

DÜZLEMSEL KOL MEKANİZMALARININ
FONKSİYON VE YÖRÜNGE SENTEZİ İÇİN
ETKİLEŞİMLİ BİR BİLGİSAYAR PAKET
PROGRAMININ GELİŞTİRİLMESİ

SEDAT KARLIDAĞ

MERSİN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNA MÜHENDİSLİĞİ
ANA BİLİM DALI

YÜKSEK LİSANS TEZİ

MERSİN
HAZİRAN - 2007

DÜZLEMSEL KOL MEKANİZMALARININ FONKSİYON VE YÖRÜNGE
SENTEZİ İÇİN ETKİLEŞİMLİ BİR BİLGİSAYAR PAKET PROGRAMININ
GELİŞTİRİLMESİ

SEDAT KARLIDAĞ

Mersin Üniversitesi
Fen Bilimleri Enstitüsü

Makina Mühendisliği
Ana Bilim Dalı

YÜKSEK LİSANS TEZİ

Tez Danışmanı
Yrd. Doç. Dr. Hüseyin MUTLU

MERSİN
HAZİRAN - 2007

Bu tezin gerek bilimsel içerik, gerekse elde edilen sonuçlar açısından tüm gerekleri sağladığı kanaatine ulaşan ve aşağıda imzaları bulunan biz jüri üyeleri, sunulan tezi oy birliği ile Yüksek Lisans Tezi olarak kabul ediyoruz.

Tez Danışmanı

Yrd. Doç Dr. Hüseyin MUTLU

Jüri Üyesi

Yrd. Doç Dr. Hüseyin CANBOLAT

Jüri Üyesi

Yrd. Doç. Dr. Bülent YILMAZ

Bu tezin Fen Bilimleri Enstitüsü yazım kurallarına uygun olarak yazıldığı Enstitü Yönetim Kurulu'nun/...../..... tarih ve/..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Mahir TURHAN

Enstitü Müdürü

NOT: Bu tezde kullanılan tüm bilgiler, şekil, çizelge ve fotoğraflardan kaynak göstermeden alıntı yapmak 5846 sayılı Fikir ve Sanat Eserleri Kanunu Hükümlerine tabidir.

ÖZ

Bu çalışmada, Düzlemsel Kol Mekanizmalarının Sentezini yapabilen Mek-Gen isimli dinamik yapıda bir bilgisayar programı geliştirilmiştir. Mekanizmalarda giriş uzvuna verilen hareket, belirli bir fonksiyonel ilişki ile çıkış uzvuna aktarılır. Yapısal hatası en aza indirgenmiş bir mekanizma tasarımı için mekanizmanın uzuvları arasındaki hareket aktarıcı fonksiyonun iyi sentez edilmesi gerekmektedir. Geliştirilen bilgisayar programı ile, girilen veriler doğrultusunda hareket edecek mekanizmanın, tasarıma en uygun yapısal hataları tespit edilir ve bu yapısal hatalara göre mekanizmanın tasarlanması sağlanır.

Mek-Gen programının tam dinamik bir yapıya sahip olması, kullanılan tüm fonksiyon ve değişkenlerin rahatlıkla değiştirilebilmesi imkanı sağlar. Sentez sonucu tasarlanan mekanizmaların animasyon ve hata sapmaları kullanıcıya aynı ekranda yansıtılmaktadır.

Mekanizma Sentez yöntemleri olarak Galerkin, Subdomain (Alt Bölge) ve Precision Point (Kesin Noktalar) metodları kullanılmıştır. Programda farklı metodların kullanılabilmesi, sentezi yapılan mekanizmanın tasarımdaki hata analizlerinin daha rahat görülebilmesini sağlar.

Matematiksel denklem ve fonksiyonların bilgisayar çözümlerinde analitik, trigonometrik ve kompleks matematik metodlarından oluşan birimler kullanılmıştır. Kağıt üzerinde üretilen fonksiyon ve çözüm yolları bu metodlar sayesinde programa işlenebilmiştir.

Mek-Gen programı, Delphi 6.0 programlama dilinde geliştirilmiştir. Nesne Yönelimli olup, tüm Windows işletim sistemlerinde çalışabilmektedir.

Anahtar kelimeler: Mekanizma Sentezi, Dört Çubuk Mekanizmaları, Krank-Biyel Mekanizmaları, Object Oriented Programming, Galerkin, Subdomain, Precision Point, Fonksiyon Sentezi, Yörünge Sentezi, Delphi.

ABSTRACT

In this study, a dynamic computer program, called by Mek-Gen which synthesis the planar linkage mechanisms is developed. In mechanisms, movement is given to an input member is transfered to the output member with a functional contact.

Complete Dynamic structure of Mek-Gen makes possible to change any variable of function easily. The result of the synthesis and the animation of the synthesis problem is displayed on the screen.

Subdomain, Galerkin and the Precision Point methods are used for the function synthesis. In this computer program, different methods provide to see analyses mistakes in the design of the mechanism whose synthesis was made more easily.

Mek-Gen Computer program is developed in Delphi 6.0 programming language and all Windows versions can run Mek-Gen due to it's object oriented nature.

Keywords: mechanism synthesis, Four Bar mechanisms, Krank-Rocker Mechanism, Object Oriented Programming, Galerkin, Subdomain, Precision Point, Delphi.

TEŐEKKÖRLER

Tez alıőmamda büyük emeđi geen ve bana her konuda yardımcı olan deđerli tez danıőmanım Yrd. Dođ. Dr. Hüseyin MUTLU' ya, beni daima destekleyip motive eden bitaneme ve aileme sonsuz teőekkürler...

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZ	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ÇİZELGELER DİZİNİ	viii
ŞEKİLLER DİZİNİ	xi
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMALARI	4
3. MATERYAL VE METOD	10
3.1. FONKSİYON ÜRETEN 4 ÇUBUK MODÜLÜ	10
3.1.1. Giriş Kolu Açısı (ψ) ile Çıkış Kolu Açısı (α) Fonksiyonel İlişisini Üreten Dört Çubuk Tasarımı	11
3.1.1.1. Kesin Nokta Metodu:	12
3.1.1.2. Altbölge Metodu:	12
3.1.1.3. Galerkin Metodu:	13
3.1.2. Giriş Kolu Açısı(ψ) ile Biyel Kolu Açısı(δ) Fonksiyonel İlişisini Üreten Dört Çubuk Tasarımı	27
3.1.2.1. Kesin Nokta Metodu	28
3.1.2.2. Altbölge Metodu	28
3.1.2.3. Galarkin Metodu	29
3.1.3. Sayısal Örnekler	32
3.1.3.1. Örnek1	32
3.1.3.2. Örnek 2.	34
3.2. FONKSİYON ÜRETEN KRANK-BİYEL MODÜLÜ	36
3.2.1. (ψ -s) Fonksiyonel İlişisini Üreten Krank-Biyel Tasarımı	36
3.2.1.1. Kesin Nokta Metodu:	38
3.2.1.2. Altbölge Metodu:	38
3.2.1.3. Galarkin Metodu:	39
3.2.2. (ψ - δ) Fonksiyonel İlişisini Üreten Krank-Biyel Tasarımı	46

3.2.2.1. Kesin Nokta Metodu:	47
3.2.2.2. Altbölge Metodu:	47
3.2.2.3. Galarkin Metodu:	48
3.2.3. Sayısal Örnekler	51
3.2.3.1. Örnek 1	51
3.2.3.2. Örnek 2.	53
4. MEK-GEN PROGRAM PAKETİ	55
4.1. NESNE YÖNELİMLİ PROGRAMLAMA	55
4.2. PROGRAMLAMA DİLİ: DELPHI 6.0. ENTERPRISE EDITION	57
4.3. MEK-GEN PROGRAMI	60
4.3.1. Giriş	60
4.3.2. Mek-Gen Program Paketi İçeriği	61
4.3.3. Mek-Gen Kod Yapısı	62
4.3.3.1. Four-Bar Programı Kod Yapısı	63
Fonksiyon Unitleri (Global ve Polkat)	65
Fonksiyon Girişi Unitleri	65
Grafik Unitleri	66
Global Değişkenler Uniti (Unit 2)	67
Veri giriş Uniti (Unit3)	67
4.3.3.2. Four-Bar Delta Programı Kod Yapısı	70
4.3.3.3. Krank-Biyel Programı Kod Yapısı	72
4.3.3.4. Krank-Biyel Delta Programı Kod Yapısı	75
4.3.4. Mek-Gen Programının Çalışma Yapısı	76
4.3.4.1. Four-Bar Programı	78
Four-Bar Programının Çalışma Yapısı ve Mekanizma Sentezi	81
Veri Giriş Menüsü ve İşleyişi	83
Fonksiyon Girişi Menüsü ve İşleyişi	86
Dosyalama Sistemi ve İşleyişi	88
Four-Bar Programı İle Mekanizma Sentezi	91
Dinamik Grafikler ve İşleyişleri	103

4.3.4.2. Four-Bar Delta Programı	111
Four-Bar Delta Programı Veri Girişi	112
Four-Bar Programı ile Mekanizma Sentezi	112
4.3.4.3. Krank-Biyel Programı	115
Krank-Biyel Programının Çalışma Yapısı ve Mekanizma Sentezi	116
Veri Giriş Menüsü ve İşleyişi	116
Fonksiyon Giriş Menüsü ve İşleyişi	119
Krank-Biyel Programı İle Mekanizma Sentezi	120
4.3.4.4. Krank-Biyel Delta Programı	135
Krank-Biyel Delta Programı Veri Girişi	136
Krank-Biyel Programının Çalışma Yapısı ve Mekanizma Sentezi	137
5. ARAŞTIRMA BULGULARI VE TARTIŞMA	139
5.1. GİRİŞ	139
5.2. DÖRT- ÇUBUK TASARIMLARI	139
5.2.1. Galerkin Metodu İle Dört Çubuk Tasarım Sonuçları	139
5.2.1.1. $Y=\tan(x)$ fonksiyonu için dört çubuk tasarımı	139
5.2.1.2. $Y=x^2$ Fonksiyonu İçin Dört Çubuk Tasarımı	144
5.2.1.3. $Y=\sin(x)$ Fonksiyonu İçin 4 Çubuk Tasarımı	149
5.2.2. Alt Bölgeler Metodu İle Dört Çubuk Tasarım Sonuçları	154
5.2.2.1. $Y=\tan(x)$ fonksiyonu için dört çubuk tasarımı	154
5.2.2.2. $Y=x^2$ Fonksiyonu İçin Dört Çubuk Tasarımı	159
5.2.2.3. $Y=\log_{10}(x)$ fonksiyonu için dört çubuk tasarımı	164
5.2.3. Kesin Noktalar Metodu ile dört çubuk tasarım sonuçları	169
5.2.3.1. $Y=\tan(x)$ Fonksiyonu İçin Dört Çubuk Tasarımı	169
5.2.3.2. $Y=x^2$ Fonksiyonu İçin Dört Çubuk Tasarımı	174
5.3. KRANK – BİYEL TASARIMLARI	179
5.3.1. Galerkin Metodu İle Krank-Biyel Tasarım Sonuçları	179
5.3.1.1. $Y=\tan(x)$ Fonksiyonu İçin Krank – Biyel Tasarımı	179
5.3.1.2. $Y=x^2$ Fonksiyonu İçin Krank-Biyel Tasarımı	184
5.3.1.3. $Y=\sin(x)$ Fonksiyonu için 4 çubuk tasarımı	189

6. SONUÇ VE ÖNERİLER	195
7. KAYNAKLAR	198
ÖZGEÇMİŞ	201
EKLER	202
EK 1	202
1. Uygulama Örnekleri	202
1.1. Akışkan Hızı Ölçme	202
1.2. Ağırlık Ölçme	204
1.3. Ağırlık Ölçme	205
1.4. İvme Ölçer	206

ÇİZELGELER DİZİNİ

ÇİZELGE	SAYFA
Çizelge 2.1. 5 noktada fonksiyon üretiminde optimum 4-çubuklar	5
Çizelge 2.2. Galerkin metodu ile 4- çubuklarda fonksiyon üretimi sonuçları.	7
Çizelge 2.3. Altbölge metodu ile 4- çubuklarda fonksiyon üretimi sonuçları.	7
Çizelge 3.1. $Y = x^2$, $0 \leq x \leq 1$, Fonksiyonunu Üreten Dört Çubuk Modülü Kinematik Boyutları	33
Çizelge 3.2. Biyel Kolu $Y = x^2$, $0 \leq x \leq 1$ Fonksiyonuna Göre Hareket Eden Dört Çubuk Modülüne ait Boyutlar	34
Çizelge 3.3. $Y = x^2$, $0 \leq x \leq 1$, Fonksiyonunu Üreten Krank-Biyel Modülü Kinematik Boyutları	52
Çizelge 3.4. $Y = x^2$, $0 \leq x \leq 1$, Fonksiyonunu Üreten Krank-Biyel Modülü Kinematik Boyutları	53

ŞEKİLLER DİZİNİ

ŞEKİL	SAYFA
Şekil 3.1. Fonksiyon Üreten Dört-Çubuk Mekanizması Tasarım Parametreleri	10
Şekil 3.2. Galarkin Metodu ile Tasarlanmış Dört Çubuk Modülü	33
Şekil 3.3. Yapısal Hata Dağılımı	34
Şekil 3.4. Kesin Nokta Metodu ile Tasarlanmış Dört Çubuk Modülü	35
Şekil 3.5. Yapısal Hata Dağılımı	36
Şekil 3.6. Fonksiyon Üreten Krank-Biyel Mekanizması Tasarım Parametreleri	37
Şekil 3.7. Galarkin Metodu ile Tasarlanmış Krank-Biyel Modülü	52
Şekil 3.8. Yapısal Hata Dağılımı	53
Şekil 3.9. Kesin Nokta Metodu ile Tasarlanmış Krank-Biyel Modülü	54
Şekil 3.10. Yapısal Hata Dağılımı	54
Şekil 4.1. Mek-Gen ana programın kod yapısı	62
Şekil 4.2. Four –Bar Programının Kod Yapısı	63
Şekil 4.3. Four Bar Programının Ana Görünüşü	64
Şekil 4.4. Dört Çubuk Mekanizması	65
Şekil 4.5. Four-Bar Programı Veri Giriş Menüsü	65
Şekil 4.6. Four-Bar Programının Çalışma Sentez Sırasındaki Görünüşü	66
Şekil 4.7. Four Bar Programı Veri Giriş Menüsü	67
Şekil 4.8. Four-Bar Programı Giriş Dosyaları	69
Şekil 4.9. Four Bar Programı Çıktı Dosyaları	70
Şekil 4.10. Four Bar Delta Programının Kod Yapısı	71
Şekil 4.11. Dört Çubuk Mekanizması	72
Şekil 4.12. Krank-Biyel Mekanizmasının Kod Yapısı	73
Şekil 4.13. Krank-Biyel Mekanizmasının Ekran Görünüşü	74
Şekil 4.14. Krank Biyel Mekanizması	74
Şekil 4.15. Krank-Biyel Delta Programının Kod Yapısı	75
Şekil 4.16. Mek-Gen Programının Ana Görüntüsü	76
Şekil 4.17. Four-Bar Programının Açıklanması	78
Şekil 4.18. Four Bar Programının Ekran Görüntüsü	82
Şekil 4.19. Four Bar Programının Veri Giriş Ekranı	83

Şekil 4.20. Sentez Yapılacak Metod Seçimi	84
Şekil 4.21. Dört Çubuk mekanizmasının temel yapısı	84
Şekil 4.22. Fonksiyon Giriş Menüsü	86
Şekil 4.23. İşlenen Fonksiyonun Görünüşü	87
Şekil 4.24. İşlenen Fonksiyonun Veri Giriş Ekranında Görülmesi	88
Şekil 4.25. Kayıt Yapılan Girdi Dosyaları	89
Şekil 4.26. Çıktı Dosyaları	90
Şekil 4.27. Sentezin Başlaması İçin Verilerin Gönderilmesi	91
Şekil 4.28. Sentezin Başlatılması	92
Şekil 4.29. Sentez Sonrası Ana Ekran	94
Şekil 4.30. Sentez Sonrası Analiz Sonuçları	95
Şekil 4.31. Sentez Sonrası Statik Grafikler	96
Şekil 4.32. Sentez sonrası detay verilerin ekrana yansımaları	96
Şekil 4.33. Fonksiyon Değeri ve X Değerinin Sentez Sırasında Aldıkları Değerler	97
Şekil 4.34. Sentez Sonrası Animasyon Çizimi	99
Şekil 4.35. Animasyonda İşlenen Çıktı Değerleri	100
Şekil 4.36. Seçilen Çözümün Hata Analizleri	101
Şekil 4.37. Seçilen Çözümün Ürettiği Çubuk Boyları	101
Şekil 4.38. Dinamik grafikler	104
Şekil 4.39. Statik grafikler	104
Şekil 4.40. Seçilen Çözüme Göre Hata Analizlerinin Değişimi	105
Şekil 4.41. X'e Göre Alac-Alth Hata Sapma Grafiği (Dinamik)	105
Şekil 4.42. X'e Göre Alac-Alth Hata Sapma Grafiği (Statik)	106
Şekil 4.43. X'e Göre Hata Sapma Grafiği (Dinamik)	107
Şekil 4.44. X'e Göre Hata Sapma Grafiği (Statik)	108
Şekil 4.45. Psi-Alac Değişim Grafiği (Dinamik)	108
Şekil 4.46. Psi-Alac Değişim Grafiği (Statik)	109
Şekil 4.47. Dört –Çubuk Mekanizması	111
Şekil 4.48. Veri Girişindeki Değişiklik	112
Şekil 4.49. Four-Bar Delta Programı Ekran Çıktısı	114
Şekil 4.50. Krank-Biyel Mekanizması Ana Görüntüsü	115

Şekil 4.51. Krank-Biyel Mekanizması Veri Girişi	116
Şekil 4.52. Krank-Biyel Mekanizması Metod Seçimi	117
Şekil 4.53. Krank-Biyel Mekanizması	117
Şekil 4.54. Krank-Biyel Mekanizması Senteze Başlama	120
Şekil 4.55. Krank-Biyel Mekanizması Sentez Çıktı Ekranı	123
Şekil 4.56. Krank-Biyel Mekanizması Hata Analizleri	124
Şekil 4.57. Krank-Biyel Mekanizması Statik Grafikler	125
Şekil 4.58. Krank-Biyel Mekanizması Detaylar Menüsü	125
Şekil 4.59. Krank-Biyel Mekanizması Fonksiyonunun Sentez Sırasındaki Değişimi	126
Şekil 4.60. Krank-Biyel Mekanizması Animasyon Sonrası Çıktı Ekranı	127
Şekil 4.61. Krank-Biyel Mekanizması Animasyonda Çubukların Yerleşimi	128
Şekil 4.62. Çözüme Göre Hata Analizlerinin Görülmesi	129
Şekil 4.63. Çözüme Göre Hata Çubuk Boylarının Görülmesi	129
Şekil 4.64. Dinamik Grafikler	130
Şekil 4.65. Statik Grafikler	131
Şekil 4.66. Sac-Sth-X Grafiği (Dinamik)	131
Şekil 4.67. Sac-Sth-X Grafiği (Statik)	132
Şekil 4.68. X-Hata Grafiği (Dinamik)	133
Şekil 4.69. X-Hata Grafiği (Statik)	133
Şekil 4.70. Psi-S Grafiği (Dinamik)	134
Şekil 4.71. Psi-Sac Grafiği (Statik)	134
Şekil 4.72. Krank-Biyel Delta İçin Mekanizma Tipi	135
Şekil 4.73. Veri Girişindeki Değişiklik	136
Şekil 4.74. Program Çıktı Ekranı	138
Şekil 5.1. $Y = \tan(x)$ Fonksiyon Girişi	139
Şekil 5.2. $Y = \tan(x)$ Veri Girişi	140
Şekil 5.3. $Y = \tan(x)$ Çıktı Ekranı	141
Şekil 5.4. $Y = \tan(x)$ Program Girdileri Çubuk Boyları ve Çözümler	141
Şekil 5.5. $Y = \tan(x)$ Text Dosyası	142
Şekil 5.6. $Y = \tan(x)$ Psi-Alac Hata Grafiği	143
Şekil 5.7. $Y = \tan(x)$ X-Hata Grafiği	143

Şekil 5.8. $Y = \tan(x)$ Psi-Alac Grafiği	144
Şekil 5.9. $Y = x^2$ Fonksiyon Girişi	144
Şekil 5.10. $Y = x^2$ Veri Girişi	145
Şekil 5.11. $Y = x^2$ Program Çıktısı	146
Şekil 5.12. $Y = x^2$ Program Girdileri Çubuk Boyları ve Çözümler	146
Şekil 5.13. $Y = x^2$ Text Dosyası	147
Şekil 5.14. $Y = x^2$ Psi-Alac Hata Grafiği	148
Şekil 5.15. $Y = x^2$ X-Hata Grafiği	148
Şekil 5.16. $Y = x^2$ Psi-Alac Grafiği	149
Şekil 5.17. $Y = \sin(x)$ Fonksiyon Girişi	149
Şekil 5.18. $Y = \sin(x)$ Veri Girişi	150
Şekil 5.19. $Y = \sin(x)$ Program Çıktısı	151
Şekil 5.20. $Y = \sin(x)$ Program Girdileri Çubuk Boyları ve Çözümler	151
Şekil 5.21. $Y = \sin(x)$ Text Dosyası	152
Şekil 5.22. $Y = \sin(x)$ X- Hata Grafiği	153
Şekil 5.23. $Y = \sin(x)$ Psi-Alac Grafiği	153
Şekil 5.24. $Y = \sin(x)$ Psi-Alac Hata Grafiği	154
Şekil 5.25. $Y = \tan(x)$ Fonksiyon Girişi	154
Şekil 5.26. $Y = \tan(x)$ Veri Girişi	155
Şekil 5.27. $Y = \tan(x)$ Çıktı Ekranı	156
Şekil 5.28. $Y = \tan(x)$ Program Girdileri Çubuk Boyları ve Çözümler	156
Şekil 5.29. $Y = \tan(x)$ Text Dosyası	157
Şekil 5.30. $Y = \tan(x)$ Psi-Alac Hata Grafiği	158
Şekil 5.31. $Y = \tan(x)$ X-Hata Grafiği	158
Şekil 5.32. $Y = \tan(x)$ Psi-Alac Grafiği	158
Şekil 5.33. $Y = x^2$ Fonksiyon Girişi	159
Şekil 5.34. $Y = x^2$ Veri Girişi	159
Şekil 5.35. $Y = x^2$ Program Çıktısı	160
Şekil 5.36. $Y = x^2$ Program Girdileri Çubuk Boyları ve Çözümler	161
Şekil 5.37. $Y = x^2$ Text Dosyası	162
Şekil 5.38. $Y = x^2$ Psi-Alac Grafiği	163
Şekil 5.39. $Y = x^2$ Psi-Alac Hata Grafiği	163

Şekil 5.40. $Y = x^2$ X-Hata Grafiği	164
Şekil 5.41. $Y = \text{Log}(x)$ Fonksiyon Girişi	164
Şekil 5.42. $Y = \text{Log}(x)$ Veri Girişi	165
Şekil 5.43. $Y = \text{Log}(x)$ Program Çıktısı	166
Şekil 5.44. $Y = \text{Log}(x)$ Program Girdileri Çubuk Boyları ve Çözümler	166
Şekil 5.45. $Y = \text{Log}(x)$ Text Dosyası	167
Şekil 5.46. $Y = \text{Log}(x)$ Psi-Alac Grafiği	168
Şekil 5.47. $Y = \text{Log}(x)$ Psi-Alac Hata Grafiği	168
Şekil 5.48. $Y = \text{Log}(x)$ X- Hata Grafiği	168
Şekil 5.49. $Y = \text{Tan}(x)$ Fonksiyon Girişi	169
Şekil 5.50. $Y = \text{Tan}(x)$ Veri Girişi	169
Şekil 5.51. $Y = \text{Tan}(x)$ Ekran Çıktısı	171
Şekil 5.52. $Y = \text{Tan}(x)$ Program Girdileri Çubuk Boyları ve Çözümler	171
Şekil 5.53. $Y = \text{Tan}(x)$ Program Text Dosyası	172
Şekil 5.54. $Y = \text{Tan}(x)$ Psi-Alac Grafiği	172
Şekil 5.55. $Y = \text{Tan}(x)$ Psi-Alac Hata Grafiği	173
Şekil 5.56. $Y = \text{Tan}(x)$ X-Hata Grafiği	173
Şekil 5.57. $Y = x^2$ Fonksiyon Girişi	174
Şekil 5.58. $Y = x^2$ Veri Girişi	174
Şekil 5.59. $Y = x^2$ Program Çıktısı	175
Şekil 5.60. $Y = x^2$ Program Girdileri Çubuk Boyları ve Çözümler	176
Şekil 5.61. $Y = x^2$ Text Dosyası	176
Şekil 5.62. $Y = x^2$ Psi-Alac Grafiği	177
Şekil 5.63. $Y = x^2$ Psi-Alac Hata Grafiği	177
Şekil 5.64. $Y = x^2$ X-Hata Grafiği	178
Şekil 5.65. $Y = \text{Tan}(x)$ Fonksiyon Girişi	179
Şekil 5.66. $Y = \text{Tan}(x)$ Veri Girişi	180
Şekil 5.67. $Y = \text{Tan}(x)$ Çıktı Ekranı	181
Şekil 5.68. $Y = \text{Tan}(x)$ Program Girdileri Çubuk Boyları ve Çözümler	182
Şekil 5.69. $Y = \text{Tan}(x)$ Text Dosyası	182
Şekil 5.70. $Y = \text{Tan}(x)$ Psi-Alac Grafiği	183
Şekil 5.71. $Y = \text{Tan}(x)$ Psi-Alac Hata Grafiği	183

Şekil 5.72. $Y = \tan(x)$ X-Hata Grafiği	184
Şekil 5.73. $Y = x^2$ Fonksiyon Girişi	184
Şekil 5.74. $Y = x^2$ Veri Girişi	185
Şekil 5.75. $Y = x^2$ Program Çıktısı	186
Şekil 5.76. $Y = x^2$ Program Girdileri Çubuk Boyları ve Çözümler	186
Şekil 5.77. $Y = x^2$ Text Dosyası	187
Şekil 5.78. $Y = x^2$ Psi-Alac Grafiği	188
Şekil 5.79. $Y = x^2$ Psi-Alac Hata Grafiği	188
Şekil 5.80. $Y = x^2$ X-Hata Grafiği	189
Şekil 5.81. $Y = \sin(x)$ Fonksiyon Girişi	189
Şekil 5.82. $Y = \sin(x)$ Veri Girişi	190
Şekil 5.83. $Y = \sin(x)$ Program Çıktısı	191
Şekil 5.84. $Y = \sin(x)$ Program Girdileri Çubuk Boyları ve Çözümler	191
Şekil 5.85. $Y = \sin(x)$ Text Dosyası	192
Şekil 5.86. $Y = \sin(x)$ X- Hata Grafiği	193
Şekil 5.87. $Y = \sin(x)$ Psi-Alac Grafiği	193
Şekil 5.88. $Y = \sin(x)$ Psi-Alac Hata Grafiği	194
Şekil 8.1. Akışkan hızı ölçen düzenek.	203
Şekil 8.2. Ağırlık ölçmek için bir düzenek	204
Şekil 8.3. Ağırlık ölçen düzenek	205
Şekil 8.4. İvme ölçen bir düzenek	206

1. GİRİŞ

Mekanizma, kuvvet veya hareket aktarmaya yarayan, yada bir cismin noktalarını belirli yörüngeler üzerinde ilerletmeye yarayan mekanik bir düzenlemedir [1]. Bu yüzden mekanizma öncelikle hareket fikrini akla getirir. Mekanizmalarda hareket aktarımı, mekanizmanın kinematik değerleri ile ilişkili bir aktarım fonksiyonuna göre gerçekleşir. Giriş uzvundan verilen bir hareket, belirlenen fonksiyonel dönüşümle çıkış uzvundan alınır. Böylelikle mekanizmalar, giriş ve çıkış uzvu hareketlerinin kontrolünü sağlayan ve bu kontrol işlevini karakterize eden bilgilerin, mekanizmaya ait kinematik büyüklüklerde saklandığı bir bilgi deposu olarak düşünülürler. Bu kinematik değerlerin, istenen hareket kontrol işlevini sağlamak üzere seçilmesi ise fonksiyon üretimi problemi olarak adlandırılır.

Makinalarda kullanılan fonksiyonel dönüşümler çok çeşitli mekanizmalar yardımı ile gerçekleştirilebilir. Kam, dişli, raçet vs. gibi mekanizmalar bunlara örnek verilebilir [2]. Kol mekanizmaları bunların içerisinde diğerlerine göre daha ekonomik ve basit olan önemli bir mekanizmadır. Kol mekanizmaları, diğer çeşitli uygulamalarda olduğu gibi, teorik ve kinematik literatürde de köklü bir tarihe sahiptir. James Watt'ın buhar makinası için yaptığı mühendislik uygulamasından; günümüzde dağ bisikletlerinden protez ayaklara kadar birçok modern uygulamada kol mekanizmaları kullanılmaktadır.

Düzlemsel Kol Mekanizmalarının tasarımında giriş ve çıkış kollarının hareket aralığının, zamanın veya giriş kolunun hareketinin bir fonksiyonu olması istenir.

Mekanizma tasarımında kullanılacak bir fonksiyon üretiminde ilk adım mekanizmaya ait giriş parametrelerinin saptanmasıdır. Bu durumu hatasız yapabilmek gayet maliyetli olduğundan fonksiyonu üretebilecek yaklaşık değerlerin hesaplanması daha kolaydır. Bu sebeple günümüzde mekanizma üretimi için yaklaşık senteze ilişkin yöntemler tercih edilmektedir.

Mekanizmalarda giriş ve çıkış uzuvları arasında çok karmaşık matematiksel ilişkilerin olması nedeniyle önceleri sentez problemlerine grafiksel yaklaşımlarla çözümler arama yoluna gidilmiştir [3]. Fakat grafiksel metotlarda çözüm, çizim hassasiyeti ile sınırlı ve her yeni veri için tekrarı gerektirdiğinden uzun ve sıkıcı bir yöntemdir. Hızlı hesaplama olanağına sahip bilgisayarın ortaya çıkması yaklaşık fonksiyon üretimi problemlerinde çok sayıda analitik yöntemin geliştirilmesini olanaklı kılmıştır.

Geliştirilen analitik yöntemlerin ortak noktası, hata analizleri yaparak minimum hatayı sağlayan çözümler bulmaktır. Fonksiyon sentez problemlerinde en çok kullanılan ve belirgin yöntemler olarak kesin nokta metodu [4], alt bölge metodu [5], galerkin metodu [6], en küçük kareler metodunu [7] sayabiliriz.

Bu çalışmada hedeflenen, düzlemsel kol mekanizmalarının fonksiyon ve yörünge sentezi için etkileşimli bir bilgisayar paket programı tasarımıdır. Mek-Gen isimli bu bilgisayar programı sayesinde mekanizmaların bilinen kinematik değerleri istenen fonksiyon ile derlenerek, seçilen hata analiz yöntemi ile hareketin sağlanabileceği aralıklar, hata analizleri ve uzuv oranları elde edilir.

Mek-Gen programında şu ana kadar gelinen aşamada, düzlemsel dört çubuk mekanizmalarının ve Krank-biyel mekanizmalarının fonksiyon sentezi; alt bölge metodu, galerkin metodu ve kesin noktalar metodu kullanılarak sağlanmaktadır. Programın yapılan bu kısmına Mek-Gen version 1.0 adı verilmiştir.

Düzlemsel dört çubuk mekanizmalarının ve Krank-Biyel mekanizmalarının fonksiyon sentezinde Mek-Gen, kullanıcının kendi fonksiyonunu görsel arayüzde oluşturma imkanı sunmaktadır. Bu özellik, Mek-Gen'i konu ile ilgili yazılmış diğer yazılımlardan ayıran en önemli özelliklerden biridir. Senteze girilen veriler ile sentez sonucu sağlanan hata sapmaları ve değerler programda kaydedilebildiği gibi, kaydedilmiş olan veriler çağrılıp yeniden sentez yapılabilmektedir.

Girilen tasarım deęerleri doęrultusunda oluřturulan mekanizmanın, her tasarım deęeri için ayrı ayrı yapılmakta, saęlanan grafiksel veriler de aynı ekranda görülebilmektedir. Animasyon sırasında kullanılan uzuvların oranları, kullanılan açısal deęerler ve fonksiyon deęerleri sentez sonrası detaylı olarak görülebilmektedir. Ayrıca animasyon ekranında büyütme-küçültme hızlı-yavaş geçişlerin kullanıcı tarafından deęiřtirilebilir yapıda olması ayrıntıların daha iyi görülebilmesini saęlamaktadır.

Geliřmiř bir grafik arayüzüne sahip olan Mek-Gen, kullanılacak tüm kinematik deęerlerin programa çok kolay bir şekilde işlenebilmesi konusunda kullanıcıya kolaylık saęlamaktadır.

Mek-Gen programı, ilk defa bu tez çalışması sonucu ortaya çıkarılmış görsel bir programdır. Programın kodlanmasında, Delphi 6.0 enterprise programlama dili kullanılmıştır. Bu editörün kullanılmasındaki en önemli etkenler; kullanıcıya windows ortamında nesneye yönelik (Object Oriented) görsel bir arayüzün sunulabilmesi ve programda kullanılan matematiksel ve fonksiyonel çıktıları grafik ortamda daha rahat görebilme imkanınıdır. Delphi 6.0, %100 Windows uyumlu bir uygulamadır ve dolayısıyla Mek-Gen , tüm NT tabanlı (Windows 2000, 2000 Server, Me, Xp, Vista ...) işletim sistemlerinde çalışabilmektedir. Ayrıca Delphi'nin dünyanın en hızlı derleyicilerinden birine sahip olması, nesne yönelimli ve VCL (Visual Component Library) özellikleri sayesinde program içerisinde mevcut sınıflardan yeni bileşenler oluşturmaya mümkün kılması, dinamik bir hafıza yapısının olması program yazımında tercih edilmesinin dięer nedenleridir.

Saęlam bir omurgaya sahip olan Mek-Gen'in kod yapısı, daha sonraki çalışmalarda geliştirilebilir şekilde tasarlandığından, yeni modüller eklenerek farklı mekanizma tasarımlarında da kullanılabilir niteliktedir. Bu özellięi sayesinde Mek-Gen, ilgili dięer mekanizma sentez programları ile baędařtırılabilir ve kullanılması hedeflenen uygun çalışma alanlarına uyumlu hale getirilebilir.

2. KAYNAK ARAŞTIRMALARI

Çeşitli mekanizma tasarımları yapmak üzere, geçmişten bugüne kadar farklı metodlarla pek çok çalışmalar yapılmıştır. Bu çalışmalar, zamanın gerektirdiği şartlara uygun ortamlarda geliştirilmiş olup, her yapılan yeni çalışma, mekanizma tasarım tekniğine yeni bulgular kazandırmıştır.

Mekanizma denince akla gelen fikirlerden birisi kinematik fikridir. Kinematik terimi ilk olarak 'Ampere' tarafından yapılan çalışmalar sonucunda ortaya çıkmıştır. 19. yüzyıl içerisinde yapılan çalışmalar ile mekanizma kinematiği farklı alanlarda gelişme sağlamıştır [8]. Mekanizmanın biçimsel sınıflandırılması ve bu sınıflandırmaların sembolik gösterimleri ile tasarım elde etme ise ilk defa 1876 yılında Reuleaux [9] tarafından gerçekleştirildi. Yine bu yüzyılın sonlarında, Burnmester tarafından dört-çubuk mekanizmalarının sentezi için kesin noktalar metodu geliştirildi.[10,11,12,13]

Düzlemsel kol mekanizmaları ile ilgili grafik alanındaki çalışmalar ise 20. Yüzyılın ilk yarısında yapıldı. Grafik metodlar, mekanizmanın önceden belirlenen komutlarının tam elde edilebilmesini sağlayacak biçimde bilinmeyen parametrelerin çizim yardımı ile bulunması esasına dayanır. Verilen nokta-konum sayısı artıca, grafiksel metodlarda çözüm güçlüğünün yanı sıra hassasiyet de azalır. Konum sayısı beşi geçtiğinde, yaklaşım deneme yanılmaya dayanmakta, çözüm yaklaşık bir nitelik almaktadır. Artık problemde önemli olan konumları kesin olarak üretmek değil, verilen bir fonksiyona en yakın fonksiyonun elde edilebileceği mekanizmayı tasarlamaktır. Deneme yanılmayı içeren grafik bir tasarım yöntemi çakıştırma (Overlay) metodudur [14]

Analitik yaklaşımlardan birisi, daha önce de sözü edilen kesin nokta sentezidir. Bu metod, mekanizmanın hareket aralığında, mekanizmaya ait parametre sayısı kadar noktada, üretilen fonksiyonla istenen fonksiyonun birbirine eşitlenmesi ve ortaya çıkan denklemin bilinmeyen parametreleri için çözülmesi esasına dayanır.

Yöntem, nokta sayısı üç ise üç-nokta, dört ise dört-nokta, beş ise beş-nokta sentezi şeklinde adlandırılır [15]. Aynı konu ile ilgili Hartenberg ve Denavit [16], Beya [17] ve Hall [18] da çalışmalar yapmıştır.

Hartenberg ve Denavit [16] beş noktada kesin nokta metodunu 4- çubukların tasarımında kullanmışlardır.

$Y = f(x)$ şeklindeki fonksiyonların 4- çubukla üretilmesinde, optimum tasarımlar olarak elde edilen sonuçlar çizelge 2.1’de görülmektedir.

Çizelge 2.1. 5 noktada fonksiyon üretiminde optimum 4-çubuklar

Fonksiyon	Log x	Sin x	Tan x	x ²	x ²
X in tanım aralığı	1 < x < 2	0 < x < 90°	0 < x < 45°	0 < x < 1	-1 < x < 1
Δθ (°)	60	90	90	90	90
Δψ (°)	60	90	90	90	60
Kesin Noktalar					
x1	1.0144678	2.76333673	1.6246052	0.03368	-0.945
x2	1.1355580	21.988925	12.8424451	0.24917	-0.525
x3	1.3842742	48.226892	27.47676	0.54280	0.049
x4	1.7004985	71.414168	38.542302	0.81636	0.672
x5	1.9627009	87.54952	44.269088	0.97863	0.976
θ1 (°)	52.6283	-62.263261	269.70917	-29.32	-21.18
ψ1 (°)	259.07749	75.606709	124.18966	233.83	-53.67
Çubuk Boy					
a1/a4	-3.23485	1.8343688	-2.6603189	2.5233	-0.610
a2/a4	0.8456021	2.2385439	7.430444	3.3295	0.565
a3/a4	3.4855187	-0.6936455	8.6856194	-0.5559	0.380
En büyük yapısal hata %	0.0062	0.21	0.042	0.0748	4.47

Çizelge 2.1. yapısal hatanın en büyük değeri ile 4- çubuk mekanizmasının kol boyutlarını de içermektedir. İyi bir yapısal hatanın yanı sıra tasarım kalitesi açısından kol boyutlarının uygunluğu önemlidir. Çalışmada yapısal hatanın tüm hareket aralığı boyunca dağılımın nasıl olduğu verilmiştir.

20. yüzyılda yapılan çalışmalarda Freudenstein tarafından, belirlenen pozisyonlarda analitik sentez yapan bir metod geliştirildi[13]. Freudenstein kinematikte yeni bir çağ başlattı. Kinematik teorisi ile ilgili değerli fikir ve

methodları ile mekanizma sentezine yardım etmiş ve hala etmektedir. O günden itibaren, Kinematik sentezle ilgili hemen hemen bütün yayınlarda Freudenstein' ın yayınları referans olarak gösterildi

Loerch [19] , 20.yy da yaptığı bir takım çalışmalarda dyadic yaklaşım ve çember - merkez nokta kavislerini kullandı. Kramer, kesin noktalar metodunun kullanılmadığı mekanizma sentezlerini yapmak amacıyla hareket, yörünge ve fonksiyon çizimi yapabilen optimizasyon sentezi üzerine çalışmalar yaptı. Wilson' un [19] yayınları gibi analitik metodu çözümlen çesitli çalışmalar geliştirildi. Claussen ve Dizioğlu [19], kesin noktalar metodu ile fonksiyon sentezi üzerine çalışmalar yaptı.

