

Program: Otonom Sürüş Teknolojileri
Uzmanlık Programı

Konu: Otonom Araçlarda Hareket Planlama
ve Kontrol

Proje: Şerit Değiştirme ve Akıllı Hız Asistanı
Fonksiyonlarının Tasarımı

Hazırlayan

Adı: Fatih

Soyadı: Küçükbiyık

Okul: Karadeniz Teknik Üniversitesi

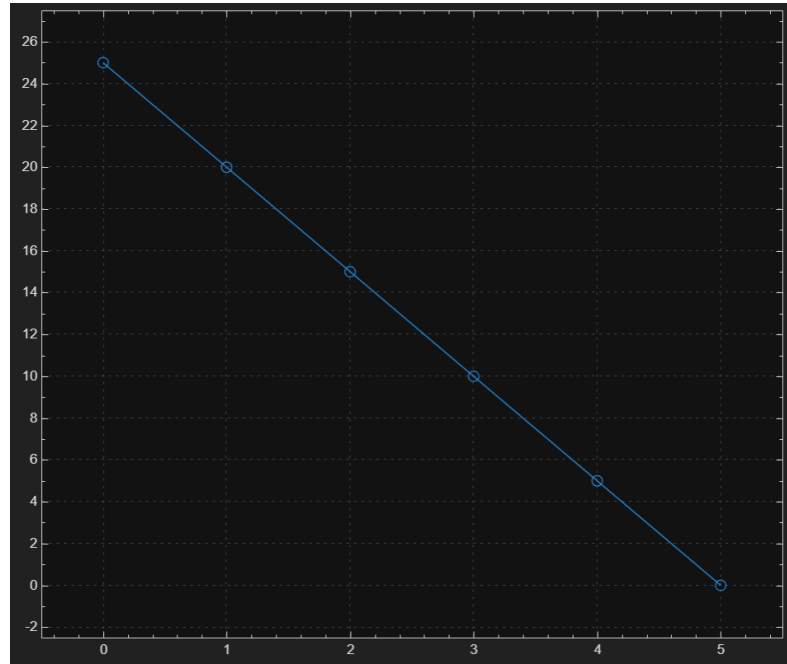
1.) Frenleme ve Dümenleme ile Kaçınma

$a_x = -5\text{m/s}^2$ ile acil frenleme yapıldığı için ay değeri 0 olur. Bu durumda a_x , max değeri de

$$\sqrt{a_x^2 + 0} = |a_x| \leq \mu g$$

Şeklinde hesaplanır. $\mu = 0.15$ ve $\mu = 0.5$ değerleri için araç fren yapamayıp kayacaktır bu yüzden sadece $\mu = 0.85$ değeri kullanılmıştır.

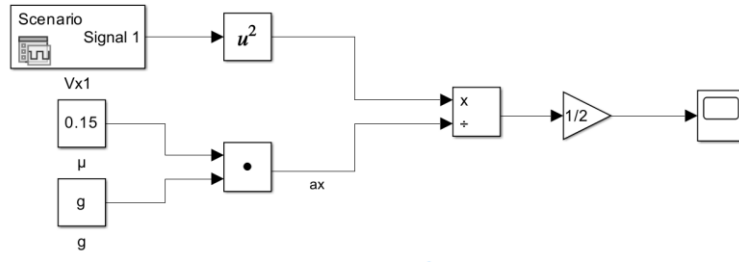
$a_x = -5\text{m/s}^2$ ile durduğu için araç 5 saniye sonunda durmuş olur. Böylece şekildeki gibi bir hız grafiği oluşturulabilir.



Araç acil frenleme yaptığı için 2. Denklemin kullanılması ile durma mesafesi hesaplanabilir.

$$s_s = \frac{1}{2} \frac{v_{x0}^2}{a_{x,max}}$$

Matlab' de bu denklem aşağıdaki şekildeki gibi modellenmiştir.

