

BİR ROLLER COASTER'IN MATLAB İLE DİNAMİK ANALİZİ

Özet

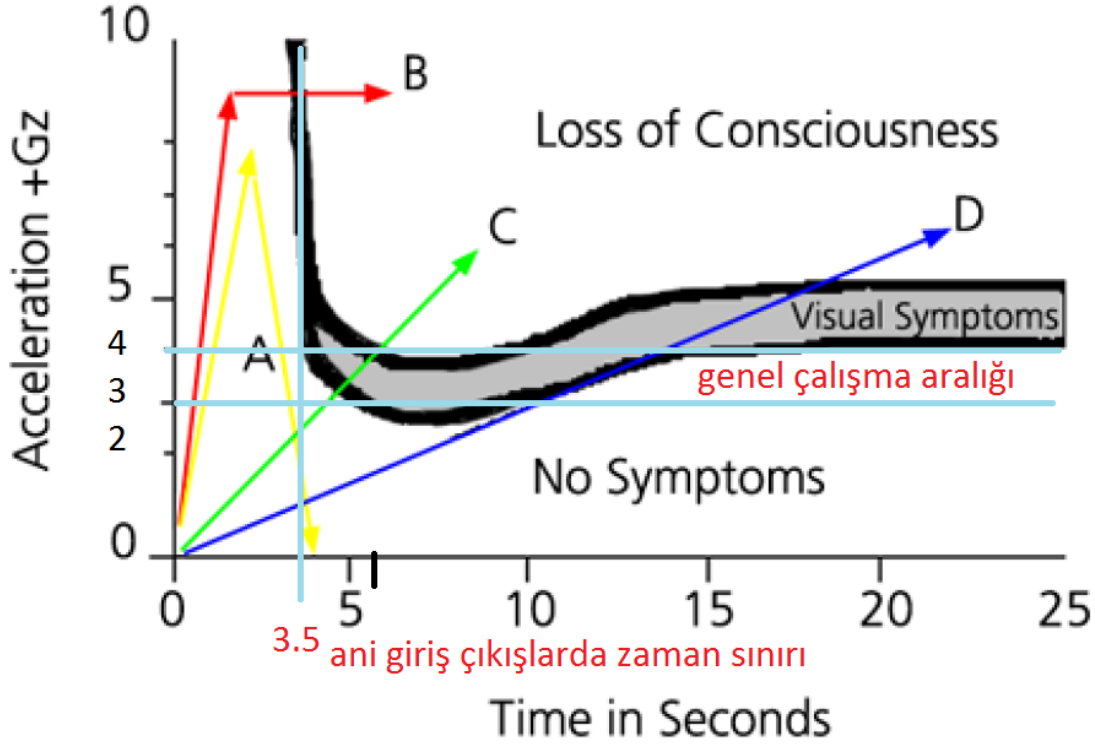
Çalışmada tren noktasal bir cisim olarak düşünülmüştür. Toplam kütle tren ve alacağı maksimum insan kütesidir. Hız trenlerinin izleyeceği pist Kartezyen koordinatta tasarlanmıştır. Trenin alması gereken maksimum hız çalışmalarla belirlenen ortalama bir insanın dayanabileceği maksimum ivme olarak düşünülmüş, bu bir sistem sınırı olarak düşünülmüştür. Burada doluluk oranındaki değişim farklı olacağından her doluluk oranı için etkiyecek hız ve ivme bulunmuş, burada ivmenin sistem sınırlarını geçmemesi için gereken yol tasarlanmıştır. Dönüşlerde tercih edilen eğrinin göstereceği ivme değişimi gözlemlenmiş, bu eğrinin kullanıcıyı nasıl etkiyeceği araştırılmıştır. Burada ağırlıklı olarak yapılan çalışmalar ile bulunan değerler karşılaştırılmıştır.

1. Giriş

Fizik yasalarının belirlenmesinin ardından bir çok çalışma insan yaşamını geliştirip niteliğini arttırmıştır. Nitelik olarak hayatın kolaylaştırılmasından yanı sıra o insanın güvenliği sağlanmış bir biçimde doğal sınırlarına yaklaştırmak da sayılabilir. Her ne kadar insan kendisi için en rahat ortamda yaşamak istese de, anlık olarak bu rahatlığından vazgeçmek isteyebilmektedir.

İnsan duyusunun en temel güdülerinden biri de hayatta kalma içgüdüsüdür. Atalarından miras kalmış bu duygular karşısında böyle bir durumda kalmamak için korkuyu bir savunma mekanizması olarak kullanır. Buna karşın böyle bir durumda kaldığındaysa o durumu kontrolü altına alıp bunun üstesinden geldiğini düşündüğü vakitte bir zafer duygusuna kapılıp bundan zevk alabilmektedir. İşte eğlence sektörü olarak eğlence parklarındaki yöneticilerin amaçladığı temel fikir de budur. Santrifüj mekanizmalar, hız trenleri, büyük G yaratan odalar insanların ivmeye karşı olan limitlerini zorlama imkanı tanımaktadır.

İnsanlara bu zevk duygusunu yaşatmak için çok çeşitli mekanizmalar tasarlanmıştır. Bu mekanizmalar, kullanıcının güvenliğinin yanında insanın fizyolojik sınırlarına da yaklaşmayı amaçladığından hassasiyetle üzerine durulması gereken bir etmen olarak görünür. Her insanın fiziksel etmenlere karşı hassasiyetleri farklıdır. Bu sebeple bu tasarımlarda en genel insan modeli kullanılıp bunlara göre işlemler belirlenmektedir. Bu sayede kullanıcıların memnuniyetinde maksimum verim sağlanmaya çalışılır.



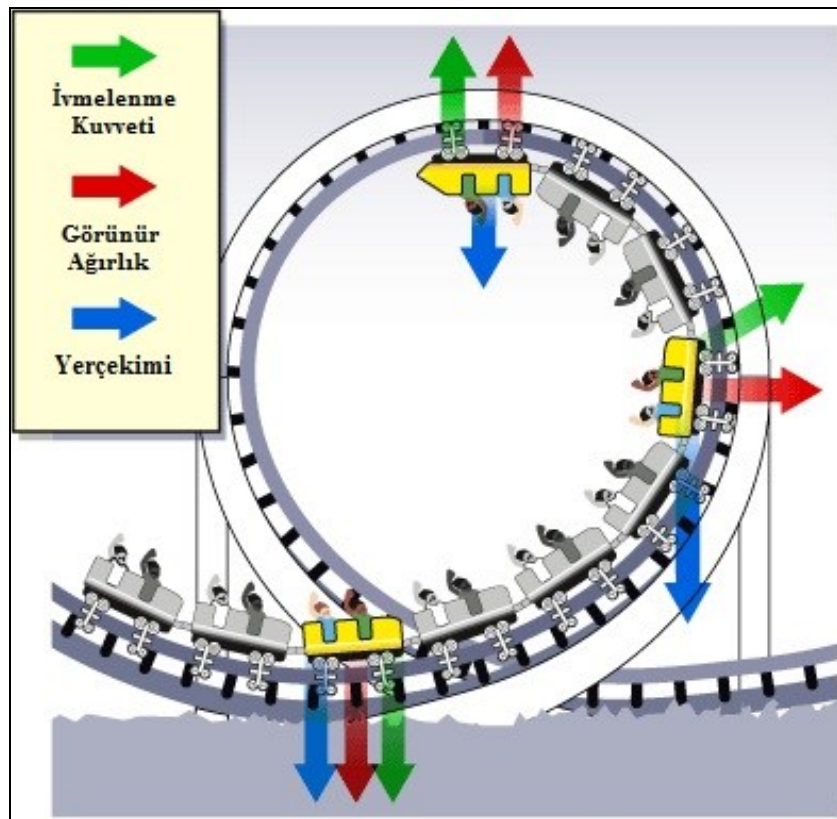
Şekil 1.1 İnsanın Fizyolojik Sınırları

Bu çalışmada hız treni olarak tabir edebileceğimiz, ‘roller coaster’ olarak geçen raylı mekanizmanın farklı yol geometrilerinde, belli bir ilk fırlatma kuvveti karşısında, yol boyunca maruz kalacağı hız ve ivme şartları bilgisayar ortamında incelenmiştir. Çalışmada Klotoid profilinin sisteme cevap olarak vereceği ivme gözlemlenmiş ve bunun kullanıcının isteyeceği koşullarını ne kadar sağladığı yorumlanmaya çalışılmıştır.