Bir başka çalışmada Akçalı ve Dittrich [20] , Galerkin metodunu 4-çubuk tasarımına uygulamıştır. Galerkin yöntemi, çözüm bölgesinde sıfır olması istenen bir 'Kalıntı Fonksiyonu' ve bu kalıntı fonksiyonunu çeşitli oranlardaki katkılarıyla oluşturan tanım bölgesinde sürekli 'ağırlık fonksiyonları' gibi iki temel öğeyi içermektedir. Kalıntı fonksiyonu içindeki parametreler, ağırlık ile kalıntı fonksiyonlarının çarpımının bölgedeki integrali sıfır olacak şekilde ayarlanır. Galerkin metoduyla gerçekleştirilen bazı 4- çubuk tasarımları çizelge 2.2 de sunulmuştur.

Fonksiyon sentez problemlerinde kullanılan bir başka metod da, daha önce anılan altbölge metodudur [21]. Bu yöntemde hareket aralığı mekanizmada varolan parametre sayısı kadar alt aralıklara bölünür. Kapalı bir hata ifadesi içindeki mekanizma parametrelerinin değerleri her alt aralıktaki ortalama yapısal hatanın sıfır olması şartından hareketle belirlenir. Akçalı ve Lindholm [21], altbölge metodunu 4-çubukla fonksiyon üretilmesine uygulamışlar ve çizelge 2.3 de sunulan bazı sonuçları elde etmişlerdir.

Çizelge 2.2. Galerkin metodu ile 4- çubuklarda fonksiyon üretimi sonuçları.

Fonksiyon	Sin x	Tan x	x ²
X in tanım aralığı	1<x<90°	0<x<45°	0<x<1
Δψ (°)	90	90	90
Δα (°)	60	60	60
Ağırlık			
Fonksiyon			
w1	1.00	1.00	1.00
w2	cos ψ'	cos ψ'	0.0174x
w3	sin ψ'	sin ψ'	0.0174x ²
w4	cos α'	cos α'	0.0174x ³
w5	sin α'	cos α' - ψ'	0.0174x ⁴
ψ0 (°)	89.8709	-89.9938	-10.289
α0 (°)	62.3125	24.8585	82.1175
Çubuk Boy			
X1	-0.7488	42.0822	-1.8934
X2	9.6568	58.7830	11.8045
X3	-0.7298	-32.0972	-1.6809
En büyük yapısal hata %	0.66	-0.15	0.78

Çizelge 2.3 Altbölge metodu ile 4- çubuklarda fonksiyon üretimi sonuçları.

Fonksiyon	Sin x	log x
X in tanım aralığı	1<x<90°	0<x<10
Δψ (°)	90	90
Δα (°)	60	60
ψ0 (°)	292.79	41.87
α0 (°)	274.31	289.73
Çubuk Boy		
X1	-0.6696	1.2833
X2	1.2202	0.7659
X3	-0.5594	-1.5378
En büyük yapısal hata %	0.22	-0.66

Bilgisayar destekli kinematik ve dinamik simülasyonlar, tasarlanan mekanizmaların üretilmeden önce görülebilmesi için kullanıcılara büyük kolaylık sağladı. Çok fonksiyonlu mekanizma simülasyonları için ilk bilgisayar destekli

alıřmalar 1960 ve 1970 li yıllarda yapıldı. Bu programlarda, mekanizma sentez ve analizleri yapılamamakta, sadece mekanizmanın simulasyonu oluşturulabilmekteydi. 1970'lerden sonra, bilgisayar grafikleri kullanabilen sentez paketleri geliřtirilmeye bařlandı. KINSYN adlı bilgisayar programı, bilgisayar grafięi kullanarak Kaufman [22] tarafından geliřtirildi.

1970'lerin sonunda, gnmzn en nemli bilgisayar paketlerinden olan LINCAGES'in ilk versiyonu Erdman [23] tarafından geliřtirildi. Bu program mekanizma sentez programlarının ilk versiyonu olarak bilinmektedir. Bu program, Windows NT ve MacOS tabanlı bilgisayarlarda Grafik arayz kullanılarak alıřtırılabilmektedir. Programın gnmzde kullanılan LINCAGES 4 versiyonu, 4-ubuklu mekanizmaların sentezini yapabilmektedir. Bu program, 3 ve 4 noktada fonksiyon, yrnge ve hareket sentezi yapabilmektedir.

Polat [10], yaptıęı alıřmalarda, kesin noktalar metodu kullanarak dzlemsel Kol mekanizmalarının sentezini yapabilen MECSYN isimli bir bilgisayar programı geliřtirmiřtir. Belirlenmiř noktalarda mekanizma sentezi yapabilen bu program genel amalı bir dizayn paketi olmuřtur. Yalnız grafik arayzndeki sorunlar, programının kullanılmasında sorunlar yaratmıřtır.

Ruth ve McCarthy [24] tarafından geliřtirilen dięer bir mekanizma sentez paketinde, drt-ubuk mekanizmalarının 4 pozisyonda sentezi gerekleřtirilmektedir. SphinxPC isimli bu bilgisayar program paketi, mekanizma sentezi iin Spherical Burmester teorisi kullanmaktadır.

Son 15 yıllık gemiře baktıęımızda, geliřtirilen bilgisayar programlama dilleri sayesinde nesnelerin bilgisayar ortamındaki hareketleri daha grsel bir nitelik kazanmıřtır. Object Oriented Programming (Nesne ynelimli programlama) sayesinde geliřtirilen yazılımlarda kullanıcı tanımlı veri tipleri, sınıflar ve ktphaneler kullanılabilmektedir. Bu durum sayesinde programcı deęiřik modellemeler yapabilen grafik yazılımlara, nmerik girdileri etkileřimli bir řekilde iřleyebilmektedir. Windows iřletim sistemi altında alıřan birok gnmz

programlama dili sayesinde, bilgisayar ortamında grafik arayüzler kullanılarak çeşitli yazılımlar geliştirilebilmektedir. Bu dillere örnek olarak, Visual Basic, Delphi, C++, Visual C++ ve Java gösterilebilir.

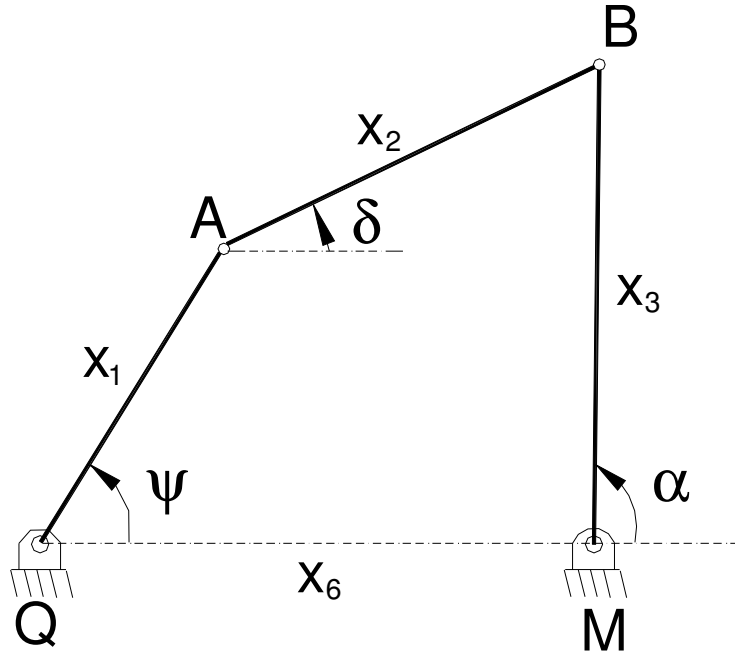
Mek - Gen (Version1.0) programı, tamamen nesneye yönelimli bir yazılım olup, girilen mekanizma boyutlarına uygun mekanizma tasarımlarını ve tasarımlara bağlı hata analizlerini üreten bir programdır. Ayrıca üretilen mekanizmanın görsel tasarımı ve işleyişi, kullanıcıya yansıtılabilmektedir. Kullanıcı etkileşimli bu program, her mekanizma tasarımcısının rahatlıkla kullanabileceği ve seçilen metodlar doğrultusunda çok iyi tasarım sonuçlarını elde edebileceği bir programdır. Programda işlenen veriler kullanıcı tarafından kaydedilebilir, kaydedilen veriler daha sonra işlenebilir niteliktedir. Tasarımı yapılan mekanizmanın daha rahat görülebilmesi için kullanıcıya animasyonu durdurma-yeniden başlatma, büyültme-küçültme, animasyonu hızlandırma - yavaşlatma, koordinat düzleminde yukarı - aşağı sağa - sola kaydırma gibi bir çok özellik sağlanmıştır. Tüm özellikleriyle Mek-Gen diğer programlardan daha farklıdır.

Mekanizma sentezi üzerine geliştirilen metodlar ve metodların kullanıldığı bilgisayar yazılımları daha da geliştirilebilir fikir ve sonuçlar içermektedir. Yapılan çalışmalarda, bu fikir ve sonuçlardan da yararlanılarak yapısal hatanın bundan önce elde edilen düzeylerinden daha düşük ve tasarım kalitesi daha yüksek çözümler, çözümlere bağlı dinamik görüntüler hedeflenmektedir.

3. MATERYAL VE METOD

3.1. FONKSİYON ÜRETEN 4 ÇUBUK MODÜLÜ

İstenen bir fonksiyonu üreten dört çubuk mekanizmasının tasarım parametreleri Şekil 3.1. de sunulmuştur. Burada, x_1 krank boyunu, x_2 biyel boyunu, x_3 çıkış kolu uzunluğunu, ψ giriş kolu açısını, α çıkış kolu açısını, ψ_0 , α_0 sırası ile giriş ve çıkış kollarının harekete başlama açısını simgelemektedir. Burada, dört çubuk mekanizmasının fonksiyon üretimi kapsamında tasarımında iki farklı fonksiyonel ilişki ele alınmıştır. Bunlar, giriş kolu dönüşü (ψ) ile çıkış kolu dönüşü (α) ve biyel kolunun yatayla yaptığı (δ) açısı arasındaki fonksiyonel ilişkidir.



Şekil 3.1. Fonksiyon Üreten Dört-Çubuk Mekanizması Tasarım Parametreleri

3.1.1. Giriş Kolu Açısı (ψ) ile Çıkış Kolu Açısı (α) Fonksiyonel İlişisini Üreten Dört Çubuk Tasarımı:

ψ - α fonksiyonel bağıntısının üretilmesi için, Şekil 3.1. deki dört çubuk mekanizmasının *QABM* kapalı döngüsünden (3.1.1-2) deki vektörel ifadeler yazılabilir:

$$x_1 \cos \psi + x_2 \cos \delta = x_6 + x_3 \sin \alpha \quad (3.1.1)$$

$$x_1 \sin \psi + x_2 \sin \delta = x_3 \sin \alpha \quad (3.1.2)$$

(3.1.1),(3.1.2) ifadelerinden (δ) açısı yok edilir ve fonksiyon değişkeni (x) ile giriş kolu açısı (ψ) ve çıkış kolu açısı (α) ile de fonksiyon bağımlı değişkeni (y) arasında (3.1.5) de verilen bağıntılar kurulursa, (3.1.3)'deki doğrusal olmayan denklem takımı elde edilir.

$$a_i Z_3 + Z_1 (b_i \cos \alpha_0 - c_i \sin \alpha_0) - Z_2 (d_i \sin(\alpha_0 - \psi_0) + e_i \cos(\alpha_0 - \psi_0)) + f_i \sin \psi_0 - g_i \cos \psi_0 = 0 \quad i=1,2,3,4,5 \quad (3.1.3)$$

Burada,

$$Z_1 = \frac{x_3}{x_1}; \quad Z_2 = \frac{x_3}{x_6}; \quad Z_3 = \frac{x_1^2 - x_2^2 + x_3^2 + x_6^2}{2x_1 x_6} \quad (3.1.4)$$

$$\begin{aligned} \psi &= \psi_0 + \psi'; & \psi' &= R_x (x - x_0); & R_x &= \frac{\Delta \psi}{\Delta x} \\ \alpha &= \alpha_0 + \alpha'; & \alpha' &= R_y (y - y_0); & R_y &= \frac{\Delta \alpha}{\Delta y} \end{aligned} \quad (3.1.5)$$

Burada, tasarımcı tarafından belirlenen, $(\Delta \psi = \psi_n - \psi_0)$, $(\Delta \alpha = \alpha_n - \alpha_0)$, sırası ile, giriş kolu ve çıkış kolu çalışma aralığı, $(\Delta x = x_n - x_0)$, $(\Delta y = y_n - y_0)$ ise üretilen fonksiyonun bağımsız ve bağımlı değişkenlerinin değişim miktarlarıdır.

Ayrıca (3.1.3) denklem takımındaki üç ayrı metoda göre hesaplanabilen $(a_i, b_i, c_i, d_i, e_i, f_i, g_i)$ katsayılar (3.1.6)'da verilmiştir.

3.1.1.1. Kesin Nokta Metodu:

$$\left. \begin{aligned} a_i &= 1 \\ b_i &= \text{Cos}\alpha'_i \\ c_i &= \text{Sin}\alpha'_i \\ d_i &= \text{Sin}(\alpha'_i - \psi'_i) \\ e_i &= \text{Cos}(\alpha'_i - \psi'_i) \\ f_i &= \text{Sin}\psi'_i \\ g_i &= \text{Cos}\psi'_i \end{aligned} \right\} i = 1,2,3,4,5 \quad (3.1.6)$$

3.1.1.2. Alt Bölge Metodu:

$$\left. \begin{aligned} a_i &= \int_{x_{i-1}}^{x_i} dx \\ b_i &= \int_{x_{i-1}}^{x_i} \text{Cos}\alpha' dx \\ c_i &= \int_{x_{i-1}}^{x_i} \text{Sin}\alpha' dx \\ d_i &= \int_{x_{i-1}}^{x_i} \text{Sin}(\alpha' - \psi') dx \\ e_i &= \int_{x_{i-1}}^{x_i} \text{Cos}(\alpha' - \psi') dx \\ f_i &= \int_{x_{i-1}}^{x_i} \text{Sin}\psi' dx \\ g_i &= \int_{x_{i-1}}^{x_i} \text{Cos}\psi' dx \end{aligned} \right\} i = 1,2,3,4,5 \quad (3.1.7)$$

3.1.1.3. Galerkin Metodu:

$$\left. \begin{aligned}
 a_i &= \int_{x_0}^{x_n} w_i dx \\
 b_i &= \int_{x_0}^{x_n} \text{Cos} \alpha' w_i dx \\
 c_i &= \int_{x_0}^{x_n} \text{Sin} \alpha' w_i dx \\
 d_i &= \int_{x_0}^{x_n} \text{Sin}(\alpha' - \psi') w_i dx \\
 e_i &= \int_{x_0}^{x_n} \text{Cos}(\alpha' - \psi') w_i dx \\
 f_i &= \int_{x_0}^{x_n} \text{Sin} \psi' w_i dx \\
 g_i &= \int_{x_0}^{x_n} \text{Cos} \psi' w_i dx
 \end{aligned} \right\} i = 1,2,3,4,5$$

(3.1.8)

Kapalı çözümlere ulaşmak amacı ile, (3.1.3) denklem kümesinden sırası ile (Z_2, Z_3) ve (Z_1) yok edilip yeniden düzenlenirse, $(\text{Tan}(\psi_0/2))$ ve $(\text{Tan}(\alpha_0/2))$ bilinmeyen parametrelerine bağlı doğrusal olmayan iki denklem elde edilir.

$$\begin{aligned}
& v^4 \left\{ (a_{pk} + b_{pk} + c_{pk} + d_{pk})t^4 + 4(f_{pk} - g_{pk})t^3 + 2(a_{pk} + b_{pk} - 3c_{pk} - 3d_{pk})t^2 + \right. \\
& \left. 4(g_{pk} - f_{pk})t + (a_{pk} + b_{pk} + c_{pk} + d_{pk}) \right\} + \\
& v^3 \left\{ -4(e_{pk} + f_{pk})t^4 + 16c_{pk}t^3 - 8(e_{pk} - 3f_{pk})t^2 - 16c_{pk}t - 4(e_{pk} + f_{pk}) \right\} + \\
& v^2 \left\{ 2(a_{pk} - 3b_{pk} - 3c_{pk} + d_{pk})t^4 - 4(3f_{pk} + g_{pk})t^3 + 4(a_{pk} - 3b_{pk} + 9c_{pk} - 3d_{pk})t^2 + \right. \\
& \left. 4(3f_{pk} + g_{pk})t + 2(a_{pk} - 3b_{pk} - 3c_{pk} + d_{pk}) \right\} + \\
& v \left\{ 4(e_{pk} + f_{pk})t^4 - 16c_{pk}t^3 + 8(e_{pk} - 3f_{pk})t^2 + 16c_{pk}t + 4(e_{pk} + f_{pk}) \right\} + \\
& \left\{ (a_{pk} + b_{pk} + c_{pk} + d_{pk})t^4 + 4(f_{pk} - g_{pk})t^3 + 2(a_{pk} + b_{pk} - 3c_{pk} - 3d_{pk})t^2 + \right. \\
& \left. 4(g_{pk} - f_{pk})t + (a_{pk} + b_{pk} + c_{pk} + d_{pk}) \right\} = 0, k = 1, 2
\end{aligned} \tag{3.1.9}$$

Burada,

$$v = \tan \frac{\alpha_0}{2}; \quad t = \tan \frac{\psi_0}{2}; \tag{3.1.10}$$

(3.1.9) ifadesindeki hesaplanabilir katsayılar aşağıda verilmiştir.

$$\begin{aligned}
a_{p1} = & a_k b_3 d_2 f_1 - a_3 b_k d_2 f_1 - a_k b_2 d_3 f_1 + a_2 b_k d_3 f_1 + a_3 b_2 d_k f_1 - a_2 b_3 d_k f_1 + \\
& a_k c_3 e_2 f_1 - a_3 c_k e_2 f_1 - a_k c_2 e_3 f_1 + a_2 c_k e_3 f_1 + a_3 c_2 e_k f_1 - a_2 c_3 e_k f_1 - \\
& a_k b_3 d_1 f_2 + a_3 b_k d_1 f_2 + a_k b_1 d_3 f_2 - a_1 b_k d_3 f_2 - a_3 b_1 d_k f_2 + a_1 b_3 d_k f_2 - \\
& a_k c_3 e_1 f_2 + a_3 c_k e_1 f_2 + a_k c_1 e_3 f_2 - a_1 c_k e_3 f_2 - a_3 c_1 e_k f_2 + a_1 c_3 e_k f_2 + \\
& a_k b_2 d_1 f_3 - a_2 b_k d_1 f_3 - a_k b_1 d_2 f_3 + a_1 b_k d_2 f_3 + a_2 b_1 d_k f_3 - a_1 b_2 d_k f_3 + \\
& a_k c_2 e_1 f_3 - a_2 c_k e_1 f_3 - a_k c_1 e_2 f_3 + a_1 c_k e_2 f_3 + a_2 c_1 e_k f_3 - a_1 c_2 e_k f_3 - \\
& a_3 b_2 d_1 f_k + a_2 b_3 d_1 f_k + a_3 b_1 d_2 f_k - a_1 b_3 d_2 f_k - a_2 b_1 d_3 f_k + a_1 b_2 d_3 f_k - \\
& a_3 c_2 e_1 f_k + a_2 c_3 e_1 f_k + a_3 c_1 e_2 f_k - a_1 c_3 e_2 f_k - a_2 c_1 e_3 f_k + a_1 c_2 e_3 f_k - \\
& a_k c_3 d_2 g_1 + a_3 c_k d_2 g_1 + a_k c_2 d_3 g_1 - a_2 c_k d_3 g_1 - a_3 c_2 d_k g_1 + a_2 c_3 d_k g_1 + \\
& a_k b_3 e_2 g_1 - a_3 b_k e_2 g_1 - a_k b_2 e_3 g_1 + a_2 b_k e_3 g_1 + a_3 b_2 e_k g_1 - a_2 b_3 e_k g_1 + \\
& a_k c_3 d_1 g_2 - a_3 c_k d_1 g_2 - a_k c_1 d_3 g_2 + a_1 c_k d_3 g_2 + a_3 c_1 d_k g_2 - a_1 c_3 d_k g_2 - \\
& a_k b_3 e_1 g_2 + a_3 b_k e_1 g_2 + a_k b_1 e_3 g_2 - a_1 b_k e_3 g_2 - a_3 b_1 e_k g_2 + a_1 b_3 e_k g_2 - \\
& a_k c_2 d_1 g_3 + a_2 c_k d_1 g_3 + a_k c_1 d_2 g_3 - a_1 c_k d_2 g_3 - a_2 c_1 d_k g_3 + a_1 c_2 d_k g_3 + \\
& a_k b_2 e_1 g_3 - a_2 b_k e_1 g_3 - a_k b_1 e_2 g_3 + a_1 b_k e_2 g_3 + a_2 b_1 e_k g_3 - a_1 b_2 e_k g_3 + \\
& a_3 c_2 d_1 g_k - a_2 c_3 d_1 g_k - a_3 c_1 d_2 g_k + a_1 c_3 d_2 g_k + a_2 c_1 d_3 g_k - a_1 c_2 d_3 g_k - \\
& a_3 b_2 e_1 g_k + a_2 b_3 e_1 g_k + a_3 b_1 e_2 g_k - a_1 b_3 e_2 g_k - a_2 b_1 e_3 g_k + a_1 b_2 e_3 g_k -
\end{aligned}$$

$$\begin{aligned}
& a_k c_2 d_1 g_3 - a_2 c_k d_1 g_3 - a_k c_1 d_2 g_3 + a_1 c_k d_2 g_3 + a_2 c_1 d_k g_3 - a_1 c_2 d_k g_3 + \\
& a_k b_2 e_1 g_3 - a_2 b_k e_1 g_3 - a_k b_1 e_2 g_3 + a_1 b_k e_2 g_3 + a_2 b_1 e_k g_3 - a_1 b_2 e_k g_3 - \\
& a_3 c_2 d_1 g_k + a_2 c_3 d_1 g_k + a_3 c_1 d_2 g_k - a_1 c_3 d_2 g_k - a_2 c_1 d_3 g_k + a_1 c_2 d_3 g_k - \\
& a_3 b_2 e_1 g_k + a_2 b_3 e_1 g_k + a_3 b_1 e_2 g_k - a_1 b_3 e_2 g_k - a_2 b_1 e_3 g_k + a_1 b_2 e_3 g_k
\end{aligned}$$

$$\begin{aligned}
c_{p1} = & a_k b_3 d_2 f_1 + a_3 b_k d_2 f_1 + a_k b_2 d_3 f_1 - a_2 b_k d_3 f_1 - a_3 b_2 d_k f_1 + a_2 b_3 d_k f_1 + \\
& a_k c_3 e_2 f_1 - a_3 c_k e_2 f_1 - a_k c_2 e_3 f_1 + a_2 c_k e_3 f_1 + a_3 c_2 e_k f_1 - a_2 c_3 e_k f_1 + \\
& a_k b_3 d_1 f_2 - a_3 b_k d_1 f_2 - a_k b_1 d_3 f_2 + a_1 b_k d_3 f_2 + a_3 b_1 d_k f_2 - a_1 b_3 d_k f_2 - \\
& a_k c_3 e_1 f_2 + a_3 c_k e_1 f_2 + a_k c_1 e_3 f_2 - a_1 c_k e_3 f_2 - a_3 c_1 e_k f_2 + a_1 c_3 e_k f_2 - \\
& a_k b_2 d_1 f_3 + a_2 b_k d_1 f_3 + a_k b_1 d_2 f_3 - a_1 b_k d_2 f_3 - a_2 b_1 d_k f_3 + a_1 b_2 d_k f_3 + \\
& a_k c_2 e_1 f_3 - a_2 c_k e_1 f_3 - a_k c_1 e_2 f_3 + a_1 c_k e_2 f_3 + a_2 c_1 e_k f_3 - a_1 c_2 e_k f_3 + \\
& a_3 b_2 d_1 f_k - a_2 b_3 d_1 f_k - a_3 b_1 d_2 f_k + a_1 b_3 d_2 f_k + a_2 b_1 d_3 f_k - a_1 b_2 d_3 f_k - \\
& a_3 c_2 e_1 f_k + a_2 c_3 e_1 f_k + a_3 c_1 e_2 f_k - a_1 c_3 e_2 f_k - a_2 c_1 e_3 f_k + a_1 c_2 e_3 f_k + \\
& a_k c_3 d_2 g_1 - a_3 c_k d_2 g_1 - a_k c_2 d_3 g_1 + a_2 c_k d_3 g_1 + a_3 c_2 d_k g_1 - a_2 c_3 d_k g_1 + \\
& a_k b_3 e_2 g_1 - a_3 b_k e_2 g_1 - a_k b_2 e_3 g_1 + a_2 b_k e_3 g_1 + a_3 b_2 e_k g_1 - a_2 b_3 e_k g_1 - \\
& a_k c_3 d_1 g_2 + a_3 c_k d_1 g_2 + a_k c_1 d_3 g_2 - a_1 c_k d_3 g_2 - a_3 c_1 d_k g_2 + a_1 c_3 d_k g_2 - \\
& a_k b_3 e_1 g_2 + a_3 b_k e_1 g_2 + a_k b_1 e_3 g_2 - a_1 b_k e_3 g_2 - a_3 b_1 e_k g_2 + a_1 b_3 e_k g_2 + \\
& a_k c_2 d_1 g_3 - a_2 c_k d_1 g_3 - a_k c_1 d_2 g_3 + a_1 c_k d_2 g_3 + a_2 c_1 d_k g_3 - a_1 c_2 d_k g_3 + \\
& a_k b_2 e_1 g_3 - a_2 b_k e_1 g_3 - a_k b_1 e_2 g_3 + a_1 b_k e_2 g_3 + a_2 b_1 e_k g_3 - a_1 b_2 e_k g_3 - \\
& a_3 c_2 d_1 g_k + a_2 c_3 d_1 g_k + a_3 c_1 d_2 g_k - a_1 c_3 d_2 g_k - a_2 c_1 d_3 g_k + a_1 c_2 d_3 g_k - \\
& a_3 b_2 e_1 g_k + a_2 b_3 e_1 g_k + a_3 b_1 e_2 g_k - a_1 b_3 e_2 g_k - a_2 b_1 e_3 g_k + a_1 b_2 e_3 g_k
\end{aligned}$$

$$\begin{aligned}
d_{p1} = & a_k b_3 d_2 f_1 + a_3 b_k d_2 f_1 + a_k b_2 d_3 f_1 - a_2 b_k d_3 f_1 - a_3 b_2 d_k f_1 + a_2 b_3 d_k f_1 - \\
& a_k c_3 e_2 f_1 + a_3 c_k e_2 f_1 + a_k c_2 e_3 f_1 - a_2 c_k e_3 f_1 - a_3 c_2 e_k f_1 + a_2 c_3 e_k f_1 + \\
& a_k b_3 d_1 f_2 - a_3 b_k d_1 f_2 - a_k b_1 d_3 f_2 + a_1 b_k d_3 f_2 + a_3 b_1 d_k f_2 - a_1 b_3 d_k f_2 + \\
& a_k c_3 e_1 f_2 - a_3 c_k e_1 f_2 - a_k c_1 e_3 f_2 + a_1 c_k e_3 f_2 + a_3 c_1 e_k f_2 - a_1 c_3 e_k f_2 - \\
& a_k b_2 d_1 f_3 + a_2 b_k d_1 f_3 + a_k b_1 d_2 f_3 - a_1 b_k d_2 f_3 - a_2 b_1 d_k f_3 + a_1 b_2 d_k f_3 - \\
& a_k c_2 e_1 f_3 + a_2 c_k e_1 f_3 + a_k c_1 e_2 f_3 - a_1 c_k e_2 f_3 - a_2 c_1 e_k f_3 + a_1 c_2 e_k f_3 + \\
& a_3 b_2 d_1 f_k - a_2 b_3 d_1 f_k - a_3 b_1 d_2 f_k + a_1 b_3 d_2 f_k + a_2 b_1 d_3 f_k - a_1 b_2 d_3 f_k + \\
& a_3 c_2 e_1 f_k - a_2 c_3 e_1 f_k - a_3 c_1 e_2 f_k + a_1 c_3 e_2 f_k + a_2 c_1 e_3 f_k - a_1 c_2 e_3 f_k - \\
& a_k c_3 d_2 g_1 + a_3 c_k d_2 g_1 + a_k c_2 d_3 g_1 - a_2 c_k d_3 g_1 - a_3 c_2 d_k g_1 + a_2 c_3 d_k g_1 + \\
& a_k b_3 e_2 g_1 - a_3 b_k e_2 g_1 - a_k b_2 e_3 g_1 + a_2 b_k e_3 g_1 + a_3 b_2 e_k g_1 - a_2 b_3 e_k g_1 + \\
& a_k c_3 d_1 g_2 - a_3 c_k d_1 g_2 - a_k c_1 d_3 g_2 + a_1 c_k d_3 g_2 + a_3 c_1 d_k g_2 - a_1 c_3 d_k g_2 - \\
& a_k b_3 e_1 g_2 + a_3 b_k e_1 g_2 + a_k b_1 e_3 g_2 - a_1 b_k e_3 g_2 - a_3 b_1 e_k g_2 + a_1 b_3 e_k g_2 - \\
& a_k c_2 d_1 g_3 + a_2 c_k d_1 g_3 + a_k c_1 d_2 g_3 - a_1 c_k d_2 g_3 - a_2 c_1 d_k g_3 + a_1 c_2 d_k g_3 + \\
& a_k b_2 e_1 g_3 - a_2 b_k e_1 g_3 - a_k b_1 e_2 g_3 + a_1 b_k e_2 g_3 + a_2 b_1 e_k g_3 - a_1 b_2 e_k g_3 - \\
& a_3 c_2 d_1 g_k - a_2 c_3 d_1 g_k - a_3 c_1 d_2 g_k + a_1 c_3 d_2 g_k + a_2 c_1 d_3 g_k - a_1 c_2 d_3 g_k - \\
& a_3 b_2 e_1 g_k + a_2 b_3 e_1 g_k + a_3 b_1 e_2 g_k - a_1 b_3 e_2 g_k - a_2 b_1 e_3 g_k + a_1 b_2 e_3 g_k
\end{aligned}$$

$$\begin{aligned}
& a_k b_2 d_1 g_3 - a_2 b_k d_1 g_3 - a_k b_1 d_2 g_3 + a_1 b_k d_2 g_3 + a_2 b_1 d_k g_3 - a_1 b_2 d_k g_3 - \\
& a_k c_2 e_1 g_3 + a_2 c_k e_1 g_3 + a_k c_1 e_2 g_3 - a_1 c_k e_2 g_3 - a_2 c_1 e_k g_3 + a_1 c_2 e_k g_3 - \\
& a_3 b_2 d_1 g_k + a_2 b_3 d_1 g_k + a_3 b_1 d_2 g_k - a_1 b_3 d_2 g_k - a_2 b_1 d_3 g_k + a_1 b_2 d_3 g_k - \\
& a_3 c_2 e_1 g_k - a_2 c_3 e_1 g_k - a_3 c_1 e_2 g_k + a_1 c_3 e_2 g_k + a_2 c_1 e_3 g_k - a_1 c_2 e_3 g_k
\end{aligned}$$

$$\begin{aligned}
f_{p1} = & a_k c_2 d_1 f_1 - a_3 c_k d_2 f_1 - a_k c_2 d_3 f_1 + a_2 c_k d_3 f_1 + a_3 c_2 d_k f_1 - a_2 c_3 d_k f_1 + \\
& a_k b_3 e_2 f_1 - a_3 b_k e_2 f_1 - a_k b_2 e_3 f_1 + a_2 b_k e_3 f_1 + a_3 b_2 e_k f_1 - a_2 b_3 e_k f_1 - \\
& a_k c_3 d_1 f_2 + a_3 c_k d_1 f_2 + a_k c_1 d_3 f_2 - a_1 c_k d_3 f_2 - a_3 c_1 d_k f_2 + a_1 c_3 d_k f_2 - \\
& a_k b_3 e_1 f_2 + a_3 b_k e_1 f_2 + a_k b_1 e_3 f_2 - a_1 b_k e_3 f_2 - a_3 b_1 e_k f_2 + a_1 b_3 e_k f_2 + \\
& a_k c_2 d_1 f_3 - a_2 c_k d_1 f_3 - a_k c_1 d_2 f_3 + a_1 c_k d_2 f_3 + a_2 c_1 d_k f_3 - a_1 c_2 d_k f_3 + \\
& a_k b_2 e_1 f_3 - a_2 b_k e_1 f_3 - a_k b_1 e_2 f_3 + a_1 b_k e_2 f_3 + a_2 b_1 e_k f_3 - a_1 b_2 e_k f_3 - \\
& a_3 c_2 d_1 f_k + a_2 c_3 d_1 f_k + a_3 c_1 d_2 f_k - a_1 c_3 d_2 f_k - a_2 c_1 d_3 f_k + a_1 c_2 d_3 f_k - \\
& a_3 b_2 e_1 f_k + a_2 b_3 e_1 f_k + a_3 b_1 e_2 f_k - a_1 b_3 e_2 f_k - a_2 b_1 e_3 f_k + a_1 b_2 e_3 f_k + \\
& a_k b_3 d_2 g_1 - a_3 b_k d_2 g_1 - a_k b_2 d_3 g_1 + a_2 b_k d_3 g_1 + a_3 b_2 d_k g_1 - a_2 b_3 d_k g_1 - \\
& a_k c_3 e_2 g_1 + a_3 c_k e_2 g_1 + a_k c_2 e_3 g_1 - a_2 c_k e_3 g_1 - a_3 c_2 e_k g_1 + a_2 c_3 e_k g_1 - \\
& a_k b_3 d_1 g_2 + a_3 b_k d_1 g_2 + a_k b_1 d_3 g_2 - a_1 b_k d_3 g_2 - a_3 b_1 d_k g_2 + a_1 b_3 d_k g_2 + \\
& a_k c_3 e_1 g_2 - a_3 c_k e_1 g_2 - a_k c_1 e_3 g_2 + a_1 c_k e_3 g_2 + a_3 c_1 e_k g_2 - a_1 c_3 e_k g_2 + \\
& a_k b_2 d_1 g_3 - a_2 b_k d_1 g_3 - a_k b_1 d_2 g_3 + a_1 b_k d_2 g_3 + a_2 b_1 d_k g_3 - a_1 b_2 d_k g_3 - \\
& a_k c_2 e_1 g_3 + a_2 c_k e_1 g_3 + a_k c_1 e_2 g_3 - a_1 c_k e_2 g_3 - a_2 c_1 e_k g_3 + a_1 c_2 e_k g_3 - \\
& a_3 b_2 d_1 g_k + a_2 b_3 d_1 g_k + a_3 b_1 d_2 g_k - a_1 b_3 d_2 g_k - a_2 b_1 d_3 g_k + a_1 b_2 d_3 g_k - \\
& a_3 c_2 e_1 g_k - a_2 c_3 e_1 g_k - a_3 c_1 e_2 g_k + a_1 c_3 e_2 g_k + a_2 c_1 e_3 g_k - a_1 c_2 e_3 g_k
\end{aligned}$$

$$\begin{aligned}
g_{p1} = & a_k c_1 d_2 f_1 - a_3 c_k d_2 f_1 - a_k c_2 d_3 f_1 + a_2 c_k d_3 f_1 + a_3 c_2 d_k f_1 - a_2 c_3 d_k f_1 - \\
& a_k b_3 e_2 f_1 + a_3 b_k e_2 f_1 + a_k b_2 e_3 f_1 - a_2 b_k e_3 f_1 - a_3 b_2 e_k f_1 + a_2 b_3 e_k f_1 - \\
& a_k c_3 d_1 f_2 + a_3 c_k d_1 f_2 + a_k c_1 d_3 f_2 - a_1 c_k d_3 f_2 - a_3 c_1 d_k f_2 + a_1 c_3 d_k f_2 + \\
& a_k b_3 e_1 f_2 - a_3 b_k e_1 f_2 - a_k b_1 e_3 f_2 + a_1 b_k e_3 f_2 + a_3 b_1 e_k f_2 - a_1 b_3 e_k f_2 + \\
& a_k c_2 d_1 f_3 - a_2 c_k d_1 f_3 - a_k c_1 d_2 f_3 + a_1 c_k d_2 f_3 + a_2 c_1 d_k f_3 - a_1 c_2 d_k f_3 - \\
& a_k b_2 e_1 f_3 + a_2 b_k e_1 f_3 + a_k b_1 e_2 f_3 - a_1 b_k e_2 f_3 - a_2 b_1 e_k f_3 + a_1 b_2 e_k f_3 - \\
& a_3 c_2 d_1 f_k + a_2 c_3 d_1 f_k + a_3 c_1 d_2 f_k - a_1 c_3 d_2 f_k - a_2 c_1 d_3 f_k + a_1 c_2 d_3 f_k + \\
& a_3 b_2 e_1 f_k - a_2 b_3 e_1 f_k - a_3 b_1 e_2 f_k + a_1 b_3 e_2 f_k + a_2 b_1 e_3 f_k - a_1 b_2 e_3 f_k - \\
& a_k b_3 d_2 g_1 + a_3 b_k d_2 g_1 + a_k b_2 d_3 g_1 - a_2 b_k d_3 g_1 - a_3 b_2 d_k g_1 + a_2 b_3 d_k g_1 - \\
& a_k c_3 e_2 g_1 + a_3 c_k e_2 g_1 + a_k c_2 e_3 g_1 - a_2 c_k e_3 g_1 - a_3 c_2 e_k g_1 + a_2 c_3 e_k g_1 + \\
& a_k b_3 d_1 g_2 - a_3 b_k d_1 g_2 - a_k b_1 d_3 g_2 + a_1 b_k d_3 g_2 + a_3 b_1 d_k g_2 - a_1 b_3 d_k g_2 + \\
& a_k c_3 e_1 g_2 - a_3 c_k e_1 g_2 - a_k c_1 e_3 g_2 + a_1 c_k e_3 g_2 + a_3 c_1 e_k g_2 - a_1 c_3 e_k g_2 - \\
& a_k b_2 d_1 g_3 + a_2 b_k d_1 g_3 + a_k b_1 d_2 g_3 - a_1 b_k d_2 g_3 - a_2 b_1 d_k g_3 + a_1 b_2 d_k g_3 - \\
& a_k c_2 e_1 g_3 + a_2 c_k e_1 g_3 + a_k c_1 e_2 g_3 - a_1 c_k e_2 g_3 - a_2 c_1 e_k g_3 + a_1 c_2 e_k g_3 - \\
& a_3 b_2 d_1 g_k - a_2 b_3 d_1 g_k - a_3 b_1 d_2 g_k + a_1 b_3 d_2 g_k + a_2 b_1 d_3 g_k - a_1 b_2 d_3 g_k - \\
& a_3 c_2 e_1 g_k - a_2 c_3 e_1 g_k - a_3 c_1 e_2 g_k + a_1 c_3 e_2 g_k + a_2 c_1 e_3 g_k - a_1 c_2 e_3 g_k
\end{aligned}$$

(3.1.9) ifadesinden (v) yok edilirse (t) ye bağılı (3.1.15) ‘deki 12. dereceden polinom bulunur.

$$\begin{aligned}
& w_0 - w_1 t + (w_2 - 2w_0) t^2 + (2w_1 - w_3) t^3 + (3w_0 - 2w_2 + w_4) t^4 + (-3w_1 + 2w_3 - w_5) t^5 + \\
& (-4w_0 + 3w_2 - 2w_4 + w_6) t^6 + (3w_1 - 2w_3 + w_5) t^7 + (3w_0 - 2w_2 + w_4) t^8 + \\
& (-2w_1 + w_3) t^9 + (w_2 - 2w_0) t^{10} + w_1 t^{11} + w_0 t^{12} = 0
\end{aligned} \tag{3.1.15}$$

Yukarıdaki polinomun katsayıları aşağıda verilmiştir.

