

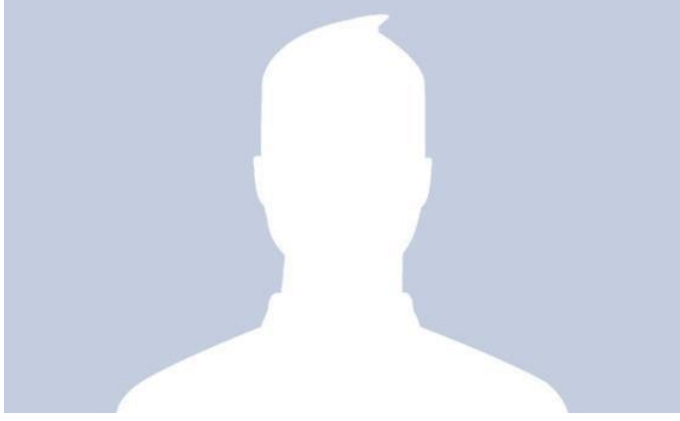


TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ

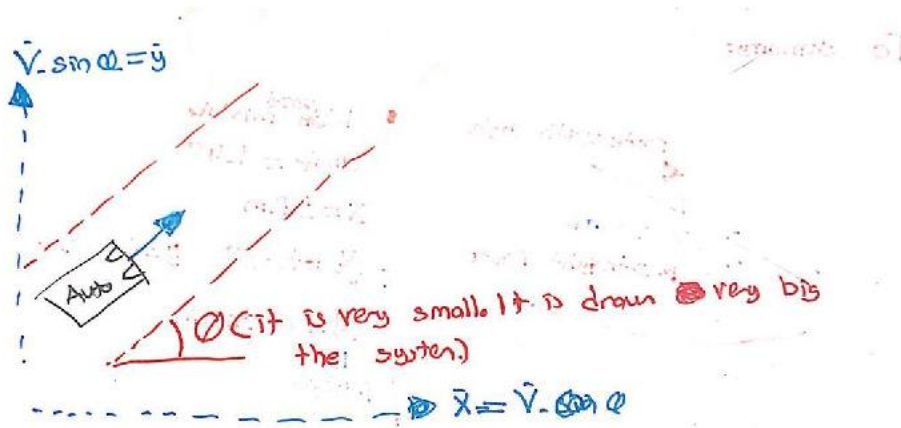
**ELE 515 – MATLAB İLE BILGISAYARLI KONTROL(2019-2020
Bahar)**

Proje Raporu

	
İsim Soyisim	Batu Kaan Özen
Öğrenci Numarası	141201079
Tarih	07/02/2020

Problem:

Dünya genelindeki düz yollar genellikle tam olarak düz değiller bundan dolayı hiç bir zaman arabamızın cruise controllünde hız sabitlenip ve direksiyonun sabitlenerek bir yerden bir yere gidilmesi gibi bir durum yapılamamaktadır. Bizim bu projedeki amacımız bu düze benzeyen ama az bir açıdan dolayı bozulan rotalarda yaptığımız kontrolcü sayesinde rotanın takip edilmesi işlemidir. Bu control sistemi yapıldıktan sonra bu açı arabamızın A_x ve A_y ivmeleri değişimi ile ayarlanıp regülasyonu yapılacaktır. Bu sayede örneğin kimsenin olmadığı düz bir yolda arabamızın direksiyonu kitleyip saatlerce bu kontrolcüyü kullanarak yolculuk yapabiliriz. Resim 1-1 de bu sistemi görmekteyiz. Raporumu daha iyi anlamak için lütfen 515projerapo.pdf dosyasını ilk okuyun.