Teknoloji ve bilimin işbirliğiyle insanoğlunun yapabileceklerinin sınırı her geçen gün hızla genişlemektedir. Roller Coaster’ın ilk patentinden (20 Ocak 1885) günümüze inanılmaz gelişmeler gelişmeler olmuştur. Hayal gücünden gerçeğe aktarılabilen hareket serbestliği ve çeşitliliği sayesinde insanlara teknolojiyi eğlence olarak sunan bir mühendislik sanatı kendini tüm dünyada sevdirmiştir. Akıl almaz Roller Coaster tiplerinden birkaçı;

- 4th Dimention Roller Coaster
- Accelerator Coaster
- Diving Machine
- Flying Roller Coaster
- Hyper Coaster
- Launched Roller Coaster
- Powered Roller Coaster
- Spinning Roller Coaster
- Steel Roller Coaster'tır.

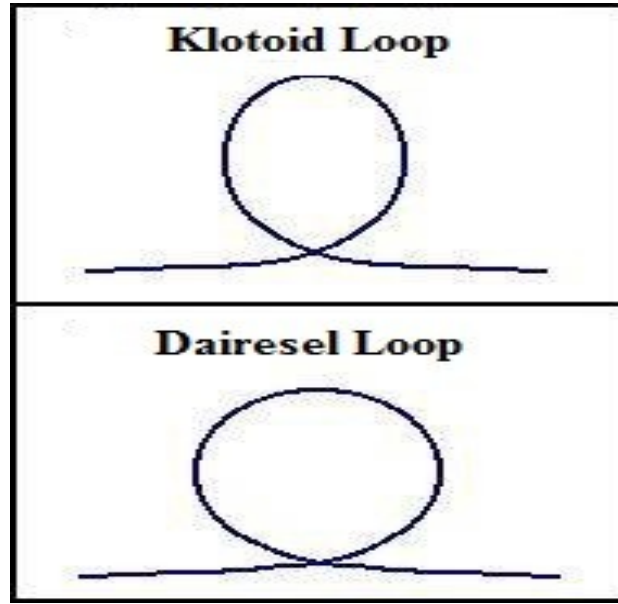
Loop: Roller Coaster'ın sonsuz radyuslu bir yoldan, sonlu rayduslu yola ilelemesi ve ardından son radyusla yarım ters çember çizmesinin ardından sonsuz radyusa devam etmesiyle sonlanan hareket. Loop hareketi süresince en büyük g kuvveti loop'un yükselmeye başladığı anında hissedilir. Bu çalışmada klotoïd loop şekli kullanılmıştır.



Şekil 1.2 Loop Hareketi Esnasında Etkiyen Kuvvetler



Şekil 1.3 Drop (Düşme) Hareketi



Şekil 1.4 Klotoid Loop ile Dairesel Loop Profilleri

Düşme (drop): Ziyaretçilere düşme hissini uyandıran dik veya yüksek eğimli yoldan aşağı inme hareketi.

Roller Coaster'ın amacı; insana, maruz kalabileceği uç durumları yaşatmak ve vücudun bu durumlara olan tepkisini insanlarda bir zevk duygusuna çevirmektir. Bu esnada çeşitli ivmelerden doğan görünmez kuvvetlerin, sağlıklı birinin başa çıkabileceği boyutlarda olması gerekir. Bir Roller Coaster'ın tasarımıındaki en büyük zorluk budur.

1920'lerde sadece Amerika'da 1500'den fazla roller coaster varken bu sayı 1970'lere gelindiğinde sadece 145'ti. Şimdi ise dünya çapında 1500'den fazla roller coaster var. Her sene 300 milyondan fazla kişi roller coaster'a biniyor ve bunun getirisi 9 milyar dolardan fazla oluyor.

İnsanlar her geçen gün daha hızlı ve korkutucu roller coaster'lar istiyorlar. Öyle ki artık NASA teknolojisine başvurulmuş durumda. Roller coaster'lar 3 saniyede 100km/h hızı geçebiliyorlar ve 23 metre gibi kısa bir mesafede durabiliyorlar.

Birkaç canavar roller coaster ismi; Titan, Thriller, Steel Eel, Poltergeist, Racing Bull, Son of Beast, Goliath.

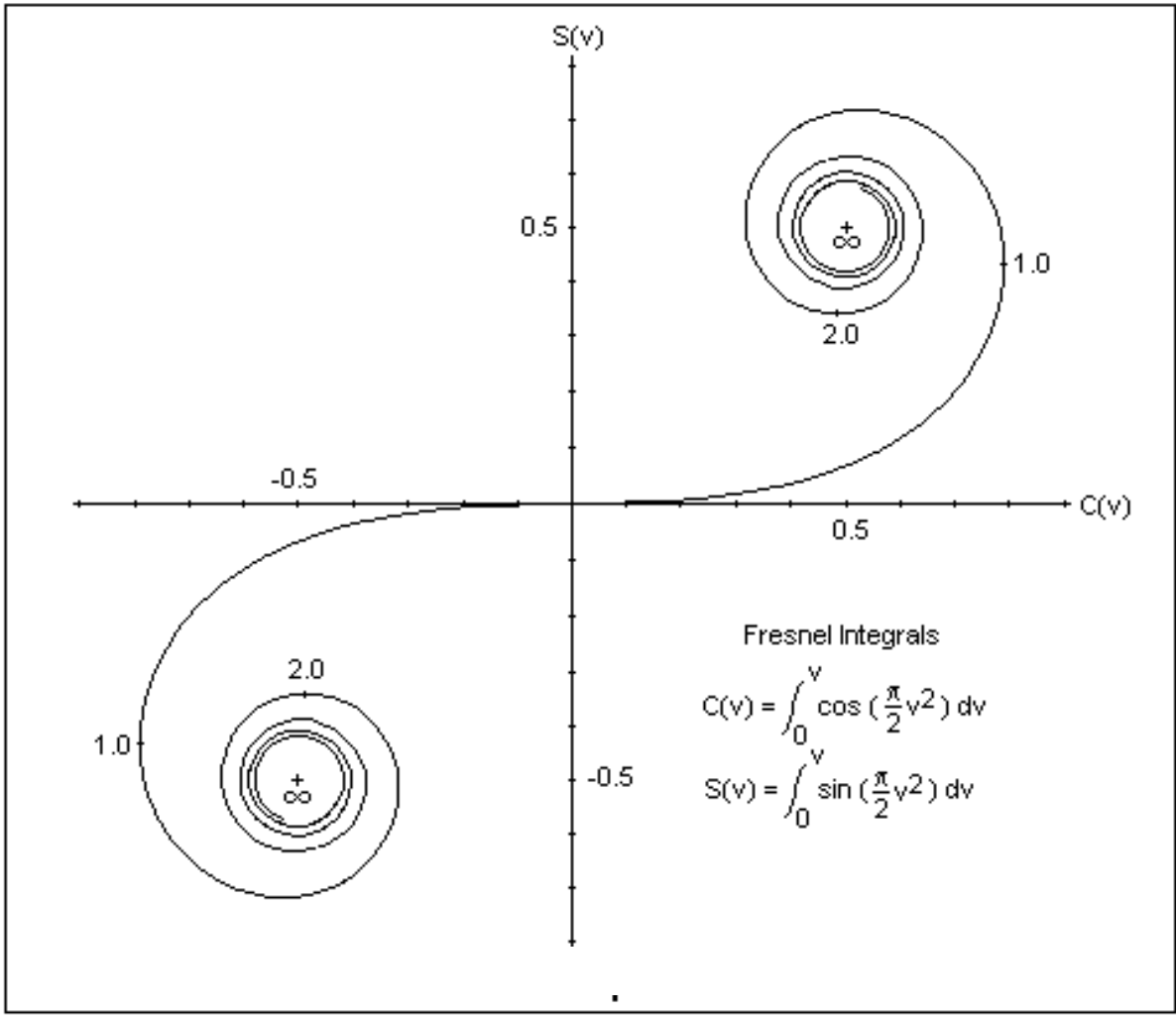
Yoldan gelen titreşimlerin ve sert hareketlerin istendiği bir durumda konfor şartlarını üst seviyede tutmak yersizdir. Aksi halde amacı eğlendirmek olan bir araç, yolcu treninden farksız olur. ISO'nun 4-8 Hz, NASA'nın 0-10 Hz olarak belirlediği ve kaçınılması gerektiği, insanın fizyolojik sınırların içine girilebilir. Bu aralıkta yaşanan; deniz tutma, nefes alma zorluğu gibi birçok etki insana bilinçli olarak yaşatılmak istenir. Korkunun ardından gelen adrenalin ve zevk duygusu, insanları bu eğlence trenine çeken en büyük etkidir.