$$\begin{aligned}
w_0 = & a_{\varphi 1}^2 b_{\varphi 1} - 2 a_{\varphi 1} a_{\varphi 2} b_{\varphi 1} b_{\varphi 2} + a_{\varphi 1}^2 b_{\varphi 2}^2 + 2 a_{\varphi 2}^2 b_{\varphi 1} c_{\varphi 1} - 2 a_{\varphi 1} a_{\varphi 2} b_{\varphi 2} c_{\varphi 1} + \\
& a_{\varphi 2}^2 c_{\varphi 1}^2 - 2 a_{\varphi 1} a_{\varphi 2} b_{\varphi 1} c_{\varphi 2} + 2 a_{\varphi 1}^2 b_{\varphi 2} c_{\varphi 2} - 2 a_{\varphi 1} a_{\varphi 2} c_{\varphi 1} c_{\varphi 2} + a_{\varphi 1}^2 c_{\varphi 2}^2 - \\
& 2 a_{\varphi 2} b_{\varphi 1} b_{\varphi 2} d_{\varphi 1} + 2 a_{\varphi 1} b_{\varphi 2}^2 d_{\varphi 1} - 2 a_{\varphi 2} b_{\varphi 2} c_{\varphi 1} d_{\varphi 1} - 2 a_{\varphi 2} b_{\varphi 1} c_{\varphi 2} d_{\varphi 1} + \\
& 4 a_{\varphi 1} b_{\varphi 2} c_{\varphi 2} d_{\varphi 1} - 2 a_{\varphi 2} c_{\varphi 1} c_{\varphi 2} d_{\varphi 1} + 2 a_{\varphi 1} c_{\varphi 2}^2 d_{\varphi 1} + b_{\varphi 2}^2 d_{\varphi 1}^2 + 2 b_{\varphi 2} c_{\varphi 2} d_{\varphi 1}^2 + \\
& c_{\varphi 2}^2 d_{\varphi 1}^2 + 2 a_{\varphi 2} b_{\varphi 1}^2 d_{\varphi 2} - 2 a_{\varphi 1} b_{\varphi 1} b_{\varphi 2} d_{\varphi 2} + 4 a_{\varphi 2} b_{\varphi 1} c_{\varphi 1} d_{\varphi 2} - 2 a_{\varphi 1} b_{\varphi 2} c_{\varphi 1} d_{\varphi 2} + \\
& 2 a_{\varphi 2} c_{\varphi 1}^2 d_{\varphi 2} - 2 a_{\varphi 1} b_{\varphi 1} c_{\varphi 2} d_{\varphi 2} - 2 a_{\varphi 1} c_{\varphi 1} c_{\varphi 2} d_{\varphi 2} - 2 b_{\varphi 1} b_{\varphi 2} d_{\varphi 1} d_{\varphi 2} - \\
& 2 b_{\varphi 2} c_{\varphi 1} d_{\varphi 1} d_{\varphi 2} - 2 b_{\varphi 1} c_{\varphi 2} d_{\varphi 1} d_{\varphi 2} - 2 c_{\varphi 1} c_{\varphi 2} d_{\varphi 1} d_{\varphi 2} + b_{\varphi 1}^2 d_{\varphi 2}^2 + 2 b_{\varphi 1} c_{\varphi 1} d_{\varphi 2}^2 + \\
& c_{\varphi 1}^2 d_{\varphi 2}^2 + a_{\varphi 2}^2 e_{\varphi 1}^2 - b_{\varphi 2}^2 e_{\varphi 1}^2 - 2 b_{\varphi 2} c_{\varphi 2} e_{\varphi 1}^2 - c_{\varphi 2}^2 e_{\varphi 1}^2 + 2 a_{\varphi 2} d_{\varphi 2} e_{\varphi 1}^2 + d_{\varphi 2}^2 e_{\varphi 1}^2 - \\
& 2 a_{\varphi 1} a_{\varphi 2} e_{\varphi 1} e_{\varphi 2} + 2 b_{\varphi 1} b_{\varphi 2} e_{\varphi 1} e_{\varphi 2} + 2 b_{\varphi 2} c_{\varphi 1} e_{\varphi 1} e_{\varphi 2} + 2 b_{\varphi 1} c_{\varphi 2} e_{\varphi 1} e_{\varphi 2} + \\
& 2 c_{\varphi 1} c_{\varphi 2} e_{\varphi 1} e_{\varphi 2} - 2 a_{\varphi 2} d_{\varphi 1} e_{\varphi 1} e_{\varphi 2} - 2 a_{\varphi 1} d_{\varphi 2} e_{\varphi 1} e_{\varphi 2} - 2 d_{\varphi 1} d_{\varphi 2} e_{\varphi 1} e_{\varphi 2} + \\
& a_{\varphi 1}^2 e_{\varphi 2}^2 - b_{\varphi 1}^2 e_{\varphi 2}^2 - 2 b_{\varphi 1} c_{\varphi 1} e_{\varphi 2}^2 - c_{\varphi 1}^2 e_{\varphi 2}^2 + 2 a_{\varphi 1} d_{\varphi 1} e_{\varphi 2}^2 + d_{\varphi 1}^2 e_{\varphi 2}^2 + \\
& 2 a_{\varphi 2}^2 e_{\varphi 1} f_{\varphi 1} - 2 b_{\varphi 2}^2 e_{\varphi 1} f_{\varphi 1} - 4 b_{\varphi 2} c_{\varphi 2} e_{\varphi 1} f_{\varphi 1} - 2 c_{\varphi 2}^2 e_{\varphi 1} f_{\varphi 1} + 4 a_{\varphi 2} d_{\varphi 2} e_{\varphi 1} f_{\varphi 1} + \\
& 2 d_{\varphi 2}^2 e_{\varphi 1} f_{\varphi 1} - 2 a_{\varphi 1} a_{\varphi 2} e_{\varphi 2} f_{\varphi 1} + 2 b_{\varphi 1} b_{\varphi 2} e_{\varphi 2} f_{\varphi 1} + 2 b_{\varphi 2} c_{\varphi 1} e_{\varphi 2} f_{\varphi 1} + \\
& 2 b_{\varphi 1} c_{\varphi 2} e_{\varphi 2} f_{\varphi 1} + 2 c_{\varphi 1} c_{\varphi 2} e_{\varphi 2} f_{\varphi 1} - 2 a_{\varphi 2} d_{\varphi 1} e_{\varphi 2} f_{\varphi 1} - 2 a_{\varphi 1} d_{\varphi 2} e_{\varphi 2} f_{\varphi 1} - \\
& 2 d_{\varphi 1} d_{\varphi 2} e_{\varphi 2} f_{\varphi 1} + a_{\varphi 2}^2 f_{\varphi 1}^2 - b_{\varphi 2}^2 f_{\varphi 1}^2 - 2 b_{\varphi 2} c_{\varphi 2} f_{\varphi 1}^2 - c_{\varphi 2}^2 f_{\varphi 1}^2 + 2 a_{\varphi 2} d_{\varphi 2} f_{\varphi 1}^2 + \\
& d_{\varphi 2}^2 f_{\varphi 1}^2 - 2 a_{\varphi 1} a_{\varphi 2} e_{\varphi 1} f_{\varphi 2} + 2 b_{\varphi 1} b_{\varphi 2} e_{\varphi 1} f_{\varphi 2} + 2 b_{\varphi 2} c_{\varphi 1} e_{\varphi 1} f_{\varphi 2} + \\
& 2 b_{\varphi 1} c_{\varphi 2} e_{\varphi 1} f_{\varphi 2} + 2 c_{\varphi 1} c_{\varphi 2} e_{\varphi 1} f_{\varphi 2} - 2 a_{\varphi 2} d_{\varphi 1} e_{\varphi 1} f_{\varphi 2} - 2 a_{\varphi 1} d_{\varphi 2} e_{\varphi 1} f_{\varphi 2} - \\
& 2 d_{\varphi 1} d_{\varphi 2} e_{\varphi 1} f_{\varphi 2} + 2 a_{\varphi 1}^2 e_{\varphi 2} f_{\varphi 2} - 2 b_{\varphi 1}^2 e_{\varphi 2} f_{\varphi 2} - 4 b_{\varphi 1} c_{\varphi 1} e_{\varphi 2} f_{\varphi 2} - \\
& 2 c_{\varphi 1}^2 e_{\varphi 2} f_{\varphi 2} + 4 a_{\varphi 1} d_{\varphi 1} e_{\varphi 2} f_{\varphi 2} + 2 d_{\varphi 1}^2 e_{\varphi 2} f_{\varphi 2} - 2 a_{\varphi 1} a_{\varphi 2} f_{\varphi 1} f_{\varphi 2} + \\
& 2 b_{\varphi 1} b_{\varphi 2} f_{\varphi 1} f_{\varphi 2} + 2 b_{\varphi 2} c_{\varphi 1} f_{\varphi 1} f_{\varphi 2} + 2 b_{\varphi 1} c_{\varphi 2} f_{\varphi 1} f_{\varphi 2} + 2 c_{\varphi 1} c_{\varphi 2} f_{\varphi 1} f_{\varphi 2} - \\
& 2 a_{\varphi 2} d_{\varphi 1} f_{\varphi 1} f_{\varphi 2} - 2 a_{\varphi 1} d_{\varphi 2} f_{\varphi 1} f_{\varphi 2} - 2 d_{\varphi 1} d_{\varphi 2} f_{\varphi 1} f_{\varphi 2} + a_{\varphi 1}^2 f_{\varphi 2}^2 - b_{\varphi 1}^2 f_{\varphi 2}^2 - \\
& 2 b_{\varphi 1} c_{\varphi 1} f_{\varphi 2}^2 - c_{\varphi 1}^2 f_{\varphi 2}^2 + 2 a_{\varphi 1} d_{\varphi 1} f_{\varphi 2}^2 + d_{\varphi 1}^2 f_{\varphi 2}^2
\end{aligned} \tag{3.1.16}$$

$w_1 =$

$$\begin{aligned}
& - 512 \quad a_{p2}^2 e_{p1} + b_{p2}^2 c_{p1} e_{p1} + a_{p1} a_{p2} c_{p2} e_{p1} - b_{p1} b_{p2} c_{p2} e_{p1} + b_{p2} c_{p1} c_{p2} e_{p1} - \\
& \quad b_{p1} c_{p2}^2 e_{p1} + a_{p2} c_{p2} d_{p1} e_{p1} - 2 a_{p2} c_{p1} d_{p2} e_{p1} + a_{p1} c_{p2} d_{p2} e_{p1} + c_{p2} d_{p1} d_{p2} e_{p1} - \\
& \quad c_{p1} d_{p2}^2 e_{p1} + a_{p1} a_{p2} c_{p1} e_{p2} - b_{p1} b_{p2} c_{p1} e_{p2} - b_{p2} c_{p1}^2 e_{p2} - a_{p1}^2 c_{p2} e_{p2} + \\
& \quad b_{p1}^2 c_{p2} e_{p2} + b_{p1} c_{p1} c_{p2} e_{p2} + a_{p2} c_{p1} d_{p1} e_{p2} - 2 a_{p1} c_{p2} d_{p1} e_{p2} - c_{p2} d_{p1}^2 e_{p2} + \\
& \quad a_{p1} c_{p1} d_{p2} e_{p2} + c_{p1} d_{p1} d_{p2} e_{p2} + a_{p2}^2 b_{p1} f_{p1} - a_{p1} a_{p2} b_{p2} f_{p1} + b_{p2}^2 c_{p1} f_{p1} - \\
& \quad b_{p1} b_{p2} c_{p2} f_{p1} + b_{p2} c_{p1} c_{p2} f_{p1} - b_{p1} c_{p2}^2 f_{p1} - a_{p2} b_{p2} d_{p1} f_{p1} + 2 a_{p2} b_{p1} d_{p2} f_{p1} - \\
& \quad a_{p1} b_{p2} d_{p2} f_{p1} - b_{p2} d_{p1} d_{p2} f_{p1} + b_{p1} d_{p2}^2 f_{p1} + b_{p2} e_{p1} e_{p2} f_{p1} + c_{p2} e_{p1} e_{p2} f_{p1} - \\
& \quad b_{p1} e_{p2}^2 f_{p1} - c_{p1} e_{p2}^2 f_{p1} + b_{p2} e_{p2} f_{p1}^2 + c_{p2} e_{p2} f_{p1}^2 - a_{p1} a_{p2} b_{p1} f_{p2} + \\
& \quad a_{p1}^2 b_{p2} f_{p2} - b_{p1} b_{p2} c_{p1} f_{p2} - b_{p2} c_{p1}^2 f_{p2} + b_{p1}^2 c_{p2} f_{p2} + b_{p1} c_{p1} c_{p2} f_{p2} - \\
& \quad a_{p2} b_{p1} d_{p1} f_{p2} + 2 a_{p1} b_{p2} d_{p1} f_{p2} + b_{p2} d_{p1}^2 f_{p2} - a_{p1} b_{p1} d_{p2} f_{p2} - b_{p1} d_{p1} d_{p2} f_{p2} - \\
& \quad b_{p2} e_{p1}^2 f_{p2} - c_{p2} e_{p1}^2 f_{p2} + b_{p1} e_{p1} e_{p2} f_{p2} + c_{p1} e_{p1} e_{p2} f_{p2} - b_{p2} e_{p1} f_{p1} f_{p2} - \\
& \quad c_{p2} e_{p1} f_{p1} f_{p2} - b_{p1} e_{p2} f_{p1} f_{p2} - c_{p1} e_{p2} f_{p1} f_{p2} + b_{p1} e_{p1} f_{p2}^2 + c_{p1} e_{p1} f_{p2}^2 + \\
& \quad a_{p2} b_{p1} b_{p2} g_{p1} - a_{p1} b_{p2}^2 g_{p1} + a_{p2} b_{p2} c_{p1} g_{p1} + a_{p2} b_{p1} c_{p2} g_{p1} - 2 a_{p1} b_{p2} c_{p2} g_{p1} + \\
& \quad a_{p2} c_{p1} c_{p2} g_{p1} - a_{p1} c_{p2}^2 g_{p1} - b_{p2}^2 d_{p1} g_{p1} - 2 b_{p2} c_{p2} d_{p1} g_{p1} - c_{p2}^2 d_{p1} g_{p1} + \\
& \quad b_{p1} b_{p2} d_{p2} g_{p1} + b_{p2} c_{p1} d_{p2} g_{p1} + b_{p1} c_{p2} d_{p2} g_{p1} + c_{p1} c_{p2} d_{p2} g_{p1} + \\
& \quad a_{p2} e_{p1} e_{p2} g_{p1} + d_{p2} e_{p1} e_{p2} g_{p1} - a_{p1} e_{p2}^2 g_{p1} - d_{p1} e_{p2}^2 g_{p1} + a_{p2} e_{p2} f_{p1} g_{p1} + \\
& \quad d_{p2} e_{p2} f_{p1} g_{p1} + a_{p2} e_{p1} f_{p2} g_{p1} + d_{p2} e_{p1} f_{p2} g_{p1} - 2 a_{p1} e_{p2} f_{p2} g_{p1} - \\
& \quad 2 d_{p1} e_{p2} f_{p2} g_{p1} + a_{p2} f_{p1} f_{p2} g_{p1} + d_{p2} f_{p1} f_{p2} g_{p1} - a_{p1} f_{p2}^2 g_{p1} - d_{p1} f_{p2}^2 g_{p1} - \\
& \quad a_{p2} b_{p1}^2 g_{p2} + a_{p1} b_{p1} b_{p2} g_{p2} - 2 a_{p2} b_{p1} c_{p1} g_{p2} + a_{p1} b_{p2} c_{p1} g_{p2} - a_{p2} c_{p1}^2 g_{p2} + \\
& \quad a_{p1} b_{p1} c_{p2} g_{p2} + a_{p1} c_{p1} c_{p2} g_{p2} + b_{p1} b_{p2} d_{p1} g_{p2} + b_{p2} c_{p1} d_{p1} g_{p2} + \\
& \quad b_{p1} c_{p2} d_{p1} g_{p2} + c_{p1} c_{p2} d_{p1} g_{p2} - b_{p1}^2 d_{p2} g_{p2} - 2 b_{p1} c_{p1} d_{p2} g_{p2} - c_{p1}^2 d_{p2} g_{p2} - \\
& \quad a_{p2} e_{p1}^2 g_{p2} - d_{p2} e_{p1}^2 g_{p2} + a_{p1} e_{p1} e_{p2} g_{p2} + d_{p1} e_{p1} e_{p2} g_{p2} - 2 a_{p2} e_{p1} f_{p1} g_{p2} - \\
& \quad 2 d_{p2} e_{p1} f_{p1} g_{p2} + a_{p1} e_{p2} f_{p1} g_{p2} + d_{p1} e_{p2} f_{p1} g_{p2} - a_{p2} f_{p1}^2 g_{p2} - g_{p2} f_{p1}^2 g_{p2} + \\
& \quad a_{p1} e_{p1} f_{p2} g_{p2} + d_{p1} e_{p1} f_{p2} g_{p2} + a_{p1} f_{p1} f_{p2} g_{p2} + d_{p1} f_{p1} f_{p2} g_{p2}
\end{aligned}$$

(3.1.17)

$w_2 =$

$$\begin{aligned}
& 512 a_{p2}^2 b_{p1} - 2 a_{p1} a_{p2} b_{p1} b_{p2} + a_{p1}^2 b_{p2}^2 + a_{p2}^2 c_{p1}^2 - 2 b_{p2}^2 c_{p1}^2 - 2 a_{p1} a_{p2} c_{p1} c_{p2} + \\
& 4 b_{p1} b_{p2} c_{p1} c_{p2} + a_{p1}^2 c_{p2}^2 - 2 b_{p1}^2 c_{p2}^2 + 2 a_{p2} b_{p2} c_{p1} d_{p1} + 2 a_{p2} b_{p1} c_{p2} d_{p1} - \\
& 4 a_{p1} b_{p2} c_{p2} d_{p1} - b_{p2}^2 d_{p1}^2 - 4 b_{p2} c_{p2} d_{p1}^2 - c_{p2}^2 d_{p1}^2 - 4 a_{p2} b_{p1} c_{p1} d_{p2} + \\
& 2 a_{p1} b_{p2} c_{p1} d_{p2} + 2 a_{p1} b_{p1} c_{p2} d_{p2} + 2 b_{p1} b_{p2} d_{p1} d_{p2} + 4 b_{p2} c_{p1} d_{p1} d_{p2} + \\
& 4 b_{p1} c_{p2} d_{p1} d_{p2} + 2 c_{p1} c_{p2} d_{p1} d_{p2} - b_{p1}^2 d_{p2}^2 - 4 b_{p1} c_{p1} d_{p2}^2 - c_{p1}^2 d_{p2}^2 + \\
& a_{p2}^2 e_{p1}^2 - b_{p2}^2 e_{p1}^2 + c_{p2}^2 e_{p1}^2 - d_{p2}^2 e_{p1}^2 - 2 a_{p1} a_{p2} e_{p1} e_{p2} + 2 b_{p1} b_{p2} e_{p1} e_{p2} - \\
& 2 c_{p1} c_{p2} e_{p1} e_{p2} + 2 d_{p1} d_{p2} e_{p1} e_{p2} + a_{p1}^2 e_{p2}^2 - b_{p1}^2 e_{p2}^2 + c_{p1}^2 e_{p2}^2 - d_{p1}^2 e_{p2}^2 - \\
& 4 a_{p2} d_{p2} e_{p1} f_{p1} - 4 d_{p2}^2 e_{p1} f_{p1} - 6 b_{p2} c_{p1} e_{p2} f_{p1} + 6 b_{p1} c_{p2} e_{p2} f_{p1} + \\
& 2 a_{p2} d_{p1} e_{p2} f_{p1} + 2 a_{p1} d_{p2} e_{p2} f_{p1} + 4 d_{p1} d_{p2} e_{p2} f_{p1} + a_{p2}^2 f_{p1}^2 + b_{p2}^2 f_{p1}^2 - \\
& c_{p2}^2 f_{p1}^2 - d_{p2}^2 f_{p1}^2 - 2 e_{p2}^2 f_{p1}^2 + 6 b_{p2} c_{p1} e_{p1} f_{p2} - 6 b_{p1} c_{p2} e_{p1} f_{p2} + \\
& 2 a_{p2} d_{p1} e_{p1} f_{p2} + 2 a_{p1} d_{p2} e_{p1} f_{p2} + 4 d_{p1} d_{p2} e_{p1} f_{p2} - 4 a_{p1} d_{p1} e_{p2} f_{p2} - \\
& 4 d_{p1}^2 e_{p2} f_{p2} - 2 a_{p1} a_{p2} f_{p1} f_{p2} - 2 b_{p1} b_{p2} f_{p1} f_{p2} + 2 c_{p1} c_{p2} f_{p1} f_{p2} + \\
& 2 d_{p1} d_{p2} f_{p1} f_{p2} + 4 e_{p1} e_{p2} f_{p1} f_{p2} + a_{p1}^2 f_{p2}^2 + b_{p1}^2 f_{p2}^2 - c_{p1}^2 f_{p2}^2 - d_{p1}^2 f_{p2}^2 - \\
& 2 e_{p1}^2 f_{p2}^2 - 4 a_{p2} c_{p2} e_{p1} g_{p1} - 4 c_{p2} d_{p2} e_{p1} g_{p1} - 4 a_{p2} c_{p1} e_{p2} g_{p1} + \\
& 8 a_{p1} c_{p2} e_{p2} g_{p1} + 8 c_{p2} d_{p1} e_{p2} g_{p1} - 4 c_{p1} d_{p2} e_{p2} g_{p1} + 4 a_{p2} b_{p2} f_{p1} g_{p1} + \\
& 4 b_{p2} d_{p2} f_{p1} g_{p1} + 4 a_{p2} b_{p1} f_{p2} g_{p1} - 8 a_{p1} b_{p2} f_{p2} g_{p1} - 8 b_{p2} d_{p1} f_{p2} g_{p1} + \\
& 4 b_{p1} d_{p2} f_{p2} g_{p1} + 2 b_{p2}^2 g_{p1}^2 + 4 b_{p2} c_{p2} g_{p1}^2 + 2 c_{p2}^2 g_{p1}^2 + 2 e_{p2}^2 g_{p1}^2 + \\
& 4 e_{p2} f_{p2} g_{p1}^2 + 2 f_{p2}^2 g_{p1}^2 + 8 a_{p2} c_{p1} e_{p1} g_{p2} - 4 a_{p1} c_{p2} e_{p1} g_{p2} - 4 c_{p2} d_{p1} e_{p1} g_{p2} + \\
& 8 c_{p1} d_{p2} e_{p1} g_{p2} - 4 a_{p1} c_{p1} e_{p2} g_{p2} - 4 c_{p1} d_{p1} e_{p2} g_{p2} - 8 a_{p2} b_{p1} f_{p1} g_{p2} + \\
& 4 a_{p1} b_{p2} f_{p1} g_{p2} + 4 b_{p2} d_{p1} f_{p1} g_{p2} - 8 b_{p1} d_{p2} f_{p1} g_{p2} + 4 a_{p1} b_{p1} f_{p2} g_{p2} + \\
& 4 b_{p1} d_{p1} f_{p2} g_{p2} - 4 b_{p1} b_{p2} g_{p1} g_{p2} - 4 b_{p2} c_{p1} g_{p1} g_{p2} - 4 b_{p1} c_{p2} g_{p1} g_{p2} - \\
& 4 c_{p1} c_{p2} g_{p1} g_{p2} - 4 e_{p1} g_{p2} g_{p1} g_{p2} - 4 e_{p2} f_{p1} g_{p1} g_{p2} - 4 e_{p1} f_{p2} g_{p1} g_{p2} - \\
& 4 f_{p1} f_{p2} g_{p1} g_{p2} + 2 b_{p1}^2 g_{p2}^2 + 4 b_{p1} c_{p1} g_{p2}^2 + 2 c_{p1}^2 g_{p2}^2 + 2 e_{p1}^2 g_{p2}^2 + \\
& 4 e_{p1} f_{p1} g_{p2}^2 + 2 f_{p1}^2 g_{p2}^2
\end{aligned}$$

(3.1.18)

$w_3 =$

$$\begin{aligned}
& - 512 \quad 5 a_{p1} c_{p1} e_{p1} + 5 b_{p2}^2 c_{p1} e_{p1} + 5 a_{p1} a_{p2} c_{p2} e_{p1} - 5 b_{p1} b_{p2} c_{p2} e_{p1} - \\
& \quad 3 b_{p2} c_{p1} c_{p2} e_{p1} + 3 b_{p1} c_{p2}^2 e_{p1} - 3 a_{p2} c_{p2} d_{p1} e_{p1} + 6 a_{p2} c_{p1} d_{p2} e_{p1} - \\
& \quad 3 a_{p1} c_{p2} d_{p2} e_{p1} - 11 c_{p2} d_{p1} d_{p2} e_{p1} + 11 c_{p1} d_{p2}^2 e_{p1} + 5 a_{p1} a_{p2} c_{p1} e_{p2} - \\
& \quad 5 b_{p1} b_{p2} c_{p1} e_{p2} + 3 b_{p2} c_{p1}^2 e_{p2} - 5 a_{p1}^2 c_{p2} e_{p2} + 5 b_{p1}^2 c_{p2} e_{p2} - 3 b_{p1} c_{p1} c_{p2} e_{p2} - \\
& \quad 3 a_{p2} c_{p1} d_{p1} e_{p2} + 6 a_{p1} c_{p2} d_{p1} e_{p2} + 11 c_{p2} d_{p1}^2 e_{p2} - 3 a_{p1} c_{p1} d_{p2} e_{p2} - \\
& \quad 11 c_{p1} d_{p1} d_{p2} e_{p2} + 5 a_{p2}^2 b_{p1} f_{p1} - 5 a_{p1} a_{p2} b_{p2} f_{p1} - 3 b_{p2}^2 c_{p1} f_{p1} + \\
& \quad 3 b_{p1} b_{p2} c_{p2} f_{p1} + 5 b_{p2} c_{p1} c_{p2} f_{p1} - 5 b_{p1} c_{p2}^2 f_{p1} + 3 a_{p2} b_{p2} d_{p1} f_{p1} - \\
& \quad 6 a_{p2} b_{p1} d_{p2} f_{p1} + 3 a_{p1} b_{p2} d_{p2} f_{p1} + 11 b_{p2} d_{p1} d_{p2} f_{p1} - 11 b_{p1} d_{p2}^2 f_{p1} + \\
& \quad 5 b_{p2} e_{p1} e_{p2} f_{p1} - 3 c_{p2} e_{p1} e_{p2} f_{p1} - 5 b_{p1} e_{p2}^2 f_{p1} + 3 c_{p1} e_{p2}^2 f_{p1} - \\
& \quad 3 b_{p2} e_{p2} f_{p1}^2 + 5 c_{p2} e_{p2} f_{p1}^2 - 5 a_{p1} a_{p2} b_{p1} f_{p2} + 5 a_{p1}^2 b_{p2} f_{p2} + 3 b_{p1} b_{p2} c_{p1} f_{p2} - \\
& \quad 5 b_{p2} c_{p1}^2 f_{p2} - 3 b_{p1}^2 c_{p2} f_{p2} + 5 b_{p1} c_{p1} c_{p2} f_{p2} + 3 a_{p2} b_{p1} d_{p1} f_{p2} - \\
& \quad 6 a_{p1} b_{p2} d_{p1} f_{p2} - 11 b_{p2} d_{p1}^2 f_{p2} + 3 a_{p1} b_{p1} d_{p2} f_{p2} + 11 b_{p1} d_{p1} d_{p2} f_{p2} - \\
& \quad 5 b_{p2} e_{p1}^2 f_{p2} + 3 c_{p2} e_{p1}^2 f_{p2} + 5 b_{p1} e_{p1} e_{p2} f_{p2} - 3 c_{p1} e_{p1} e_{p2} f_{p2} + \\
& \quad 3 b_{p2} e_{p1} f_{p1} f_{p2} - 5 c_{p2} e_{p1} f_{p1} f_{p2} + 3 b_{p1} e_{p2} f_{p1} f_{p2} - 5 c_{p1} e_{p2} f_{p1} f_{p2} - \\
& \quad 3 b_{p1} e_{p1} f_{p2}^2 + 5 c_{p1} e_{p1} f_{p2}^2 + 5 a_{p2} b_{p1} b_{p2} g_{p1} - 5 a_{p1} b_{p2}^2 g_{p1} - 3 a_{p2} b_{p2} c_{p1} g_{p1} - \\
& \quad 3 a_{p2} b_{p1} c_{p2} g_{p1} + 6 a_{p1} b_{p2} c_{p2} g_{p1} + 5 a_{p2} c_{p1} c_{p2} g_{p1} - 5 a_{p1} c_{p2}^2 g_{p1} + \\
& \quad 3 b_{p2}^2 d_{p1} g_{p1} + 22 b_{p2} c_{p2} d_{p1} g_{p1} + 3 c_{p2}^2 d_{p1} g_{p1} - 3 b_{p1} b_{p2} d_{p2} g_{p1} - \\
& \quad 11 b_{p2} c_{p1} d_{p2} g_{p1} - 11 b_{p1} c_{p2} d_{p2} g_{p1} - 3 c_{p1} c_{p2} d_{p2} g_{p1} + 5 a_{p2} e_{p1} e_{p2} g_{p1} - \\
& \quad 3 d_{p2} e_{p1} e_{p2} g_{p1} - 5 a_{p1} e_{p2}^2 g_{p1} + 3 d_{p1} e_{p2}^2 g_{p1} - 3 a_{p2} e_{p2} f_{p1} g_{p1} - \\
& \quad 11 d_{p2} e_{p2} f_{p1} g_{p1} - 3 a_{p2} e_{p1} f_{p2} g_{p1} - 11 d_{p2} e_{p1} f_{p2} g_{p1} + 6 a_{p1} e_{p2} f_{p2} g_{p1} + \\
& \quad 22 d_{p1} e_{p2} f_{p2} g_{p1} + 5 a_{p2} f_{p1} f_{p2} g_{p1} - 3 d_{p2} f_{p1} f_{p2} g_{p1} - 5 a_{p1} f_{p2}^2 g_{p1} + \\
& \quad 3 d_{p1} f_{p2}^2 g_{p1} - 16 c_{p2} e_{p2} g_{p1}^2 + 16 b_{p2} f_{p2} g_{p1}^2 - 5 a_{p2} b_{p1}^2 g_{p2} + 5 a_{p1} b_{p1} b_{p2} g_{p2} + \\
& \quad 6 a_{p2} b_{p1} c_{p1} g_{p2} - 3 a_{p1} b_{p2} c_{p1} g_{p2} - 5 a_{p2} c_{p1}^2 g_{p2} - 3 a_{p1} b_{p1} c_{p2} g_{p2} + \\
& \quad 5 a_{p1} c_{p1} c_{p2} g_{p2} - 3 b_{p1} b_{p2} d_{p1} g_{p2} - 11 b_{p2} c_{p1} d_{p1} g_{p2} - 11 b_{p1} c_{p2} d_{p1} g_{p2} - \\
& \quad 3 c_{p1} c_{p2} d_{p1} g_{p2} + 3 b_{p1}^2 d_{p2} g_{p2} + 22 b_{p1} c_{p1} d_{p2} g_{p2} + 3 c_{p1}^2 d_{p2} g_{p2} - \\
& \quad 5 a_{p2} e_{p1}^2 g_{p2} + 3 d_{p2} e_{p1}^2 g_{p2} + 5 a_{p1} e_{p1} e_{p2} g_{p2} - 3 d_{p1} e_{p1} e_{p2} g_{p2} + \\
& \quad 6 a_{p2} e_{p1} f_{p1} g_{p2} + 22 d_{p2} e_{p1} f_{p1} g_{p2} - 3 a_{p1} e_{p2} f_{p1} g_{p2} - 11 d_{p1} e_{p2} f_{p1} g_{p2} - \\
& \quad 5 a_{p2} f_{p1}^2 g_{p2} + 3 d_{p2} f_{p1}^2 g_{p2} - 3 a_{p1} e_{p1} f_{p2} g_{p2} - 11 d_{p1} e_{p1} f_{p2} g_{p2} + \\
& \quad 5 a_{p1} f_{p1} f_{p2} g_{p2} - 3 d_{p1} f_{p1} f_{p2} g_{p2} + 16 c_{p2} e_{p1} g_{p1} g_{p2} + 16 c_{p1} e_{p2} g_{p1} g_{p2} - \\
& \quad 16 b_{p2} f_{p1} g_{p1} g_{p2} - 16 b_{p1} f_{p2} g_{p1} g_{p2} - 16 c_{p1} e_{p1} g_{p2}^2 + 16 b_{p1} f_{p1} g_{p2}^2
\end{aligned}$$

(3.1.19)

$w_4 =$

$$\begin{aligned}
256 & a_{p2}^2 a_{p1}^2 - 14 a_{p1} a_{p2} b_{p1} b_{p2} + 7 a_{p1}^2 b_{p2}^2 - 10 a_{p2}^2 b_{p1} c_{p1} + 10 a_{p1} a_{p2} b_{p2} c_{p1} + \\
& 7 a_{p2}^2 c_{p1}^2 - 8 b_{p2}^2 c_{p1}^2 + 10 a_{p1} a_{p2} b_{p1} c_{p2} - 10 a_{p1}^2 b_{p2} c_{p2} - 14 a_{p1} a_{p2} c_{p1} c_{p2} + \\
& 16 b_{p1} b_{p2} c_{p1} c_{p2} + 7 a_{p1}^2 c_{p2}^2 - 8 b_{p1}^2 c_{p2}^2 + 10 a_{p2} b_{p1} b_{p2} d_{p1} - 10 a_{p1} b_{p2}^2 d_{p1} + \\
& 2 a_{p2} b_{p2} c_{p1} d_{p1} + 2 a_{p2} b_{p1} c_{p2} d_{p1} - 4 a_{p1} b_{p2} c_{p2} d_{p1} + 10 a_{p2} c_{p1} c_{p2} d_{p1} - \\
& 10 a_{p1} c_{p2}^2 d_{p1} - b_{p2}^2 d_{p1}^2 + 38 b_{p2} c_{p2} d_{p1}^2 - c_{p2}^2 d_{p1}^2 - 10 a_{p2} b_{p1}^2 d_{p2} + \\
& 10 a_{p1} b_{p1} b_{p2} d_{p2} - 4 a_{p2} b_{p1} c_{p1} d_{p2} + 2 a_{p1} b_{p2} c_{p1} d_{p2} - 10 a_{p2} c_{p1}^2 d_{p2} + \\
& 2 a_{p1} b_{p1} c_{p2} d_{p2} + 10 a_{p1} c_{p1} c_{p2} d_{p2} + 2 b_{p1} b_{p2} d_{p1} d_{p2} - 38 b_{p2} c_{p1} d_{p1} d_{p2} - \\
& 38 b_{p1} c_{p2} d_{p1} d_{p2} + 2 c_{p1} c_{p2} d_{p1} d_{p2} - b_{p1}^2 d_{p2}^2 + 38 b_{p1} c_{p1} d_{p2}^2 - c_{p1}^2 d_{p2}^2 + \\
& 7 a_{p2}^2 e_{p1}^2 - 7 b_{p2}^2 e_{p1}^2 + 10 b_{p2} c_{p2} e_{p1}^2 + c_{p2}^2 e_{p1}^2 - 10 a_{p2} d_{p2} e_{p1}^2 - d_{p2}^2 e_{p1}^2 - \\
& 14 a_{p1} a_{p2} e_{p1} e_{p2} + 14 b_{p1} b_{p2} e_{p1} e_{p2} - 10 b_{p2} c_{p1} e_{p1} e_{p2} - 10 b_{p1} c_{p2} e_{p1} e_{p2} - \\
& 2 c_{p1} c_{p2} e_{p1} e_{p2} + 10 a_{p2} d_{p1} e_{p1} e_{p2} + 10 a_{p1} d_{p2} e_{p1} e_{p2} + 2 d_{p1} d_{p2} e_{p1} e_{p2} + \\
& 7 a_{p1}^2 e_{p2}^2 - 7 b_{p1}^2 e_{p2}^2 + 10 b_{p1} c_{p1} e_{p2}^2 + c_{p1}^2 e_{p2}^2 - 10 a_{p1} d_{p1} e_{p2}^2 - d_{p1}^2 e_{p2}^2 - \\
& 10 a_{p2}^2 e_{p1} f_{p1} + 10 b_{p2}^2 e_{p1} f_{p1} - 12 b_{p2} c_{p2} e_{p1} f_{p1} + 10 c_{p2}^2 e_{p1} f_{p1} - \\
& 4 a_{p2} d_{p2} e_{p1} f_{p1} + 38 d_{p2}^2 e_{p1} f_{p1} + 10 a_{p1} a_{p2} e_{p2} f_{p1} - 10 b_{p1} b_{p2} e_{p2} f_{p1} - \\
& 18 b_{p2} c_{p1} e_{p2} f_{p1} + 30 b_{p1} c_{p2} e_{p2} f_{p1} - 10 c_{p1} c_{p2} e_{p2} f_{p1} + 2 a_{p2} d_{p1} e_{p2} f_{p1} + \\
& 2 a_{p1} d_{p2} e_{p2} f_{p1} - 38 d_{p1} d_{p2} e_{p2} f_{p1} + 7 a_{p2}^2 f_{p1}^2 + b_{p2}^2 f_{p1}^2 + 10 b_{p2} c_{p2} f_{p1}^2 - \\
& 7 c_{p2}^2 f_{p1}^2 - 10 a_{p2} d_{p2} f_{p1}^2 - d_{p2}^2 f_{p1}^2 - 8 e_{p2}^2 f_{p1}^2 + 10 a_{p1} a_{p2} e_{p1} f_{p2} - \\
& 10 b_{p1} b_{p2} e_{p1} f_{p2} + 30 b_{p2} c_{p1} e_{p1} f_{p2} - 18 b_{p1} c_{p2} e_{p1} f_{p2} - 10 c_{p1} c_{p2} e_{p1} f_{p2} + \\
& 2 a_{p2} d_{p1} e_{p1} f_{p2} + 2 a_{p1} d_{p2} e_{p1} f_{p2} - 38 d_{p1} d_{p2} e_{p1} f_{p2} - 10 a_{p2}^2 e_{p2} f_{p2} + \\
& 10 b_{p1}^2 e_{p2} f_{p2} - 12 b_{p1} c_{p1} e_{p2} f_{p2} + 10 c_{p1}^2 e_{p2} f_{p2} - 4 a_{p1} d_{p1} e_{p2} f_{p2} + \\
& 38 d_{p1}^2 e_{p2} f_{p2} - 14 a_{p1} a_{p2} f_{p1} f_{p2} - 2 b_{p1} b_{p2} f_{p1} f_{p2} - 10 b_{p2} c_{p1} f_{p1} f_{p2} - \\
& 10 b_{p1} c_{p2} f_{p1} f_{p2} + 14 c_{p1} c_{p2} f_{p1} f_{p2} + 10 a_{p2} d_{p1} f_{p1} f_{p2} + 10 a_{p1} d_{p2} f_{p1} f_{p2} + \\
& 2 d_{p1} d_{p2} f_{p1} f_{p2} + 16 e_{p1} e_{p2} f_{p1} f_{p2} + 7 a_{p1}^2 f_{p2}^2 + b_{p1}^2 f_{p2}^2 + 10 b_{p1} c_{p1} f_{p2}^2 - \\
& 7 c_{p1}^2 f_{p2}^2 - 10 a_{p1} d_{p1} f_{p2}^2 - d_{p1}^2 f_{p2}^2 - 8 e_{p1}^2 f_{p2}^2 - 16 a_{p2} c_{p2} e_{p1} g_{p1} + \\
& 48 c_{p2} d_{p2} e_{p1} g_{p1} - 16 a_{p2} c_{p1} e_{p2} g_{p1} + 32 a_{p1} c_{p2} e_{p2} g_{p1} - 96 c_{p2} d_{p1} e_{p2} g_{p1} + \\
& 48 c_{p1} d_{p2} e_{p2} g_{p1} + 16 a_{p2} b_{p2} f_{p1} g_{p1} - 48 b_{p2} d_{p2} f_{p1} g_{p1} + 16 a_{p2} b_{p1} f_{p2} g_{p1} - \\
& 32 a_{p1} b_{p2} f_{p2} g_{p1} + 96 b_{p2} d_{p1} f_{p2} g_{p1} - 48 b_{p1} d_{p2} f_{p2} g_{p1} + 8 b_{p2}^2 g_{p1}^2 - \\
& 48 b_{p2} c_{p2} g_{p1}^2 + 8 c_{p2}^2 g_{p1}^2 + 8 e_{p2}^2 g_{p1}^2 - 48 e_{p2} f_{p2} g_{p1}^2 + 8 f_{p2}^2 g_{p1}^2 + \\
& 32 a_{p2} c_{p1} e_{p1} g_{p2} - 16 a_{p1} c_{p2} e_{p1} g_{p2} + 48 c_{p2} d_{p1} e_{p1} g_{p2} - 96 c_{p1} d_{p2} e_{p1} g_{p2} - \\
& 16 a_{p1} c_{p1} e_{p2} g_{p2} + 48 c_{p1} d_{p1} e_{p2} g_{p2} - 32 a_{p2} b_{p1} f_{p1} g_{p2} + 16 a_{p1} b_{p2} f_{p1} g_{p2} - \\
& 48 b_{p2} d_{p1} f_{p1} g_{p2} + 96 b_{p1} d_{p2} f_{p1} g_{p2} + 16 a_{p1} b_{p1} f_{p2} g_{p2} - 48 b_{p1} d_{p1} f_{p2} g_{p2} - \\
& 16 b_{p1} b_{p2} g_{p1} g_{p2} + 48 b_{p2} c_{p1} g_{p1} g_{p2} + 48 b_{p1} c_{p2} g_{p1} g_{p2} - 16 c_{p1} c_{p2} g_{p1} g_{p2} - \\
& 16 e_{p1} e_{p2} g_{p1} g_{p2} + 48 e_{p2} f_{p1} g_{p1} g_{p2} + 48 e_{p1} f_{p2} g_{p1} g_{p2} - 16 f_{p1} f_{p2} g_{p1} g_{p2} + \\
& 8 b_{p1}^2 g_{p2}^2 - 48 b_{p1} c_{p1} g_{p2}^2 + 8 c_{p1}^2 g_{p2}^2 + 8 e_{p1}^2 g_{p2}^2 - 48 e_{p1} f_{p1} g_{p2}^2 + 8 f_{p1}^2 g_{p2}^2
\end{aligned}$$