Resim 1-1

Sistemizin Mekanik Eşitlikleri:

Resim 1.2 de sistemimizin genel denklemlerini görmekteyiz.

$$Ay = A * \sin(Q)$$

$$Ax = A * \cos(Q)$$

$$\frac{dv(t)}{dt} = A$$

$$\frac{dX(t)}{dt} = V$$

$$\frac{dQ(t)}{dt} = W$$

$$\frac{dw(t)}{dt} = a$$

$$\tan(Q) = Vy/Vx$$

$$\tan(w) = Ay/Ax$$

$$Ax = A * \cos(Q)$$

Resim 1.2

Sistemimizin Lineerleştirilmesi:

Sistemi lineerleştirmek için aşağıdaki varsayımı yapıyoruz.

$$\sin(Q) = Q$$

$$\cos(Q) = 1$$

Sistem lineerleştikten sonra:

Sistemimizin hızının 80km/h ve zaman değişiminin 0.001 sn olduğunu varsayıp, ivmenin sabit bir değer olduğunu varsayıyoruz. Aşağıdaki denklem sistemi ile açısal hız, açısal ivme ve açısal konum aşağıdaki denklemlerle bulunuyo.

$$a(K) = a(K + 1)$$

$$w(k + 1) = w(k) + a(k)$$

$$Q(k + 1) = Q(k) + W(k) + a(K)/w$$

Ve daha sonra Resim 1.3 deki state space değerleri olan Am,Bm,Cm,Dm değerlerini yukardaki denklemlerden elde ediyoruz.

$A_m =$

$$\begin{bmatrix} 1.0000 & 1.0000 & 0.5000 \\ 0 & 1.0000 & 1.0000 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$B_m =$

$$\begin{bmatrix} 0 \\ 0 \\ 0.0125 \end{bmatrix}$$

$C_m =$

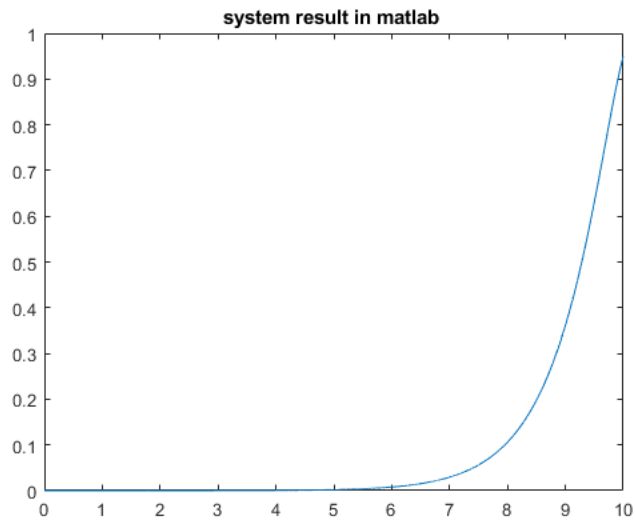
$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$D_m =$

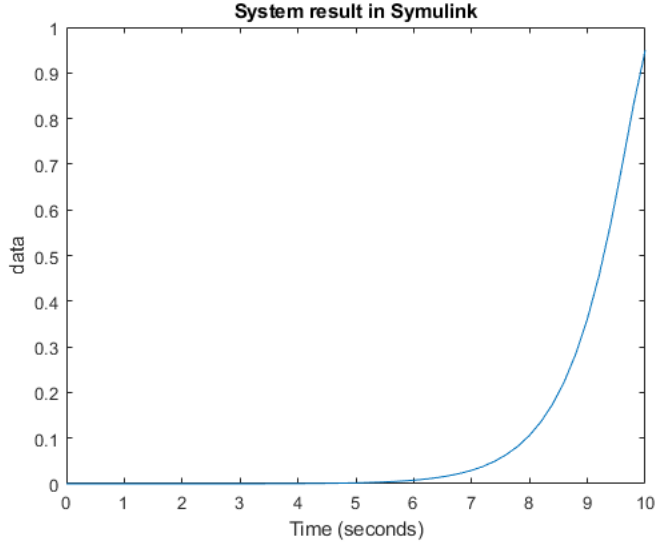
$$1$$

Resim 1.3

Bu aşamada sistemimizin 0.0001 derecelik bir hatadan dolayı matlab ve smulink ortalamında yaklaşık olarak 1m lik acculatif hata bulduk, Sonuçları Resim 1.4 ve Resim 1.5 de görebilirsiniz.