Bilgisayar modellemesi ve biyomekanik mühendisliği sayesinde daha hızlı gitmek, daha fazla heyecan, daha fazla eğlence ve daha fazla güvenlik sağlanabilmektedir. Hareketler arası geçişler ve twistler şimdilerde daha yumuşak geçişlerle yapılabiliyor. Matematik, fizik ve bilgisayarın bir mühendisin elinde nelere dönüşebileceğinin bir göstergesi de roller coaster'lardır.

1.1. Klotoid Eğrisi

Klotoid eğrisi, bir doğrudan daireye geçişte en mükemmel geçişi sağlamak için kullanılan geçiş eğrisidir. Bu çalışmada klotoidin uygun bir geometrik ve matematiksel çözüm olaması durumu, fiziksel olarak incelenmekte ve simulinkte çözümü yapılmaktadır.

Klotoid eğrisi, ankastre mesnetle sabitlenen bir çubuğun ucuna bir F kuvveti veya bir M kütlesi konulduğunda elastik sınırlar içinde mesnetten uç kısma kadar olan eğilme eğrisiyle özdeşdir.



Şekil 1.5 Euler Spirali (Birim Klotoid Profili)

Tanım:

$L \times R = A^2$ olsun. L geçiş eğrisinin uzunluğu ile R kurp yarıçapının çarpımı bir A sayısının karesine eşittir. $A=1$ alınırsa bu klotoidde birim klotoid adı verilir. Klotoid, eğriliği $K = 0$ 'dan $K = \infty$ 'a giden ve sürekli doğrusal artan bir eğridir. Her defasında ilgili yarıçap ile gelişigüzel eğri uzunluğunun çarpımı sabittir. Boyutsal formül çıkarılırsa;

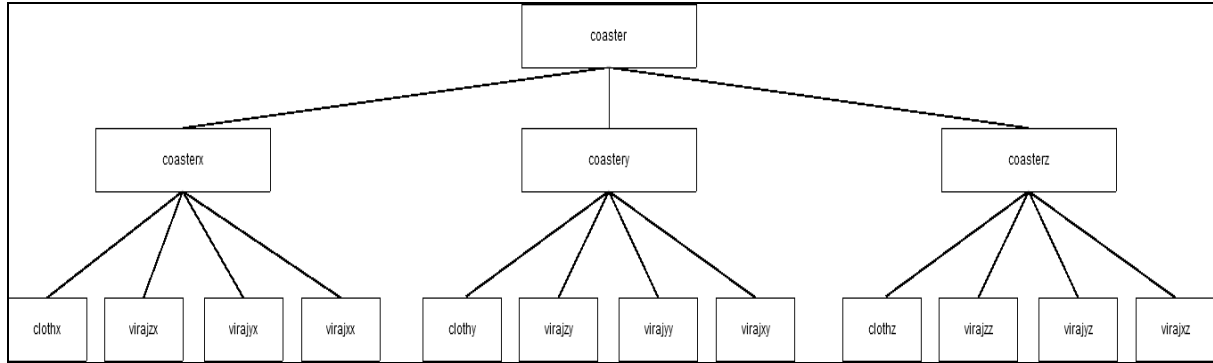
$$A^2 = R \times L, \quad A = \sqrt{R \times L}$$

A Klotoid parametresi [m], R Klotoidin sonundaki yarıçap [m], L Klotoid uzunluğu [m]

$$K = 1/R = L / A^2 \quad (K = \text{eğrilik})$$

2. Uygulama

Roller Coaster için tasarlanmış yolun çalışma diagramı Şekil de gösterilmiştir.



Şekil 2.1 Yol Profilini Oluşturan Programın Çalışma Diyagramı

Buradaki fonksiyonlar şu şekilde açıklanmalıdır.

coaster: Hiyerarşinin en üstündeki programdır. Görevi altındaki birimlerde çalışacak fonksiyonlar için gereken parametrelerin kullanıcı tarafından girişini sağlamaktır. Alt birimleri **coasterx**, **coastery** ve **coasterz** dir.

coasterx: Roller coaster yolunun X bileşenlerini oluşturan birimdir. Yolun gidişatına göre fonksiyon sayısında artış olabilir. Tasarlanan yolda temel olarak 4 fonksiyon yeterlidir. Dolayısıyla yapısında 4 fonksiyon bulundurulur. Bunun yanında fonksiyonlar arası geçiş yolları ayrı olarak tanımlanmıştır. Y bileşenini tanımlayacak olan **coastery** ve Z bileşenini tanımlayacak olan **coasterz**, **coasterx** ile benzerlik göstermekle birlikte, program içindeki geçiş yolları birbirlerinin çalışmasını aksatmaması için birbirlerine senkronudur.

virajzx: **virajzy** ve **virajzz** fonksiyonları ile vektör z virajına ait yolu tanımlayan fonksiyondur. **virajzx**(R, alfa, b, d) şeklinde tanımlanır. Burada ‘R’, virajı tanımlayan klotoid eğrisinin birim klotoide göre kaç kat büyük olduğunu belirler. ‘alfa’ taranacak açının bitiş noktasını belirler. Burada 2,5065 rad klotoid eğrinin referans noktasıdır. ‘b’ virajın yola oturması için, kendisinden bir önceki noktayı kartezyen ekseninde klotoidin başlangıç noktası yapan parametredir. ‘d’ ise klotoid oluşumunun yukarıdan aşağıya mı yoksa aşağıdan yukarıya mı olacağını belirleyen parametredir. Sistemi $2.5065:d*0.0001:alfa$ şekliyle etkiler. İşlemlerde 1 veya -1 değerini alır. X vektör virajını tanımlayan **virajxx**, **virajxy** ve **virajxz** ile Z vektör virajını tanımlayan **virajzx**, **virajzy** ve **virajzz**’nin çalışma şekli aynıdır.