(3.1.20)

$w_5 =$

$$\begin{aligned}
& - 512 \quad 9 a_{p1} c_{p1} e_{p1} + 9 b_{p2}^2 c_{p1} e_{p1} + 9 a_{p1} a_{p2} c_{p2} e_{p1} - 9 b_{p1} b_{p2} c_{p2} e_{p1} - \\
& \quad 15 b_{p2} c_{p1} c_{p2} e_{p1} + 15 b_{p1} c_{p2}^2 e_{p1} - 15 a_{p2} c_{p2} d_{p1} e_{p1} + 30 a_{p2} c_{p1} d_{p2} e_{p1} - \\
& \quad 15 a_{p1} c_{p2} d_{p2} e_{p1} + 25 c_{p2} d_{p1} d_{p2} e_{p1} - 25 c_{p1} d_{p2}^2 e_{p1} + 9 a_{p1} a_{p2} c_{p1} e_{p2} - \\
& \quad 9 b_{p1} b_{p2} c_{p1} e_{p2} + 15 b_{p2} c_{p1}^2 e_{p2} - 9 a_{p1}^2 c_{p2} e_{p2} + 9 b_{p1}^2 c_{p2} e_{p2} - \\
& \quad 15 b_{p1} c_{p1} c_{p2} e_{p2} - 15 a_{p2} c_{p1} d_{p1} e_{p2} + 30 a_{p1} c_{p2} d_{p1} e_{p2} - 25 c_{p2} d_{p1}^2 e_{p2} - \\
& \quad 15 a_{p1} c_{p1} d_{p2} e_{p2} + 25 c_{p1} d_{p1} d_{p2} e_{p2} + 9 a_{p2}^2 b_{p1} f_{p1} - 9 a_{p1} a_{p2} b_{p2} f_{p1} - \\
& \quad 15 b_{p2}^2 c_{p1} f_{p1} + 15 b_{p1} b_{p2} c_{p2} f_{p1} + 9 b_{p2} c_{p1} c_{p2} f_{p1} - 9 b_{p1} c_{p2}^2 f_{p1} + \\
& \quad 15 a_{p2} b_{p2} d_{p1} f_{p1} - 30 a_{p2} b_{p1} d_{p2} f_{p1} + 15 a_{p1} b_{p2} d_{p2} f_{p1} - 25 b_{p2} d_{p1} d_{p2} f_{p1} + \\
& \quad 25 b_{p1} d_{p2}^2 f_{p1} + 9 b_{p2} e_{p1} e_{p2} f_{p1} - 15 c_{p2} e_{p1} e_{p2} f_{p1} - 9 b_{p1} e_{p2}^2 f_{p1} + \\
& \quad 15 c_{p1} e_{p2}^2 f_{p1} - 15 b_{p2} e_{p2} f_{p1}^2 + 9 c_{p2} e_{p2} f_{p1}^2 - 9 a_{p1} a_{p2} b_{p1} f_{p2} + 9 a_{p1}^2 b_{p2} f_{p2} + \\
& \quad 15 b_{p1} b_{p2} c_{p1} f_{p2} - 9 b_{p2} c_{p1}^2 f_{p2} - 15 b_{p1}^2 c_{p2} f_{p2} + 9 b_{p1} c_{p1} c_{p2} f_{p2} + \\
& \quad 15 a_{p2} b_{p1} d_{p1} f_{p2} - 30 a_{p1} b_{p2} d_{p1} f_{p2} + 25 b_{p2} d_{p1}^2 f_{p2} + 15 a_{p1} b_{p1} d_{p2} f_{p2} - \\
& \quad 25 b_{p1} d_{p1} d_{p2} f_{p2} - 9 b_{p2} e_{p1}^2 f_{p2} + 15 c_{p2} e_{p1}^2 f_{p2} + 9 b_{p1} e_{p1} e_{p2} f_{p2} - \\
& \quad 15 c_{p1} e_{p1} e_{p2} f_{p2} + 15 b_{p2} e_{p1} f_{p1} f_{p2} - 9 c_{p2} e_{p1} f_{p1} f_{p2} + 15 b_{p1} e_{p2} f_{p1} f_{p2} - \\
& \quad 9 c_{p1} e_{p2} f_{p1} f_{p2} - 15 b_{p1} e_{p1} f_{p2}^2 + 9 c_{p1} e_{p1} f_{p2}^2 + 9 a_{p2} b_{p1} b_{p2} g_{p1} - \\
& \quad 9 a_{p1} b_{p2}^2 g_{p1} - 15 a_{p2} b_{p2} c_{p1} g_{p1} - 15 a_{p2} b_{p1} c_{p2} g_{p1} + 30 a_{p1} b_{p2} c_{p2} g_{p1} + \\
& \quad 9 a_{p2} c_{p1} c_{p2} g_{p1} - 9 a_{p1} c_{p2}^2 g_{p1} + 15 b_{p2}^2 d_{p1} g_{p1} - 50 b_{p2} c_{p2} d_{p1} g_{p1} + \\
& \quad 15 c_{p2}^2 d_{p1} g_{p1} - 15 b_{p1} b_{p2} d_{p2} g_{p1} + 25 b_{p2} c_{p1} d_{p2} g_{p1} + 25 b_{p1} c_{p2} d_{p2} g_{p1} - \\
& \quad 15 c_{p1} c_{p2} d_{p2} g_{p1} + 9 a_{p2} e_{p1} e_{p2} g_{p1} - 15 d_{p2} e_{p1} e_{p2} g_{p1} - 9 a_{p1} e_{p2}^2 g_{p1} + \\
& \quad 15 d_{p1} e_{p2}^2 g_{p1} - 15 a_{p2} e_{p2} f_{p1} g_{p1} + 25 d_{p2} e_{p2} f_{p1} g_{p1} - 15 a_{p2} e_{p1} f_{p2} g_{p1} + \\
& \quad 25 d_{p2} e_{p1} f_{p2} g_{p1} + 30 a_{p1} e_{p2} f_{p2} g_{p1} - 50 d_{p1} e_{p2} f_{p2} g_{p1} + 9 a_{p2} f_{p1} f_{p2} g_{p1} - \\
& \quad 15 d_{p2} f_{p1} f_{p2} g_{p1} - 9 a_{p1} f_{p2}^2 g_{p1} + 15 d_{p1} f_{p2}^2 g_{p1} + 16 c_{p2} e_{p2} g_{p1}^2 - \\
& \quad 16 b_{p2} f_{p2} g_{p1}^2 - 9 a_{p2} b_{p1}^2 g_{p2} + 9 a_{p1} b_{p1} b_{p2} g_{p2} + 30 a_{p2} b_{p1} c_{p1} g_{p2} - \\
& \quad 15 a_{p1} b_{p2} c_{p1} g_{p2} - 9 a_{p2} c_{p1}^2 g_{p2} - 15 a_{p1} b_{p1} c_{p2} g_{p2} + 9 a_{p1} c_{p1} c_{p2} g_{p2} - \\
& \quad 15 b_{p1} b_{p2} d_{p1} g_{p2} + 25 b_{p2} c_{p1} d_{p1} g_{p2} + 25 b_{p1} c_{p2} d_{p1} g_{p2} - 15 c_{p1} c_{p2} d_{p1} g_{p2} + \\
& \quad 15 b_{p1}^2 d_{p2} g_{p2} - 50 b_{p1} c_{p1} d_{p2} g_{p2} + 15 c_{p1}^2 d_{p2} g_{p2} - 9 a_{p2} e_{p1}^2 g_{p2} + \\
& \quad 15 d_{p2} e_{p1}^2 g_{p2} + 9 a_{p1} e_{p1} e_{p2} g_{p2} - 15 d_{p1} e_{p1} e_{p2} g_{p2} + 30 a_{p2} e_{p1} f_{p1} g_{p2} - \\
& \quad 50 d_{p2} e_{p1} f_{p1} g_{p2} - 15 a_{p1} e_{p2} f_{p1} g_{p2} + 25 d_{p1} e_{p2} f_{p1} g_{p2} - 9 a_{p2} f_{p1}^2 g_{p2} + \\
& \quad 15 d_{p2} f_{p1}^2 g_{p2} - 15 a_{p1} e_{p1} f_{p2} g_{p2} + 25 d_{p1} e_{p1} f_{p2} g_{p2} + 9 a_{p1} f_{p1} f_{p2} g_{p2} - \\
& \quad 15 d_{p1} f_{p1} f_{p2} g_{p2} - 16 c_{p2} e_{p1} g_{p1} g_{p2} - 16 c_{p1} e_{p2} g_{p1} g_{p2} + 16 b_{p2} f_{p1} g_{p1} g_{p2} + \\
& \quad 16 b_{p1} f_{p2} g_{p1} g_{p2} + 16 c_{p1} e_{p1} g_{p2}^2 - 16 b_{p1} f_{p1} g_{p2}^2
\end{aligned}$$

(3.1.21)

$w_6 =$

$$\begin{aligned}
& 512 a_{p2}^2 b_{p1}^2 - 14 a_{p1} a_{p2} b_{p1} b_{p2} + 7 a_{p1}^2 b_{p2}^2 - 16 a_{p2}^2 b_{p1} c_{p1} + 16 a_{p1} a_{p2} b_{p2} c_{p1} + \\
& 7 a_{p2}^2 c_{p1}^2 + 2 b_{p2}^2 c_{p1}^2 + 16 a_{p1} a_{p2} b_{p1} c_{p2} - 16 a_{p1}^2 b_{p2} c_{p2} - 14 a_{p1} a_{p2} c_{p1} c_{p2} - \\
& 4 b_{p1} b_{p2} c_{p1} c_{p2} + 7 a_{p1}^2 c_{p2}^2 + 2 b_{p1}^2 c_{p2}^2 + 16 a_{p2} b_{p1} b_{p2} d_{p1} - 16 a_{p1} b_{p2}^2 d_{p1} - \\
& 18 a_{p2} b_{p2} c_{p1} d_{p1} - 18 a_{p2} b_{p1} c_{p2} d_{p1} + 36 a_{p1} b_{p2} c_{p2} d_{p1} + 16 a_{p2} c_{p1} c_{p2} d_{p1} - \\
& 16 a_{p1} c_{p2}^2 d_{p1} + 9 b_{p2}^2 d_{p1}^2 - 12 b_{p2} c_{p2} d_{p1}^2 + 9 c_{p2}^2 d_{p1}^2 - 16 a_{p2} b_{p1}^2 d_{p2} + \\
& 16 a_{p1} b_{p1} b_{p2} d_{p2} + 36 a_{p2} b_{p1} c_{p1} d_{p2} - 18 a_{p1} b_{p2} c_{p1} d_{p2} - 16 a_{p2} c_{p1}^2 d_{p2} - \\
& 18 a_{p1} b_{p1} c_{p2} d_{p2} + 16 a_{p1} c_{p1} c_{p2} d_{p2} - 18 b_{p1} b_{p2} d_{p1} d_{p2} + 12 b_{p2} c_{p1} d_{p1} d_{p2} + \\
& 12 b_{p1} c_{p2} d_{p1} d_{p2} - 18 c_{p1} c_{p2} d_{p1} d_{p2} + 9 b_{p1}^2 d_{p2}^2 - 12 b_{p1} c_{p1} d_{p2}^2 + 9 c_{p1}^2 d_{p2}^2 + \\
& 7 a_{p2}^2 e_{p1}^2 - 7 b_{p2}^2 e_{p1}^2 + 16 b_{p2} c_{p2} e_{p1}^2 - 9 c_{p2}^2 e_{p1}^2 - 16 a_{p2} d_{p2} e_{p1}^2 + 9 d_{p2}^2 e_{p1}^2 - \\
& 14 a_{p1} a_{p2} e_{p1} e_{p2} + 14 b_{p1} b_{p2} e_{p1} e_{p2} - 16 b_{p2} c_{p1} e_{p1} e_{p2} - 16 b_{p1} c_{p2} e_{p1} e_{p2} + \\
& 18 c_{p1} c_{p2} e_{p1} e_{p2} + 16 a_{p2} d_{p1} e_{p1} e_{p2} + 16 a_{p1} d_{p2} e_{p1} e_{p2} - 18 d_{p1} d_{p2} e_{p1} e_{p2} + \\
& 7 a_{p1}^2 e_{p2}^2 - 7 b_{p1}^2 e_{p2}^2 + 16 b_{p1} c_{p1} e_{p2}^2 - 9 c_{p1}^2 e_{p2}^2 - 16 a_{p1} d_{p1} e_{p2}^2 + 9 d_{p1}^2 e_{p2}^2 - \\
& 16 a_{p2}^2 e_{p1} f_{p1} + 16 b_{p2}^2 e_{p1} f_{p1} - 32 b_{p2} c_{p2} e_{p1} f_{p1} + 16 c_{p2}^2 e_{p1} f_{p1} + \\
& 36 a_{p2} d_{p2} e_{p1} f_{p1} - 12 d_{p2}^2 e_{p1} f_{p1} + 16 a_{p1} a_{p2} e_{p2} f_{p1} - 16 b_{p1} b_{p2} e_{p2} f_{p1} + \\
& 22 b_{p2} c_{p1} e_{p2} f_{p1} + 10 b_{p1} c_{p2} e_{p2} f_{p1} - 16 c_{p1} c_{p2} e_{p2} f_{p1} - 18 a_{p2} d_{p1} e_{p2} f_{p1} - \\
& 18 a_{p1} d_{p2} e_{p2} f_{p1} + 12 d_{p1} d_{p2} e_{p2} f_{p1} + 7 a_{p2}^2 f_{p1}^2 - 9 b_{p2}^2 f_{p1}^2 + 16 b_{p2} c_{p2} f_{p1}^2 - \\
& 7 c_{p2}^2 f_{p1}^2 - 16 a_{p2} d_{p2} f_{p1}^2 + 9 d_{p2}^2 f_{p1}^2 + 2 e_{p2}^2 f_{p1}^2 + 16 a_{p1} a_{p2} e_{p1} f_{p2} - \\
& 16 b_{p1} b_{p2} e_{p1} f_{p2} + 10 b_{p2} c_{p1} e_{p1} f_{p2} + 22 b_{p1} c_{p2} e_{p1} f_{p2} - 16 c_{p1} c_{p2} e_{p1} f_{p2} - \\
& 18 a_{p2} d_{p1} e_{p1} f_{p2} - 18 a_{p1} d_{p2} e_{p1} f_{p2} + 12 d_{p1} d_{p2} e_{p1} f_{p2} - 16 a_{p1}^2 e_{p2} f_{p2} + \\
& 16 b_{p1}^2 e_{p2} f_{p2} - 32 b_{p1} c_{p1} e_{p2} f_{p2} + 16 c_{p1}^2 e_{p2} f_{p2} + 36 a_{p1} d_{p1} e_{p2} f_{p2} - \\
& 12 d_{p1}^2 e_{p2} f_{p2} - 14 a_{p1} a_{p2} f_{p1} f_{p2} + 18 b_{p1} b_{p2} f_{p1} f_{p2} - 16 b_{p2} c_{p1} f_{p1} f_{p2} - \\
& 16 b_{p1} c_{p2} f_{p1} f_{p2} + 14 c_{p1} c_{p2} f_{p1} f_{p2} + 16 a_{p2} d_{p1} f_{p1} f_{p2} + 16 a_{p1} d_{p2} f_{p1} f_{p2} - \\
& 18 d_{p1} d_{p2} f_{p1} f_{p2} - 4 e_{p1} e_{p2} f_{p1} f_{p2} + 7 a_{p1}^2 f_{p2}^2 - 9 b_{p1}^2 f_{p2}^2 + 16 b_{p1} c_{p1} f_{p2}^2 - \\
& 7 c_{p1}^2 f_{p2}^2 - 16 a_{p1} d_{p1} f_{p2}^2 + 9 d_{p1}^2 f_{p2}^2 + 2 e_{p1}^2 f_{p2}^2 + 4 a_{p2} c_{p2} e_{p1} g_{p1} + \\
& 4 c_{p2} d_{p2} e_{p1} g_{p1} + 4 a_{p2} c_{p1} e_{p2} g_{p1} - 8 a_{p1} c_{p2} e_{p2} g_{p1} - 8 c_{p2} d_{p1} e_{p2} g_{p1} + \\
& 4 c_{p1} d_{p2} e_{p2} g_{p1} - 4 a_{p2} b_{p2} f_{p1} g_{p1} - 4 b_{p2} d_{p2} f_{p1} g_{p1} - 4 a_{p2} b_{p1} f_{p2} g_{p1} + \\
& 8 a_{p1} b_{p2} f_{p2} g_{p1} + 8 b_{p2} d_{p1} f_{p2} g_{p1} - 4 b_{p1} d_{p2} f_{p2} g_{p1} - 2 b_{p2}^2 g_{p1}^2 - \\
& 4 b_{p2} c_{p2} g_{p1}^2 - 2 c_{p2}^2 g_{p1}^2 - 2 e_{p2}^2 g_{p1}^2 - 4 e_{p2} f_{p2} g_{p1}^2 - 2 f_{p2}^2 g_{p1}^2 - 8 a_{p2} c_{p1} e_{p1} g_{p2} + \\
& 4 a_{p1} c_{p2} e_{p1} g_{p2} + 4 c_{p2} d_{p1} e_{p1} g_{p2} - 8 c_{p1} d_{p2} e_{p1} g_{p2} + 4 a_{p1} c_{p1} e_{p2} g_{p2} + \\
& 4 c_{p1} d_{p1} e_{p2} g_{p2} + 8 a_{p2} b_{p1} f_{p1} g_{p2} - 4 a_{p1} b_{p2} f_{p1} g_{p2} - 4 b_{p2} d_{p1} f_{p1} g_{p2} + \\
& 8 b_{p1} d_{p2} f_{p1} g_{p2} - 4 a_{p1} b_{p1} f_{p2} g_{p2} - 4 b_{p1} d_{p1} f_{p2} g_{p2} + 4 b_{p1} b_{p2} g_{p1} g_{p2} + \\
& 4 b_{p2} c_{p1} g_{p1} g_{p2} + 4 b_{p1} c_{p2} g_{p1} g_{p2} + 4 c_{p1} c_{p2} g_{p1} g_{p2} + 4 e_{p1} g_{p1} g_{p1} g_{p2} + \\
& 4 e_{p2} f_{p1} g_{p1} g_{p2} + 4 e_{p1} f_{p2} g_{p1} g_{p2} + 4 f_{p1} f_{p2} g_{p1} g_{p2} - 2 b_{p1}^2 g_{p1}^2 - \\
& 4 b_{p1} c_{p1} g_{p2}^2 - 2 c_{p1}^2 g_{p2}^2 - 2 e_{p1}^2 g_{p2}^2 - 4 e_{p1} f_{p1} g_{p2}^2 - 2 f_{p1}^2 g_{p2}^2
\end{aligned}$$

(3.1.22)

T 'ye bađlı (3.1.11) polinomunun tm kklerinin bulunması sonucunda ψ_0 ađısına ait, gerek veya sanal olmak zere, 12 adet zm kmesi

$$\Psi_{oi} = 2 \arctan t_i; \quad i = 1, 2, \dots, 12 \quad (3.1.23)$$

olarak bulunur.

(3.1.23) ifadesinden elde edilen çözüm kümesi (3.1.9) bağıntısında yerine yazılıp v_4 terimi yok edilirse, çıkış kolu açısı α_0 in çözümü için (3.1.24)'deki ikinci dereceden polinomun köklerinden faydalanılır.

$$(aa_2bb_1 - aa_1bb_2) + (aa_1cc_2 - aa_2cc_1)v + (aa_1bb_2 - aa_2bb_1)v^2 = 0 \quad (3.1.24)$$

Burada,

$$\left. \begin{aligned} aa_k &= a_{pk} + b_{pk} + (c_{pk} + d_{pk})\cos 2\Psi_{0i} + (-f_{pk} + g_{pk})\sin 2\Psi_{0i} \\ bb_k &= -4(c_{pk}\sin 2\Psi_{0i} + f_{pk}\cos 2\Psi_{0i} + e_{pk}) \\ cc_k &= 2(a_{pk} - 3b_{pk} + (-3c_{pk} + d_{pk})\cos 2\Psi_{0i} + (3f_{pk} + g_{pk})\sin 2\Psi_{0i}) \end{aligned} \right\} \begin{array}{l} k = 1, 2 \\ i = 1, 2, \dots, 12 \end{array} \quad (3.1.25)$$

olarak hesaplanır.

Ayrıca (3.24) denkleminin kökleri yardımı ile çıkış kolu açısı α_0 (3.1.26)'daki gibi olur.

$$\alpha_{0i\pm} = \text{Arctan } v_{i\pm} \quad i = 1, 2, \dots, 12 \quad (3.1.26)$$

Fonksiyon üreten dört çubuğa ait diğer boyut (x_1, x_2, x_3) parametrelerinin belirlenmesi için (3.1.27)'deki (Z_1, Z_2, Z_3) büyüklüklerinden faydalanılır.

$$\begin{aligned}
Z_{2i\pm} = & \frac{\left\{ \begin{array}{l} \text{Cos}(\alpha_{0i\pm} - \Psi_{0i}) \left[\begin{array}{l} a_k(-c_2f_1 + c_1f_2 - b_2g_1 + b_1g_2) + \\ a_2(c_kf_1 - c_1f_k + b_kg_1 - b_1g_k) + \\ a_1(-c_kf_2 + c_2f_k - b_kg_2 + b_2g_k) \end{array} \right] + \\ \text{Cos}(\alpha_{0i\pm} + \Psi_{0i}) \left[\begin{array}{l} a_k(c_2f_1 - c_1f_2 - b_2g_1 + b_1g_2) + \\ a_2(-c_kf_1 + c_1f_k + b_kg_1 - b_1g_k) + \\ a_1(c_kf_2 - c_2f_k - b_kg_2 + b_2g_k) \end{array} \right] + \\ \text{Sin}(\alpha_{0i\pm} - \Psi_{0i}) \left[\begin{array}{l} a_k(-b_2f_1 + b_1f_2 + c_2g_1 - c_1g_2) + \\ a_2(b_kf_1 - b_1f_k - c_kg_1 + c_1g_k) + \\ a_1(-b_kf_2 + b_2f_k + c_kg_2 - c_2g_k) \end{array} \right] + \\ \text{Sin}(\alpha_{0i\pm} + \Psi_{0i}) \left[\begin{array}{l} a_k(b_2f_1 - b_1f_2 + c_2g_1 - c_1g_2) + \\ a_2(-b_kf_1 + b_1f_k - c_kg_1 + c_1g_k) + \\ a_1(b_kf_2 - b_2f_k + c_kg_2 - c_2g_k) \end{array} \right] \end{array} \right\}}{i = 1, \dots, 12} \\
& \frac{\left\{ \begin{array}{l} \text{Cos}(2\alpha_{0i\pm} - \Psi_{0i}) \left[\begin{array}{l} a_k(-c_2d_1 + c_1d_2 - b_2e_1 + b_1e_2) + \\ a_2(c_kd_1 - c_1d_k + b_ke_1 - b_1e_k) + \\ a_1(-c_kd_2 + c_2d_k - b_ke_2 + b_2e_k) \end{array} \right] + \\ \text{Cos}\Psi_{0i} \left[\begin{array}{l} a_k(c_2d_1 - c_1d_2 - b_2e_1 + b_1e_2) + \\ a_2(-c_kd_1 + c_1d_k + b_ke_1 - b_1e_k) + \\ a_1(c_kd_2 - c_2d_k - b_ke_2 + b_2e_k) \end{array} \right] + \\ \text{Sin}(2\alpha_{0i\pm} - \Psi_{0i}) \left[\begin{array}{l} a_k(-b_2d_1 + b_1d_2 + c_2e_1 - c_1e_2) + \\ a_2(b_kd_1 - b_1d_k - c_ke_1 + c_1e_k) + \\ a_1(-b_kd_2 + b_2d_k + c_ke_2 - c_2e_k) \end{array} \right] + \\ \text{Sin}\Psi_{0i} \left[\begin{array}{l} a_k(b_2d_1 - b_1d_2 + c_2e_1 - c_1e_2) + \\ a_2(-b_kd_1 + b_1d_k - c_ke_1 + c_1e_k) + \\ a_1(b_kd_2 - b_2d_k + c_ke_2 - c_2e_k) \end{array} \right] \end{array} \right\}}{k = 3/4/5}
\end{aligned} \tag{3.1.27}$$

$$\begin{aligned}
Z_{1i\pm} = & \frac{Z_{2i\pm} \{ \text{Sin}(\alpha_{0i\pm} - \Psi_{0i})(a_kd_1 - c_kd_k) + \text{Cos}(\alpha_{0i\pm} - \Psi_{0i})(a_ke_1 - a_1e_k) \} + \\ & \text{Sin}\Psi_{0i}(a_1f_k - a_kf_1) + \text{Cos}\Psi_{0i}(a_kg_1 - a_1g_k)}{\text{Cos}\alpha_{0i\pm}(a_1b_k - a_kb_1) + \text{Sin}\alpha_{0i\pm}(a_kc_1 - a_1c_k)} \quad \begin{array}{l} i = 1, \dots, 12 \\ k = 2/3/4/5 \end{array}
\end{aligned} \tag{3.1.28}$$

$$\begin{aligned}
Z_{3i\pm} = & \frac{Z_{1i\pm}(b_k \text{Cos}\alpha_{0i\pm} - c_k \text{Sin}\alpha_{0i\pm}) - \\ & Z_{2i\pm} [d_k \text{Sin}(\alpha_{0i\pm} - \Psi_{0i}) + e_k \text{Cos}(\alpha_{0i\pm} - \Psi_{0i})] + \\ & f_k \text{Sin}\Psi_{0i} - g_k \text{Cos}\Psi_{0i}}{a_k} \quad \begin{array}{l} i = 1, \dots, 12 \\ k = 1/2/3/4/5 \end{array}
\end{aligned}$$

(3.1.29)

$$\left. \begin{aligned} x_{3i\pm} &= Z_{2i\pm} x_6 \\ x_{1i\pm} &= \frac{x_{3i\pm}}{Z_{1i\pm}} \\ x_{2i\pm} &= \sqrt{x_{1i\pm}^2 + x_{3i\pm}^2 + x_6^2 - 2Z_{3i\pm} x_{1i\pm} x_6} \end{aligned} \right\} i = 1, 2, \dots, 12 \quad (3.1.30)$$

(3.1.30) ifadesindeki, Dört çubuk mekanizmasının gövde mafsalı noktaları arasındaki mesafe olan, x_6 uzunluğunun bir ölçek faktörü olması sebebiyle keyfi olarak seçilebilir.

Elde edilen $(x_{1i\pm}, x_{2i\pm}, x_{3i\pm}, \alpha_{0i\pm}, \psi_{0i}, i = 1, 2, \dots, 12)$ çözüm kümesinde yer alan gerçek çözümlerin istenilen fonksiyona yakınlığını belirleme amacıyla (3.1.31)'deki yapısal hata (e) tanımlanabilir.

$$e = \alpha_{th} - \alpha_{ac} \quad (3.1.31)$$

Burada, α_{th} (3.1.5) bağıntısından hesaplanan teorik piston yer değiştirmesini, α_{ac} ise dört çubuk mekanizmasının analizinden hesaplanan ve (3.1.32)'de verilen gerçek değeri göstermektedir.

$$\alpha_{ac\pm} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (3.1.32)$$

Burada:

$$\begin{aligned}
A &= x_1^2 - x_2^2 + x_3^2 - 2x_3x_6 + x_6^2 + (2x_1x_3 - 2x_1x_6) \cos(\psi_0 + \psi') \\
B &= -4x_1x_3 \sin(\psi_0 + \psi') \\
C &= x_1^2 - x_2^2 + x_3^2 + 2x_3x_6 + x_6^2 - (2x_1x_3 + 2x_1x_6) \cos(\psi_0 + \psi')
\end{aligned} \tag{3.1.33}$$

Ayrıca tasarlanan dört çubuk mekanizmasının istenen yörüngeye uygunluğunun belirlenmesi için (3.1.31) ifadesinde tanımlanan yapısal hatanın mekanizmanın çalışma aralığındaki dağılımına bakılması gereklidir. Öte yandan (3.1.32) ifadesinde yer alan $\alpha_{ac\pm}$ açısından sadece bir tanesi istenilen fonksiyona uygunluk gösterir.

3.1.2. Giriş Kolu Açısı (ψ) ile Biyel Kolu Açısı (δ) Fonksiyonel İlişkisini Üreten Dört Çubuk Tasarımı

Bu kısımda (ψ - δ) arasındaki fonksiyonel bağıntının verilmesi halinde, bu verilen fonksiyonu üreten dört çubuk mekanizmasının tasarımı yapılacaktır. Bunun için, (3.1.1) , (3.1.2) bağıntılarından çıkış kolu açısı α yok edilirse (3.1.34)'deki denklem takımına ulaşılır.

$$\begin{aligned}
a_i Z_3 + Z_1 (b_i \cos \delta_0 - c_i \sin \delta_0) - Z_2 (d_i \sin(\delta_0 - \psi_0) + e_i \cos(\delta_0 - \psi_0)) + \\
f_i \sin \psi_0 - g_i \cos \psi_0 = 0 \quad i = 1, 2, 3, 4, 5
\end{aligned} \tag{3.1.34}$$

Burada,

$$Z_1 = -\frac{x_2}{x_1}; \quad Z_2 = -\frac{x_2}{x_6}; \quad Z_3 = \frac{x_1^2 + x_2^2 - x_3^2 + x_6^2}{2x_1x_6} \tag{3.1.35}$$

$$\begin{aligned}
\psi &= \psi_0 + \psi'; & \psi' &= R_x (x - x_0); & R_x &= \frac{\Delta \psi}{\Delta x} \\
\delta &= \delta_0 + \delta'; & \delta' &= R_y (y - y_0); & R_y &= \frac{\Delta \delta}{\Delta y}
\end{aligned} \tag{3.1.36}$$

Burada, tasarımcı tarafından belirlenen, $(\Delta\psi = \psi_n - \psi_0)$, $(\Delta\delta = \delta_n - \delta_0)$, sırası ile, giriş kolu ve çıkış kolu çalışma aralığı, $(\Delta x = x_n - x_0)$, $(\Delta y = y_n - y_0)$ ise üretilen fonksiyonun bağımsız ve bağımlı değişkenlerinin değişim miktarlarıdır.

Ayrıca (3.1.3) denklem takımındaki üç ayrı metoda göre hesaplanabilen $(a_i, b_i, c_i, d_i, e_i, f_i, g_i)$ katsayılar (3.1.37)' de verilmiştir.

3.1.2.1. Kesin Nokta Metodu:

$$\left. \begin{aligned} a_i &= 1 \\ b_i &= \text{Cos}\delta'_i \\ c_i &= \text{Sin}\delta'_i \\ d_i &= \text{Sin}(\delta'_i - \psi'_i) \\ e_i &= \text{Cos}(\delta'_i - \psi'_i) \\ f_i &= \text{Sin}\psi'_i \\ g_i &= \text{Cos}\psi' \end{aligned} \right\} i = 1,2,3,4,5 \quad (3.1.37)$$

3.1.2.2. Altbölge Metodu:

$$\left. \begin{aligned} a_i &= \int_{x_{i-1}}^{x_i} dx \\ b_i &= \int_{x_{i-1}}^{x_i} \text{Cos}\delta' dx \\ c_i &= \int_{x_{i-1}}^{x_i} \text{Sin}\delta' dx \\ d_i &= \int_{x_{i-1}}^{x_i} \text{Sin}(\delta' - \psi') dx \\ e_i &= \int_{x_{i-1}}^{x_i} \text{Cos}(\delta' - \psi') dx \\ f_i &= \int_{x_{i-1}}^{x_i} \text{Sin}\psi' dx \\ g_i &= \int_{x_{i-1}}^{x_i} \text{Cos}\psi' dx \end{aligned} \right\} i = 1,2,3,4,5 \quad (3.1.38)$$

3.1.2.3. Galerkin Metodu:

$$\left. \begin{aligned}
 a_i &= \int_{x_0}^{x_n} w_i dx \\
 b_i &= \int_{x_0}^{x_n} \text{Cos} \delta' w_i dx \\
 c_i &= \int_{x_0}^{x_n} \text{Sin} \delta' w_i dx \\
 d_i &= \int_{x_0}^{x_n} \text{Sin}(\delta' - \psi') w_i dx \\
 e_i &= \int_{x_0}^{x_n} \text{Cos}(\delta' - \psi') w_i dx \\
 f_i &= \int_{x_0}^{x_n} \text{Sin} \psi' w_i dx \\
 g_i &= \int_{x_0}^{x_n} \text{Cos} \psi' w_i dx
 \end{aligned} \right\} i = 1,2,3,4,5$$

(3.1.39)

(3.1.37 - 3.1.39) katsayıları (3.1.11 - 3.1.14) de yerine konulduğunda (3.1.9) bağıntısı tekrar elde edilir. Burada değişen sadece (3.1.40)' da verilen (v) büyüklüğüdür.

$$v = \text{Tan} \frac{\delta_0}{2}; \quad t = \text{Tan} \frac{\Psi_0}{2}; \quad (3.1.40)$$

Benzer şekilde, (3.1.37-3.1.39) katsayıları (3.1.16-3.1.22) de yerine konulması suretiyle (3.1.15) ifadesindeki 12. dereceden polinom elde edilir. Bu polinomun kökleri yardımı ile giriş kolu açısı ψ_0 (3.1.41)'deki gibi yeniden hesaplanır.

$$\Psi_{oi} = 2 \arctan t_i; \quad i = 1,2,\dots,12 \quad (3.1.41)$$

(3.1.24-3.1.25) bağıntıları kullanılarak bulunan (v) değeri yardımı ile biyel kolu açısı δ_0 ,

$$\delta_{0i\pm} = \text{Arctan } v_{i\pm} \quad i=1,2,\dots,12 \quad (3.1.42)$$

olarak bulunur.

Fonksiyon üreten dört çubuğa ait diğer boyut (x_1, x_2, x_3) parametrelerinin belirlenmesi için gerekli (Z_1, Z_2, Z_3) büyüklükleri (3.1.43)'deki gibi hesaplanır.

$$Z_{2i\pm} = - \left\{ \begin{array}{l} \left. \begin{array}{l} \text{Cos}(\delta_{0i\pm} - \Psi_{0i}) \left[\begin{array}{l} a_k(-c_2f_1 + c_1f_2 - b_2g_1 + b_1g_2) + \\ a_2(c_kf_1 - c_1f_k + b_kg_1 - b_1g_k) + \\ a_1(-c_kf_2 + c_2f_k - b_kg_2 + b_2g_k) \end{array} \right] + \\ \text{Cos}(\delta_{0i\pm} + \Psi_{0i}) \left[\begin{array}{l} a_k(c_2f_1 - c_1f_2 - b_2g_1 + b_1g_2) + \\ a_2(-c_kf_1 + c_1f_k + b_kg_1 - b_1g_k) + \\ a_1(c_kf_2 - c_2f_k - b_kg_2 + b_2g_k) \end{array} \right] + \\ \text{Sin}(\delta_{0i\pm} - \Psi_{0i}) \left[\begin{array}{l} a_k(-b_2f_1 + b_1f_2 + c_2g_1 - c_1g_2) + \\ a_2(b_kf_1 - b_1f_k - c_kg_1 + c_1g_k) + \\ a_1(-b_kf_2 + b_2f_k + c_kg_2 - c_2g_k) \end{array} \right] + \\ \text{Sin}(\delta_{0i\pm} + \Psi_{0i}) \left[\begin{array}{l} a_k(b_2f_1 - b_1f_2 + c_2g_1 - c_1g_2) + \\ a_2(-b_kf_1 + b_1f_k - c_kg_1 + c_1g_k) + \\ a_1(b_kf_2 - b_2f_k + c_kg_2 - c_2g_k) \end{array} \right] \end{array} \right\} \quad i=1,\dots,12 \\ \left. \begin{array}{l} \text{Cos}(2\delta_{0i\pm} - \Psi_{0i}) \left[\begin{array}{l} a_k(-c_2d_1 + c_1d_2 - b_2e_1 + b_1e_2) + \\ a_2(c_kd_1 - c_1d_k + b_ke_1 - b_1e_k) + \\ a_1(-c_kd_2 + c_2d_k - b_ke_2 + b_2e_k) \end{array} \right] + \\ \text{Cos}\Psi_{0i} \left[\begin{array}{l} a_k(c_2d_1 - c_1d_2 - b_2e_1 + b_1e_2) + \\ a_2(-c_kd_1 + c_1d_k + b_ke_1 - b_1e_k) + \\ a_1(c_kd_2 - c_2d_k - b_ke_2 + b_2e_k) \end{array} \right] + \\ \text{Sin}(2\delta_{0i\pm} - \Psi_{0i}) \left[\begin{array}{l} a_k(-b_2d_1 + b_1d_2 + c_2e_1 - c_1e_2) + \\ a_2(b_kd_1 - b_1d_k - c_ke_1 + c_1e_k) + \\ a_1(-b_kd_2 + b_2d_k + c_ke_2 - c_2e_k) \end{array} \right] + \\ \text{Sin}\Psi_{0i} \left[\begin{array}{l} a_k(b_2d_1 - b_1d_2 + c_2e_1 - c_1e_2) + \\ a_2(-b_kd_1 + b_1d_k - c_ke_1 + c_1e_k) + \\ a_1(b_kd_2 - b_2d_k + c_ke_2 - c_2e_k) \end{array} \right] \end{array} \right\} \quad k=3/4/5 \end{array} \right.$$

$$(3.1.43)$$

$$Z_{2i\pm} \{ \text{Sin}(\delta_{0i\pm} - \psi_{0i})(a_k d_1 - c_1 d_k) + \text{Cos}(\delta_{0i\pm} - \psi_{0i})(a_k e_1 - a_1 e_k) \} +$$

$$Z_{1i\pm} = - \frac{\text{Sin}\psi_{0i}(a_1 f_k - a_k f_1) + \text{Cos}\psi_{0i}(a_k g_1 - a_1 g_k)}{\text{Cos}\delta_{0i\pm}(a_1 b_k - a_k b_1) + \text{Sin}\delta_{0i\pm}(a_k c_1 - a_1 c_k)} \quad \begin{matrix} i=1,..,12 \\ k=2/3/4/5 \end{matrix} \quad (3.1.44)$$

$$Z_{1i\pm}(b_k \text{Cos}\delta_{0i\pm} - c_k \text{Sin}\delta_{0i\pm}) -$$

$$Z_{2i\pm}[d_k \text{Sin}(\delta_{0i\pm} - \psi_{0i}) + e_k \text{Cos}(\delta_{0i\pm} - \psi_{0i})] +$$

$$Z_{3i\pm} = - \frac{f_k \text{Sin}\psi_{0i} - g_k \text{Cos}\psi_{0i}}{a_k} \quad \begin{matrix} i=1,..,12 \\ k=1/2/3/4/5 \end{matrix} \quad (3.1.45)$$

$$\left. \begin{aligned} x_{2i\pm} &= -Z_{2i\pm} x_6 \\ x_{1i\pm} &= \frac{x_{2i\pm}}{Z_{1i\pm}} \\ x_{3i\pm} &= \sqrt{x_{1i\pm}^2 + x_{2i\pm}^2 + x_6^2 - 2Z_{3i\pm} x_{1i\pm} x_6} \end{aligned} \right\} i=1,2,\dots,12 \quad (3.1.46)$$

(3.1.46) ifadesindeki, Dört çubuk mekanizmasının gövde mafsalı noktaları arasındaki mesafe olan x_6 uzunluğunun bir ölçek faktörü olması sebebiyle keyfi olarak seçilebilir.