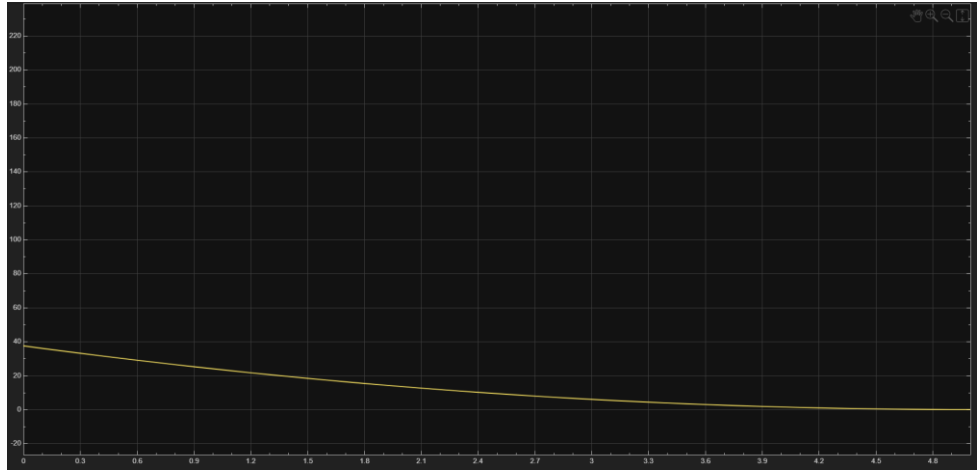


Kullanılan sabitlerin script dosyası aşağıdaki gibidir.

```

1      ax = -5;
2      Vx0 = 25;
3      g = 9.81;
4      d = 3.5;
5      mu = 0.85;
6      sim("Odev1.slx")
  
```

Ss değerinin zamana bağlı olarak grafiksel gösterimi:



2 Yörünge Üretme

$T = 3$ ve $\Delta s = 2$ için

Bizim Arah

$$s_0 = 0$$

$$\dot{s}_0 = 25$$

$$\ddot{s}_0 = 0$$

Çadeki Arah

$$s_{IV}(0) = 50m$$

$$\dot{s}_{IV} = 20m/s$$

$$\ddot{s}_{IV} = 0$$

$$\Delta s = 5 \text{ (Güvenli mesafe)}, T = 1,5 \text{ s}, \Delta s = 2$$

$$s_{IV}(3) = 50 + 20 \cdot 3 = 110m$$

$$s_{target}(3) = 110 - (5 + 1,5 \cdot 20) = 110 - 35 = 75m$$

$$s_1 = 75m$$

olarak bulunur.

$$\dot{s}_1 = 20m/s$$

$$\ddot{s}_1 = 0m/s^2$$

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

yörüngeyi belirleyen katsayıları aşağıdaki denklemler ile bulabiliriz

$$s_0 = a_0 = 0$$

$$s(3) = 75$$

$$s(3) = 27a_3 + 81a_4 + 243a_5 = 0$$

$$\dot{s}_0 = a_1 = 25$$

$$\dot{s}(3) = 20$$

$$\dot{s}(3) = 27a_3 + 108a_4 + 405a_5 = -5$$

$$\ddot{s}_0 = a_2 = 0$$

$$\ddot{s}(3) = 0$$

$$\ddot{s}(3) = 18a_3 + 108a_4 + 540a_5 = 0$$

Bundan

$$a_3 = 2,219$$

$$a_4 = -1,235$$

$$a_5 = 0,185$$

olarak bulunur.

$$s(t) = 25t + 2,219t^3 - 1,235t^4 + 0,185t^5$$

$$\dot{s}(t) = 12,314 - 31,08t + 11,1t^2 \text{ denklemini ile Jark maliyetini}$$

hesaplırsak

$$J_t(P) = \int_0^3 (12,314 - 31,08t + 11,1t^2)^2 dt = 1676,412$$

olarak bulunur.

$$C_t = k_5 (1674,412) + k_t 3 + k_s (2)^2$$

$$C_t = k_5 (1674,412) + k_t 3 + k_s 4$$

Sıra

	k_5	k_t	k_s
1	10	1	1
2	1	10	1
3	1	1	10

2. Değerler için:

$$C_t = 16747,412$$

2. Değerler için

$$C_t = 1708,412$$

3. Değerler için

$$C_t = 1717$$

Şeklinde bulunur.