clothx: Loop hareketi sırasında roller coasterın izleyeceği yolu belirleyen fonksiyondur. $[x]=clothx(R, \alpha, c, d, a)$ şeklinde tanımlanır. Burada ‘R’ loop hareketine ait klotoide eğrisinin birim klotoide göre kaç kat büyük olduğunu belirler. ‘ α ’ 2.5065 rad başlangıç açısının kaç radyanda biteceğini tanımlar. Sürekliliğin sağlanması için α açısı 3.529 rad ile sınırlandırılmıştır. ‘c’ loop girişi ile loop çıkışı arasındaki mesafeyi tanımlar. ‘d’ parametresi ise klotoidler arası geçişte geçilen çember yayının merkezinin referans y mesafesine uzaklığını tanımlar. Dolayısıyla geçiş anındaki taranan açı ‘fi’ ile ‘pi-fi’ arasında ve

$$fi = \arctan((y(s)-d)/(x(s)-c/2)) \quad (1)$$

şeklinde tanımlanır. Taranacak olan yarıçap,

$$ro = \sqrt{(y(s)-d)^2 + (x(s)-c/2)^2} \quad (2)$$

şeklinde tanımlanır. Klotoidler ile çember hareketinin eşit miktarda z üzerinden çizilmesi için bir düzeltme katsayısı tanımlanması gerekir. Bu düzeltme katsayısı sonradan sayacı etkileyip istenen durumu yaratır. Bu da

$$duzeltme = (\pi - 2 \cdot fi) / (\alpha - 2.44) \quad (3)$$

ile ifade edilir. $fi:duzeltme \cdot 0.0001 : (\pi - fi)$ şeklinde sisteme etki eder. Dizi elemanı ‘s’ ise giriş klotoidinin son noktasıdır. Çember hareketinden sonra yukarıdan aşağıya olman üzere klotoid tanımlanır. Yolun alacağı sonra nokta x ekseninde başlangıç noktasından c mesafesi uzaklıkta olacaktır. ‘a’ parametresi ise loop sırasında yolun z bileşkesinde ne kadar öteleneceğini tanımlar.

2.1. Hareket Denklemleri

Yol tasarımının ardından trenin bu yolun her noktasında sahip olacağı hızların hesaplanmasında enerjinin korunumu yasasından faydalanılmıştır.

$$\sum E = E_k + E_p \quad (4)$$

$$E_p = m \times g \times h \quad (5)$$

$$E_k = \frac{1}{2} \times m \times V^2 \quad (6)$$

Burada toplam enerji değişmediğinden (kayıplar ihmal edilmiştir) yüksekliği bilinen her konum için hız bulunabilmektedir.

Trenin sahip olduğu ivmeler normal ve teğetsel ivme olarak iki çeşittir. Normal ivme taşıtın virajlardaki hızından kaynaklanan ve virajın o bölgesinin dönme noktasına doğru olan ivmedir. Diğer bir ivme çeşidi olan teğetsel ivme ise taşıtın iki ardışık nokta arasında hareket doğrultusunda olan ivmesidir ve hız değişiminden kaynaklanır. Bu ivmenin değeri normal ivmenin yanında küçük kaldığından ve maksimum değeri yerçekimi ivmesi kadar olduğundan sonuçlara dahil edilmemiştir. Yalnızca kalkış anında dışarıdan verilen kuvvet sonucu oluşan teğetsel ivme dış kaynaklı bir ivme olduğundan sonuçlara dahil edilmiştir.

$$a_n = \frac{V^2}{R} \quad (7)$$

$$a_t = \frac{V_s^2 - V_i^2}{2 \times \Delta s} \quad (8)$$

$$a_{ilk} = \frac{F}{m} \quad (9)$$

2.2. Tasarlanan Roller Coaster'ın Matlab Kodları

1. `clc`
2. `clear all`
3. `buyukluk=input('clothoid ne kadar büyük?');`
4. `f=0;`
5. `while f==0`
6. `alfa=input('bitiş eğimi nedir?');`
7. `if alfa<3.529`
8. `f=1;`
9. `else`
10. `disp('eğim 3.529 dan küçük olmalı')`
11. `end`
12. `end`
13. `f=0;`
14. `dugum=input('başlangıç ve bitiş noktası arasındaki mesafe nedir?');`
15. `viraj=input('viraj yarıçapı orjine ne kadar uzaklıkta?');`

```

16. oteleme=input('oteleme ne kadar?')
17. viraj1=input('viraj buyukluđu nedir');
18. alfa1=3.03;
19. giris=input('giris pistinin uzunluđu nedir?');
20. viraj2=input('rampa giriş viraj büyüklüđu nedir?');
21. alfa2=3.03;
22. viraj3=viraj1 %rampa çıkış virajı
23. alfa3=3.03;
24. coasterx;
25. coastery;
26. coasterz;

```

2.2.1. Roller Coastera ait x bileşenlerini veren Matlab kodu

```

1. clc ,referans_loop_girisi=1 ,disp('loopun girişı')
2. clothx(buyukluk,alfa,dugum,viraj,oteleme) %loop'un x bileşeni
3. x=ans;
4. a=length(x); ,referans_loop_cikisi=a ,disp('loopun çıkışı')
5. for I=1:buyukluk*2 %looptan sonraki ilerleme
6. a=a+1;
7. x(a)=x(a-1)+1;
8. end
9. c=x(a); ,referans_son_viraj_giris=a ,disp('son viraj girişı') %c bir önceki fonksiyonun
   x teki son noktası
10. virajyx(viraj1,alfa1,c,1) %looptan sonraki ilk virajın x bileşeni
11. b=length(ans); ,referans_son_viraj_cikis=a+b ,disp('son viraj çıkışı')
12. for I=1:b
13. a=a+1;
14. x(a)=ans(I);
15. end
16. for I=1:25 %frenleme hattı
17. a=a+1;
18. x(a)=x(a-1);
19. end
20. c=x(a); ,referans_gise_viraji_giris=a ,disp('gişe virajı girişı')

```

```

21. virajyx(viraj1,alfa1,c,-1) %gişeye dönüş virajı
22. b=length(ans); ,referans_gise_viraji_cikis=a+b ,disp('gişe virajı çıkışı')
23. for I=1:b
24. a=a+1;
25. x(a)=ans(I);
26. end
27. for I=1:gisir %giriş
28. a=a+1;
29. x(a)=x(a-1)-1;
30. end
31. c=x(a); ,referans_1viraj_giris=a ,disp('yukarı çıkış virajı başlangıcı')
32. virajzx(viraj2,alfa2,c,1) %rampa çıkış virajı
33. b=length(ans); ,referans_1viraj_cikis=a+b ,disp('yukarı çıkış virajı sonu')
34. for I=1:b
35. a=a+1;
36. x(a)=ans(I);
37. end
38. for I=1:25 %rampa
39. a=a+1;
40. x(a)=x(a-1);
41. end
42. c=x(a); ,referans_2viraj_giris=a ,disp('tepe çıkış virajı başlangıcı')
43. virajxx(alfa3,c); %rampa tepe noktası düzelme virajı
44. b=length(ans); ,referans_2viraj_cikis=a+b ,disp('tepe çıkış virajı sonu')
45. for I=1:b
46. a=a+1;
47. x(a)=ans(I);
48. end
49. for I=1:(25+oteleme) %tepe noktası
50. a=a+1;
51. x(a)=x(a-1);
52. end
53. c=x(a); ,referans_3viraj_giris=a ,disp('tepe iniş virajı başlangıcı')
54. virajxx(alfa3,c) %rampa tepe noktası bozulma virajı

```

```

55. b=length(ans); ,referans_3viraj_cikis=a+b ,disp('tepe iniş virajı sonu')
56. for I=1:b
57. a=a+1;
58. x(a)=ans(I);
59. end
60. for I=1:25 %rampa
61. a=a+1;
62. x(a)=x(a-1);
63. end
64. c=x(a); ,referans_4viraj_giris=a ,disp('aşağı iniş virajı başlangıcı')
65. virajzx(viraj2,alfa2,c,-1) %rampa çıkış virajı
66. b=length(ans); ,referans_4viraj_cikis=a+b ,disp('aşağı iniş virajı sonu')
67. for I=1:b
68. a=a+1;
69. x(a)=ans(I);
70. end
71. for I=x(a):x(1) %bağlanma yolu
72. a=a+1;
73. x(a)=I;
74. end
75. son=a;