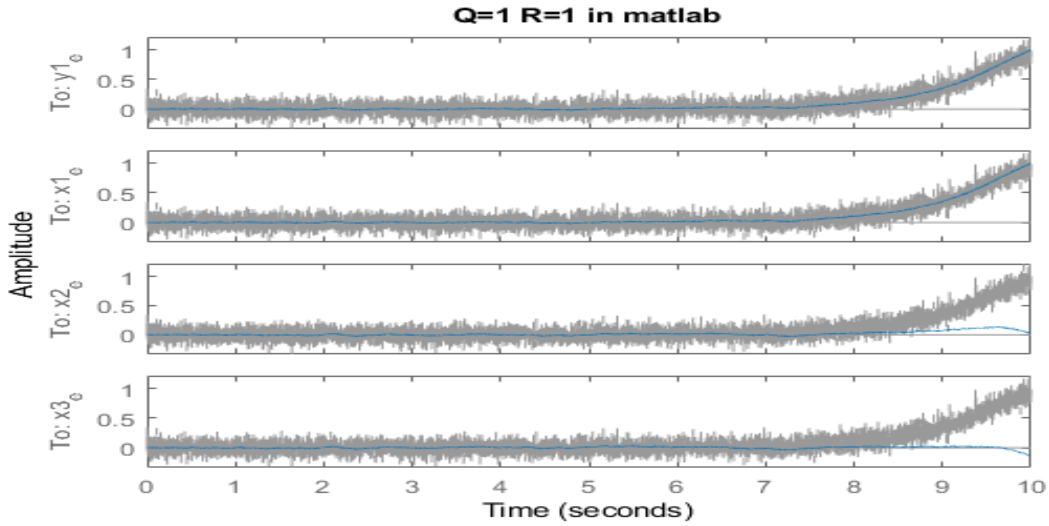
Resim 1.4



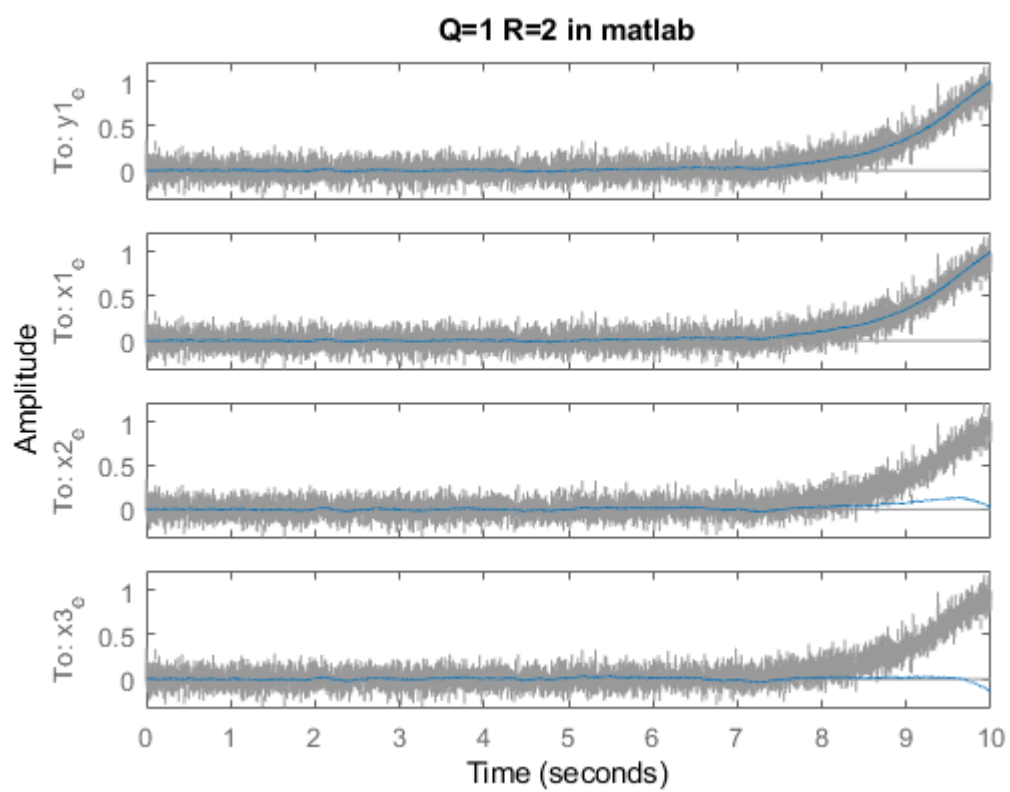
Resim 1.5

İkinci aşama:

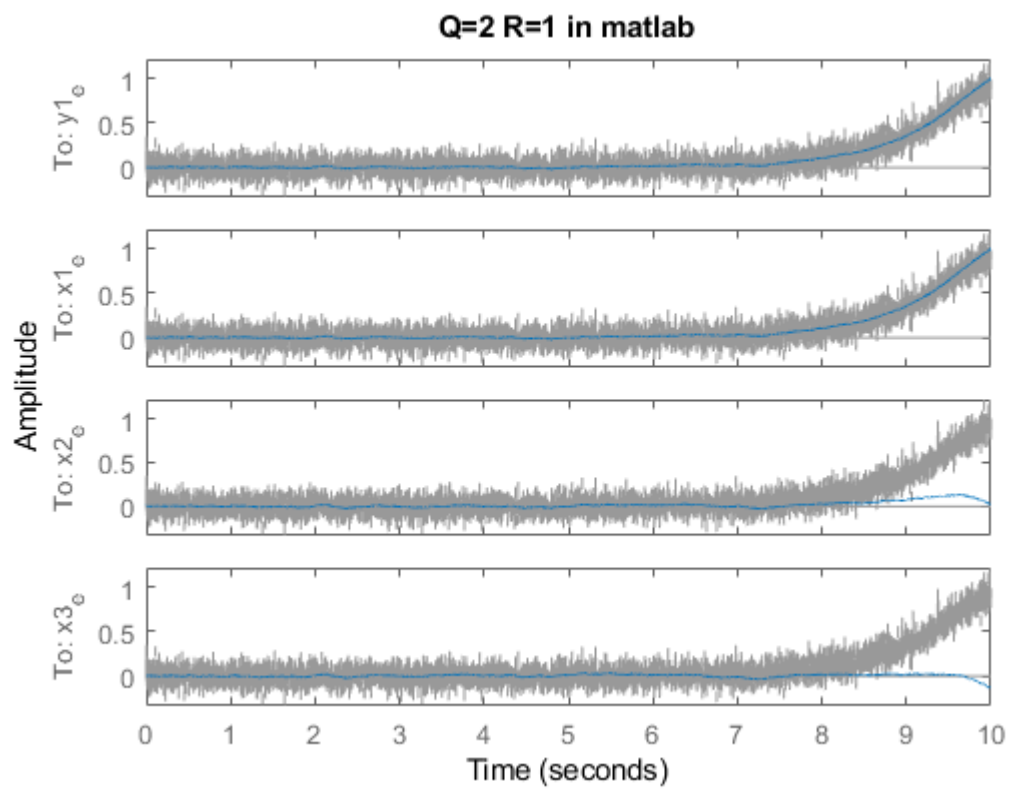
Bu aşamada sistemimizi yukarda anlatığımız gibi linearize edip daha sonra Kalman filteri uygulayıp, farklı değerler için ve linear edilmemiş sonucunu Resim 1.6, Resim 1.7, Resim 1.8 ve Resim 1.9 da bulabilirsiniz da bulabiliriz burda $Q>R$ alınca modelle ilgili sıkıntıların baskın olduğunu ve $R<Q$ olunca gürültü hatasının baskın olduğunu söylüyoruz ve sonuçlarda beklediği gibi gelmektedir.