Elde edilen $(x_{1i\pm}, x_{2i\pm}, x_{3i\pm}, \delta_{0i\pm}, \psi_{0i\pm}; i:1,2,\dots,12)$ çözüm kümesinde yer alan gerçek çözümlerin istenilen fonksiyona yakınlığını belirleme amacıyla (3.1.47)'deki gerçeğe yakın hata (e) tanımlanabilir.

$$e = \delta_{th} - \delta_{ac} \quad (3.1.47)$$

Burada, δ_{th} (3.36) bağıntısından hesaplanan teorik piston yer değiştirmesini, δ_{ac} ise dört çubuk mekanizmasının analizinden hesaplanan ve (3.1.48)'de verilen gerçeğe değeri göstermektedir.

$$\delta_{ac\pm} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (3.1.48)$$

Burada:

$$\begin{aligned} A &= x_1^2 + x_2^2 - x_3^2 + 2x_2x_6 + x_6^2 - (2x_1x_2 + 2x_1x_6) \cos(\psi_0 + \psi') \\ B &= 4x_1x_2 \sin(\psi_0 + \psi') \\ C &= x_1^2 + x_2^2 - x_3^2 - 2x_2x_6 + x_6^2 + (2x_1x_2 - 2x_1x_6) \cos(\psi_0 + \psi') \end{aligned} \quad (3.1.49)$$

Ayrıca tasarlanan dört çubuk mekanizmasının istenilen yörüngeye uygunluğunun belirlenmesi için (3.1.47) ifadesinde tanımlanan yapısal hatanın mekanizmanın çalışma aralığındaki dağılımına bakılması gereklidir. Öte yandan (3.1.47) ifadesinde yer alan $\delta_{ac\pm}$ açısından sadece bir tanesi istenen fonksiyona uygunluk gösterir.

3.1.3. Sayısal Örnekler

Yukarıdaki kapalı çözümler veren yaklaşımlar kullanılarak geliştirilen bilgisayar programı yardımı ile seçilen bazı tipik fonksiyonları üreten dört çubuk modülü tasarımları yapılmıştır. Elde edilen farklı çözümler arasındaki gerçek çözümler seçilip, bu çözümlere ait yapısal hata değerlerinin küçük olması ve boyut oranlarının imalata uygunlukları göz önüne alınmıştır. Bunun sonucunda elde edilen uygun çözümlere ait dört çubuk mekanizmalarının kinematik diyagramlar ve üç farklı metoda ait yapısal hata dağılımları gösterilmiştir.

3.1.3.1. Örnek 1

$Y = x^2$, $0 \leq x \leq 1$, fonksiyonunu üreten dört çubuk modülü tasarımı üç metoda bağlı olarak elde edilen kinematik boyutları çizelge 3.1. de, kinematik diyagramı Şekil 3.2. de ve yapısal hata dağılımları Şekil 3.3. de verilmiştir. Burada, çıkış kolu en fazla dönme miktarı $\Delta\alpha = 60.000$, giriş kolu dönü miktarı $\Delta\psi = 90.000$ ve gövde mafsalları arası mesafe $x_6 = 1.000000$ birim olarak seçilmiştir.

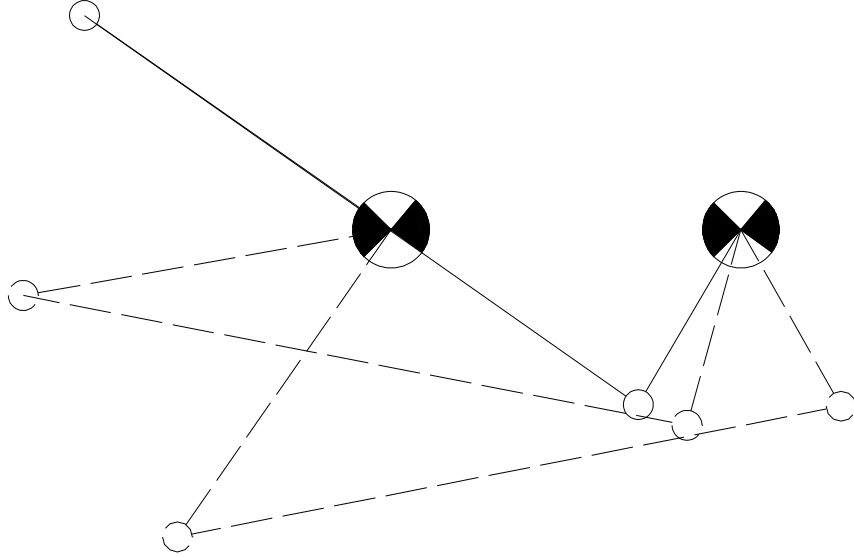
Çizelge 3.1. $Y = x^2$, $0 \leq x \leq 1$, Fonksiyonunu Üreten Dört Çubuk Modülü Kinematik Boyutları

Yöntem	x1	x2	x3	$\psi_0(0)$	$\alpha_0(0)$	$\delta_0(0)$	emax
Kesin Nokta	0.9915 34	1.844513	0.555468	146.4105	-121.5306	-33.6476	0.000902
Alt bölge	1.4264 88	2.248630	0.662497	140.1858	-125.0119	-40.3238	0.002325
Galerkin	1.0682 26	1.933302	0.579052	145.1251	-120.4289	-35.0391	0.000384

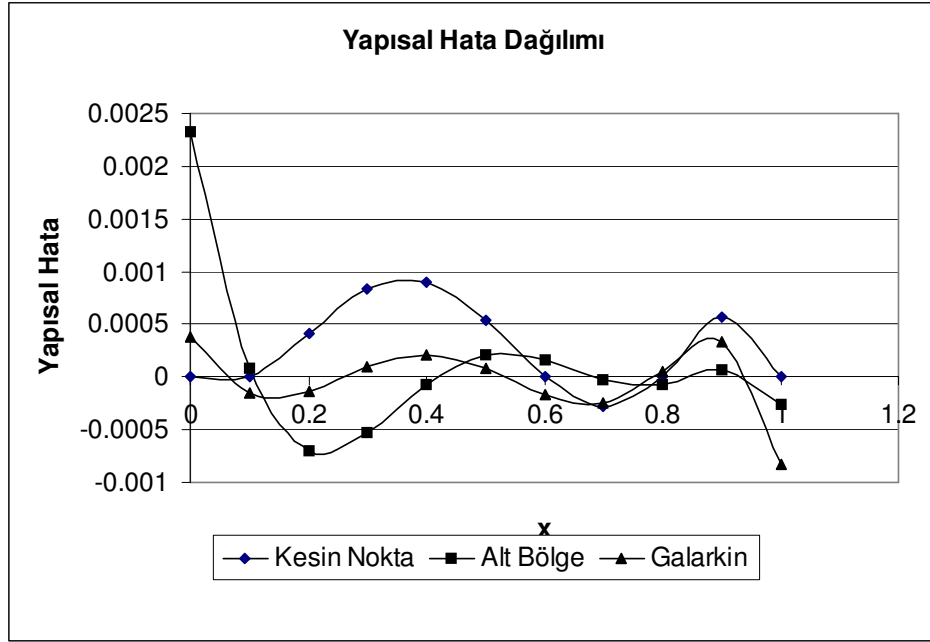
Kesin Noktalar: $(x_{pi}, i=1,2,3,4,5)$ 0.00, 0.10, 0.60, 0.80, 1.00

Alt Bölgeler: $(x_{si}, i=1,2,3,4,5,6)$ 0.00, 0.30, 0.60, 0.80, 0.90, 1.00

Ağırlık Fonksiyonları: $(w_i, i=1,2,3,4,5)$ $\sin x, \cos x, \sin 2x, \cos 2x, \sin x \cos x$



Şekil 3.2. Galerkin Metodu ile Tasarlanmış Dört Çubuk Modülü



Şekil 3.3. Yapısal Hata Dağılımı

3.1.3.2. Örnek 2.

Bu örnekte, Dört Çubuk mekanizmasının biyel kolunun açisal yer değiştirmesinin $y = x_2$, $0 \leq x \leq 1$, fonksiyonu olmaması koşulu ele alınmıştır. Buna göre elde edilen mekanizma boyutları üç metoda bağlı olarak çizelge 3.2. de, kinematik diyagramı Şekil 3.4. de ve yapısal hata dağılımları Şekil 3.5. de verilmiştir. Biyel kolunun en fazla dönü miktarı $\Delta\delta = 60.000$, giriş kolu dönü miktarı $\Delta\psi = 90.000$ ve çıkış kolu uzunluğu $x_6 = 1.00000$ olarak seçilmiştir.

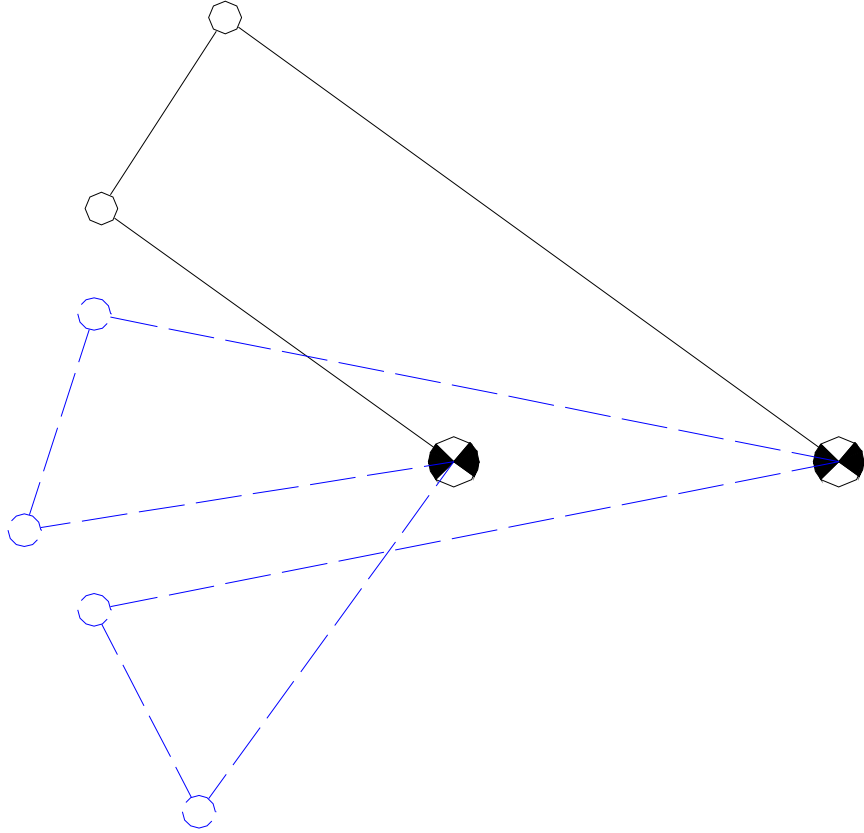
Çizelge 3.2. Biyel Kolu $y = x_2$, $0 \leq x \leq 1$ Fonksiyonuna Göre Hareket Eden Dört Çubuk Modülüne ait Kinematik Boyutlar

Yöntem	X_1	x_2	X_3	$\psi_0(0)$	$\delta_0(0)$	$\alpha_0(0)$	E_{\max} (rad)
Kesin Nokta	1.129624	0.593949	1.972273	144.1268	57.2464	143.9213	0.000622
Altbölge	1.130983	0.595540	1.986998	144.1448	58.5668	143.9124	0.000704
Galerkin	1.068226	0.579052	1.933302	145.1251	59.5710	144.9609	0.000835

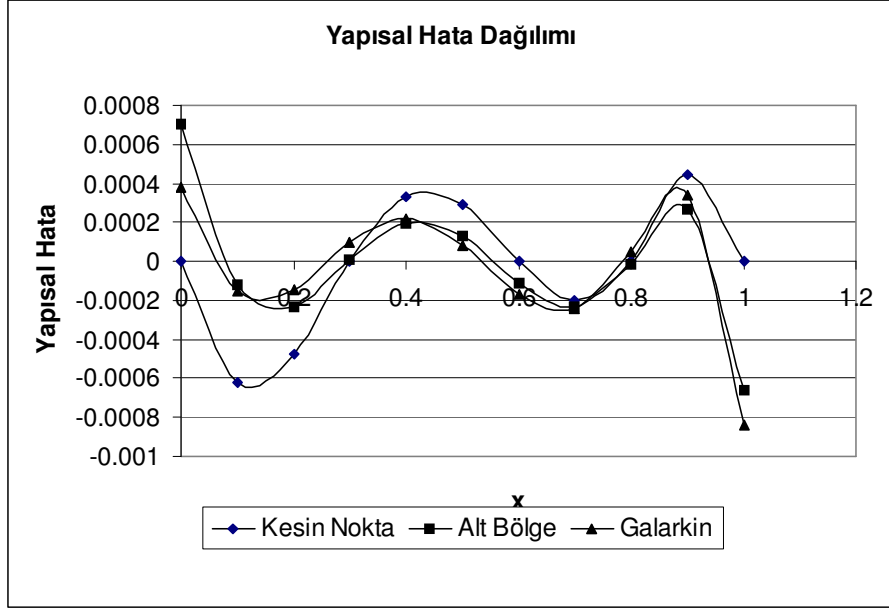
Kesin Noktalar: $(x_{pi}, i=1,2,3,4)$ 0.00, 0.30, 0.60, 0.80,1.00

Alt Bölgeler: $(x_{si}, i=1,2,3,4,5)$ 0.00, 0.20, 0.40, 0.70, 0.90, 1.00

Ağırlık Fonksiyonları: $(w_i, i=1,2,3,4)$ $\sin x, \cos x, \sin 2x, \cos 2x, \sin x \cos x$



Şekil 3.4. Kesin Nokta Metodu ile Tasarlanmış Dört Çubuk Modülü



Şekil 3.5. Yapısal Hata Dağılımı

3.2. FONKSİYON ÜRETEN KRANK-BİYEL MODÜLÜ

İstenen bir fonksiyonu üreten krank-biyel mekanizmasının tasarım parametreleri Şekil 3.6. da sunulmuştur. Burada, x_1 krank, x_2 biyel boyu, x_3 pistonun hareket doğrultusunun krank pim merkezine olan dik uzaklığı, ψ krank kolunun harekete başlama açısı ve s_0 da krank açısının başlangıç konumuna karşılık gelen piston pim merkezinin *AOI* doğrusuna dik uzaklığını simgelemektedir. Krank-Biyel mekanizmasının fonksiyon üretimi tasarımında iki farklı fonksiyonel ilişkiden söz edilebilir. Bunlar, giriş kolu dönüşü (ψ) ile çıkış uzvu olan pistonun (s) doğrusal yer değiştirmesi arasındaki fonksiyonel ilişki ile , giriş kolu dönüşü (ψ) ile biyel uzvunun yatayla yaptığı açı (δ) arasındaki fonksiyonel bağıntıdır.

3.2.1. (ψ - s) Fonksiyonel İlişisini Üreten Krank-Biyel Tasarımı

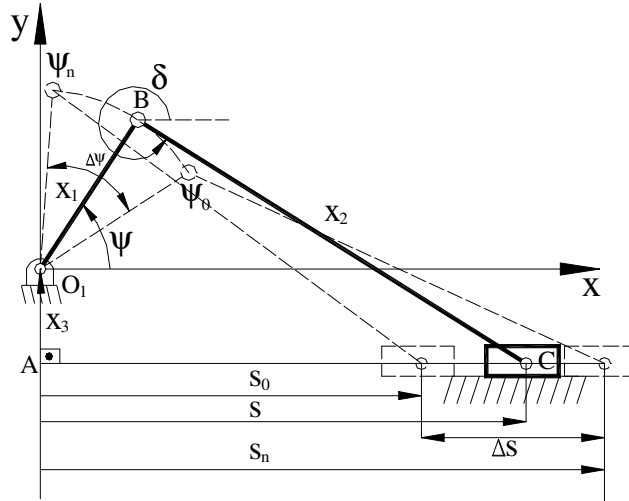
ψ - s fonksiyonel bağıntısı ele alınırsa, Şekil 3.6. dan *AOIBC* kapalı döngüsünden yazılan vektörel ifadenin kareleri alınıp taraf tarafa toplanması sonucunda biyel kolunun yatayla yaptığı (δ) açısı yok edildiğinde, (3.2.1)'deki hareket denklemi bulunur.

$$x_1^2 - 2sx_1 \cos \psi + 2x_1x_3 \sin \psi + x_3^2 + s^2 - x_2^2 = 0 \quad (3.2.1)$$

Krank-biyel mekanizmasının (ψ) açısı ile fonksiyon değişkeni (x) ve piston deplasmanı (s) ile fonksiyon bağımlı değişkeni (y) arasında (3.2.2)' deki doğrusal bağıntılar kurulabilir.

$$\begin{aligned} \psi &= \psi_0 + \psi'; & \psi' &= R_x(x - x_0); & R_x &= \frac{\Delta\psi}{\Delta x} \\ s &= s_0 + s'; & s' &= R_y(y - y_0); & R_y &= \frac{\Delta s}{\Delta y} \end{aligned} \quad (3.2.2)$$

Burada, tasarımcı tarafından belirlenen, ($\Delta\psi = \psi_n - \psi_0$), ($\Delta s = s_n - s_0$) sırası ile, krank kolu ve pistonun çalışma aralığı, ($\Delta x = x_n - x_0$), ($\Delta y = y_n - y_0$) ise üretilen fonksiyonun bağımsız ve bağımlı değişkenlerinin değişim miktarlarıdır.



Şekil 3.6. Fonksiyon Üreten Krank-Biyel Mekanizması Tasarım Parametreleri

(3.2.2) eşitlikleri (3.2.1) de yerine yazıldığında $G(s_0, \psi_0, x_1, x_2, x_3, x, y) = 0$ şeklindeki hareket denklemi;

$$2s'_0 + s'^2 + Z_3(\sin \psi_0 \cos \psi' + \cos \psi_0 \sin \psi') + Z_1(-s_0 \cos \psi_0 \cos \psi' + s_0 \sin \psi_0 \sin \psi' - s' \cos \psi_0 \cos \psi' + s' \sin \psi_0 \sin \psi') + Z_2 = 0 \quad (3.2.3)$$

Burada; $Z_1 = 2x_1$; $Z_2 = x_1^2 + x_3^2 + s_0^2 - x_2^2$; $Z_3 = 2x_1 x_3$ olarak tanımlanmıştır.

(3.2.3) denkleminde, $(s_0, \psi_0, x_1, x_2, x_3)$ bilinmeyen büyüklüklerin çözümü için, katsayıları Kesin Nokta, Altbölge ve Galerkin metotlarına göre hesaplanabilen beş bilinmeyenli doğrusal olmayan denklem kümesi (3.2.4)' deki gibidir.

$$(a_i s_0 + g_i) + Z_3(c_i \sin \psi_0 + b_i \cos \psi_0) + Z_1(-c_i s_0 \cos \psi_0 + b_i s_0 \sin \psi_0 - d_i \cos \psi_0 + e_i \sin \psi_0) + f_i Z_2 = 0 \quad i = 1, 2, 3, 4, 5 \quad (3.2.4)$$

$(a_i, b_i, c_i, d_i, e_i, f_i, g_i)$ katsayıları üç metoda göre (3.2.5)' deki gibi hesaplanır.

3.2.1.1. Kesin Nokta Metodu:

$$a_i = 2s'_i; \quad b_i = \sin \psi'_i; \quad c_i = \cos \psi'_i; \quad d_i = s' \cos \psi'_i; \\ e_i = s'_i \sin \psi'_i; \quad f_i = 1; \quad g_i = s_i'^2; \quad i = 1, 2, 3, 4, 5 \quad (3.2.5)$$

3.2.1.2. Altbölge Metodu:

$$a_i = 2 \int_{x_{i-1}}^{x_i} s' dx; \quad b_i = \int_{x_{i-1}}^{x_i} \sin \psi' dx; \quad c_i = \int_{x_{i-1}}^{x_i} \cos \psi' dx; \quad d_i = \int_{x_{i-1}}^{x_i} s' \cos \psi' dx; \quad e_i = \int_{x_{i-1}}^{x_i} s' \sin \psi' dx \\ f_i = \int_{x_{i-1}}^{x_i} dx; \quad g_i = \int_{x_{i-1}}^{x_i} s'^2 dx \quad i = 1, 2, 3, 4, 5 \quad (3.2.6)$$

3.2.1.3. Galerkin Metodu:

$$\begin{aligned} a_i &= 2 \int_{x_0}^{x_n} s' w_i(x) dx; & b_i &= \int_{x_0}^{x_n} \text{Sin} \psi' w_i(x) dx; & c_i &= \int_{x_0}^{x_n} \text{Cos} \psi' w_i(x) dx; & d_i &= \int_{x_0}^{x_n} s' \text{Cos} \psi' w_i(x) dx \\ e_i &= \int_{x_0}^{x_n} s' \text{Sin} \psi' w_i(x) dx; & f_i &= \int_{x_0}^{x_n} w_i(x) dx; & g_i &= \int_{x_0}^{x_n} s'^2 w_i(x) dx \quad i = 1, 2, 3, 4, 5 \end{aligned}$$

(3.2.7)

Kapalı çözümlere ulaşmak amacı ile, (3.2.4) denklem kümesinden sırası ile (Z_2, Z_3) ve (Z_1) yok edilip yeniden düzenlenirse, $(\tan \psi_0)$ ve (s_0) bilinmeyen parametrelerine bağlı doğrusal olmayan iki denklem elde edilir.

$$a_{pk} s_0^2 + (b_{pk} + c_{pk} \cos 2\psi_0 + d_{pk} \sin 2\psi_0) s_0 + e_{pk} + f_{pk} \cos 2\psi_0 + g_{pk} \sin 2\psi_0 = 0 \quad k = 1, 2$$

(3.2.8)

(3.2.8) ifadesindeki hesaplanabilir katsayılar aşağıda verilmiştir.

$$\begin{aligned}
a_{pi} = & 2 a_k b_3 c_2 f_1 - 2 a_3 b_k c_2 f_1 - 2 a_k b_2 c_3 f_1 + 2 a_2 b_k c_3 f_1 + 2 a_3 b_2 c_k f_1 - \\
& 2 a_2 b_3 c_k f_1 - 2 a_k b_3 c_1 f_2 + 2 a_3 b_k c_1 f_2 + 2 a_k b_1 c_3 f_2 - 2 a_1 b_k c_3 f_2 - \\
& 2 a_3 b_1 c_k f_2 + 2 a_1 b_3 c_k f_2 + 2 a_k b_2 c_1 f_3 - 2 a_2 b_k c_1 f_3 - 2 a_k b_1 c_2 f_3 + \\
& 2 a_1 b_k c_2 f_3 + 2 a_2 b_1 c_k f_3 - 2 a_1 b_2 c_k f_3 - 2 a_3 b_2 c_1 f_k + 2 a_2 b_3 c_1 f_k + \\
& 2 a_3 b_1 c_2 f_k - 2 a_1 b_3 c_2 f_k - 2 a_2 b_1 c_3 f_k + 2 a_1 b_2 c_3 f_k;
\end{aligned}$$

$$\begin{aligned}
b_{pi} = & a_k b_3 d_2 f_1 - a_3 b_k d_2 f_1 - a_k b_2 d_3 f_1 + a_2 b_k d_3 f_1 + a_3 b_2 d_k f_1 - a_2 b_3 d_k f_1 - \\
& a_k c_3 e_2 f_1 + a_3 c_k e_2 f_1 + a_k c_2 e_3 f_1 - a_2 c_k e_3 f_1 - a_3 c_2 e_k f_1 + a_2 c_3 e_k f_1 - \\
& a_k b_3 d_1 f_2 + a_3 b_k d_1 f_2 + a_k b_1 d_3 f_2 - a_1 b_k d_3 f_2 - a_3 b_1 d_k f_2 + a_1 b_3 d_k f_2 + \\
& a_k c_3 e_1 f_2 - a_3 c_k e_1 f_2 - a_k c_1 e_3 f_2 + a_1 c_k e_3 f_2 + a_3 c_1 e_k f_2 - a_1 c_3 e_k f_2 + \\
& a_k b_2 d_1 f_3 - a_2 b_k d_1 f_3 - a_k b_1 d_2 f_3 + a_1 b_k d_2 f_3 + a_2 b_1 d_k f_3 - a_1 b_2 d_k f_3 - \\
& a_k c_2 e_1 f_3 + a_2 c_k e_1 f_3 + a_k c_1 e_2 f_3 - a_1 c_k e_2 f_3 - a_2 c_1 e_k f_3 + a_1 c_2 e_k f_3 - \\
& a_3 b_2 d_1 f_k + a_2 b_3 d_1 f_k + a_3 b_1 d_2 f_k - a_1 b_3 d_2 f_k - a_2 b_1 d_3 f_k + a_1 b_2 d_3 f_k + \\
& a_3 c_2 e_1 f_k - a_2 c_3 e_1 f_k - a_3 c_1 e_2 f_k + a_1 c_3 e_2 f_k + a_2 c_1 e_3 f_k - a_1 c_2 e_3 f_k - \\
& 2 b_k c_3 f_2 g_1 + 2 b_3 c_k f_2 g_1 + 2 b_k c_2 f_3 g_1 - 2 b_2 c_k f_3 g_1 - 2 b_3 c_2 f_k g_1 + \\
& 2 b_2 c_3 f_k g_1 + 2 b_k c_3 f_1 g_2 - 2 b_3 c_k f_1 g_2 - 2 b_k c_1 f_3 g_2 + 2 b_1 c_k f_3 g_2 + \\
& 2 b_3 c_1 f_k g_2 - 2 b_1 c_3 f_k g_2 - 2 b_k c_2 f_1 g_3 + 2 b_2 c_k f_1 g_3 + 2 b_k c_1 f_2 g_3 - \\
& 2 b_1 c_k f_2 g_3 - 2 b_2 c_1 f_k g_3 + 2 b_1 c_2 f_k g_3 + 2 b_3 c_2 f_1 g_k - 2 b_2 c_3 f_1 g_k - \\
& 2 b_3 c_1 f_2 g_k + 2 b_1 c_3 f_2 g_k + 2 b_2 c_1 f_3 g_k - 2 b_1 c_2 f_3 g_k;
\end{aligned}$$

$$\begin{aligned}
c_{pi} = & a_k b_3 d_2 f_1 - a_3 b_k d_2 f_1 - a_k b_2 d_3 f_1 + a_2 b_k d_3 f_1 + a_3 b_2 d_k f_1 - a_2 b_3 d_k f_1 + \\
& a_k c_3 e_2 f_1 - a_3 c_k e_2 f_1 - a_k c_2 e_3 f_1 + a_2 c_k e_3 f_1 + a_3 c_2 e_k f_1 - a_2 c_3 e_k f_1 - \\
& a_k b_3 d_1 f_2 + a_3 b_k d_1 f_2 + a_k b_1 d_3 f_2 - a_1 b_k d_3 f_2 - a_3 b_1 d_k f_2 + a_1 b_3 d_k f_2 - \\
& a_k c_3 e_1 f_2 + a_3 c_k e_1 f_2 + a_k c_1 e_3 f_2 - a_1 c_k e_3 f_2 - a_3 c_1 e_k f_2 + a_1 c_3 e_k f_2 + \\
& a_k b_2 d_1 f_3 - a_2 b_k d_1 f_3 - a_k b_1 d_2 f_3 + a_1 b_k d_2 f_3 + a_2 b_1 d_k f_3 - a_1 b_2 d_k f_3 + \\
& a_k c_2 e_1 f_3 - a_2 c_k e_1 f_3 - a_k c_1 e_2 f_3 + a_1 c_k e_2 f_3 + a_2 c_1 e_k f_3 - a_1 c_2 e_k f_3 - \\
& a_3 b_2 d_1 f_k + a_2 b_3 d_1 f_k + a_3 b_1 d_2 f_k - a_1 b_3 d_2 f_k - a_2 b_1 d_3 f_k + a_1 b_2 d_3 f_k - \\
& a_3 c_2 e_1 f_k + a_2 c_3 e_1 f_k + a_3 c_1 e_2 f_k - a_1 c_3 e_2 f_k - a_2 c_1 e_3 f_k + a_1 c_2 e_3 f_k;
\end{aligned}$$

$$\begin{aligned}
d_{pi} = & a_k c_3 d_2 f_1 - a_3 c_k d_2 f_1 - a_k c_2 d_3 f_1 + a_2 c_k d_3 f_1 + a_3 c_2 d_k f_1 - a_2 c_3 d_k f_1 - \\
& a_k b_3 e_2 f_1 + a_3 b_k e_2 f_1 + a_k b_2 e_3 f_1 - a_2 b_k e_3 f_1 - a_3 b_2 e_k f_1 + a_2 b_3 e_k f_1 - \\
& a_k c_3 d_1 f_2 + a_3 c_k d_1 f_2 + a_k c_1 d_3 f_2 - a_1 c_k d_3 f_2 - a_3 c_1 d_k f_2 + a_1 c_3 d_k f_2 + \\
& a_k b_3 e_1 f_2 - a_3 b_k e_1 f_2 - a_k b_1 e_3 f_2 + a_1 b_k e_3 f_2 + a_3 b_1 e_k f_2 - a_1 b_3 e_k f_2 + \\
& a_k c_2 d_1 f_3 - a_2 c_k d_1 f_3 - a_k c_1 d_2 f_3 + a_1 c_k d_2 f_3 + a_2 c_1 d_k f_3 - a_1 c_2 d_k f_3 - \\
& a_k b_2 e_1 f_3 + a_2 b_k e_1 f_3 + a_k b_1 e_2 f_3 - a_1 b_k e_2 f_3 - a_2 b_1 e_k f_3 + a_1 b_2 e_k f_3 - \\
& a_3 c_2 d_1 f_k + a_2 c_3 d_1 f_k + a_3 c_1 d_2 f_k - a_1 c_3 d_2 f_k - a_2 c_1 d_3 f_k + a_1 c_2 d_3 f_k + \\
& a_3 b_2 e_1 f_k - a_2 b_3 e_1 f_k - a_3 b_1 e_2 f_k + a_1 b_3 e_2 f_k + a_2 b_1 e_3 f_k - a_1 b_2 e_3 f_k;
\end{aligned}$$

$$\begin{aligned}
e_{pi} = & - b_k d_3 f_2 g_1 + b_3 d_k f_2 g_1 + c_k e_3 f_2 g_1 - c_3 e_k f_2 g_1 + b_k d_2 f_3 g_1 - b_2 d_k f_3 g_1 - \\
& c_k e_2 f_3 g_1 + c_2 e_k f_3 g_1 - b_3 d_2 f_k g_1 + b_2 d_3 f_k g_1 + c_3 e_2 f_k g_1 - c_2 e_3 f_k g_1 + \\
& b_k d_3 f_1 g_2 - b_3 d_k f_1 g_2 - c_k e_3 f_1 g_2 + c_3 e_k f_1 g_2 - b_k d_1 f_3 g_2 + b_1 d_k f_3 g_2 + \\
& c_k e_1 f_3 g_2 - c_1 e_k f_3 g_2 + b_3 d_1 f_k g_2 - b_1 d_3 f_k g_2 - c_3 e_1 f_k g_2 + c_1 e_3 f_k g_2 - \\
& b_k d_2 f_1 g_3 + b_2 d_k f_1 g_3 + c_k e_2 f_1 g_3 - c_2 e_k f_1 g_3 + b_k d_1 f_2 g_3 - b_1 d_k f_2 g_3 - \\
& c_k e_1 f_2 g_3 + c_1 e_k f_2 g_3 - b_2 d_1 f_k g_3 + b_1 d_2 f_k g_3 + c_2 e_1 f_k g_3 - c_1 e_2 f_k g_3 + \\
& b_3 d_2 f_1 g_k - b_2 d_3 f_1 g_k - c_3 e_2 f_1 g_k + c_2 e_3 f_1 g_k - b_3 d_1 f_2 g_k + b_1 d_3 f_2 g_k + \\
& c_3 e_1 f_2 g_k - c_1 e_3 f_2 g_k + b_2 d_1 f_3 g_k - b_1 d_2 f_3 g_k - c_2 e_1 f_3 g_k + c_1 e_2 f_3 g_k;
\end{aligned}$$

$$\begin{aligned}
f_{pi} = & -b_k d_3 f_2 g_1 + b_3 d_k f_2 g_1 - c_k e_3 f_2 g_1 + c_3 e_k f_2 g_1 + b_k d_2 f_3 g_1 - b_2 d_k f_3 g_1 + \\
& c_k e_2 f_3 g_1 - c_2 e_k f_3 g_1 - b_3 d_2 f_k g_1 + b_2 d_3 f_k g_1 - c_3 e_2 f_k g_1 + c_2 e_3 f_k g_1 + \\
& b_k d_3 f_1 g_2 - b_3 d_k f_1 g_2 + c_k e_3 f_1 g_2 - c_3 e_k f_1 g_2 - b_k d_1 f_3 g_2 + b_1 d_k f_3 g_2 - \\
& c_k e_1 f_3 g_2 + c_1 e_k f_3 g_2 + b_3 d_1 f_k g_2 - b_1 d_3 f_k g_2 + c_3 e_1 f_k g_2 - c_1 e_3 f_k g_2 - \\
& b_k d_2 f_1 g_3 + b_2 d_k f_1 g_3 - c_k e_2 f_1 g_3 + c_2 e_k f_1 g_3 + b_k d_1 f_2 g_3 - b_1 d_k f_2 g_3 + \\
& c_k e_1 f_2 g_3 - c_1 e_k f_2 g_3 - b_2 d_1 f_k g_3 + b_1 d_2 f_k g_3 - c_2 e_1 f_k g_3 + c_1 e_2 f_k g_3 + \\
& b_3 d_2 f_1 g_k - b_2 d_3 f_1 g_k + c_3 e_2 f_1 g_k - c_2 e_3 f_1 g_k - b_3 d_1 f_2 g_k + b_1 d_3 f_2 g_k - \\
& c_3 e_1 f_2 g_k + c_1 e_3 f_2 g_k + b_2 d_1 f_3 g_k - b_1 d_2 f_3 g_k + c_2 e_1 f_3 g_k - c_1 e_2 f_3 g_k;
\end{aligned}$$

$$\begin{aligned}
g_{pi} = & -c_k d_3 f_2 g_1 + c_3 d_k f_2 g_1 + b_k e_3 f_2 g_1 - b_3 e_k f_2 g_1 + c_k d_2 f_3 g_1 - c_2 d_k f_3 g_1 - \\
& b_k e_2 f_3 g_1 + b_2 e_k f_3 g_1 - c_3 d_2 f_k g_1 + c_2 d_3 f_k g_1 + b_3 e_2 f_k g_1 - b_2 e_3 f_k g_1 + \\
& c_k d_3 f_1 g_2 - c_3 d_k f_1 g_2 - b_k e_3 f_1 g_2 + b_3 e_k f_1 g_2 - c_k d_1 f_3 g_2 + c_1 d_k f_3 g_2 + \\
& b_k e_1 f_3 g_2 - b_1 e_k f_3 g_2 + c_3 d_1 f_k g_2 - c_1 d_3 f_k g_2 - b_3 e_1 f_k g_2 + b_1 e_3 f_k g_2 - \\
& c_k d_2 f_1 g_3 + c_2 d_k f_1 g_3 + b_k e_2 f_1 g_3 - b_2 e_k f_1 g_3 + c_k d_1 f_2 g_3 - c_1 d_k f_2 g_3 - \\
& b_k e_1 f_2 g_3 + b_1 e_k f_2 g_3 - c_2 d_1 f_k g_3 + c_1 d_2 f_k g_3 + b_2 e_1 f_k g_3 - b_1 e_2 f_k g_3 + \\
& c_3 d_2 f_1 g_k - c_2 d_3 f_1 g_k - b_3 e_2 f_1 g_k + b_2 e_3 f_1 g_k - c_3 d_1 f_2 g_k + c_1 d_3 f_2 g_k + \\
& b_3 e_1 f_2 g_k - b_1 e_3 f_2 g_k + c_2 d_1 f_3 g_k - c_1 d_2 f_3 g_k - b_2 e_1 f_3 g_k + b_1 e_2 f_3 g_k;
\end{aligned}$$

(3.2.10)

(3.2.8) ifadesinden s_0 yok edilir ve ψ_0 açısı için yarım açı tanjant formülü kullanılırsa, (3.2.11)'deki 12. dereceden polinom bulunur.

$$w_0 + w_1 t + w_2 t^2 + w_3 t^3 + w_4 t^4 + w_5 t^5 + w_6 t^6 - w_5 t^7 + w_4 t^8 - w_3 t^9 + w_2 t^{10} - w_1 t^{11} + w_0 t^{12} = 0$$

(3.2.11)

Burada,

$$t = \tan \frac{\Psi_0}{2}$$

(3.2.12)

Yukarıdaki polinomun katsayıları aşağıda verilmiştir.