```

2.2.2. Roller Coaster’a ait y bileşenini veren Matlab Kodları

```

1. clc
2. cloth(buyukluk,alfa,dugum,viraj,oteleme) %loop'un y bileşeni
3. y=ans;
4. a=length(y);
5. for I=1:buyukluk*2 %looptan sonraki ilerleme
6. a=a+1;
7. y(a)=y(a-1);
8. end
9. c=y(a); %c bir önceki fonksiyonun y teki son noktası
10. virajyy(alfa1,c) %looptan sonraki ilk virajın y bileşeni
11. b=length(ans);

```

```

12. for I=1:b
13. a=a+1;
14. y(a)=ans(I);
15. end
16. for I=1:25 %frenleme hattı
17. a=a+1;
18. y(a)=y(a-1);
19. end
20. c=y(a);
21. virajyy(alfa1,c) %gişeye dönüş virajı
22. b=length(ans);
23. for I=1:b
24. a=a+1;
25. y(a)=ans(I);
26. end
27. for I=1:gisir %giris
28. a=a+1;
29. y(a)=y(a-1);
30. end
31. c=y(a);
32. virajzy(viraj2,alfa2,c,1) %rampa çıkış virajı
33. b=length(ans);
34. for I=1:b
35. a=a+1;
36. y(a)=ans(I);
37. end
38. for I=1:25 %rampa
39. a=a+1;
40. y(a)=y(a-1)+1;
41. end
42. c=y(a);
43. virajxy(viraj3,alfa3,c,1); %rampa tepe noktası düzelme virajı
44. b=length(ans);
45. for I=1:b

```

```

46. a=a+1;
47. y(a)=ans(I);
48. end
49. for I=1:(25+oteleme) %tepe noktası
50. a=a+1;
51. y(a)=y(a-1);
52. end
53. c=y(a);
54. virajxy(viraj3,alfa3,c,-1) %rampa tepe noktası bozulma virajı
55. b=length(ans);
56. for I=1:b
57. a=a+1;
58. y(a)=ans(I);
59. end
60. for I=1:25 %rampa
61. a=a+1;
62. y(a)=y(a-1)-1;
63. end
64. c=y(a);
65. virajzy(viraj2,alfa2,c,-1) %rampa çıkış virajı
66. b=length(ans);
67. for I=1:b
68. a=a+1;
69. y(a)=ans(I);
70. end
71. for I=y(a):10000 %bağlanma yolu
72. if a==son
73. break
74. end
75. a=a+1;
76. y(a)=y(1);
77. end

```

2.2.3. Roller Coaster'a ait z Bileşenini Veren Matlab kodları

```
1. clc
2. clothz(buyukluk,alfa,dugum,viraj,oteleme) %loop'un z bileşeni
3. z=ans;
4. a=length(z);
5. for I=1:buyukluk*2 %looptan sonraki ilerleme
6. a=a+1;
7. z(a)=z(a-1);
8. end
9. c=z(a); %c bir önceki fonksiyonun z teki son noktası
10. virajyz(viraj1,alfa1,c,1) %looptan sonraki ilk virajın z bileşeni
11. b=length(ans);
12. for I=1:b
13. a=a+1;
14. z(a)=ans(I);
15. end
16. for I=1:25 %frenleme hattı
17. a=a+1;
18. z(a)=z(a-1)+1;
19. end
20. c=z(a);
21. virajyz(viraj1,alfa1,c,-1) %gişeye dönüş virajı
22. b=length(ans);
23. for I=1:b
24. a=a+1;
25. z(a)=ans(I);
26. end
27. for I=1:gisir %gisir hattı
28. a=a+1;
29. z(a)=z(a-1);
30. end
31. c=z(a);
32. virajzz(alfa2,c) %rampa çıkış virajı
```

```

33. b=length(ans);
34. for I=1:b
35. a=a+1;
36. z(a)=ans(I);
37. end
38. for I=1:25 %rampa
39. a=a+1;
40. z(a)=z(a-1);
41. end
42. c=z(a);
43. virajxz(viraj3,alfa3,c,1); %rampa tepe noktası düzelme virajı
44. b=length(ans);
45. for I=1:b
46. a=a+1;
47. z(a)=ans(I);
48. end
49. for I=1:(25+oteleme) %tepe noktası
50. a=a+1;
51. z(a)=z(a-1)-1;
52. end
53. c=z(a);
54. virajxz(viraj3,alfa3,c,-1) %rampa tepe noktası bozulma virajı
55. b=length(ans);
56. for I=1:b
57. a=a+1;
58. z(a)=ans(I);
59. end
60. for I=1:25 %rampa
61. a=a+1;
62. z(a)=z(a-1);
63. end
64. c=z(a);
65. virajzz(alfa2,c) %rampa çıkış virajı
66. b=length(ans);

```



```

67. for I=1:b
68. a=a+1;
69. z(a)=ans(I);
70. end
71. for I=z(a):10000 %bağlanma yolu
72. if a==son;
73. break
74. end
75. a=a+1;
76. z(a)=z(1);
77. end

```

2.2.4. Dönüş Vektörü Z olan bir virajın fonksiyon kodları

a)Z ekseninde değişim

```

1. function [z]=virajzz(alfa,d)
2. s=0;
3. for I=2.5065:0.0001:alfa
4. s=s+1;
5. z(s)=d;
6. end

```

b)Y ekseninde değişim

```

1. function [y]=virajzy(R,alfa,b,d)
2. s=0;
3. if d==1
4. for I=2.5065:d*0.0001:alfa
5. s=s+1;
6. teta=I^2/2;
7. r=2*R/I;
8. y(s)=b-(2*R/2.5065)*cos(2.5065^2/2)+r*(cos(teta));
9. end
10. elseif d==-1
11. for I=alfa:d*0.0001:2.5065
12. s=s+1;

```

```

13. teta=I^2/2;
14. r=2*R/I;
15. y(s)=b-(2*R/alfa)*cos(alfa^2/2)+r*(cos(teta));
16. end
17. end

```

c)x Ekseninde Değişim

```

1. function [x]=virajzx(R,alfa,c,d)
2. s=0;
3. if d==1
4. for I=2.5065:d*0.0001:alfa
5. s=s+1;
6. teta=I^2/2;
7. r=2*R/I;
8. x(s)=c-(2*R/2.5065)*sin(2.5065^2/2)+r*(sin(teta));
9. end
10. elseif d==-1
11. for I=alfa:d*0.0001:2.5065
12. s=s+1;
13. teta=I^2/2;
14. r=2*R/I;
15. x(s)=c-(2*R/alfa)*sin(alfa^2/2)+r*(sin(teta));
16. end
17. end

```

2.2.5. Dönüş Vektörü Y olan bir virajın fonksiyon kodları

a)Z bileşenin değişimi

```

1. function [z]=virajyz(R,alfa,b,d)
2. s=0;
3. if d==1
4. for I=2.5065:d*0.0001:alfa
5. s=s+1;
6. teta=I^2/2;
7. r=2*R/I;

```

```

8. z(s)=b-(2*R/2.5065)*cos(2.5065^2/2)+r*(cos(teta));
9. end
10. elseif d==-1
11. for I=alfa:d*0.0001:2.5065
12. s=s+1;
13. teta=I^2/2;
14. r=2*R/I;
15. z(s)=b+(2*R/alfa)*cos(alfa^2/2)-r*(cos(teta));
16. end
17. end

```

b)Y bileşeninin değişimi

```

1. function [y]=virajyy(alfa,d)
2. s=0;
3. for I=2.5065:0.0001:alfa
4. s=s+1;
5. y(s)=d;
6. end

```

c)X bileşeninin değişimi

```

1. function [x]=virajyx(R,alfa,c,d)
2. s=0;
3. if d==1
4. for I=2.5065:d*0.0001:alfa
5. s=s+1;
6. teta=I^2/2;
7. r=2*R/I;
8. x(s)=c+(2*R/2.5065)*sin(2.5065^2/2)-r*(sin(teta));
9. end
10. elseif d==-1
11. for I=alfa:d*0.0001:2.5065
12. s=s+1;
13. teta=I^2/2;
14. r=2*R/I;
15. x(s)=c+(2*R/alfa)*sin(alfa^2/2)-r*(sin(teta));

