalman_z(i))/(kalman_x(i)^2 + kalman_y(i)^2 +
kalman_z(i)^2)^(5/2);
    J(6,3)=-(deltaT*m_u*(kalman_x(i)^2 + kalman_y(i)^2 - 2*kalman_z(i)^2))/(kalman_x(i)^2 +
kalman_y(i)^2 + kalman_z(i)^2)^(5/2);
    J(6,4)=0; J(6,5)=0; J(6,6)=1;
    x_prime = kalman_x(i) + deltaT*(kalman_u(i)); y_prime = kalman_y(i) +
deltaT*(kalman_v(i)); z_prime = kalman_z(i) + deltaT*(kalman_w(i));
    u_prime = kalman_u(i) + deltaT * (((-m_u)*kalman_x(i))/((kalman_r_b(i))^3));
    v_prime = kalman_v(i) + deltaT * (((-m_u)*kalman_y(i))/((kalman_r_b(i))^3));
    w_prime = kalman_w(i) + deltaT * (((-m_u)*kalman_z(i))/((kalman_r_b(i))^3));
    prime_v = [x_prime;y_prime;z_prime;u_prime;v_prime;w_prime];
    P_t=((J)*(P)*(J'))+((G)*(Q)*(G')); S=(H*P_t*H'+R); K=P_t*H'*(S^-1);
    Z(:,i)=olcum_v(:,(i+1))-prime_v; kalman_son=prime_v+(K*Z(:,i)); N_i(:,i)=(S^(-
1/2))*Z(:,i);
    kalman_x(i+1) = kalman_son(1); kalman_y(i+1) = kalman_son(2); kalman_z(i+1) =
kalman_son(3);
    kalman_u(i+1) = kalman_son(4); kalman_v(i+1) = kalman_son(5); kalman_w(i+1) =
kalman_son(6);
    kalman_r_v(:,(i+1))=[kalman_x(i+1);kalman_y(i+1);kalman_z(i+1)];
kalman_r_b(i+1)=norm(kalman_r_v(:,i+1));
    kalman_V_v(:,(i+1))=[kalman_u(i+1);kalman_v(i+1);kalman_w(i+1)];
kalman_V_b(i+1)=norm(kalman_V_v(:,i+1));
    kalan_P_son=((eye(6)-(K)*(H))*P_t); P=kalan_P_son;
    P_diag(i+1,:)=[P(1,1) P(2,2) P(3,3) P(4,4) P(5,5) P(6,6)];
end
```

- Uydu Hareket Simülasyonu Kodları

https://github.com/alpersanli/Satellite-Motion-Simulation/blob/master/UyduHareketVeOlcumSimulasyonu.py

https://github.com/alpersanli/Satellite-Motion-Simulation/tree/master

7 Kaynaklar

[1] Kalman Filtsi, https://tr.wikipedia.org/wiki/Kalman_Filtresi

[2] Kalman Filtsi Temeli, https://tr.wikipedia.org/wiki/Dosya:Kalman_filtresinin_temeli.svg

[3] Kalman Filtsi Dersi, https://kasnakoglu.files.wordpress.com/2014/01/ders7_ck01.pdf

[4] Kalman Filtsi Nedir?, https://medium.com/@syndrome/kalman-filter-nedir-51c38a12c423

[5] Hacıyev, C., (2010). Deney Verilerinin İşlenme Yöntemleri ve Mühendislik Uygulamaları.

[6] Kalman Filtsi, https://guraysonugur.aku.edu.tr/wp-content/uploads/sites/11/2018/05/Kalman_Filtresi_IbrahimCayiroglu.pdf

[7]http://www.emo.org.tr/ekler/9afa3a1bba5280a_ek.pdf

[8] Kalman Filtsi ve Bir Programlama Örneği, http://www.ibrahimcayiroglu.com/Dokumanlar/Makale_BilgiPaylasim/(1-2012)-Kalman_Filtresi_Ve_Bir_Programlama_Ornegi-Ibrahim_CAYIROGLU.pdf