$$\begin{aligned}
w_0 = & -4 a_{p2} b_{p1} b_{p2} e_{p1} + 4 a_{p1} b_{p2}^2 e_{p1} - 4 a_{p2} b_{p2} c_{p1} e_{p1} - 4 a_{p2} b_{p1} c_{p2} e_{p1} + \\
& 8 a_{p1} b_{p2} c_{p2} e_{p1} - 4 a_{p2} c_{p1} c_{p2} e_{p1} + 4 a_{p1} c_{p2}^2 e_{p1} + 4 a_{p2}^2 e_{p1}^2 + 4 a_{p2} b_{p1}^2 e_{p2} - \\
& 4 a_{p1} b_{p1} b_{p2} e_{p2} + 8 a_{p2} b_{p1} c_{p1} e_{p2} - 4 a_{p1} b_{p2} c_{p1} e_{p2} + 4 a_{p2} c_{p1}^2 e_{p2} - \\
& 4 a_{p1} b_{p1} c_{p2} e_{p2} - 4 a_{p1} c_{p1} c_{p2} e_{p2} - 8 a_{p1} a_{p2} e_{p1} e_{p2} + 4 a_{p1}^2 e_{p2}^2 - 4 a_{p2} b_{p1} b_{p2} f_{p1} + \\
& 4 a_{p1} b_{p2}^2 f_{p1} - 4 a_{p2} b_{p2} c_{p1} f_{p1} - 4 a_{p2} b_{p1} c_{p2} f_{p1} + 8 a_{p1} b_{p2} c_{p2} f_{p1} - \\
& 4 a_{p2} c_{p1} c_{p2} f_{p1} + 4 a_{p1} c_{p2}^2 f_{p1} + 8 a_{p2}^2 e_{p1} f_{p1} - 8 a_{p1} a_{p2} e_{p2} f_{p1} + 4 a_{p2}^2 f_{p1}^2 + \\
& 4 a_{p2} b_{p1}^2 f_{p2} - 4 a_{p1} b_{p1} b_{p2} f_{p2} + 8 a_{p2} b_{p1} c_{p1} f_{p2} - 4 a_{p1} b_{p2} c_{p1} f_{p2} + 4 a_{p2} c_{p1}^2 f_{p2} - \\
& 4 a_{p1} b_{p1} c_{p2} f_{p2} - 4 a_{p1} c_{p1} c_{p2} f_{p2} - 8 a_{p1} a_{p2} e_{p1} f_{p2} + 8 a_{p1}^2 e_{p2} f_{p2} - \\
& 8 a_{p1} a_{p2} f_{p1} f_{p2} + 4 a_{p1}^2 f_{p2}^2
\end{aligned} \tag{3.2.13}$$

$$\begin{aligned}
w_1 = & -16 a_{p2} b_{p2} d_{p1} e_{p1} - 16 a_{p2} c_{p2} d_{p1} e_{p1} - 16 a_{p2} b_{p1} d_{p2} e_{p1} + 32 a_{p1} b_{p2} d_{p2} e_{p1} - \\
& 16 a_{p2} c_{p1} d_{p2} e_{p1} + 32 a_{p1} c_{p2} d_{p2} e_{p1} + 32 a_{p2} b_{p1} d_{p1} e_{p2} - 16 a_{p1} b_{p2} d_{p1} e_{p2} + \\
& 32 a_{p2} c_{p1} d_{p1} e_{p2} - 16 a_{p1} c_{p2} d_{p1} e_{p2} - 16 a_{p1} b_{p1} d_{p2} e_{p2} - 16 a_{p1} c_{p1} d_{p2} e_{p2} - \\
& 16 a_{p2} b_{p2} d_{p1} f_{p1} - 16 a_{p2} c_{p2} d_{p1} f_{p1} - 16 a_{p2} b_{p1} d_{p2} f_{p1} + 32 a_{p1} b_{p2} d_{p2} f_{p1} - \\
& 16 a_{p2} c_{p1} d_{p2} f_{p1} + 32 a_{p1} c_{p2} d_{p2} f_{p1} + 32 a_{p2} b_{p1} d_{p1} f_{p2} - 16 a_{p1} b_{p2} d_{p1} f_{p2} + \\
& 32 a_{p2} c_{p1} d_{p1} f_{p2} - 16 a_{p1} c_{p2} d_{p1} f_{p2} - 16 a_{p1} b_{p1} d_{p2} f_{p2} - 16 a_{p1} c_{p1} d_{p2} f_{p2} - \\
& 16 a_{p2} b_{p1} b_{p2} g_{p1} + 16 a_{p1} b_{p2}^2 g_{p1} - 16 a_{p2} b_{p2} c_{p1} g_{p1} - 16 a_{p2} b_{p1} c_{p2} g_{p1} + \\
& 32 a_{p1} b_{p2} c_{p2} g_{p1} - 16 a_{p2} c_{p1} c_{p2} g_{p1} + 16 a_{p1} c_{p2}^2 g_{p1} + 32 a_{p2}^2 e_{p1} g_{p1} - \\
& 32 a_{p1} a_{p2} e_{p2} g_{p1} + 32 a_{p2}^2 f_{p1} g_{p1} - 32 a_{p1} a_{p2} f_{p2} g_{p1} + 16 a_{p2} b_{p1}^2 g_{p2} - \\
& 16 a_{p1} b_{p1} b_{p2} g_{p2} + 32 a_{p2} b_{p1} c_{p1} g_{p2} - 16 a_{p1} b_{p2} c_{p1} g_{p2} + 16 a_{p2} c_{p1}^2 g_{p2} - \\
& 16 a_{p1} b_{p1} c_{p2} g_{p2} - 16 a_{p1} c_{p1} c_{p2} g_{p2} - 32 a_{p1} a_{p2} e_{p1} g_{p2} + 32 a_{p1}^2 e_{p2} g_{p2} - \\
& 32 a_{p1} a_{p2} f_{p1} g_{p2} + 32 a_{p1}^2 f_{p2} g_{p2}
\end{aligned} \tag{3.2.14}$$

$$\begin{aligned}
w_2 = & -24 a_{p2} b_{p1} b_{p2} e_{p1} + 24 a_{p1} b_{p2}^2 e_{p1} + 8 a_{p2} b_{p2} c_{p1} e_{p1} + 8 a_{p2} b_{p1} c_{p2} e_{p1} - \\
& 16 a_{p1} b_{p2} c_{p2} e_{p1} + 40 a_{p2} c_{p1} c_{p2} e_{p1} - 40 a_{p1} c_{p2}^2 e_{p1} - 64 a_{p2} d_{p1} d_{p2} e_{p1} + \\
& 64 a_{p1} d_{p2}^2 e_{p1} + 24 a_{p2}^2 e_{p1}^2 + 24 a_{p2} b_{p1}^2 e_{p2} - 24 a_{p1} b_{p1} b_{p2} e_{p2} - 16 a_{p2} b_{p1} c_{p1} e_{p2} + \\
& 8 a_{p1} b_{p2} c_{p1} e_{p2} - 40 a_{p2} c_{p1}^2 e_{p2} + 8 a_{p1} b_{p1} c_{p2} e_{p2} + 40 a_{p1} c_{p1} c_{p2} e_{p2} + \\
& 64 a_{p2} d_{p1}^2 e_{p2} - 64 a_{p1} d_{p1} d_{p2} e_{p2} - 48 a_{p1} a_{p2} e_{p1} e_{p2} + 24 a_{p1}^2 e_{p2}^2 + \\
& 8 a_{p2} b_{p1} b_{p2} f_{p1} - 8 a_{p1} b_{p2}^2 f_{p1} + 40 a_{p2} b_{p2} c_{p1} f_{p1} + 40 a_{p2} b_{p1} c_{p2} f_{p1} - \\
& 80 a_{p1} b_{p2} c_{p2} f_{p1} + 72 a_{p2} c_{p1} c_{p2} f_{p1} - 72 a_{p1} c_{p2}^2 f_{p1} - 64 a_{p2} d_{p1} d_{p2} f_{p1} + \\
& 64 a_{p1} d_{p2}^2 f_{p1} - 16 a_{p2}^2 e_{p1} f_{p1} + 16 a_{p1} a_{p2} e_{p2} f_{p1} - 40 a_{p2}^2 f_{p1}^2 - 8 a_{p2} b_{p1}^2 f_{p2} + \\
& 8 a_{p1} b_{p1} b_{p2} f_{p2} - 80 a_{p2} b_{p1} c_{p1} f_{p2} + 40 a_{p1} b_{p2} c_{p1} f_{p2} - 72 a_{p2} c_{p1}^2 f_{p2} + \\
& 40 a_{p1} b_{p1} c_{p2} f_{p2} + 72 a_{p1} c_{p1} c_{p2} f_{p2} + 64 a_{p2} d_{p1}^2 f_{p2} - 64 a_{p1} d_{p1} d_{p2} f_{p2} + \\
& 16 a_{p1} a_{p2} e_{p1} f_{p2} - 16 a_{p1}^2 e_{p2} f_{p2} + 80 a_{p1} a_{p2} f_{p1} f_{p2} - 40 a_{p1}^2 f_{p2}^2 - \\
& 64 a_{p2} b_{p2} d_{p1} g_{p1} - 64 a_{p2} c_{p2} d_{p1} g_{p1} - 64 a_{p2} b_{p1} d_{p2} g_{p1} + 128 a_{p1} b_{p2} d_{p2} g_{p1} - \\
& 64 a_{p2} c_{p1} d_{p2} g_{p1} + 128 a_{p1} c_{p2} d_{p2} g_{p1} + 64 a_{p2}^2 g_{p1}^2 + 128 a_{p2} b_{p1} d_{p1} g_{p2} - \\
& 64 a_{p1} b_{p2} d_{p1} g_{p2} + 128 a_{p2} c_{p1} d_{p1} g_{p2} - 64 a_{p1} c_{p2} d_{p1} g_{p2} - 64 a_{p1} b_{p1} d_{p2} g_{p2} - \\
& 64 a_{p1} c_{p1} d_{p2} g_{p2} - 128 a_{p1} a_{p2} g_{p1} g_{p2} + 64 a_{p1}^2 g_{p2}^2
\end{aligned} \tag{3.2.15}$$

$$\begin{aligned}
w_3 = & - 48 a_{p2} b_{p2} d_{p1} e_{p1} + 80 a_{p2} c_{p2} d_{p1} e_{p1} - 48 a_{p2} b_{p1} d_{p2} e_{p1} + 96 a_{p1} b_{p2} d_{p2} e_{p1} + \\
& 80 a_{p2} c_{p1} d_{p2} e_{p1} - 160 a_{p1} c_{p2} d_{p2} e_{p1} + 96 a_{p2} b_{p1} d_{p1} e_{p2} - 48 a_{p1} b_{p2} d_{p1} e_{p2} - \\
& 160 a_{p2} c_{p1} d_{p1} e_{p2} + 80 a_{p1} c_{p2} d_{p1} e_{p2} - 48 a_{p1} b_{p1} d_{p2} e_{p2} + 80 a_{p1} c_{p1} d_{p2} e_{p2} + \\
& 80 a_{p2} b_{p2} d_{p1} f_{p1} + 208 a_{p2} c_{p2} d_{p1} f_{p1} + 80 a_{p2} b_{p1} d_{p2} f_{p1} - 160 a_{p1} b_{p2} d_{p2} f_{p1} + \\
& 208 a_{p2} c_{p1} d_{p2} f_{p1} - 416 a_{p1} c_{p2} d_{p2} f_{p1} - 160 a_{p2} b_{p1} d_{p1} f_{p2} + 80 a_{p1} b_{p2} d_{p1} f_{p2} - \\
& 416 a_{p2} c_{p1} d_{p1} f_{p2} + 208 a_{p1} c_{p2} d_{p1} f_{p2} + 80 a_{p1} b_{p1} d_{p2} f_{p2} + 208 a_{p1} c_{p1} d_{p2} f_{p2} - \\
& 48 a_{p2} b_{p1} b_{p2} g_{p1} + 48 a_{p1} b_{p2}^2 g_{p1} + 80 a_{p2} b_{p2} c_{p1} g_{p1} + 80 a_{p2} b_{p1} c_{p2} g_{p1} - \\
& 160 a_{p1} b_{p2} c_{p2} g_{p1} + 208 a_{p2} c_{p1} c_{p2} g_{p1} - 208 a_{p1} c_{p2}^2 g_{p1} - 256 a_{p2} d_{p1} d_{p2} g_{p1} + \\
& 256 a_{p1} d_{p2}^2 g_{p1} + 96 a_{p2}^2 e_{p1} g_{p1} - 96 a_{p1} a_{p2} e_{p2} g_{p1} - 160 a_{p2}^2 f_{p1} g_{p1} + \\
& 160 a_{p1} a_{p2} f_{p2} g_{p1} + 48 a_{p2} b_{p1}^2 g_{p2} - 48 a_{p1} b_{p1} b_{p2} g_{p2} - 160 a_{p2} b_{p1} c_{p1} g_{p2} + \\
& 80 a_{p1} b_{p2} c_{p1} g_{p2} - 208 a_{p2} c_{p1}^2 g_{p2} + 80 a_{p1} b_{p1} c_{p2} g_{p2} + 208 a_{p1} c_{p1} c_{p2} g_{p2} + \\
& 256 a_{p2} d_{p1}^2 g_{p2} - 256 a_{p1} d_{p1} d_{p2} g_{p2} - 96 a_{p1} a_{p2} e_{p1} g_{p2} + 96 a_{p1}^2 e_{p2} g_{p2} + \\
& 160 a_{p1} a_{p2} f_{p1} g_{p2} - 160 a_{p1}^2 f_{p2} g_{p2}
\end{aligned}
\tag{3.2.16}$$

$$\begin{aligned}
w_4 = & - 60 a_{p2} b_{p1} b_{p2} e_{p1} + 60 a_{p1} b_{p2}^2 e_{p1} + 68 a_{p2} b_{p2} c_{p1} e_{p1} + 68 a_{p2} b_{p1} c_{p2} e_{p1} - \\
& 136 a_{p1} b_{p2} c_{p2} e_{p1} - 60 a_{p2} c_{p1} c_{p2} e_{p1} + 60 a_{p1} c_{p2}^2 e_{p1} + 60 a_{p2}^2 e_{p1}^2 + \\
& 60 a_{p2} b_{p1}^2 e_{p2} - 60 a_{p1} b_{p1} b_{p2} e_{p2} - 136 a_{p2} b_{p1} c_{p1} e_{p2} + 68 a_{p1} b_{p2} c_{p1} e_{p2} + \\
& 60 a_{p2} c_{p1}^2 e_{p2} + 68 a_{p1} b_{p1} c_{p2} e_{p2} - 60 a_{p1} c_{p1} c_{p2} e_{p2} - 120 a_{p1} a_{p2} e_{p1} e_{p2} + \\
& 60 a_{p1}^2 e_{p2}^2 + 68 a_{p2} b_{p1} b_{p2} f_{p1} - 68 a_{p1} b_{p2}^2 f_{p1} - 60 a_{p2} b_{p2} c_{p1} f_{p1} - \\
& 60 a_{p2} b_{p1} c_{p2} f_{p1} + 120 a_{p1} b_{p2} c_{p2} f_{p1} - 444 a_{p2} c_{p1} c_{p2} f_{p1} + 444 a_{p1} c_{p2}^2 f_{p1} + \\
& 512 a_{p2} d_{p1} d_{p2} f_{p1} - 512 a_{p1} d_{p2}^2 f_{p1} - 136 a_{p2}^2 e_{p1} f_{p1} + 136 a_{p1} a_{p2} e_{p2} f_{p1} + \\
& 60 a_{p2}^2 f_{p1}^2 - 68 a_{p2} b_{p1}^2 f_{p2} + 68 a_{p1} b_{p1} b_{p2} f_{p2} + 120 a_{p2} b_{p1} c_{p1} f_{p2} - \\
& 60 a_{p1} b_{p2} c_{p1} f_{p2} + 444 a_{p2} c_{p1}^2 f_{p2} - 60 a_{p1} b_{p1} c_{p2} f_{p2} - 444 a_{p1} c_{p1} c_{p2} f_{p2} - \\
& 512 a_{p2} d_{p1}^2 f_{p2} + 512 a_{p1} d_{p1} d_{p2} f_{p2} + 136 a_{p1} a_{p2} e_{p1} f_{p2} - 136 a_{p1}^2 e_{p2} f_{p2} - \\
& 120 a_{p1} a_{p2} f_{p1} f_{p2} + 60 a_{p1}^2 f_{p2}^2 + 512 a_{p2} c_{p2} d_{p1} g_{p1} + 512 a_{p2} c_{p1} d_{p2} g_{p1} - \\
& 1024 a_{p1} c_{p2} d_{p2} g_{p1} - 1024 a_{p2} c_{p1} d_{p1} g_{p2} + 512 a_{p1} c_{p2} d_{p1} g_{p2} + 512 a_{p1} c_{p1} d_{p2} g_{p2}
\end{aligned}
\tag{3.2.17}$$

$$\begin{aligned}
w_5 = & - 32 a_{p2} b_{p2} d_{p1} e_{p1} + 96 a_{p2} c_{p2} d_{p1} e_{p1} - 32 a_{p2} b_{p1} d_{p2} e_{p1} + 64 a_{p1} b_{p2} d_{p2} e_{p1} + \\
& 96 a_{p2} c_{p1} d_{p2} e_{p1} - 192 a_{p1} c_{p2} d_{p2} e_{p1} + 64 a_{p2} b_{p1} d_{p1} e_{p2} - 32 a_{p1} b_{p2} d_{p1} e_{p2} - \\
& 192 a_{p2} c_{p1} d_{p1} e_{p2} + 96 a_{p1} c_{p2} d_{p1} e_{p2} - 32 a_{p1} b_{p1} d_{p2} e_{p2} + 96 a_{p1} c_{p1} d_{p2} e_{p2} + \\
& 96 a_{p2} b_{p2} d_{p1} f_{p1} - 800 a_{p2} c_{p2} d_{p1} f_{p1} + 96 a_{p2} b_{p1} d_{p2} f_{p1} - 192 a_{p1} b_{p2} d_{p2} f_{p1} - \\
& 800 a_{p2} c_{p1} d_{p2} f_{p1} + 1600 a_{p1} c_{p2} d_{p2} f_{p1} - 192 a_{p2} b_{p1} d_{p1} f_{p2} + 96 a_{p1} b_{p2} d_{p1} f_{p2} + \\
& 1600 a_{p2} c_{p1} d_{p1} f_{p2} - 800 a_{p1} c_{p2} d_{p1} f_{p2} + 96 a_{p1} b_{p1} d_{p2} f_{p2} - 800 a_{p1} c_{p1} d_{p2} f_{p2} - \\
& 32 a_{p2} b_{p1} b_{p2} g_{p1} + 32 a_{p1} b_{p2}^2 g_{p1} + 96 a_{p2} b_{p2} c_{p1} g_{p1} + 96 a_{p2} b_{p1} c_{p2} g_{p1} - \\
& 192 a_{p1} b_{p2} c_{p2} g_{p1} - 800 a_{p2} c_{p1} c_{p2} g_{p1} + 800 a_{p1} c_{p2}^2 g_{p1} + 768 a_{p2} d_{p1} d_{p2} g_{p1} - \\
& 768 a_{p1} d_{p2}^2 g_{p1} + 64 a_{p2}^2 e_{p1} g_{p1} - 64 a_{p1} a_{p2} e_{p2} g_{p1} - 192 a_{p2}^2 f_{p1} g_{p1} + \\
& 192 a_{p1} a_{p2} f_{p2} g_{p1} + 32 a_{p2} b_{p1}^2 g_{p2} - 32 a_{p1} b_{p1} b_{p2} g_{p2} - 192 a_{p2} b_{p1} c_{p1} g_{p2} + \\
& 96 a_{p1} b_{p2} c_{p1} g_{p2} + 800 a_{p2} c_{p1}^2 g_{p2} + 96 a_{p1} b_{p1} c_{p2} g_{p2} - 800 a_{p1} c_{p1} c_{p2} g_{p2} - \\
& 768 a_{p2} d_{p1}^2 g_{p2} + 768 a_{p1} d_{p1} d_{p2} g_{p2} - 64 a_{p1} a_{p2} e_{p1} g_{p2} + 64 a_{p1}^2 e_{p2} g_{p2} + \\
& 192 a_{p1} a_{p2} f_{p1} g_{p2} - 192 a_{p1}^2 f_{p2} g_{p2}
\end{aligned}$$

(3.2.18)

$$\begin{aligned}
w_6 = & - 80 a_{p2} b_{p1} b_{p2} e_{p1} + 80 a_{p1} b_{p2}^2 e_{p1} + 112 a_{p2} b_{p2} c_{p1} e_{p1} + 112 a_{p2} b_{p1} c_{p2} e_{p1} - \\
& 224 a_{p1} b_{p2} c_{p2} e_{p1} - 208 a_{p2} c_{p1} c_{p2} e_{p1} + 208 a_{p1} c_{p2}^2 e_{p1} + 128 a_{p2} d_{p1} d_{p2} e_{p1} - \\
& 128 a_{p1} d_{p2}^2 e_{p1} + 80 a_{p2}^2 e_{p1}^2 + 80 a_{p2} b_{p1}^2 e_{p2} - 80 a_{p1} b_{p1} b_{p2} e_{p2} - 224 a_{p2} b_{p1} c_{p1} e_{p2} + \\
& 112 a_{p1} b_{p2} c_{p1} e_{p2} + 208 a_{p2} c_{p1}^2 e_{p2} + 112 a_{p1} b_{p1} c_{p2} e_{p2} - 208 a_{p1} c_{p1} c_{p2} e_{p2} - \\
& 128 a_{p2} d_{p1}^2 e_{p2} + 128 a_{p1} d_{p1} d_{p2} e_{p2} - 160 a_{p1} a_{p2} e_{p1} e_{p2} + 80 a_{p1}^2 e_{p2}^2 + \\
& 112 a_{p2} b_{p1} b_{p2} f_{p1} - 112 a_{p1} b_{p2}^2 f_{p1} - 208 a_{p2} b_{p2} c_{p1} f_{p1} - 208 a_{p2} b_{p1} c_{p2} f_{p1} + \\
& 416 a_{p1} b_{p2} c_{p2} f_{p1} + 1008 a_{p2} c_{p1} c_{p2} f_{p1} - 1008 a_{p1} c_{p2}^2 f_{p1} - 896 a_{p2} d_{p1} d_{p2} f_{p1} + \\
& 896 a_{p1} d_{p2}^2 f_{p1} - 224 a_{p2}^2 e_{p1} f_{p1} + 224 a_{p1} a_{p2} e_{p2} f_{p1} + 208 a_{p2}^2 f_{p1}^2 - \\
& 112 a_{p2} b_{p1}^2 f_{p2} + 112 a_{p1} b_{p1} b_{p2} f_{p2} + 416 a_{p2} b_{p1} c_{p1} f_{p2} - 208 a_{p1} b_{p2} c_{p1} f_{p2} - \\
& 1008 a_{p2} c_{p1}^2 f_{p2} - 208 a_{p1} b_{p1} c_{p2} f_{p2} + 1008 a_{p1} c_{p1} c_{p2} f_{p2} + 896 a_{p2} d_{p1}^2 f_{p2} - \\
& 896 a_{p1} d_{p1} d_{p2} f_{p2} + 224 a_{p1} a_{p2} e_{p1} f_{p2} - 224 a_{p1}^2 e_{p2} f_{p2} - 416 a_{p1} a_{p2} f_{p1} f_{p2} + \\
& 208 a_{p1}^2 f_{p2}^2 + 128 a_{p2} b_{p2} d_{p1} g_{p1} - 896 a_{p2} c_{p2} d_{p1} g_{p1} + 128 a_{p2} b_{p1} d_{p2} g_{p1} - \\
& 256 a_{p1} b_{p2} d_{p2} g_{p1} - 896 a_{p2} c_{p1} d_{p2} g_{p1} + 1792 a_{p1} c_{p2} d_{p2} g_{p1} - 128 a_{p2}^2 g_{p1}^2 - \\
& 256 a_{p2} b_{p1} d_{p1} g_{p2} + 128 a_{p1} b_{p2} d_{p1} g_{p2} + 1792 a_{p2} c_{p1} d_{p1} g_{p2} - 896 a_{p1} c_{p2} d_{p1} g_{p2} + \\
& 128 a_{p1} b_{p1} d_{p2} g_{p2} - 896 a_{p1} c_{p1} d_{p2} g_{p2} + 256 a_{p1} a_{p2} g_{p1} g_{p2} - 128 a_{p1}^2 g_{p2}^2
\end{aligned}$$

(3.2.19)

T 'ye bağı (3.2.11) nolu polinomun tüm köklerinin bulunması sonucunda ψ_0 açısına ait, gerçek veya sanal olmak üzere, 12 adet çözüm kümesi

$$\Psi_{oi} = 2 \arctan t_i; \quad i = 1, 2, \dots, 12 \quad (3.2.20)$$

olarak bulunur.

Fonksiyon üreten ikili makas modülünün diğer (s_0, x_1, x_2, x_3) kinematik büyüklüklerinin hesaplanması için önce (3.2.8) ifadesinden s_0^2 terimi yok edilirse, s_0 (3.2.21)' deki gibi çözülür.

$$s_{oi} = - \frac{a_{p1} e_{p2} - a_{p2} e_{p1} + (a_{p1} f_{p2} - a_{p2} f_{p1}) \cos 2\Psi_{oi} + (a_{p1} g_{p2} - a_{p2} g_{p1}) \sin 2\Psi_{oi}}{a_{p1} b_{p2} - a_{p2} b_{p1} + (a_{p1} c_{p2} - a_{p2} c_{p1}) \cos 2\Psi_{oi} + (a_{p1} d_{p2} - a_{p2} d_{p1}) \sin 2\Psi_{oi}}; i = 1, \dots, 12$$

(3.2.21)

Diğer üç adet boyut parametresi olan (x_1, x_2, x_3) ün hesaplanması içinde (3.2.22-23-24)' deki yol izlenir.

$$Z_{3i} = - \left. \begin{aligned} & \left\{ \begin{aligned} & 2\cos\Psi_{0i}(-d_k f_2 g_1 + d_2 f_k g_1 + d_k f_1 g_2 - d_1 f_k g_2 - d_2 f_1 g_k + d_1 f_2 g_k) + \\ & 2\sin\Psi_{0i}(e_k f_2 g_1 - e_2 f_k g_1 - e_k f_1 g_2 + e_1 f_k g_2 + e_2 f_1 g_k - e_1 f_2 g_k) + \\ & s_0(2\sin\Psi_{0i}(a_k e_2 f_1 - a_2 e_k f_1 - a_k e_1 f_2 + a_1 e_k f_2 + a_2 e_1 f_k - a_1 e_2 f_k + \\ & b_k f_2 g_1 - b_2 f_k g_1 - b_k f_1 g_2 + b_1 f_k g_2 + b_2 f_1 g_k - b_1 f_2 g_k) + \\ & 2\cos\Psi_{0i}(-a_k d_2 f_1 + a_2 d_k f_1 + a_k d_1 f_2 - a_1 d_k f_2 - a_2 d_1 f_k + a_1 d_2 f_k - \\ & c_k f_2 g_1 + c_2 f_k g_1 + c_k f_1 g_2 - c_1 f_k g_2 - c_2 f_1 g_k + c_1 f_2 g_k) + \\ & s_0^2(2\sin\Psi_{0i}(a_k b_2 f_1 - a_2 b_k f_1 - a_k b_1 f_2 + a_1 b_k f_2 + a_2 b_1 f_k - a_1 b_2 f_k) + \\ & 2\cos\Psi_{0i}(-a_k c_2 f_1 + a_2 c_k f_1 + a_k c_1 f_2 - a_1 c_k f_2 - a_2 c_1 f_k + a_1 c_2 f_k)) + \end{aligned} \right\} & i=1,\dots,12 \\ & \left. \begin{aligned} & - \left\{ \begin{aligned} & -b_k d_2 f_1 + b_2 d_k f_1 + c_k e_2 f_1 - c_2 e_k f_1 + b_k d_1 f_2 - b_1 d_k f_2 - c_k e_1 f_2 + \\ & c_1 e_k f_2 - b_2 d_1 f_k + b_1 d_2 f_k + c_2 e_1 f_k - c_1 e_2 f_k + \\ & \sin 2\Psi_{0i}(-c_k d_2 f_1 + c_2 d_k f_1 + b_k e_2 f_1 - b_2 e_k f_1 + c_k d_1 f_2 - c_1 d_k f_2 - b_k e_1 f_2 + \\ & b_1 e_k f_2 - c_2 d_1 f_k + c_1 d_2 f_k + b_2 e_1 f_k - b_1 e_2 f_k) + \\ & \cos 2\Psi_{0i}(-b_k d_2 f_1 + b_2 d_k f_1 - c_k e_2 f_1 + c_2 e_k f_1 + b_k d_1 f_2 - b_1 d_k f_2 + c_k e_1 f_2 - \\ & c_1 e_k f_2 - b_2 d_1 f_k + b_1 d_2 f_k - c_2 e_1 f_k + c_1 e_2 f_k) + \\ & 2s_{0i}(-b_k c_2 f_1 + b_2 c_k f_1 + b_k c_1 f_2 - b_1 c_k f_2 - b_2 c_1 f_k + b_1 c_2 f_k) \end{aligned} \right\} & k=3/4/5 \end{aligned} \right\} \end{aligned} \quad (3.2.22)$$

$$Z_{1i} = - \frac{\left\{ \begin{aligned} & f_1 g_k - f_k g_1 + s_{0i}(a_k f_1 - a_1 f_k) + Z_{3i} \cos\Psi_{0i}(b_k f_1 - b_1 f_k) + \\ & Z_{3i} \sin\Psi_{0i}(c_k f_1 - c_1 f_k) \end{aligned} \right\}}{\left\{ \begin{aligned} & \cos\Psi_{0i}(s_{0i}(c_1 f_k - c_k f_1) + d_1 f_k - d_k f_1) + \\ & \sin\Psi_{0i}(s_{0i}(b_k f_1 - b_1 f_k) + e_k f_1 - e_1 f_k) \end{aligned} \right\}} \quad \begin{aligned} & i=1,\dots,12 \\ & k=2/3/4/5 \end{aligned} \quad (3.2.23)$$

$$Z_{2i} = - \frac{\left\{ \begin{aligned} & (a_k s_{0i} + g_k) + Z_{3i}(c_k \sin\Psi_{0i} + b_k \cos\Psi_{0i}) + \\ & Z_{1i}(-c_k s_{0i} \cos\Psi_{0i} + b_k s_{0i} \sin\Psi_{0i} - d_k \cos\Psi_{0i} + e_k \sin\Psi_{0i}) \end{aligned} \right\}}{f_k} \quad \begin{aligned} & i=1,\dots,12 \\ & k=1/2/3/4/5 \end{aligned} \quad (3.2.24)$$

$$x_{1i} = \frac{Z_{1i}}{2} \quad i=1,2,\dots,12 \quad (3.2.25)$$

$$x_{3i} = \frac{Z_{3i}}{2x_{1i}} \quad i=1,2,\dots,12 \quad (3.2.26)$$

$$x_{2i} = \sqrt{x_{1i}^2 + x_{3i}^2 + s_{0i}^2 - Z_{2i}} \quad i = 1, 2, \dots, 12 \quad (3.2.27)$$

Elde edilen $(x_{1j}, x_{2j}, x_{3j}, s_{0j}, \psi_{0j} \quad j=1, 2, \dots, 12)$ çözüm kümesinde yer alan gerçek çözümlerin istenilen fonksiyonu hangi hassasiyette sağlandığını belirleme amacıyla (3.2.28)'de yapısal hata (e) tanımlanabilir.

$$e = s_{th} - s_{ac} \quad (3.2.28)$$

Burada, s_{th} (3.2.2) bağıntısından hesaplanan teorik piston yer değiştirmesini, s_{ac} (3.2.29)'daki bağıntıdan hesaplanan gerçek piston yer değiştirmesini simgelemektedir.

$$s_{ac\pm} = \frac{2x_1 \cos(\psi_0 + \psi') \pm \sqrt{(2x_1 \cos(\psi_0 + \psi'))^2 - 4(x_1^2 + x_3^2 - x_2^2 + 2x_1x_3 \sin(\psi_0 + \psi'))}}{2} \quad (3.2.29)$$

Ayrıca, tasarlanan Krank-Biyel mekanizmasının istenen yörüngeye uygunluğunun belirlenmesi için (3.2.28) ifadesinde tanımlanan yapısal hatanın mekanizmanın çalışma aralığındaki dağılımına bakılması gereklidir.

(3.2.29) ifadesinde yer alan $s_{ac\pm}$ açısından sadece bir tanesi istenilen fonksiyona uygunluk gösterir.

3.2.2. $(\psi-\delta)$ Fonksiyonel İlişisini Üreten Krank-Biyel Tasarımı

Bu kısımda $(\psi-\delta)$ arasındaki fonksiyonel bağıntının verilmesi halinde, bu verilen fonksiyonu üreten Krank-Biyel mekanizmasının tasarımı yapılacaktır. Bunun için, Şekil 3.6. da görülen mekanizmanın kapalı döngü vektörel ifadeleri aşağıdaki gibi yazılabilir.

$$x_1 \cos \psi + x_2 \cos \delta = s \quad (3.2.30)$$

$$x_1 \sin \psi + x_2 \sin \delta + x_3 = 0 \quad (3.2.31)$$

Krank-biyel mekanizmasının giriş kolunun yatayla yaptığı (ψ) açısı ile istenilen fonksiyonun bağımsız değişkeni (x) ve biyel kolunun yatayla yaptığı (δ) ile fonksiyon bağımlı değişkeni (y) arasında (3.2.32)' deki doğrusal bağıntılar kurulabilir.

$$\begin{aligned} \psi &= \psi_0 + \psi'; & \psi' &= R_x(x - x_0); & R_x &= \frac{\Delta\psi}{\Delta x} \\ \delta &= \delta_0 + \delta'; & \delta' &= R_y(y - y_0); & R_y &= \frac{\Delta\delta}{\Delta y} \end{aligned} \quad (3.2.32)$$

Verilen fonksiyonu üretecek Krank-Biyel mekanizmasının tasarımını yapabilmek için (3.2.31) nolu ifade (3.2.23)' deki gibi yazılabilir.

$$\begin{aligned} F(S_1, S_2, \psi_0, \delta_0) &= e_k S_1 + S_2(a_k \cos \delta_0 + b_k \sin \delta_0) + \\ (c_k \cos \psi_0 + d_k \sin \psi_0) &= 0 \quad i=1,2,3,4 \end{aligned} \quad (3.2.33)$$

Burada,

$$S_1 = \frac{x_3}{x_1}; \quad S_2 = \frac{x_2}{x_1} \quad (3.2.33)$$

(3.2.33) ifadesindeki hesaplanabilir katsayılar kullanılan metoda bağlı olarak (3.2.35)' de verilmiştir.

3.2.2.1. Kesin Nokta Metodu:

$$a_i = \sin \delta'_i; \quad b_i = \cos \delta'_i; \quad c_i = \sin \psi'_i; \quad d_i = \cos \psi'_i; \quad e_i = 1; i=1,2,3,4 \quad (3.2.34)$$

3.2.2.2. Altbölge Metodu:

$$a_i = \int_{x_{i-1}}^{x_i} \sin \delta' dx; \quad b_i = \int_{x_{i-1}}^{x_i} \cos \delta' dx; \quad c_i = \int_{x_{i-1}}^{x_i} \sin \psi' dx; \quad d_i = \int_{x_{i-1}}^{x_i} \cos \psi' dx; \quad e_i = \int_{x_{i-1}}^{x_i} dx; i=1,2,3,4 \quad (3.2.35)$$

3.2.2.3. Galerkin Metodu:

$$\begin{aligned}
 a_i &= \int_{x_0}^{x_n} \text{Sin} \delta' w_i(x) dx; & b_i &= \int_{x_0}^{x_n} \text{Cos} \delta' w_i(x) dx; & c_i &= \int_{x_0}^{x_n} \text{Sin} \psi' w_i(x) dx; \\
 d_i &= \int_{x_0}^{x_n} s' \text{Cos} \psi' w_i(x) dx; & e_i &= \int_{x_0}^{x_n} w_i(x) dx; & i &= 1, 2, 3, 4
 \end{aligned}
 \tag{3.2.36}$$

Kapalı çözümlere ulaşmak amacı ile, (3.2.33) denklem kümesinden sırası ile (S_1, S_2) yok edilip yeniden düzenlenirse, (ψ_0) ve (δ_0) bilinmeyen parametrelerine bağlı (3.2.37)' de doğrusal olmayan iki denklem elde edilir.

$$\begin{aligned}
 &\tan \delta_0 (\cos \psi_0 (-b_k c_2 e_1 + b_2 c_k e_1 + b_k c_1 e_2 - b_1 c_k e_2 - b_2 c_1 e_k + b_1 c_2 e_k) + \\
 &\sin \psi_0 (-b_k d_2 e_1 + b_2 d_k e_1 + b_k d_1 e_2 - b_1 d_k e_2 - b_2 d_1 e_k + b_1 d_2 e_k)) + \\
 &\cos \psi_0 (-a_k c_2 e_1 + a_2 c_k e_1 + a_k c_1 e_2 - a_1 c_k e_2 - a_2 c_1 e_k + a_1 c_2 e_k) + \\
 &\sin \psi_0 (-a_k d_2 e_1 + a_2 d_k e_1 + a_k d_1 e_2 - a_1 d_k e_2 - a_2 d_1 e_k + a_1 d_2 e_k) = 0 \quad k = 3, 4
 \end{aligned}
 \tag{3.2.37}$$

(3.2.37) ifadesinden $\tan \delta_0$ terimi yok edilirse,

$$w_2 t^2 + w_1 t + w_0 = 0
 \tag{3.2.38}$$

elde edilir.