```

16. end

17. end

2.2.6. Dönüş Vektörü X olan bir virajın fonksiyon kodları

a) Z bileşenin değişimi

```
1. function [z]=virajxz(R,alfa,c,d)
2. s=0;
3. if d==1
4. for I=2.5065:d*0.0001:alfa
5. s=s+1;
6. teta=I^2/2;
7. r=2*R/I;
8. z(s)=c+(2*R/2.5065)*cos(2.5065^2/2)-r*(cos(teta));
9. end
10. elseif d==-1
11. for I=alfa:d*0.0001:2.5065
12. s=s+1;
13. teta=I^2/2;
14. r=2*R/I;
15. z(s)=c-(2*R/alfa)*cos(alfa^2/2)+r*(cos(teta));
16. end
17. end
```

b) Y bileşeninin değişimi

```
1. function [y]=virajxy(R,alfa,b,d)
2. s=0;
3. if d==1
4. for I=2.5065:d*0.0001:alfa
5. s=s+1;
6. teta=I^2/2;
7. r=2*R/I;
8. y(s)=b+(2*R/2.5065)*sin(2.5065^2/2)-r*(sin(teta));
9. end
10. elseif d==-1
```

```

11. for I=alfa:d*0.0001:2.5065
12. s=s+1;
13. teta=I^2/2;
14. r=2*R/I;
15. y(s)=b+(2*R/alfa)*sin(alfa^2/2)-r*(sin(teta));
16. end
17. end

```

c)X bileşeninin değişimi

function [x]=virajxx(alfa,d)

```

1. s=0;
2. for I=2.5065:0.0001:alfa
3. s=s+1;
4. x(s)=d;
5. end

```

2.2.7. Klotoid Loop Fonksiyonu

X,Y,Z Bileşeninin çekilmesi:

```

1. function [x,y,z]=cloth...(R,alfa,c,d,a)
2. s=0;
3. for I=2.5065:0.0001:alfa
4. if alfa>=3.529
5. break
6. end
7. s=s+1;
8. teta=I^2/2;
9. r=2*R/I;
10. y(s)=-2*R*cos(2.5065^2/2)/2.5065+r*(cos(teta));
11. x(s)=2*R*sin(2.5065^2/2)/2.5065-r*(sin(teta));
12. end
13. fi=atan((y(s)-d)/(x(s)-c/2));
14. ro=sqrt((y(s)-d)^2+(x(s)-c/2)^2);
15. duzeltme=(pi-2*fi)/(alfa-2.5065);
16. p=s;

```

```

17. for J=fi:duzeltme*0.0001:(pi-fi);
18. x(p)=c/2+ro*cos(J);
19. y(p)=d+ro*sin(J);
20. p=p+1;
21. end
22. p1=p;
23. for I=alfa:-0.0001:2.5065
24. teta=I^2/2;
25. r=2*R/I;
26. y(p1)=-2*R*cos(2.5065^2/2)/2.5065+r*(cos(teta));
27. x(p1)=c-2*R*sin(2.5065^2/2)/2.5065+r*(sin(teta));
28. p1=p1+1;
29. end
30. h=a/(p1-2);
31. z=0:h:a;

```

... ile belirtilmiş yere hangi bileşen çekilmek istenirse o yazılır. X, Y, Z aynı fonksiyondan çekilmektedir.

X için ilk satır: [x]=function clothx(...)

Y için ilk satır: [y]=function clothxy(...)

Z için ilk satır: [z]=function clothz(...)

2.3. Kuvvet ve İvme Hesapların Kullanılan Programın Matlab Kodları

```

1. m=input('aracın kütlesi (kg cinsinden) nedir?'); g=9.81;
2. V0=sqrt(2*g*y(referans_2viraj_cikis)+5^2); t=0;
3. kalkisyolu=x(referans_1viraj_giris)-x(referans_gise_viraji_cikis);
4. F=1/2*m*V0^2/kalkisyolu;
5. ivme=F/m;
6. E=1/2*m*V0^2;
7. for i=referans_gise_viraji_cikis:1:referans_1viraj_giris-1
8. t=t+1;
9. yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);

```

```

10. yol(t)=sum(yolparcalari);
11. if t==1
12. V(t)=sqrt(2*ivme*(x(i+1)-x(i)));
13. else
14. V(t)=sqrt(V(t-1)^2+2*ivme*(x(i+1)-x(i)));
15. end
16. at(t)=ivme;
17. an(t)=abs(ivme);
18. atop(t)=sqrt(at(t)^2+an(t)^2);
19. end
20. s=0;
21. for I=2.5065+0.0001:0.0001:alfa1
22. s=s+1;
23. r1(s)=2*viraj2/I;
24. r2(s)=2*viraj3/I;
25. r3(s)=2*viraj1/I;
26. end
27. s=0;
28. for I=alfa1:-0.0001:2.5065+0.0001;
29. s=s+1;
30. r11(s)=2*viraj2/I;
31. r22(s)=2*viraj3/I;
32. r33(s)=2*viraj1/I;
33. end
34. s=0;
35. for I=2.5065+0.0001:0.0001:alfa
36. s=s+1;
37. rl(s)=2*buyukluk/I;
38. end
39. for I=2.5065+0.0001:0.0001:alfa
40. s=s+1;
41. rl(s)=sqrt((y(s+1)-viraj)^2+(x(s+1)-dugum/2)^2);
42. end
43. for I=alfa:-0.0001:2.5065+0.0001;

```

```

44. s=s+1;
45. rl(s)=2*buyukluk/I;
46. end
47. s=0;
48. for i=referans_1viraj_giris+1:1:referans_1viraj_cikis-1
49. t=t+1;
50. s=s+1;
51. yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
52. yol(t)=sum(yolparcalari);
53. Ek=E-m*g*y(i);
54. V(t)=sqrt(2*Ek/m);
55. at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));
56. an(t)=V(t)^2/rl(s);
57. atop(t)=sqrt(at(t)^2+an(t)^2);
58. end
59. for i=referans_1viraj_cikis+1:1:referans_2viraj_giris-1
60. t=t+1;
61. yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
62. yol(t)=sum(yolparcalari);
63. Ek=E-m*g*y(i);
64. V(t)=sqrt(2*Ek/m);
65. at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));
66. an(t)=0;
67. atop(t)=sqrt(at(t)^2+an(t)^2);
68. end
69. s=0;
70. for i=referans_2viraj_giris+1:1:referans_2viraj_cikis-1
71. t=t+1;
72. s=s+1;
73. yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
74. yol(t)=sum(yolparcalari);
75. Ek=E-m*g*y(i);
76. V(t)=sqrt(2*Ek/m);
77. at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));

```



```

78. an(t)=V(t)^2/r2(s);
79. atop(t)=sqrt(at(t)^2+an(t)^2);
80. end
81. for i=referans_2viraj_cikis+1:1:referans_3viraj_giris-1
82. t=t+1;
83. yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
84. yol(t)=sum(yolparcalari);
85. Ek=E-m*g*y(i);
86. V(t)=sqrt(2*Ek/m);
87. at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));
88. an(t)=0;
89. atop(t)=sqrt(at(t)^2+an(t)^2);
90. end
91. s=0;
92. for i=referans_3viraj_giris+1:1:referans_3viraj_cikis-1
93. t=t+1;
94. s=s+1;
95. yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
96. yol(t)=sum(yolparcalari);
97. Ek=E-m*g*y(i);
98. V(t)=sqrt(2*Ek/m);
99. at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));
100.     an(t)=V(t)^2/r22(s);
101.     atop(t)=sqrt(at(t)^2+an(t)^2);
102.     end
103.     for i=referans_3viraj_cikis+1:1:referans_4viraj_giris-1
104.         t=t+1;
105.         yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
106.         yol(t)=sum(yolparcalari);
107.         Ek=E-m*g*y(i);
108.         V(t)=sqrt(2*Ek/m);
109.         at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));
110.         an(t)=0;
111.         atop(t)=sqrt(at(t)^2+an(t)^2);