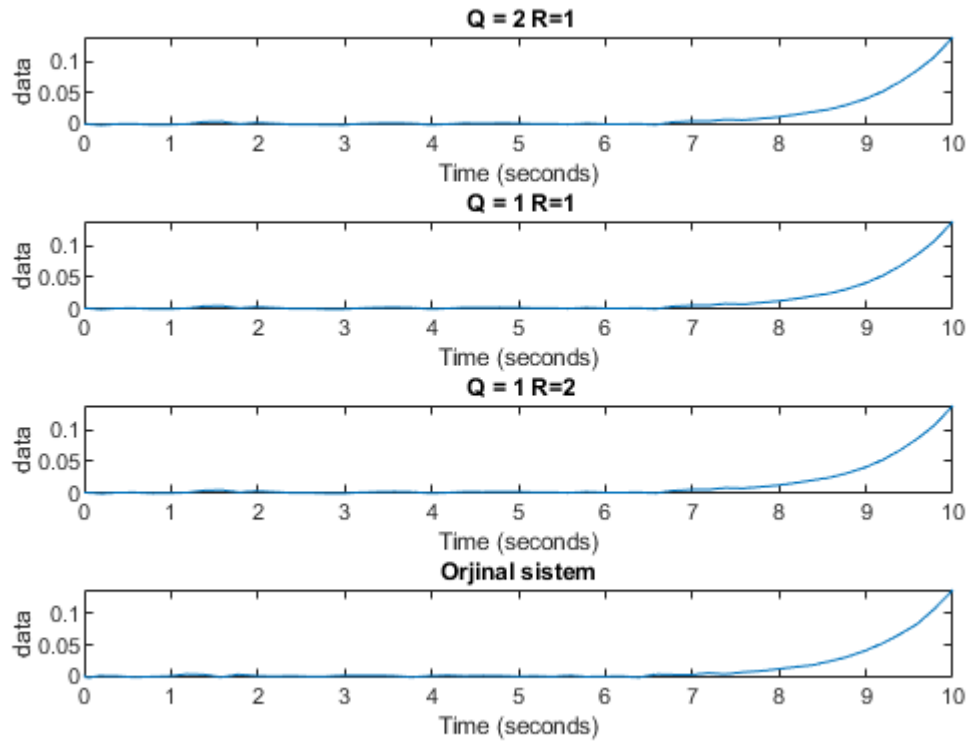
Resim 1.6



Resim 1.7



Resim 1.8

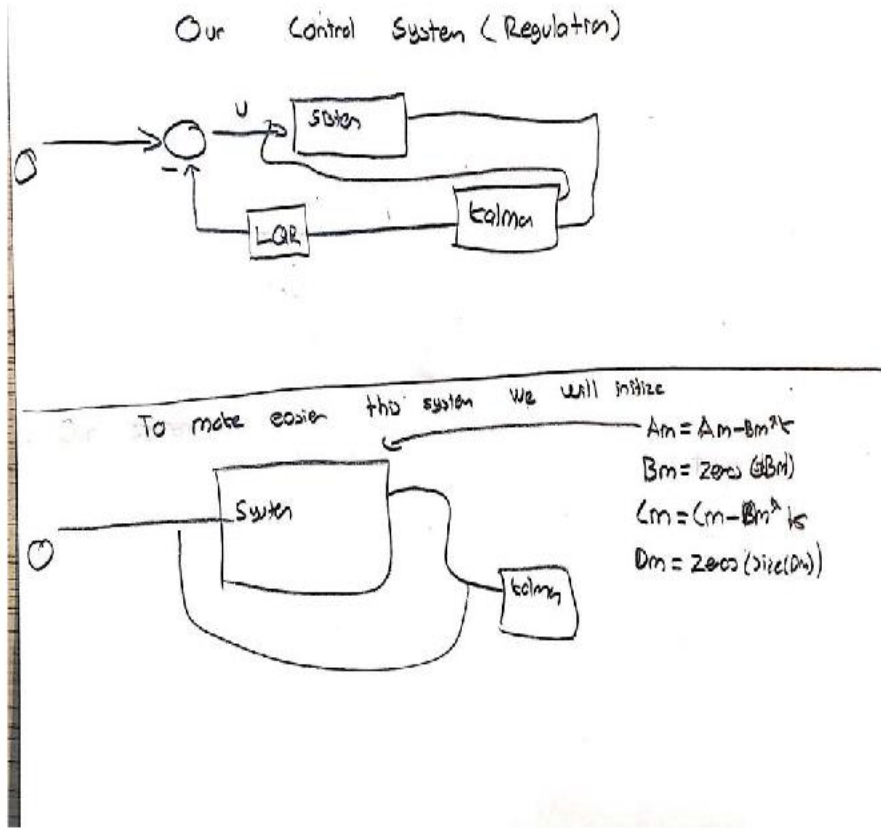


Resim 1.9

Projenin Son aşaması:

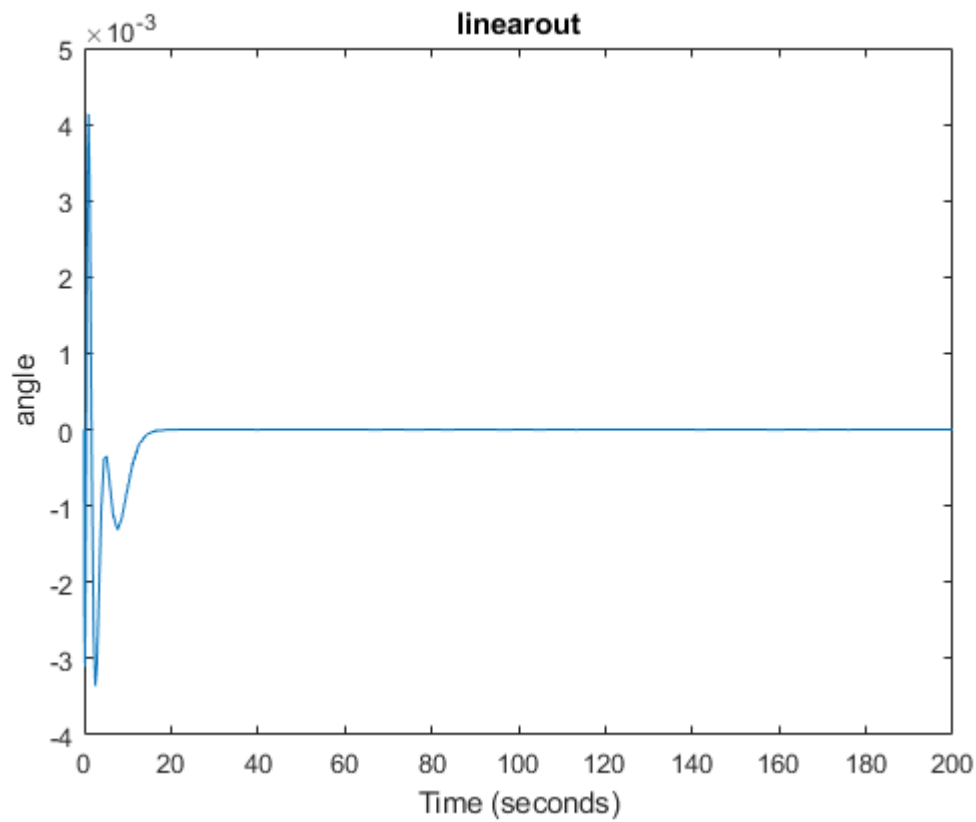
Projenin bu aşamasında ise Resim 1.10 da gördüğünü kontrolcü sistemini tasarladım ve sistemi basitleştirmek için State space elamanlarına Resresyon çeklinde aşağıdaki gibi tanımladım bu sayede simulinkte gereksiz bir büyüklükte çizmekten kurtuldum

```
G= ss (Am-Bm*K , zeros (size (Bm)) , Cm-Dm*K , zeros (size (Dm)) );
```