Burada;

$$t = \tan \frac{\psi_0}{2} = 0
 \tag{3.2.39}$$

Burada, w_0, w_1, w_2 katsayıları:

$$\begin{aligned}
w_0 = & - 2 a_4 b_3 c_2^2 e_1^2 + 2 a_3 b_4 c_2^2 e_1^2 + 2 a_4 b_2 c_2 c_3 e_1^2 - 2 a_2 b_4 c_2 c_3 e_1^2 - 2 a_3 b_2 c_2 c_4 e_1^2 + \\
& 2 a_2 b_3 c_2 c_4 e_1^2 + 4 a_4 b_3 c_1 c_2 e_1 e_2 - 4 a_3 b_4 c_1 c_2 e_1 e_2 - 2 a_4 b_2 c_1 c_3 e_1 e_2 + \\
& 2 a_2 b_4 c_1 c_3 e_1 e_2 - 2 a_4 b_1 c_2 c_3 e_1 e_2 + 2 a_1 b_4 c_2 c_3 e_1 e_2 + 2 a_3 b_2 c_1 c_4 e_1 e_2 - \\
& 2 a_2 b_3 c_1 c_4 e_1 e_2 + 2 a_3 b_1 c_2 c_4 e_1 e_2 - 2 a_1 b_3 c_2 c_4 e_1 e_2 - 2 a_4 b_3 c_1^2 e_2^2 + \\
& 2 a_3 b_4 c_1^2 e_2^2 + 2 a_4 b_1 c_1 c_3 e_2^2 - 2 a_1 b_4 c_1 c_3 e_2^2 - 2 a_3 b_1 c_1 c_4 e_2^2 + 2 a_1 b_3 c_1 c_4 e_2^2 - \\
& 2 a_4 b_2 c_1 c_2 e_1 e_3 + 2 a_2 b_4 c_1 c_2 e_1 e_3 + 2 a_4 b_1 c_2^2 e_1 e_3 - 2 a_1 b_4 c_2^2 e_1 e_3 - \\
& 2 a_2 b_1 c_2 c_4 e_1 e_3 + 2 a_1 b_2 c_2 c_4 e_1 e_3 + 2 a_4 b_2 c_1^2 e_2 e_3 - 2 a_2 b_4 c_1^2 e_2 e_3 - \\
& 2 a_4 b_1 c_1 c_2 e_2 e_3 + 2 a_1 b_4 c_1 c_2 e_2 e_3 + 2 a_2 b_1 c_1 c_4 e_2 e_3 - 2 a_1 b_2 c_1 c_4 e_2 e_3 + \\
& 2 a_3 b_2 c_1 c_2 e_1 e_4 - 2 a_2 b_3 c_1 c_2 e_1 e_4 - 2 a_3 b_1 c_2^2 e_1 e_4 + 2 a_1 b_3 c_2^2 e_1 e_4 + \\
& 2 a_2 b_1 c_2 c_3 e_1 e_4 - 2 a_1 b_2 c_2 c_3 e_1 e_4 - 2 a_3 b_2 c_1^2 e_2 e_4 + 2 a_2 b_3 c_1^2 e_2 e_4 + \\
& 2 a_3 b_1 c_1 c_2 e_2 e_4 - 2 a_1 b_3 c_1 c_2 e_2 e_4 - 2 a_2 b_1 c_1 c_3 e_2 e_4 + 2 a_1 b_2 c_1 c_3 e_2 e_4
\end{aligned}$$

(3.2.40)

$$\begin{aligned}
w_1 = & - 4 a_4 b_3 c_2 d_2 e_1^2 + 4 a_3 b_4 c_2 d_2 e_1^2 + 2 a_4 b_2 c_3 d_2 e_1^2 - 2 a_2 b_4 c_3 d_2 e_1^2 - \\
& 2 a_3 b_2 c_4 d_2 e_1^2 + 2 a_2 b_3 c_4 d_2 e_1^2 + 2 a_4 b_2 c_2 d_3 e_1^2 - 2 a_2 b_4 c_2 d_3 e_1^2 - \\
& 2 a_3 b_2 c_2 d_4 e_1^2 + 2 a_2 b_3 c_2 d_4 e_1^2 + 4 a_4 b_3 c_2 d_1 e_1 e_2 - 4 a_3 b_4 c_2 d_1 e_1 e_2 - \\
& 2 a_4 b_2 c_3 d_1 e_1 e_2 + 2 a_2 b_4 c_3 d_1 e_1 e_2 + 2 a_3 b_2 c_4 d_1 e_1 e_2 - 2 a_2 b_3 c_4 d_1 e_1 e_2 + \\
& 4 a_4 b_3 c_1 d_2 e_1 e_2 - 4 a_3 b_4 c_1 d_2 e_1 e_2 - 2 a_4 b_1 c_3 d_2 e_1 e_2 + 2 a_1 b_4 c_3 d_2 e_1 e_2 + \\
& 2 a_3 b_1 c_4 d_2 e_1 e_2 - 2 a_1 b_3 c_4 d_2 e_1 e_2 - 2 a_4 b_2 c_1 d_3 e_1 e_2 + 2 a_2 b_4 c_1 d_3 e_1 e_2 - \\
& 2 a_4 b_1 c_2 d_3 e_1 e_2 + 2 a_1 b_4 c_2 d_3 e_1 e_2 + 2 a_3 b_2 c_1 d_4 e_1 e_2 - 2 a_2 b_3 c_1 d_4 e_1 e_2 + \\
& 2 a_3 b_1 c_2 d_4 e_1 e_2 - 2 a_1 b_3 c_2 d_4 e_1 e_2 - 4 a_4 b_3 c_1 d_1 e_2^2 + 4 a_3 b_4 c_1 d_1 e_2^2 + \\
& 2 a_4 b_1 c_3 d_1 e_2^2 - 2 a_1 b_4 c_3 d_1 e_2^2 - 2 a_3 b_1 c_4 d_1 e_2^2 + 2 a_1 b_3 c_4 d_1 e_2^2 + \\
& 2 a_4 b_1 c_1 d_3 e_2^2 - 2 a_1 b_4 c_1 d_3 e_2^2 - 2 a_3 b_1 c_1 d_4 e_2^2 + 2 a_1 b_3 c_1 d_4 e_2^2 - \\
& 2 a_4 b_2 c_2 d_1 e_1 e_3 + 2 a_2 b_4 c_2 d_1 e_1 e_3 - 2 a_4 b_2 c_1 d_2 e_1 e_3 + 2 a_2 b_4 c_1 d_2 e_1 e_3 + \\
& 4 a_4 b_1 c_2 d_2 e_1 e_3 - 4 a_1 b_4 c_2 d_2 e_1 e_3 - 2 a_2 b_1 c_4 d_2 e_1 e_3 + 2 a_1 b_2 c_4 d_2 e_1 e_3 - \\
& 2 a_2 b_1 c_2 d_4 e_1 e_3 + 2 a_1 b_2 c_2 d_4 e_1 e_3 + 4 a_4 b_2 c_1 d_1 e_2 e_3 - 4 a_2 b_4 c_1 d_1 e_2 e_3 - \\
& 2 a_4 b_1 c_2 d_1 e_2 e_3 + 2 a_1 b_4 c_2 d_1 e_2 e_3 + 2 a_2 b_1 c_4 d_1 e_2 e_3 - 2 a_1 b_2 c_4 d_1 e_2 e_3 - \\
& 2 a_4 b_1 c_1 d_2 e_2 e_3 + 2 a_1 b_4 c_1 d_2 e_2 e_3 + 2 a_2 b_1 c_1 d_4 e_2 e_3 - 2 a_1 b_2 c_1 d_4 e_2 e_3 + \\
& 2 a_3 b_2 c_2 d_1 e_1 e_4 - 2 a_2 b_3 c_2 d_1 e_1 e_4 + 2 a_3 b_2 c_1 d_2 e_1 e_4 - 2 a_2 b_3 c_1 d_2 e_1 e_4 - \\
& 4 a_3 b_1 c_2 d_2 e_1 e_4 + 4 a_1 b_3 c_2 d_2 e_1 e_4 + 2 a_2 b_1 c_3 d_2 e_1 e_4 - 2 a_1 b_2 c_3 d_2 e_1 e_4 + \\
& 2 a_2 b_1 c_2 d_3 e_1 e_4 - 2 a_1 b_2 c_2 d_3 e_1 e_4 - 4 a_3 b_2 c_1 d_1 e_2 e_4 + 4 a_2 b_3 c_1 d_1 e_2 e_4 + \\
& 2 a_3 b_1 c_2 d_1 e_2 e_4 - 2 a_1 b_3 c_2 d_1 e_2 e_4 - 2 a_2 b_1 c_3 d_1 e_2 e_4 + 2 a_1 b_2 c_3 d_1 e_2 e_4 + \\
& 2 a_3 b_1 c_1 d_2 e_2 e_4 - 2 a_1 b_3 c_1 d_2 e_2 e_4 - 2 a_2 b_1 c_1 d_3 e_2 e_4 + 2 a_1 b_2 c_1 d_3 e_2 e_4
\end{aligned}$$

(3.2.41)

$$\begin{aligned}
w_2 = & - 2 a_4 b_3 d_2^2 e_1^2 + 2 a_3 b_4 d_2^2 e_1^2 + 2 a_4 b_2 d_2 d_3 e_1^2 - 2 a_2 b_4 d_2 d_3 e_1^2 - 2 a_3 b_2 d_2 d_4 e_1^2 + \\
& 2 a_2 b_3 d_2 d_4 e_1^2 + 4 a_4 b_3 d_1 d_2 e_1 e_2 - 4 a_3 b_4 d_1 d_2 e_1 e_2 - 2 a_4 b_2 d_1 d_3 e_1 e_2 + \\
& 2 a_2 b_4 d_1 d_3 e_1 e_2 - 2 a_4 b_1 d_2 d_3 e_1 e_2 + 2 a_1 b_4 d_2 d_3 e_1 e_2 + 2 a_3 b_2 d_1 d_4 e_1 e_2 - \\
& 2 a_2 b_3 d_1 d_4 e_1 e_2 + 2 a_3 b_1 d_2 d_4 e_1 e_2 - 2 a_1 b_3 d_2 d_4 e_1 e_2 - 2 a_4 b_3 d_1^2 e_2^2 + \\
& 2 a_3 b_4 d_1^2 e_2^2 + 2 a_4 b_1 d_1 d_3 e_2^2 - 2 a_1 b_4 d_1 d_3 e_2^2 - 2 a_3 b_1 d_1 d_4 e_2^2 + 2 a_1 b_3 d_1 d_4 e_2^2 - \\
& 2 a_4 b_2 d_1 d_2 e_1 e_3 + 2 a_2 b_4 d_1 d_2 e_1 e_3 + 2 a_4 b_1 d_2^2 e_1 e_3 - 2 a_1 b_4 d_2^2 e_1 e_3 - \\
& 2 a_2 b_1 d_2 d_4 e_1 e_3 + 2 a_1 b_2 d_2 d_4 e_1 e_3 + 2 a_4 b_2 d_1^2 e_2 e_3 - 2 a_2 b_4 d_1^2 e_2 e_3 - \\
& 2 a_4 b_1 d_1 d_2 e_2 e_3 + 2 a_1 b_4 d_1 d_2 e_2 e_3 + 2 a_2 b_1 d_1 d_4 e_2 e_3 - 2 a_1 b_2 d_1 d_4 e_2 e_3 + \\
& 2 a_3 b_2 d_1 d_2 e_1 e_4 - 2 a_2 b_3 d_1 d_2 e_1 e_4 - 2 a_3 b_1 d_2^2 e_1 e_4 + 2 a_1 b_3 d_2^2 e_1 e_4 + \\
& 2 a_2 b_1 d_2 d_3 e_1 e_4 - 2 a_1 b_2 d_2 d_3 e_1 e_4 - 2 a_3 b_2 d_1^2 e_2 e_4 + 2 a_2 b_3 d_1^2 e_2 e_4 + \\
& 2 a_3 b_1 d_1 d_2 e_2 e_4 - 2 a_1 b_3 d_1 d_2 e_2 e_4 - 2 a_2 b_1 d_1 d_3 e_2 e_4 + 2 a_1 b_2 d_1 d_3 e_2 e_4
\end{aligned}$$

(3.2.42)

T 'ye bağı (3.2.38) polinomunun tüm köklerinin bulunması sonucunda ψ_0 açısına ait , gerçek veya sanal olmak üzere, çözüm kümesi

$$\psi_i = 2 \arctan t_i; \quad i=1,2 \quad (3.2.43)$$

olarak bulunur.

Fonksiyon üreten krank-biyel modülünün (δ_0, x_2, x_3) kinematik büyüklüklerinin hesaplanması amacıyla, ilk olarak (3.2.37) ifadesinden (δ_0) (3.2.44)'deki gibi hesaplanır.

$$\delta_{0i} = \tan^{-1} \frac{\begin{cases} \cos \Psi_{0i}(-a_k c_2 e_1 + a_2 c_k e_1 + a_k c_1 e_2 - a_1 c_k e_2 - a_2 c_1 e_k + a_1 c_2 e_k) + \\ \sin \Psi_{0i}(-a_k d_2 e_1 + a_2 d_k e_1 + a_k d_1 e_2 - a_1 d_k e_2 - a_2 d_1 e_k + a_1 d_2 e_k) \end{cases}}{\begin{cases} \cos \Psi_{0i}(-b_k c_2 e_1 + b_2 c_k e_1 + b_k c_1 e_2 - b_1 c_k e_2 - b_2 c_1 e_k + b_1 c_2 e_k) + \\ \sin \Psi_{0i}(-b_k d_2 e_1 + b_2 d_k e_1 + b_k d_1 e_2 - b_1 d_k e_2 - b_2 d_1 e_k + b_1 d_2 e_k) \end{cases}} \quad \begin{matrix} i=1,2 \\ k=3/4 \end{matrix} \quad (3.2.44)$$

İkinci olarak (3.2.33) ifadesinden (S_I) in yok edilmesi sonucunda (S_2) ,

$$S_{2i} = -\frac{\cos \Psi_{0i}(c_k e_1 - c_1 e_k) + \sin \Psi_{0i}(d_k e_1 - d_1 e_k)}{\cos \delta_{0i}(a_k e_1 - a_1 e_k) + \sin \delta_{0i}(b_k e_1 - b_1 e_k)} \quad \begin{matrix} i=1,2 \\ k=2/3/4 \end{matrix} \quad (3.2.45)$$

olarak bulunur.

S_I ise (3.2.46)'daki gibi hesaplanır.

$$S_{1i} = -\frac{S_{2i}(a_k \cos \delta_{0i} + b_k \sin \delta_{0i}) + (c_k \cos \Psi_{0i} + d_k \sin \Psi_{0i})}{e_k} \quad \begin{matrix} i=1,2 \\ k=1/2/3/4 \end{matrix} \quad (3.2.46)$$

Krank-Biyel mekanizmasının krank kolu uzunluğu (x_I) bir ölçek faktörü olduğundan keyfi seçilebilmesi nedeniyle, x_2 ve x_3 uzunlukları $(S_{Ij}, S_{2j}, j=1,2)$ yardımı ile (3.2.47)'deki gibi hesaplanır.

$$x_{2i} = S_{1i}x_1; \quad x_{3i} = S_{2i}x_1; \quad i=1,2 \quad (3.2.47)$$

Elde edilen çözüm kümesinin istenen fonksiyona olan yakınlığı için, mekanizmanın çalışma aralığı içerisinde hesaplanması gerekli olan (3.2.48)'deki yapısal hata (e) ile belirlenebilir.

$$e = \delta_{th} - \delta_{ac} \quad (3.2.48)$$

Burada δ_{th} (3.2.32) bağıntısından hesaplanan teorik değeri, δ_{ac} ise, (3.2.49)'da verildiği gibi, mekanizmanın konum analizinden elde edilen gerçek açı değerini simgelemektedir.

$$\delta_{ac\pm} = 2 \tan^{-1} \left\{ \frac{-2x_2 \pm \sqrt{4x_2^2 - 4(x_3 + x_1 \sin(\psi_0 + \psi'))^2}}{2(x_3 + x_1 \sin(\psi_0 + \psi'))} \right\} \quad (3.2.49)$$

(3.2.49) ifadesinde yer alan $\delta_{ac\pm}$ açısından sadece bir tanesi istenilen fonksiyona uygunluk gösterir.

3.2.3. Sayısal Örnekler

Yukarıdaki kapalı çözümler veren yaklaşımlar kullanılarak geliştirilen bilgisayar programı yardımı seçilen bazı tipik fonksiyonları üreten Krank-Biyel modülü tasarımları yapılmıştır. Elde edilen farklı çözümler arasındaki gerçek çözümlere ait , yapısal hata değerinin küçük olması ve boyut oranlarının imalata uygunlukları göz önüne alınarak , kinematik diyagramlar ve üç farklı metoda ait yapısal hata dağılımları çizilmiştir.

3.2.3.1. Örnek 1.

$Y = x_2$, $0 \leq x \leq 1$, fonksiyonunu üreten Krank-Biyel modülü tasarımı üç metoda bağlı olarak elde edilen kinematik boyutları çizelge 3.3 de, kinematik diyagramı Şekil 3.7. de ve yapısal hata dağılımları Şekil 3.8. de verilmiştir. Burada,

Pistonun en fazla yer deęiřtirme miktarı $\Delta s = -2.56$ ve giriř kolu dn miktarı $\Delta \psi = 90.000$ olarak seilmiřtir.

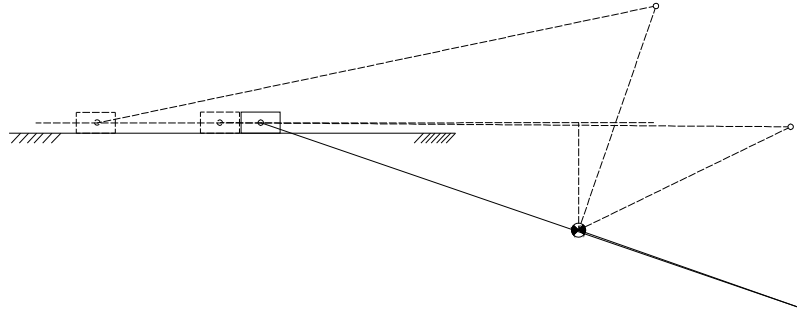
izelge 3.3. $Y = x^2$, $0 \leq x \leq 1$, Fonksiyonunu reten Krank-Biyel Modl Kinematik Boyutları

Yntem	X_1	X_2	X_3	$\psi_0(0)$	S_0	$\delta_0(0)$	E_{max}
Kesin Nokta	-1.166280	3.571874	2.454331	149.3090	4.052874	-31.3638	0.006614
Kesin Nokta	-3.818860	8.721568	-1.808737	157.9935	-4.556910	158.1943	0.004481
Altblge	-1.042497	5.976213	4.991932	136.1496	4.932921	-45.5982	0.003535
Altblge	-4.039391	9.029687	-2.340895	150.5643	-4.407518	151.3740	0.004690
Galerkin	-1.147591	3.039668	1.950732	153.3380	3.702830	-28.1865	0.010394
Galerkin	-3.681181	8.920099	-1.666460	160.7983	-4.965523	161.1828	0.001697

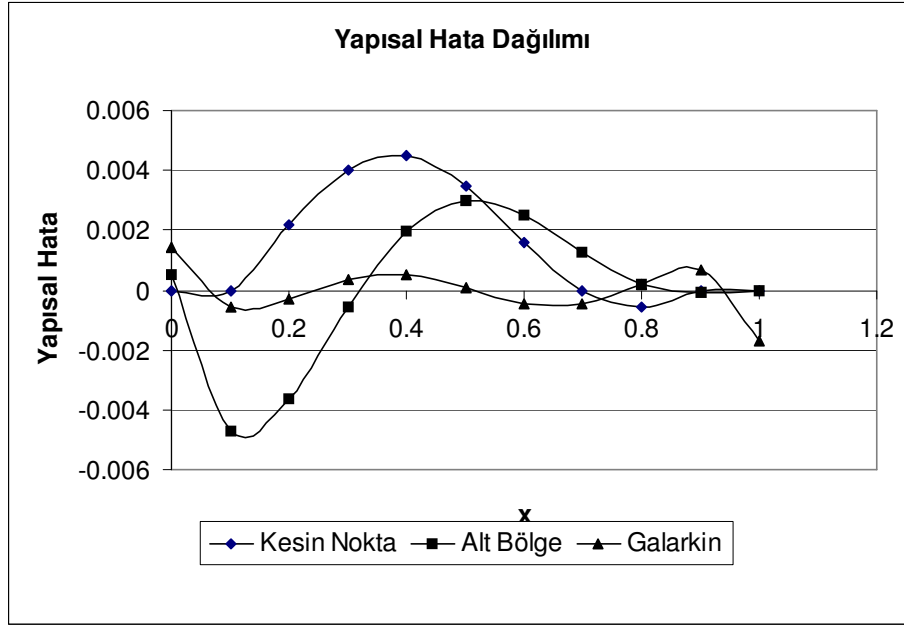
Kesin Noktalar: $(x_{pi}, i=1,2,3,4,5)$ 0.00, 0.10, 0.70, 1.90,1.00

Alt Blgeler: $(x_{si}, i=1,2,3,4,5,6)$ 0.00, 0.01, 0.80, 0.90,0.99, 1.00

Aęrlık Fonksiyonları: $(w_i, i=1,2,3,4,5)$ $\sin x, \cos x, \sin 2x, \cos 2x, \sin x \cos x$



řekil 3.7. Galerkin Metodu ile Tasarlanmıř Krank-Biyel Modl



Şekil 3.8. Yapısal Hata Dağılımı

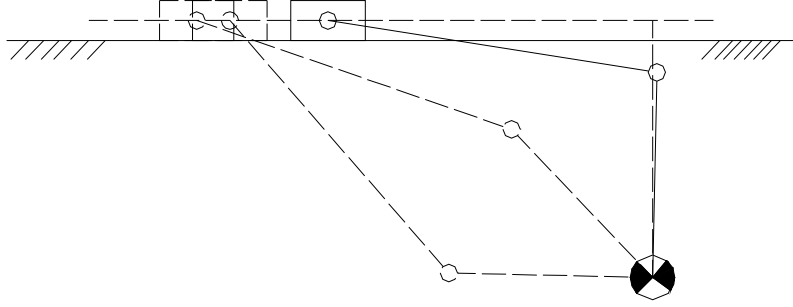
3.2.3.2. Örnek 2.

Bu örnekte, Krank-Biyel mekanizmasının biyel kolunun açısız yer değiştirmesinin $y = x^2$, $0 \leq x \leq 1$, fonksiyonu olmaması koşulu verilmiştir. Buna göre, elde edilen Krank-Biyel modülü boyutları üç metoda bağlı olarak çizelge 3.4. de, kinematik diyagramı Şekil 3.9. da ve yapısal hata dağılımları Şekil 3.10. da verilmiştir. Ayrıca, biyel kolunun en fazla dönü miktarı $\Delta\delta = -40.000$, giriş kolu dönü miktarı $\Delta\psi = 90.000$ ve keyfi olarak seçilebilen krank kolu uzunluğu $x_1 = 1.00$ olarak seçilmiştir.

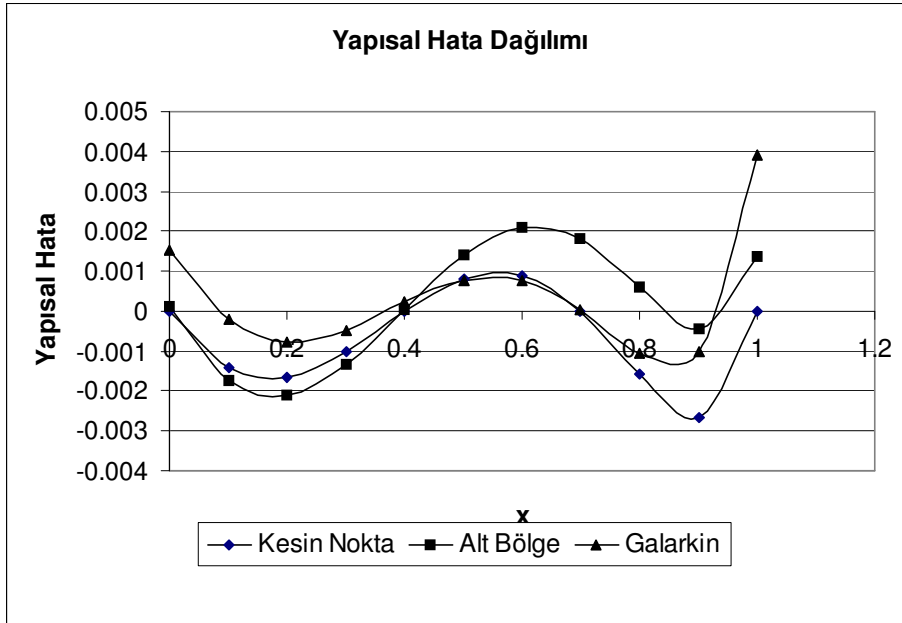
Çizelge 3.4. $Y = x^2$, $0 \leq x \leq 1$, Fonksiyonunu Üreten Krank-Biyel Modülü Kinematik Boyutları

Yöntem	X_1	X_2	X_3	$\psi_0(0)$	$\delta_0(0)$	S_0	$E_{max}(rad)$
Kesin Nokta	1.000000	- 1.633761	-1.252658	88.7720	-8.9046	-1.59260	0.001655
Altbölge	1.000000	- 1.593032	-1.193240	88.4418	-6.9732	-1.55400	0.001799
Galerkin	1.000000	- 1.644000	-1.288664	88.5951	- 10.0351	-1.59390	0.003914

Kesin Noktalar: $(x_{pi}, i=1,2,3,4)$ 0.00, 0.40, 0.70, 1.00
 Alt Bölgeler: $(x_{si}, i=1,2,3,4,5)$ 0.00, 0.01, 0.80, 0.90, 1.00
 Ağırlık Fonksiyonları: $(w_i, i=1,2,3,4)$ $\sin x, \cos x, \sin 2x, \cos 2x$



Şekil3.9. Kesin Nokta Metodu ile Tasarlanmış Krank-Biyel Modülü



Şekil 3.10. Yapısal Hata Dağılımı

4. MEK-GEN PROGRAM PAKETİ

4.1. NESNE YÖNELİMLİ PROGRAMLAMA

Nesne yönelimli programlama (NYP) (İngilizce - Object Oriented Programming (OOP)) özetle bir bilgisayar programlama yaklaşımıdır.

1960'lı yılların sonuna doğru ortaya çıkan bu yaklaşım, o dönemin yazılım dünyasında beliren bir bunalımın sonucudur. Yazılımların karmaşıklığı ve boyutları sürekli artıyor, ancak belli bir nitelik düzeyi korumak için gereken bakımın maliyeti (zaman ve çaba olarak) daha da hızlı artıyordu. NYP 'yi bu soruna karşı bir çözüm haline getiren ana özelliği yazılımda birimselliği (İngilizce - modularity) benimsemesidir.[25]

NYP 'nin altında yatan birimselliğin ana fikri, her bilgisayar programının etkileşim içerisinde olan birimler veya nesnelere kümesinden oluştuğu varsayımdır. Bu nesnelere her biri, kendi içerisinde veri işleyebilir ve diğer nesnelere ile çift yönlü veri alışverişinde bulunabilir. Halbuki NYP 'dan önce var olan tek yaklaşımda (Yordamsal programlama), programlar sadece bir komut dizisi veya birer işlev (fonksiyon) kümesi olarak görülmektedirler.[26]

Bu temele dayanan NYP 'nin, bilimsel çevreler tarafından geçmişe göre daha yüksek esneklik ve bakım kolaylığı sunduğu iddia edilmektedir, ve bu nedenle günümüzde geniş çaplı yazılım geliştirme tasarımlarında yaygın kullanılmaktadır.

Nesne tabanlı programlama dili, nesne kullanımını destekleyen ve sarma ilkesine uyan programlama dillerini tanımlar. Ancak nesne tabanlı programlama dilleri, kalıtım gibi nesne yönelimli programlama dillerine özel özellikleri taşımazlar. Bu durumda nesne yönelimli programlama dilleri nesne tabanlı programlama dillerinin altkümesini oluşturur.

Nesne yönelimli olan nesne tabanlı programlama dillerinden Simula, Smalltalk, C++, Object Pascal, Objective-C, Eiffel, Python, Java, C Sharp programlama dili, Visual Basic .NET sayabiliriz.

Çoğu modern programlama dilinde olduğu gibi Delphi de nesne yönelimli programlamayı (OOP: Object Oriented Programming) destekler. Bu tür diller sınıflar, kalıtım ve çok biçimlilik[(polimorphism)(geç tutturma)] olmak üzere üç temel kavrama dayanırlar. Eğer NYP(Nesne Yönelimli Programlama) 'yi en iyi şekilde kullanmak istiyorsak sürekli amacımız keşfetmek olmalıdır. Bunu da tüm nesnelerin özelliklerini ortaya çıkararak sağlayabiliriz. Satırlarca yazıp elde edebileceğimiz bir nesneyi, tek bir tuşla programa ekleyebiliriz.

4.2.PROGRAMLAMA DİLİ: DELPHI 6.0. ENTERPRISE EDITION

Mek-Gen programının kodlanmasında, Delphi 6.0 enterprise programlama dili kullanılmıştır. Bu editörün kullanılmasındaki en önemli etkenler; kullanıcıya windows ortamında nesneye yönelik (object oriented) görsel bir arayüzün sunulabilmesi ve programda kullanılan matematiksel ve fonksiyonel çıktıları grafik ortamda daha rahat görebilme imkanınıdır. Delphi 6.0, %100 Windows uyumlu bir uygulamadır ve dolayısıyla Mek-Gen , tüm Windows tabanlı (Windows 2000,2000 Server,Me,Xp,Vista...) işletim sistemlerinde çalışabilmektedir. Ayrıca Delphi'nin dünyanın en hızlı derleyicilerinden birine sahip olması, Object Oriented(nesne yönelimli) ve VCL (Visual Component Library) özellikleri sayesinde program içerisinde mevcut sınıflardan yeni bileşenler oluşturmayı mümkün kılması, dinamik bir hafıza yapısının olması program yazımında tercih edilmesinin diğer nedenleridir.

Delphi programlama dili, temeli Pascal dilidir. Özellikle nesne yönelimli programlama anlayışıyla yapılandırılmış Turbo Pascal dilinin görsel sürümü denebilir. Nesne, sınıf, kalıtım, fonksiyon aşırı yükleme(overloading) gibi temel NYP tekniklerini ve daha fazlasını içeren ve C++ den aşağı kalmayan güçlü ve esnek bir programlama dilidir. Borland tarafından geliştirilmektedir. Win32 ve .NET platformları üzerinde yazılım geliştirmeye olanak sağlar. GNU/Linux platformu üzerinde geliştirme imkanı sağlayan Kylix isimli bir sürümü de bulunmaktadır. [26]

Delphi programlama dili nesne yönelimli (Object Oriented) bir dil olduğu için eklenen bütün nesnelerin (Formlar da dahil) kodlarını oluşturur. Bu kodları "Unit" ler içerisinde barındırır.

Aşağıda örnek bir Delphi kaynak kodu görülmektedir. Kod parçacıklarının temeli bu şekildedir.

```
unit Unit1;  
interface  
uses
```

```

    Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
    Dialogs, StdCtrls;
type
TForm1 = class(TForm)
Button1: TButton;
procedure Button1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
Form1: TForm1;
implementation
    {$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
begin
    ShowMessage('Merhaba'); { programcının eklediği
satır}
end;
end.

```

Delphi 6, Mayıs 2001 de programcılara sunulmuştur. Delphi 6 ile birlikte gelen Modelmaker modelleme aracı ile UML-Modellerinden Delphi kodları oluşturulabilmektedir. Modelmaker nesne tabanlı sistemleri modelleyen standart bir dil olan Unified Modeling Language UML ile Diagram oluşturmayı sağlıyor. Böylece kod yazımı en aza indirgenmiştir.

Delphi 6 kurulumu, Mek-Gen programının delphide derlenmesi ve geliştirilmesi için gerekli bilgisayar donanım özellikleri minimum aşağıdaki gibi olmalıdır:

- Intel Pentium 166 MHz veya üstü (P2 400 MHz önerilir)
- Microsoft Windows 98, 2000, Me, NT 4.0 with Service Pack 5 yada sonraki sürümler
- 64 Mb RAM (128 Mb önerilir)
- 115 Mb boş alan (kısa kurulum)
- 350 Mb boş alan (tam kurulum)

4.3. MEK-GEN PROGRAMI

4.3.1. Giriş

Bu çalışmada, düzlemsel kol mekanizmalarından dört çubuk mekanizması ve Krank-biyel mekanizmasının fonksiyon sentezini gerçekleştiren etkileşimli bir bilgisayar program paketi geliştirilmiştir.

Mek-Gen (V.1.0.) bilgisayar program paketi ile, girilen tasarım değerleri ve hareket fonksiyonuna uygun mekanizma sentezi yapılarak, kullanıcıya mekanizma tasarımı, mekanizma uzuvlarına ait sayısal değerler, tasarımın görsel simülasyonu ve bu tasarıma bağlı hata analizleri sunulmaktadır. Programda geliştirilen statik ve dinamik grafikler, sentez çıktılarının kıyaslanmasında önem taşımaktadır.

Mek-Gen (V.1.0.) ile, Düzlemsel dört çubuk mekanizmalarının ve Krank-biyel mekanizmalarının fonksiyon sentezi; Alt Bölge Metodu, Galerkin Metodu ve Kesin Noktalar Metodu kullanılarak sağlanmaktadır.

Düzlemsel Dört Çubuk mekanizmalarının ve Krank-Biyel mekanizmalarının fonksiyon sentezinde Mek-Gen, kullanıcının kendi fonksiyonunu görsel arayüzde oluşturma imkanı sunmaktadır. Bu özellik, Mek-Gen 'i konu ile ilgili yazılmış diğer yazılımlardan ayırt eden en önemli özelliklerden biridir. Senteze girilen veriler ile sentez sonucu sağlanan hata sapmaları ve değerler programda kaydedilebildiği gibi, kaydedilmiş olan veriler çağrılıp yeniden sentez yapılabilir.

Girilen tasarım değerleri doğrultusunda oluşturulan mekanizma, her tasarım değeri için ayrı ayrı ekranda simule edilmekte, sağlanan grafiksel veriler de aynı ekranda görülebilmektedir. Simülasyon sırasında kullanılan uzuvların oranları, kullanılan açısız değerler ve fonksiyon değerleri sentez sonrası detaylı olarak görülebilmektedir. Ayrıca simülasyon ekranında büyütme-küçültme hızlı-yavaş geçişlerin kullanıcı tarafından değiştirilebilir yapıda olması ayrıntıların daha iyi görülebilmesini sağlamaktadır.

Gelişmiş bir grafik arayüzüne sahip olan Mek-Gen, kullanılacak tüm kinematik değerlerin programa çok kolay bir şekilde işlenebilmesi konusunda kullanıcıya kolaylık sağlamaktadır.

Mek-Gen Version1.0, Delphi 6.0 programlama dilinde geliştirilmiştir. Delphi 6.0 editörünün kullanımı ile, programa Object - Oriented (nesneye yönelik) ve tüm Windows işletim sistemlerinde çalıştırılabilir özellik sağlanmıştır. Sağlam bir omurgaya sahip olan Mek-Gen, daha sonraki çalışmalarda geliştirilebilir şekilde tasarlandığından, yeni modüller eklenerek farklı mekanizma tasarımlarında da kullanılabilir niteliktedir.

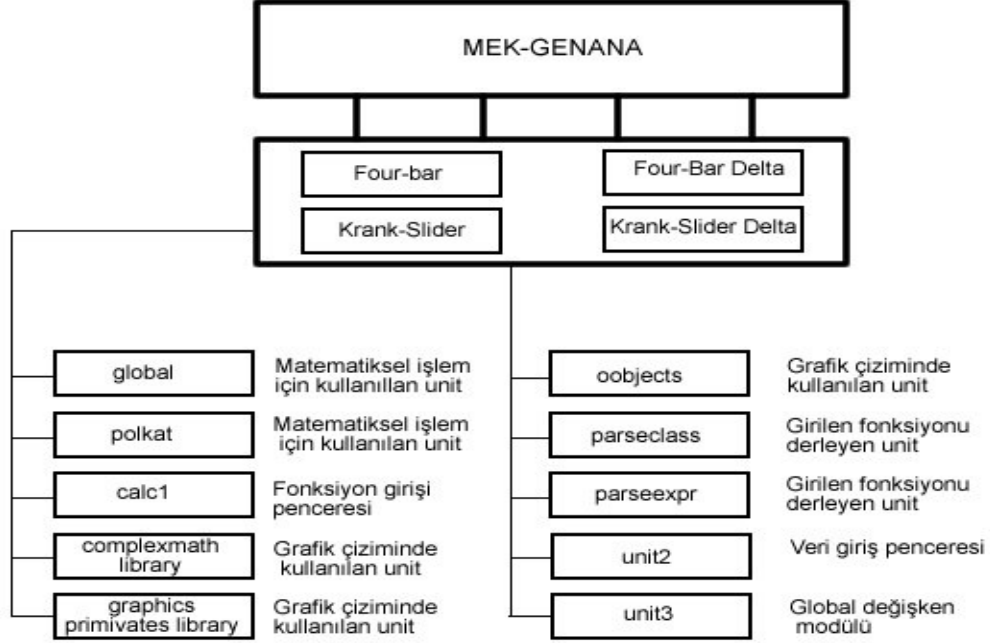
4.3.2. Mek-Gen Program Paketi İçeriği

Mek-Gen Version 1.0. Bilgisayar destekli Mekanizma Sentezi programı, *mekgen.exe* isimli exe dosyası olarak derlenmiştir. Program başka hiçbir dosyaya gerek kalmadan çalışabilmektedir. Herhangi bir “dll” kütüphanesi olmadan sadece tek dosya halinde taşınabilir ve kullanılabilir olması Mek-Gen programının bir diğer kolaylığıdır. Tüm Windows İşletim sistemlerinde çalışabilir bir yapıya sahiptir. Windows 95/98/NT/2000/2000 Server/ XP/ Vista işletim sistemlerinde %100 uyumlu olarak çalışabilmektedir. Mek-Gen programı işletimi sırasında bulunduğu dizine ‘in’ ve ‘out’ isimli 2 adet dosya açar. Bu dosyalar, programın sentez sonucu oluşturduğu giriş ve çıkış verilerini taşıyan dosyaları barındırır.

Mek-Gen Version 1.0 paket programı, Delphi 6.0 dilinde geliştirilmiştir. Sağlam bir omurga yapısına sahip olmasından dolayı, yeni Delphi versiyonlarına ufak değişiklikler vasıtasıyla adapte edilebilir. Ayrıca yine güçlü bir algoritma yapısının olması ve sağlam omurgası sayesinde kodlayıcı tarafından başka dillere de dönüştürülebilir geliştirilebilir.

4.3.3. Mek-Gen Kod Yapısı

Mek-Gen programı; “*Four-Bar*”, “*Four-Bar Delta*”, “*Krank-Slider*” ve “*Krank-Slider Delta*” olmak üzere toplam 4 alt programdan oluşmaktadır. 4 alt programın da kullanmış oldukları birim ve kütüphaneler aynı, fakat giriş değerleri ve birimler içerisinde derlenen kodlar farklılık göstermektedir.



Şekil 4.1. Mek-Gen ana programının kod yapısı

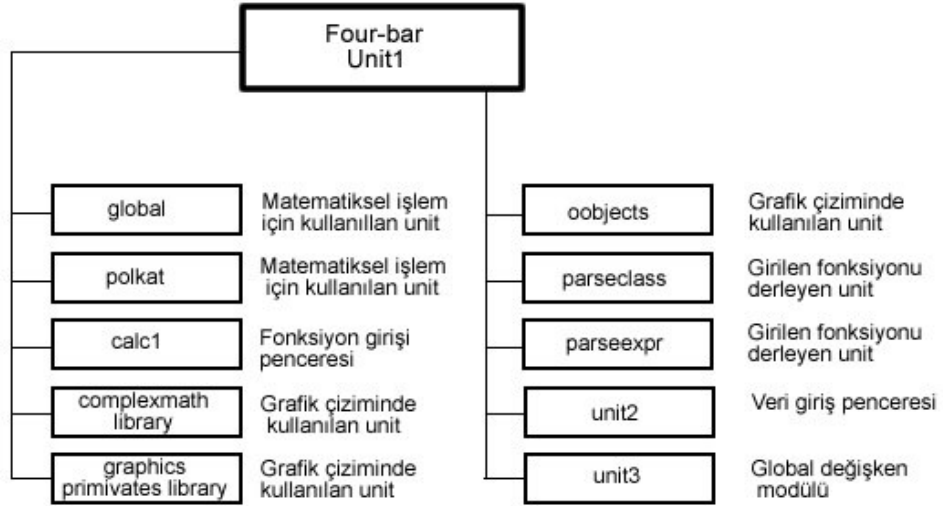
Mek-Gen programındaki her bir alt program toplam (*Four-Bar*, *Four-Bar Delta*, *Krank-Slider*, *Krank-Slider Delta*) 28 birimden oluşmaktadır. Bu birimlerden 8 tanesi programda kullanılan matematiksel, analitik ve vektörel fonksiyonları barındıran kütüphanelerdir.

Programların çalıştığı ana menüyü gösterir dosya ‘*Mekgenana*’ isimli delphi birim dosyasıdır. Bu birim dosyasında, fonksiyon sentezi yapılacak olan mekanizma tipi seçimi yapılmaktadır. Seçim menüsünden 4 alt program olan *Four-Bar*, *Four-Bar Delta*, *Krank-Slider* ve *Krank-Slider Delta* programlarından birisi seçilir. Seçim sonucunda ise çağırılan programın dosyaları işlenir.

Seçilen programa ait birimlerin mimarileri, işleyiş yapıları ve kullandıkları birimler aynıdır.

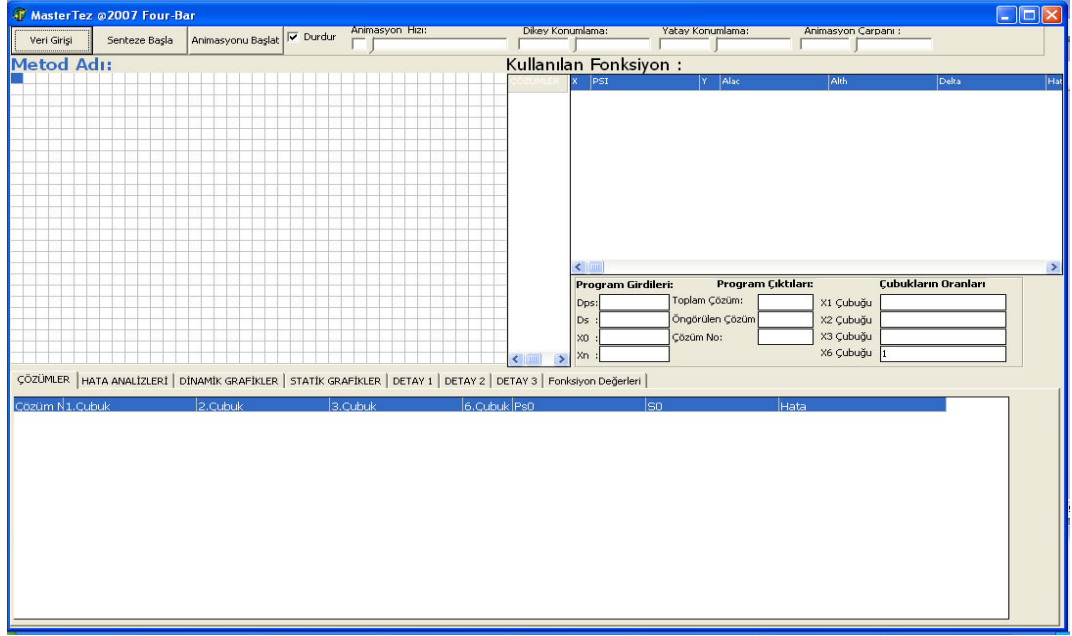
4.3.3.1. Four-Bar Programı Kod Yapısı

Four - Bar alt programı toplam 28 birim kullanarak çalışmaktadır. Bu birimlerden 8 tanesi programda kullanılan matematiksel, analitik ve vektörel fonksiyonları barındıran kütüphanelerdir. ‘Unit1’ isimli birim, programın ana modülü olup, diğer birimler bu modüle bağlıdır.



Şekil 4.2. Four –Bar Programının Kod Yapısı

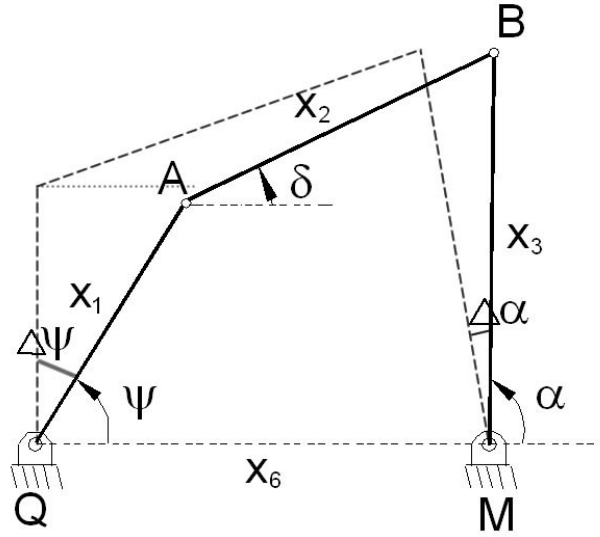
Şekilde de görüldüğü gibi, ‘Unit1’ isimli birim, Four - Bar programının ana birimi olup, diğer birimler bu birim tarafından çağırılmaktadır.



Şekil 4.3. Four Bar Programının Ana Görünüşü

Programın ana birimi, yaklaşık 2300 satır koddan oluşmaktadır. Toplamda kullanılan(delphi programının kendi birimleri hariç program için özellikle yazılan) birimler ve kütüphaneler de sayıldığında programı yaklaşık 9500 satır kod oluşturmaktadır.