```

```

112.     end
113.     s=0;
114.     for i=referans_4viraj_giris+1:1:referans_4viraj_cikis-1
115.         t=t+1;
116.         s=s+1;
117.         yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
118.         yol(t)=sum(yolparcalari);
119.         Ek=E-m*g*y(i);
120.         V(t)=sqrt(2*Ek/m);
121.         at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));
122.         an(t)=V(t)^2/rl(s);
123.         atop(t)=sqrt(at(t)^2+an(t)^2);
124.     end
125.     for i=referans_4viraj_cikis+1:1:a-1
126.         t=t+1;
127.         yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
128.         yol(t)=sum(yolparcalari);
129.         Ek=E-m*g*y(i);
130.         V(t)=sqrt(2*Ek/m);
131.         at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));
132.         an(t)=0;
133.         atop(t)=sqrt(at(t)^2+an(t)^2);
134.     end
135.     s=0;
136.     for i=referans_loop_girisi+1:1:referans_loop_cikisi-1
137.         t=t+1;
138.         s=s+1;
139.         yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
140.         yol(t)=sum(yolparcalari);
141.         Ek=E-m*g*y(i);
142.         V(t)=sqrt(2*Ek/m);
143.         at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));
144.         an(t)=V(t)^2/rl(s);
145.         atop(t)=sqrt(at(t)^2+an(t)^2);

```

```

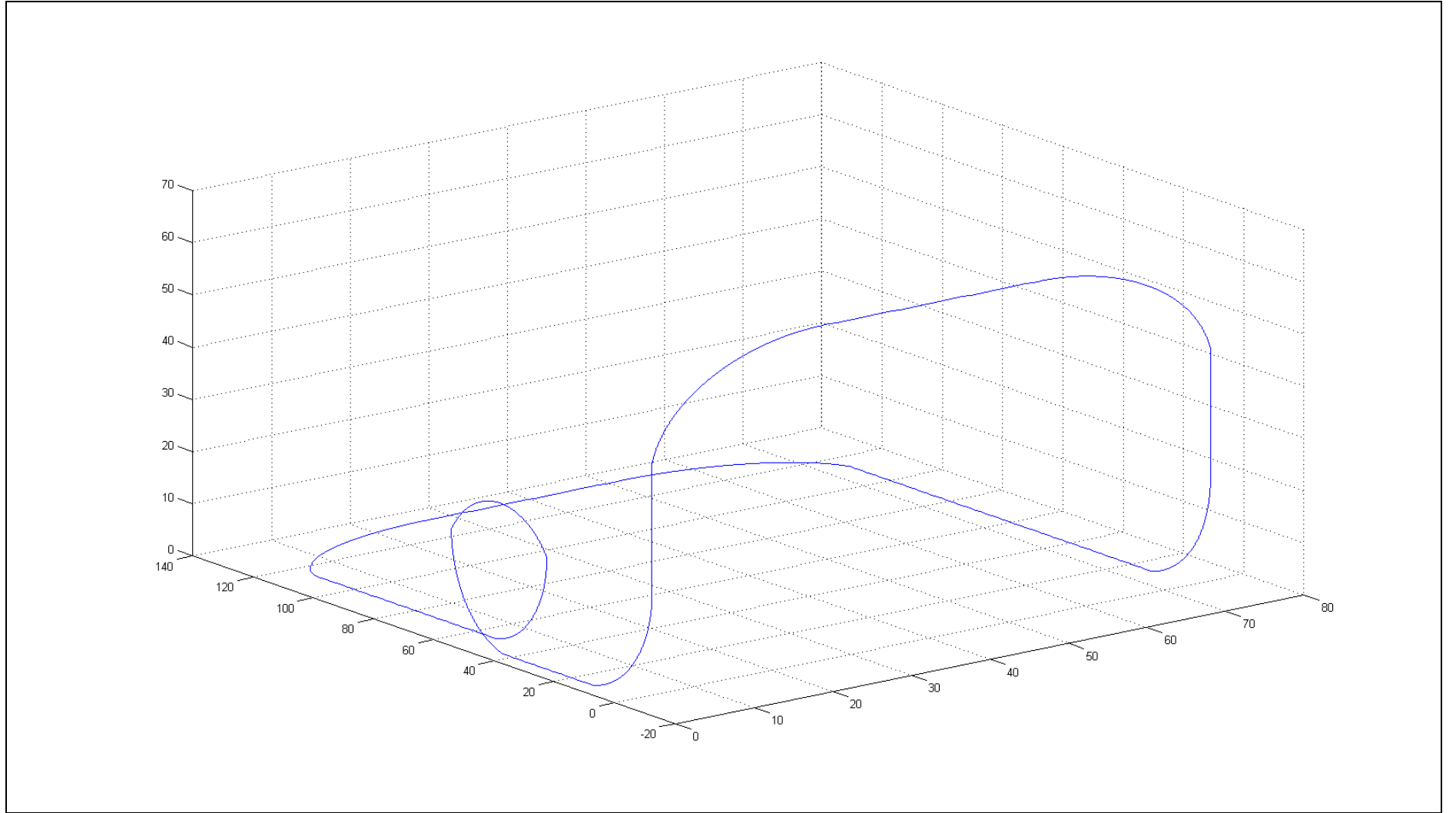
146.     end
147.     for i=referans_loop_cikisi+1:1:referans_son_viraj_giris-1
148.         t=t+1;
149.         yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
150.         yol(t)=sum(yolparcalari);
151.         Ek=E-m*g*y(i);
152.         V(t)=sqrt(2*Ek/m);
153.         at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));
154.         an(t)=0;
155.         atop(t)=sqrt(at(t)^2+an(t)^2);
156.     end
157.     s=0;
158.     for i=referans_son_viraj_giris+1:1:referans_son_viraj_cikis-1
159.         t=t+1;
160.         s=s+1;
161.         yolparcalari(t)=sqrt((x(i+1)-x(i))^2+(y(i+1)-y(i))^2+(z(i+1)-z(i))^2);
162.         yol(t)=sum(yolparcalari);
163.         Ek=E-m*g*y(i);
164.         V(t)=sqrt(2*Ek/m);
165.         at(t)=(V(t)^2-V(t-1)^2)/(2*yolparcalari(t));
166.         an(t)=V(t)^2/r3(s);
167.         atop(t)=sqrt(at(t)^2+an(t)^2);
168.     end