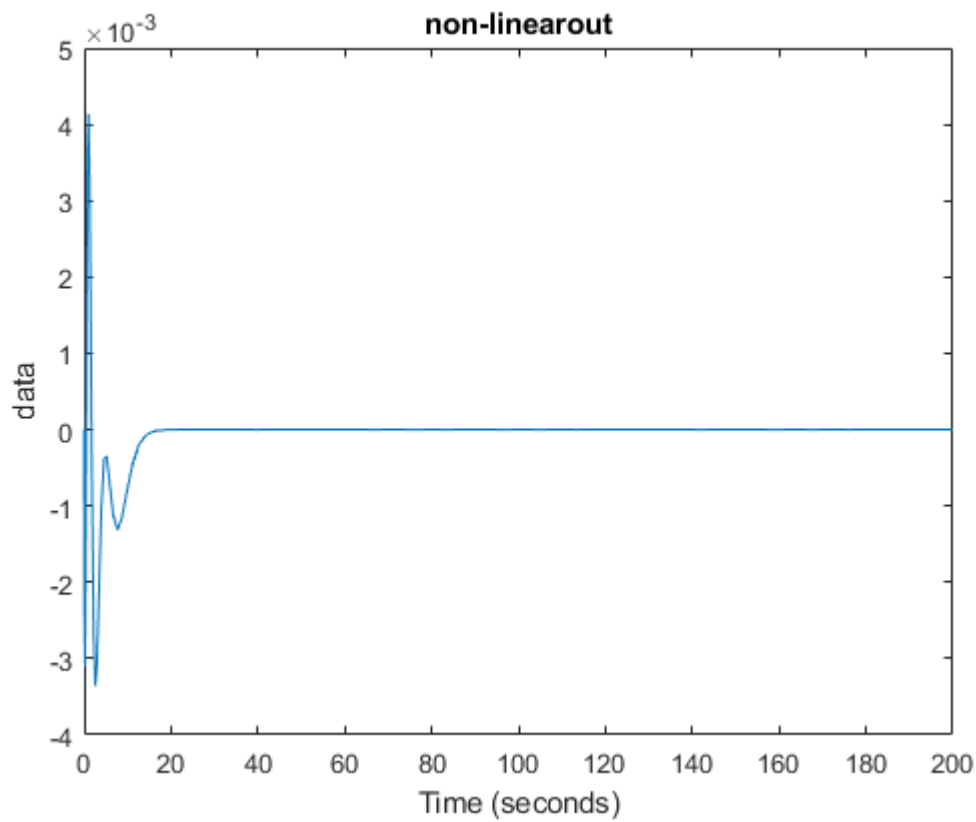


Resim 1.10

Regresyon problemimizin sonuçlarının da linear ve non linear sistemler için Resim 1.11 ve Resim 1.12 deki gibi buldum.



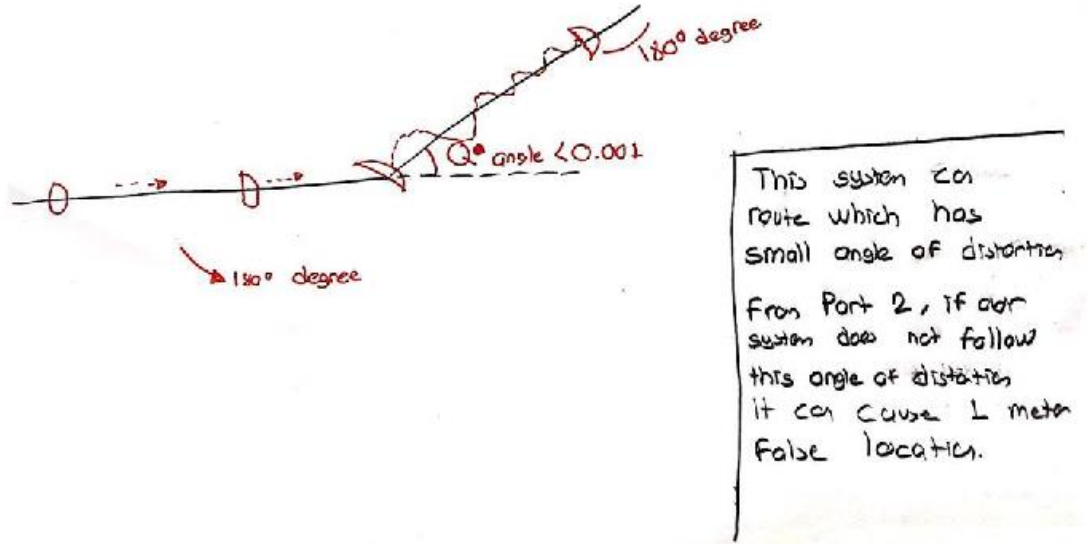
Resim 1.11



Resim 1.12

Sonuç:

Sonuç olarak, bir unicyler arababının düz bir yol üzerinde sabit gitmesi problemini çözmüş olduk, yoldaki 0.0001 derecelik açı hatasından dolayı sürekli olarak bir yolda çıkma uzun sürede çok büyük hatalara sebep olmaktaydı bu durumdan dolayı sistemimiz sürekli olarak dümdüz az açı hatası olan yolda gidememekteydi bunu otonom yapmak için bir adet kontrolcü tasarlayıp regülasyon problemi şeklinde çözdük bu işlem yapılırken Y eksenindeki ivme ve X eksenindeki ivmenin değerleri oynatarak acısak ivme elde edilip o acısal ivme kullanarak sistemdeki hatayı 0 a getirdik değer tahminlerinin de noiseden etkilenmemesi ve modellemedeki hatatadan etkilenmemesi için kalman ile yaptık. Resim 1.13 de sistemin özet olarak amacını görebilirsiniz. Sistemi daha iyi anlamak için



Resim 1.13