Burada maliyeti en çok arttıran k_5 değeri olduğu görülmektedir.

```

1  % 1. Parametrelerin Tanımlanması
2  %Referans yol parametreleri
3  dy1 = 3.5;
4  dy2 = 3.5;
5  dx1 = 25;
6  dx2 = 25;
7  ds1 = 60;
8  ds2 = 130;
9  alpha = 2.4;
10
11 % Araç hızı
12 vx = 20;
13
14 % Kazançlar (oransal kontrol)
15 ky = 0.8;
16 kpsi = 1.2;
17
18 % 2. Referans Yol Fonksiyonlarının Tanımlanması
19 X = 0:0.5:200; % Aracın ilerlediği X konumları
20
21 z1 = (alpha/dx1)*(X - ds1) - alpha/2;
22 z2 = (alpha/dx2)*(X - ds2) - alpha/2;
23
24 Yref = (dy1/2) * (1 + tanh(z1)) - (dy2/2)*(1 + tanh(z2));
25
26 % 3. Yönelim Açıları
27 dYref_dX = (dy1 * (1 ./ cosh(z1)).^2) * (alpha / (2*dx1)) - (dy2 * (1 ./ cosh(z2)).^2*(alpha/
28 (2*dx2)));
29 psiref = atan(dYref_dX); %Referans yönelim açısı(radyan)
30
31
32 % 4. Hata Hesaplama
33
34 Y_arac = zeros(size(X)); % Araç y konumu (hep 0)
35 psi_arac = zeros(size(X)); % Araç yönelimi (hep 0)
36
37 ey = Yref - Y_arac;
38 epsi = psiref - psi_arac;
39
40 % 5. Eğrilik Talebi Hesaplanması
41 kappa_des = ky * ey + kpsi * epsi;
42
43 % 6. Sonuçların çizdirilmesi
44 figure;
45 subplot(3,1,1)
46 plot(X, Yref); title('Yref (Referans Yol)'); xlabel('X [m]'); ylabel('Y [m]');
47
48 subplot(3,1,2)
49 plot(X, psiref*180/pi); title('Psi_ref (Referans Yönelim)'); xlabel('X [m]'); ylabel('psi
50 [deg]');
51
52 subplot(3,1,3)
53 plot(X, kappa_des); title('Eğrilik Talebi - \kappa_{des}'); xlabel('X [m]'); ylabel('\kappa');
54

```

```

1 % 1. Parametrelerin Tanımlanması
2 s = 0:1:3000; % yol boyunca konumlar
3 ay_comfort = 2; % konforlu yanıl ivme
4 ax_comfort = 1.5; % konforlu yavaşlama (örnek değeri)
5
6 % Eğrilik bilgisi
7 kappa = zeros(size(s));
8 kappa(s > 400 & s <= 1000) = 0.005;
9 kappa(s > 1800 & s <= 2400) = 0.0025;
10
11 % Trafik hız limiti
12 vlim_traffic = zeros(size(s));
13 vlim_traffic(s <= 400) = 90;
14 vlim_traffic(s > 400 & s <= 1000) = 50;
15 vlim_traffic(s > 1000 & s <= 1800) = 70;
16 vlim_traffic(s > 1800 & s <= 2400) = 50;
17 vlim_traffic(s > 2400) = 90;
18
19 % 2.Yol eğriliğine göre hız limiti
20 vlim_road = zeros(size(s));
21 for i = 1:length(s)
22     if kappa(i) ~= 0
23         vlim_road(i) = sqrt(ay_comfort / kappa(i));
24     else
25         vlim_road(i) = inf; % eğrilik yoksa sınırsız
26     end
27 end
28
29 % 3. Gerçek hız limiti (min alınır)
30 vlim = min(vlim_road, vlim_traffic);
31
32 % 4. Yavaşlamaya Başlama Mesafesi (dtrig)
33 v_curr = 25; % mevcut hız [m/s]
34 v_target = 13.8; % yeni hız limiti [m/s]
35
36 d_trig = (v_curr^2 - v_target^2) / (2 * ax_comfort);
37 fprintf('Yavaşlamaya başlanacak mesafe: %.2f metre\n', d_trig);
38
39 % 5. Grafik Çizdir (Hız Limitleri ve Eğrilik)
40 figure;
41
42 subplot(3,1,1)
43 plot(s, kappa); title('Yol Eğriliği κ'); xlabel('Mesafe [m]'); ylabel('κ');
44
45 subplot(3,1,2)
46 plot(s, vlim_road); title('Yola göre hız limiti (v_{lim,road})'); ylabel('v [m/s]');
47
48 subplot(3,1,3)
49 plot(s, vlim); title('Son hız limiti (min trafik/yol)'); ylabel('v_{lim} [m/s]'); xlabel('Mesafe [m]');
50

```