```

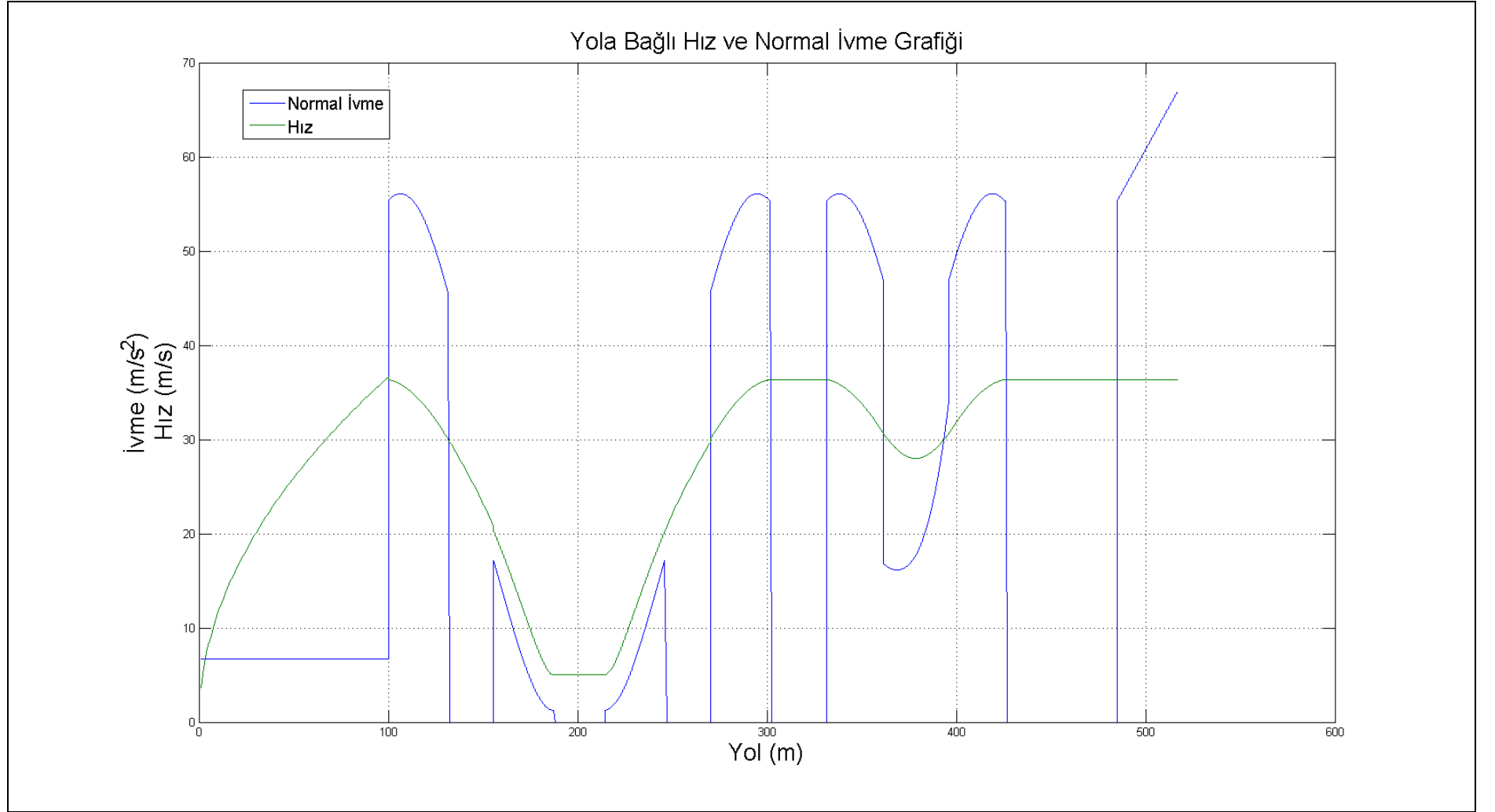
3. Sonuç

Yapılan çalışma sonucunda elde edilen veriler eşliğinde incelenen roller coasterın hızının ortalama bir roller coaster için yeterli düzeyde olduğu bununla beraber virajlarda ve loopta sahip olduğu normal ivmelerin insanların fizyolojik sınırlarına yaklaştığı ama aşmadığı görülmüştür. Buradan yola çıkarak bu roller coasterın birçok insan için güvenli ve aynı zamanda eğlendirici olduğu sonucuna varılmıştır.

Ayrıca klotoid profiline sahip yolun sonucunda oluşan ivmelenen ivme yapısının da oluştuğu görülmüştür.



Şekil 3.1 Roller Coaster'ın Yol Profili



Şekil 3.2 Roller Coaster'ın Yola Bağlı Hız ve Normal İvme Grafiği

4. Kaynaklar

- Pendrill Ann-Marie, Rollercoaster Loop Shapes, Department of Physics, Göteborg University, SE-41296 Göteborg, Sweden
- Schott Klaus ve Tobolář Jakub, Development of a Spatial Track Module in SIMPACK and Application to a Simple Roller Coaster Example, DLR Institute of Robotics and Mechatronics and Vehicle System Dynamics, SIMPACK User Meeting 2004
- James Palmer McCrae (2008), Sketch-Based Path Design, Department of Computer Science University of Toronto
- Raph Levien (30 Ağustos 2008), The Euler spiral: a mathematical history
- Ryan Seng ve Molly Severdia (9 Aralık 2007), The Clothoid
- Dong Hun Shin ve Sanjiv Singh (Aralık 1990), Path Generation for Robot Vehicles Using Composite Clothoid Segments, The Robotics Institute Carnegie-Mellon University
- Hatch, Michael R. (2001), Vibration Simulation Using Matlab and Ansys, by Chapman & Hall/CRC
- D. T. Pham, M. H. Saw ve F. Cao, Biomechanical analysis of human body segments in human-robot interaction, Manufacturing Engineering Centre, Cardiff University, Cardiff CF24 3AA, UK
- Edward J. Heiden, Ph.D. ve Stephen McGonegal (June/July 2003), NSC Injury Report of Roller Coaster, National Safety Council Research & Statistical Services
- Martin Voshell (28 Kasım 2004), High Acceleration and the Human Body
- Richard M. Morency, Glen A. Livesay, ve Eric A. Nauman (31 Ocak 2011), Application of the Serret-Frenet Basis to the Analysis of Roller Coaster Dynamics: Variations in Rider Accelerations
- Maurer Söhne (2004), Freedom on tracks – new possibilities in roller coaster design, München/Germany
- Prof. Alfred Clark, Jr., A Primer on Roller Coaster Dynamics Part I – Plane and Fancy, Part II – You Can Bank on Them, Part III – Passengers Entrained, Department of Mechanical Engineering University of Rochester, NY
- Pendrill Ann-Marie (2008), Acceleration in one, two, and three dimensions in launched roller coasters, Department of Physics, Göteborg University, Göteborg, Sweden

- James Alan Witmer, Lebanon, NH(US) (26 Mayıs 2011), Method of Applying Clothoid Curve Values to Roadways in a Geographic Data Information System, United States Patent Application Publication
- H. Ulaş AKOVA, E. İlhan KONUKSEVEN, Ulaş YAMAN, ve Suat DEMİRER (12 - 15 Ekim 2011), Dağ Treni Hidrolik Fırlatma Sistemi Tasarımı, *VI. Ulusal Hidrolik Pnömatik Kongresi*
- Roller Coaster Lab 3 Energy Analysis, Intro to Engineering AA – Lab Procedure
- João Pombo and Jorge Ambrósio (2004), Modelling Tracks For Roller Coaster Dynamics, IDMEC – Instituto Superior Técnico, Lisboa, Portugal
- D. S. MEEK ve D. J. WALTON (Temmuz 1992), Clothoid Spline Transition Spirals, *Mathematics of Computation* Volume 59, Number 199, Page 117-133
- B. Bostancı (2005), Klotoid Eğrisinde Yol Dinamiğinin İncelenmesi, TMMOB Harita ve Kadastro Mühendisleri Odası 10. Türkiye Harita Bilimsel ve Teknik Kurultayı Mart 2005, Ankara
- Jeff Rhoads ve Chuck Krousgrill (2010), Roller Coaster Dynamics, *Lecture 1 (Introduction and Loops), Lecture 2 (Loops), Lecture 3 (Airtime Hills), Lecture 4 (Transitions), Lecture 6 (Splines)*, Purdue University West Lafayette, Indiana, USA (Güz Dönemi 2010)

4.1. İnternet Kaynakları

- Ivers John (17 Şubat 2002), The Blue Flash Backyard Coaster, <http://www.negative-g.com/BlueFlash/February2002/Blue-Flash-2002-2-1.html>
- Harris Tom, How Roller Coasters Work
<http://science.howstuffworks.com/engineering/structural/roller-coaster.htm>
- Kingda Ka (29.09.2004), Six Flags Great Adventure, Power Launch Roller Coaster Facts, <http://www.rcdb.com/2832.htm>