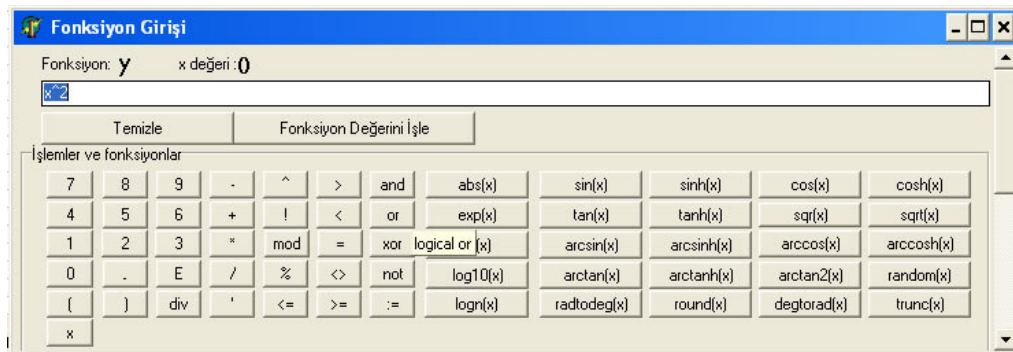
Bu alt program, mekanizmanın giriş kolu dönme miktarı $\Delta\psi$ ve çıkış kolu dönme miktarını $\Delta\alpha$ esas alarak sentez yapar.



Şekil 4.4. Dört Çubuk Mekanizması

Fonksiyon Birimleri (Global ve Polkat): Global ve Polkat birimlerinde tanımlı küçük fonksiyonlar, mekanizma hata analizlerinin bulunmasında kullanılmaktadır. Bu birimler sentezde kullanılan polinom katsayılarını bulmada ve çözüm üretmede, ayrıca diğer integral ve türev fonksiyonlarını çözümede kullanılır.

Fonksiyon Girişi Birimleri (Calc1 - Parseexpress ve Parseclass): Call birimi, programa istenen fonksiyon girişinin yapıldığı birimdir. Tasarımı ise Şekil 4.5’ deki gibidir.



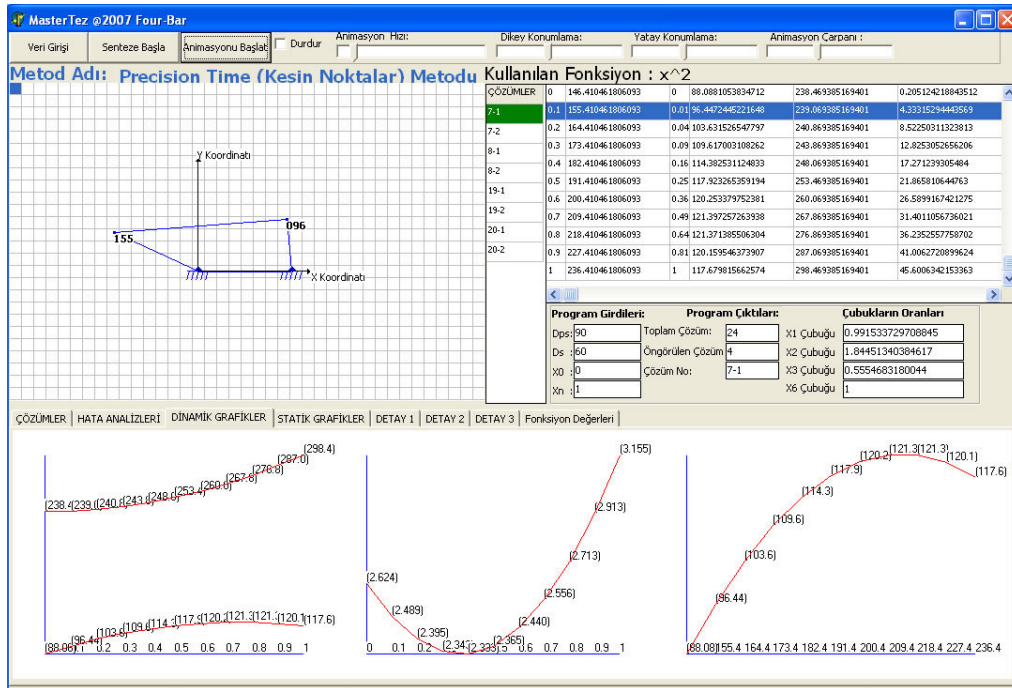
Şekil 4.5. Four-Bar Programı Veri Giriş Menüsü

Bu form sayesinde sentezi yapılacak olan fonksiyon değeri kullanıcı tarafından istenen şekilde girilebilmektedir. Kullanıcı fonksiyonu ister manuel yazabilmekte, isterse de tuşlar yardımı ile metin kutucuğuna işleyebilmektedir. $Y = f(x)$ tanımlı fonksiyonunda Y için kutucuktaki değeri ana programa göndermektedir.

Calc1 birimi, girilen kompleks fonksiyonu, bilgisayarın işlem yapabileceği formata çevirebilmek için *parseexpress* ve *parseclass* birimlerini kullanmaktadır. Derlenen girdi, global biriminde derlenerek, ana programda kullanılmak üzere unit1 birimine gönderilmektedir.

Grafik Birimleri (*Complexmathlibrary* , *Graphicsprimivateslibrary*):

Complexmathlibrary , *Graphicsprimivateslibrary* ve *objects* birimleri, ana programda sentez sonucu oluşan dinamik grafiklerin işlev kazanması için gerekli birimlerdir. Bu birimler, vektörel ve numerik fonksiyonlar içermekte olup, grafiklerde kullanılan *imagebox* nesnelерinin *canvas* özellikleri üzerinde işlem yapmaktadırlar. Ana birim tarafından kullanılmaktadırlar.



Şekil 4.6. Four-Bar Programının Çalışma Sentez Sırasındaki Görünüşü

Şeklin altında görülen dinamik grafikler mekanizmanın simülasyonuna göre çizim yapmaktadır.

Global Değişkenler Birimi (Unit 2): Diğer bir kullanılan birim ise, program içerisinde etkileşimi sağlayan global değişkenlerin tanımlandığı *unit2* modülüdür. Program içerisinde kullanılan değişkenler bu modülde tanımlanmıştır. Verilerin sağlıklı bir şekilde bir birimden, diğer bir birime taşınması için, veriyi taşıyıcı değişkenlerin global olarak tanımlanması gerekmektedir. Böylece her birim, bu değişkenleri kendi içerisinde tanımlanmamış olsa bile tanır ve işlem yapar.

Veri giriş birimi (Unit3): Bu birim, sentez programına tüm verilerin giriş yapıldığı formu oluşturur. Bu formda sentez yapılacak fonksiyon değerini giriş yapmak için *calc* birimi çağrılır. Buradan global değişkenle gelen fonksiyon, form içerisinde işlem yapılana kadar barındırılır. Bu form sayesinde programa girilecek tüm tasarım değerleri kullanıcı tarafından tanımlanabilmektedir. Şekilde form detaylı bir şekilde görülmektedir.

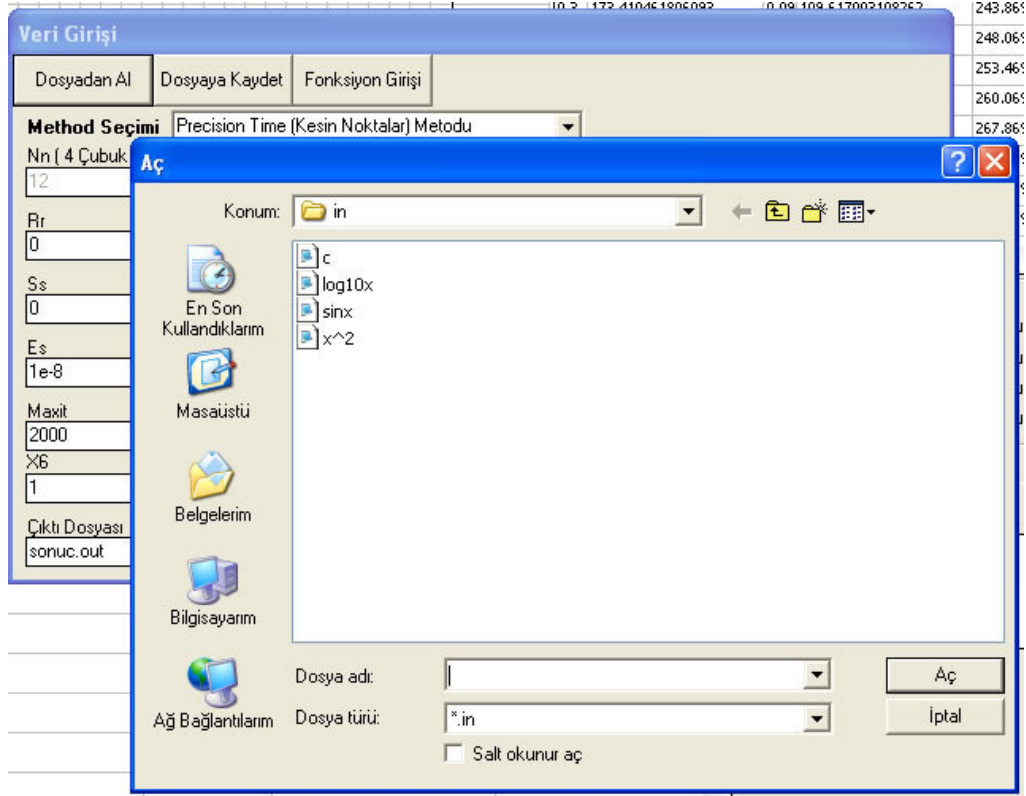
Şekil 4.7. Four Bar Programı Veri Giriş Menüsü

Fonksiyon giriři butonu ile çağrılabilen fonksiyon giriř menüsünde tanımlanan fonksiyon deęeri fonksiyon kutucuęuna işlenmektedir. *Combobox* sayesinde, tasarımı yapılacak mekanizmanın fonksiyon sentezinin hangi yöntem kullanılarak yapılabileceęi seçilir. *Nn* deęeri dört çubuk mekanizmaları için standart bir deęerdir. *D_{ps}*, 1. kol giriř açısını, *ds* 2. kol giriř açısını x_0 ve x_n , sentezin yapılması istenen nokta başlangıç ve bitiş noktalarını, aralık sayısı, başlangıç ve bitiş deęerleri bilinen kesin noktaların aralık sayısını belirtir.

Programda metin kutularında yazılı deęerler program çalıştıęında önceden tanımlı deęerler olarak gelmektedir. Çıktı ve girdi dosyaları kutucuęu, sentezi kaydetmek istedięimiz durumda dosya ismi tanımlamak amacıyla kullanılır. Mek-Gen programı, kullanıcıya istedięi takdirde yapılan sentezin program girdi ve sentez çıktılarını kaydetme imkanı vermektedir. Ayrıca *dosyadan al* butonu ile, daha önce kaydedilmiş program girdileri tekrar çağırılarak, yeniden sentez yapabılme imkanı kullanıcıya sunulmuştur. Girilen tüm deęerler, sentez yapılmak üzere tanımlı global deęişkenlerle ana programa gönderilir.

Mek-Gen Version1.0. programı, girdi verilerini *.in uzantı şeklinde kayıt eder. Tüm girdi verileri, programın çalıştıęı dizin içerisinden bulunun 'in' dosyasının içerisinde barındırılır. Bu önceden tanımlanmış bir seçimdir. Yani kullanıcı sentez yapacağı verileri, farklı yerlere farklı dosya isimleri ile ve farklı uzantılarda kaydetme imkanına sahiptir. Program ile çalışma yapıldıktan sonra giriř ve çıkış deęerlerine toplu olarak rahatça erişebilmesi için bu sistem önceden tanımlı olarak tanımlanmıştır.

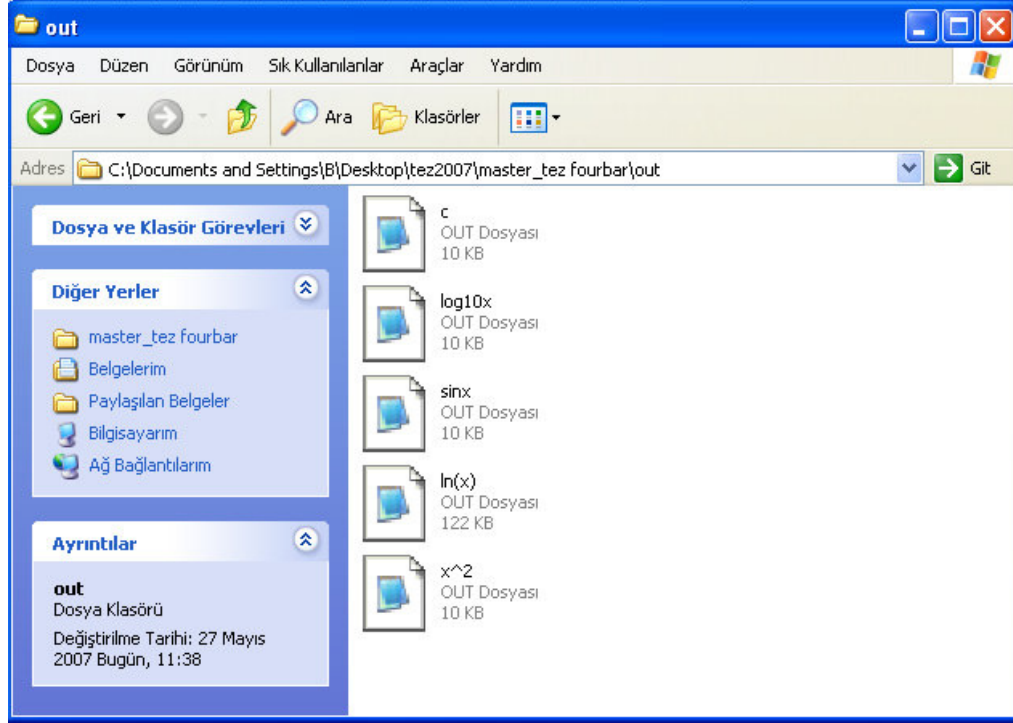
Şekil 4.8'de *dosyadan al* butonuna basılması sonucu, program otomatik olarak önceden belirlenmiş olan 'in' dosyasını çağırarak ve içerisindeki girdi dosyalarını gösterecektir.



Şekil 4.8. Four-Bar Programı Giriş Dosyaları

Veri giriş dosyaları olarak tanımladığımız *.in dosyalarının içeriğini bilgisayarımızdaki herhangi bir metin editöründe açarak işleyebiliriz. Şekilde *notepad* programında açılabilir olarak tanımlanmış in dosyaları görülmektedir.

Mek-Gen programı, veri girişinde olduğu gibi, aynı şekilde veri çıkışında da fonksiyon sentezinin tüm verilerini barındırabilen dosyaları saklama imkanı vermektedir. *.out uzantılı bu dosyalar programda kullanıcının istediğine göre kayıt edilebilir. Program içerisinde, sentezde oluşan ve mekanizma tasarımı için gerekli tüm veriler bulunmaktadır. *.out uzantısı yine tanımlı olarak verilmiştir. Ve yine veri giriş dosyalarında olduğu gibi, program işlev klasörünün içerisindeki 'out' dosyalarının içerisinde tanımlı olarak barındırılırlar. Kullanıcı bu dosyalara istediği takdirde başka uzantılar verebilir ve farklı dizinlerde saklayabilir.



Şekil 4.9. Four Bar Programı Çıktı Dosyaları

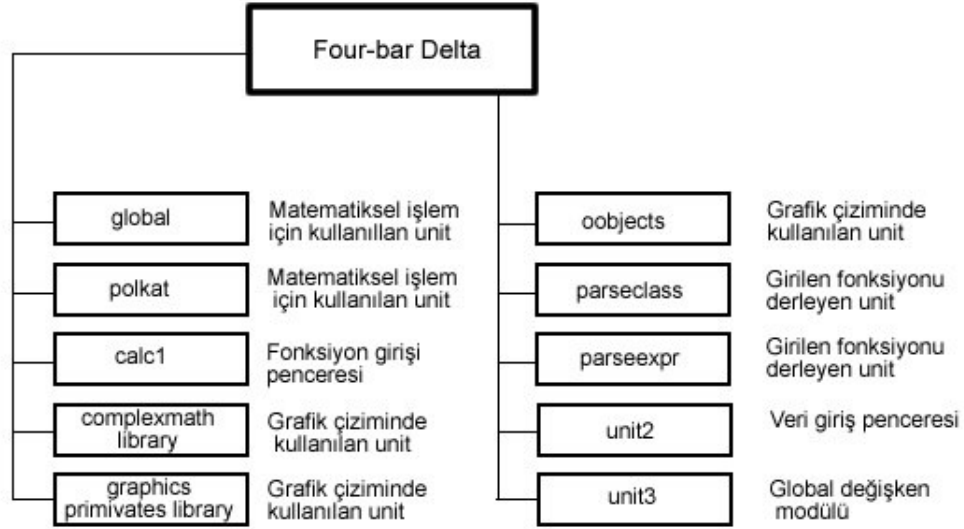
Şekil 4.9 'da görüldüğü gibi, program ana klasörü içerisinde bulunan *out* dosyasının içerisinde bulunan ve daha önce yapılaş mekanizma sentezlerinin verilerini bulunduran '.out' uzantılı dosyalar görülmektedir. Bu dosyalar da yine herhangi bir metin editörü yardımı ile açılabilir.

Hem veri giriş, hem de veri çıkış dosyaları, programda kayıt sırasında .xls uzantısında kayıt edilirse, dosyaları Excel veri sayfasına aktarmış oluruz. Aynı şekilde verileri, farklı programlarda kullanılması amacıyla programın tanıyabileceği uzantıda kayıt etmemiz, bu dosya verilerini başka programlarda da kullanabilme imkanı sağlamaktadır.

4.3.3.2. Four-Bar Delta Programı Kod Yapısı

Four-Bar delta programı toplam 28 birim kullanarak çalışmaktadır. Bu birimlerden 8 tanesi programda kullanılan matematiksel, analitik ve vektörel

fonksiyonları barındıran kütüphanelerdir. ‘Unit1’ isimli birim, programın ana modülü olup, diğer birimler bu modüle bağlıdır.



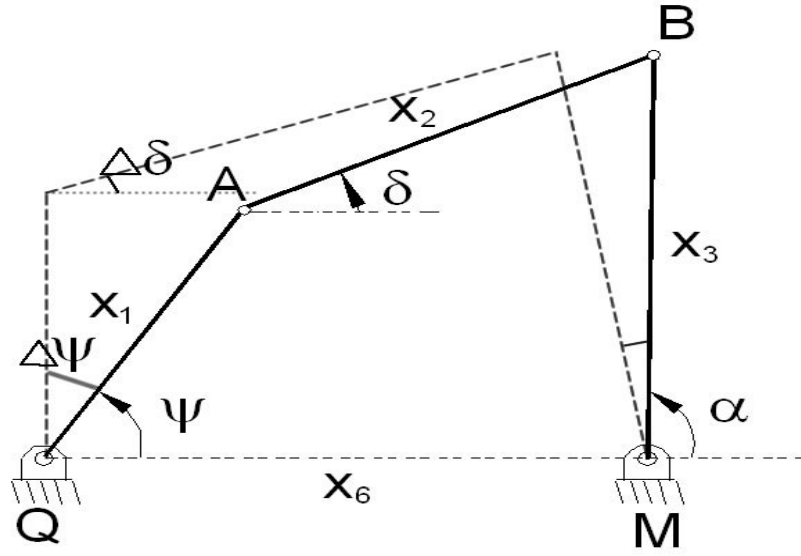
Şekil 4.10. Four Bar Delta Programının Kod Yapısı

Four-Bar Delta birimi alt programın ana birimi olup, diğer birimler Four-Bar delta program birimine bağlıdır.

Genel olarak four-bar delta programının çalışma yapısı ve dizaynı Four-Bar ile aynıdır.

Programın ana kod birimi, yaklaşık 2000 satır koddan oluşmaktadır. Toplamda kullanılan(delphi programının kendi birimleri hariç program için özellikle yazılan) birimler ve kütüphaneler de sayıldığında programı yaklaşık 9200 satır kod oluşturmaktadır.

Four bar delta programı, verilen bir dört çubuk mekanizmasının sentezini, giriş kolunun dönüş miktarı $\Delta\psi$ ve bağlantı kolunun dönüş miktarına $\Delta\delta$ göre yapar.

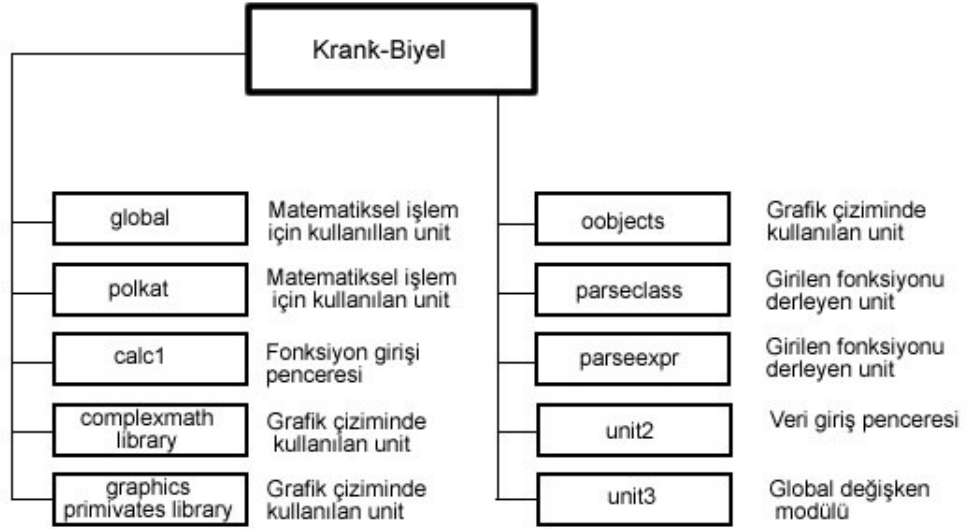


Şekil 4.11. Dört Çubuk Mekanizması

Programda kullanılan birimlerin senkronizasyonu ve çalışma yapıları Four-bar programı ile aynıdır. Sadece ana birimde, girilen değerlere bağlı olarak programın simülasyon kısmında değişiklik vardır.

4.3.3.3. Krank-Biyel Programı Kod Yapısı

Krank biyel alt programı da yukarıda bahsedilen diğer iki program gibi toplam 28 birim kullanarak çalışmaktadır. Bu birimlerden 8 tanesi programda kullanılan matematiksel, analitik ve vektörel fonksiyonları barındıran kütüphanelerdir. 'Unit1' isimli birim, programın ana modülü olup, diğer birimler bu modüle bağlıdır. Bu program, diğer programlardan farklı olarak krank-biyel mekanizmalarının fonksiyon sentezini yapmaktadır.

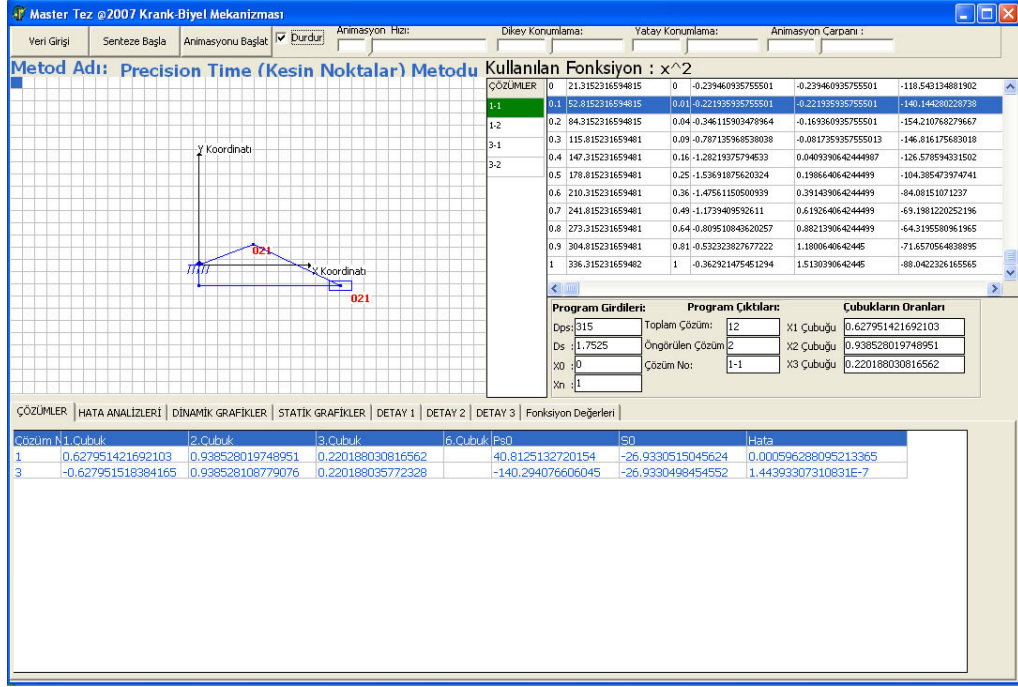


Şekil 4.12. Krank-Biyel Mekanizmasının Kod Yapısı

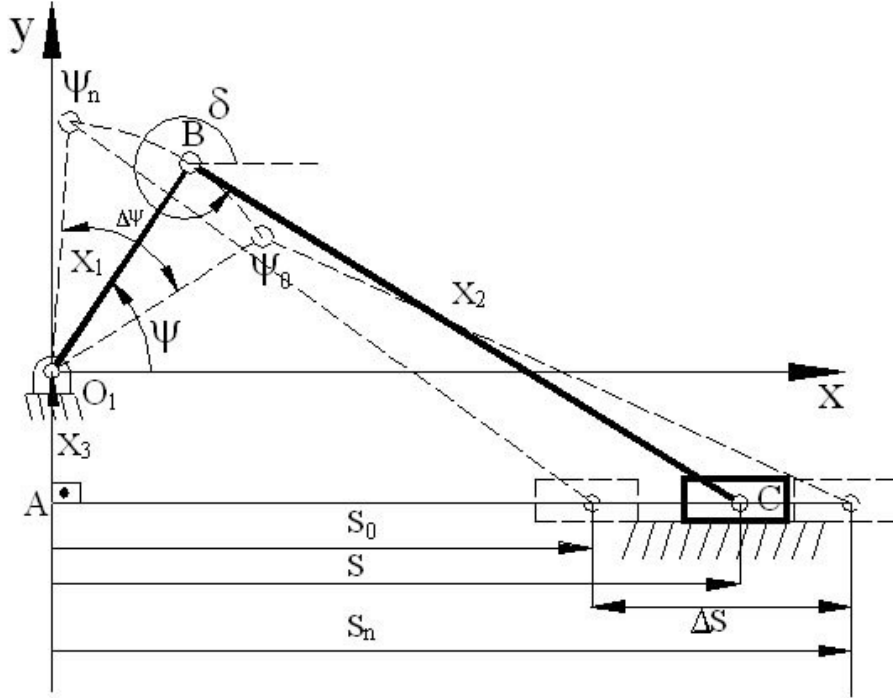
Şekilde de görüldüğü gibi, krank - biyel isimli birim, krank - biyel programının ana birimi olup, diğer birimler bu birime bağlı olarak kullanılmaktadır.

Programın ana birimi, yaklaşık 2.500 satır koddan oluşmaktadır. Toplamda kullanılan(delphi programının kendi birimleri hariç program için özellikle yazılan) birimler ve kütüphaneler de sayıldığında programı yaklaşık 10.000 satır kod oluşturmaktadır.

Krank - biyel alt programı, giriş kolu dönüş miktarı ve çıkış kolunun düzlemsel hareket miktarı verilen bir mekanizmanın fonksiyon sentezini yapmaktadır. Şekil 4.13' de krank biyel mekanizması sentezi yapan krank - biyel alt programının ekran görünüşü verilmektedir.



Şekil 4.13. Krank-Biyel Mekanizmasının Ekran Görünüşü

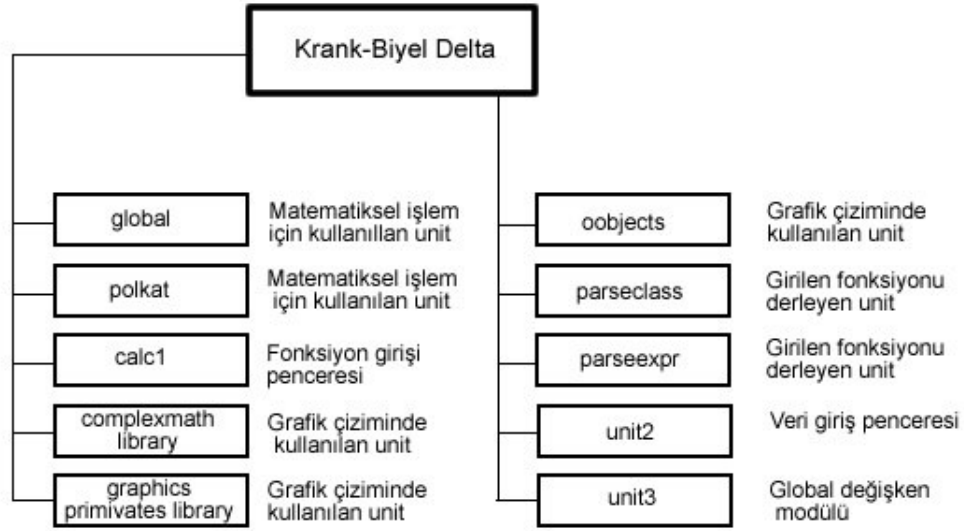


Şekil 4.14. Krank Biyel Mekanizması

Krank-Biyel alt programı, XI çubuğunun dönüş miktarı $\Delta\psi$ ve pistonun hareket miktarı ΔS 'a göre krank-biyel mekanizmasının sentezini yapar.

4.3.3.4. Krank-Biyel Delta Programı Kod Yapısı

Krank biyel delta alt programı krank-biyel programı gibi toplam 28 birim kullanarak çalışmaktadır. Bu birimlerden 8 tanesi programda kullanılan matematiksel, analitik ve vektörel fonksiyonları barındıran kütüphanelerdir. 'Unit1' isimli birim, programın ana modülü olup, diğer birimler bu modüle bağlıdır. Bu program, krank-biyel mekanizmalarının fonksiyon sentezini yapmaktadır.



Şekil 4.15. Krank-Biyel Delta Programının Kod Yapısı

Şekilde de görüldüğü gibi, krank-biyel delta isimli birim, krank-biyel delta programının ana birimi olup, diğer birimler bu birime bağlı olarak kullanılmaktadır.

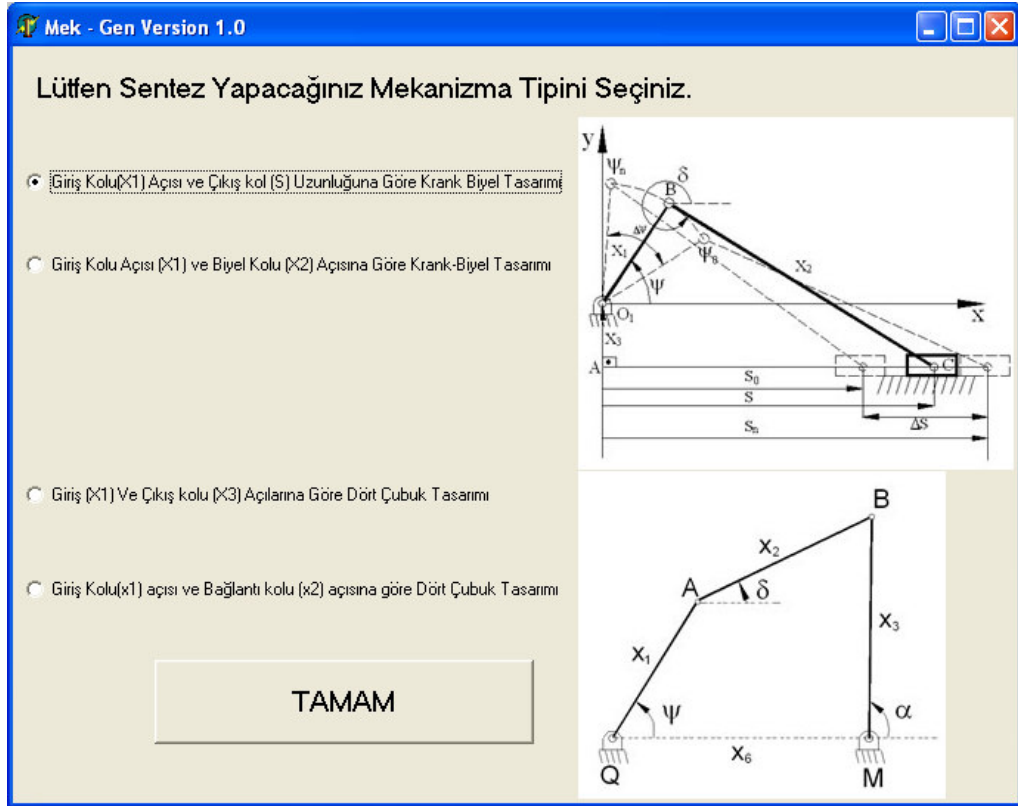
Programın ana birimi, yaklaşık 2.500 satır koddan oluşmaktadır. Toplamda kullanılan(delphi programının kendi birimleri hariç program için özellikle yazılan)

birimler ve kütüphaneler de sayıldığında programı yaklaşık 10.000 satır kod oluşturmaktadır.

Krank-biyel delta alt programı, giriş kolu($X1$ çubuğu) dönüş miktarı $\Delta\psi$ ve biyel kolunun ($X2$ çubuğu) dönüş miktarı ($\Delta\delta$) verilen bir mekanizmanın fonksiyon sentezini yapmaktadır.

4.3.4. Mek-Gen Programının Çalışma Yapısı

Mek-Gen version 1.0 adlı mekanizma sentez programımızın 4 alt programdan oluştuğundan bahsetmiştik. Programın .exe dosyası olan 'mekgen' adlı dosyayı çalıştırdığımızda karşımıza, Şekil 4.16' daki gibi bir menü gelecektir.



Şekil 4.16. Mek-Gen Programının Ana Görüntüsü

Bu form, sentezini yapacağımız mekanizma tipinin seçimini yapmamızı sağlayan karşılama ekranımızdır. Ana programa bağlı toplam 4 adet alt program mevcuttur.

İlk butona tıkladığımızda, girişi kolu dönme açısı miktarı $\Delta\psi$ ve çıkış kolu hareket aralığı ΔS verilen bir krank-biyel mekanizmasının sentezini yapmak üzere krank-Biyel isimli alt program çağırılır.

İkinci buton, giriş kolu dönüş açısı miktarı $\Delta\psi$ ve bağlantı kolu dönüş açısı miktarı $\Delta\delta$ bilinen bir krank-biyel mekanizmasının sentezini yapmak üzere krank-biyel delta isimli alt programı çağırır.

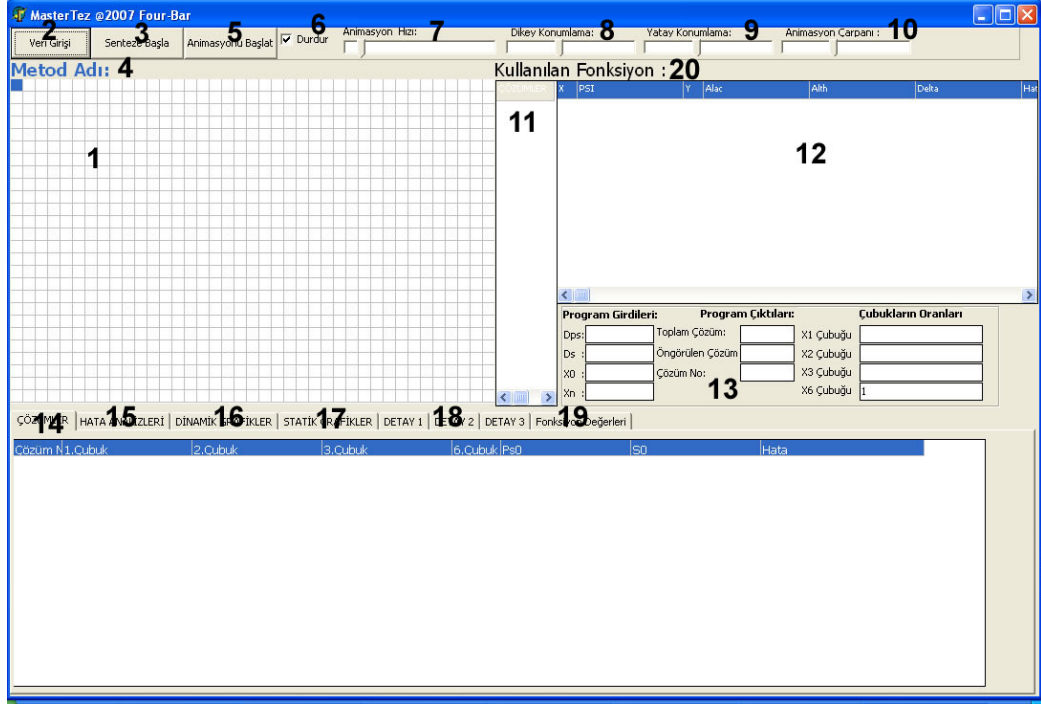
Üçüncü buton, giriş $\Delta\psi$ ve çıkış $\Delta\alpha$ kolları dönüş miktarları bilinen bir düzlemsel dört çubuk mekanizmasının sentezini yapmak üzere four-bar isimli alt programı çağırır.

Dördüncü ve son butonumuz, giriş kolu dönme miktarı $\Delta\psi$ ve bağlantı kolu dönme miktarları $\Delta\delta$ bilinen bir mekanizmanın sentezini yapmak üzere four-bar delta isimli alt programımızı çağırılmaktadır.

Şekil 4.17 'de bahse konu alt programlar detaylı olarak incelenmiştir.

4.3.4.1. Four-Bar Programı

Mek-Gen programından, 3 nolu seçeneği işaretleyip tamam butonuna basmak, kullanıcıyı dört çubuk mekanizma sentezi yapma üzere four-bar adlı alt programa götürmektedir.



Şekil 4.17. Four-Bar Programının Açıklanması

Programın ana menüsü şekilde görüldüğü gibidir. 1 numaralı yer, sentezi yapılan mekanizmanın hareketinin görsel olarak çizildiği ve gösterildiği yerdir. Tam dinamik bir yapıya sahip olan '*stringgrid*' nesnesi ve bu nesnenin '*canvas*' özellikleri kullanılmaktadır. *Canvas* özelliği, bir cisme *X-Y* koordinatlarına müdahale edebilip, gerekli grafik ve resim çizimi yapabilme özelliği sunar. Bu özellik sayesinde, gerçekleşen sentezin animasyonu için eldeki veriler vektörel olarak hesaplanarak *stringgrid*'de çizim yapılmaktadır.

2 numaralı yer, programa veri girişinin yapılabildiği formu çağırır. Programda veri girişi yapmadan sentez yapmak mümkün değildir.

3 numaralı yer, girilen deęerlere gre programın sentezini bařlatır. Bu buton, programın en can alıcı noktalarından birini, sentezin bařlamasını saęlar. Girilen deęerlere gre mekanizmanın sentezi yapılır ve ilgili yerlere deęerler çıktı verilir.

4 numaralı yer, fonksiyon sentezinde kullanılan metodu gsterir. Veri giriř kısmında seilen metod, sentez sırasında kullanıcıya bilgi vermek amacı ile buraya yazılır.

5 numaralı yer, seilen zmn animasyonunu 1 numaralı yere izimini bařlatır. Ayrıca dinamik grafiklerin de 16 nolu blgede izimini bařlatır.

6 numaralı yer, animasyonu durdurur.

7 numaralı yer, animasyonun hızını ayarlar.

8 numaralı yer, animasyonu dikeyde hareket ettirmeye yarar.

9 numaralı yer, animasyonu yatay dzlemde hareket ettirmeye yarar.

10 numaralı yer, animasyonun byklk ve kklęn ayarlar.

11 numaralı yer, zmleri liste halinde gsterir. Yapılan seime gre, 1 nolu blgede animasyon izdirir.

12 numaralı yer, seilen zmn(11'deki) hata analizlerini detaylı olarak gsterir.

13 numaralı yer, program girdi ve ıktılarından animasyon sırasında grlmesi gerekenleri gsterir.

14 numaralı yer, hata değeri en az olan çözümleri gösterir. Bunlar, cismin çubuk boyları, harekete başlangıç açıları ve hata oranlarıdır.

15 numaralı yer, sentez sonucu oluşan tüm hata analizlerini çözüm sayılarıyla birlikte detaylı olarak gösterir.

16 numaralı yer, mekanizmanın seçilen hata analizine bağlı değerlere göre dinamik grafikleri çizer.

17 numaralı yer, mekanizmanın seçilen hata analizine bağlı değerlere göre statik grafikleri çizer.

18 numaralı yer, sentez sırasında program modülleri içerisinde hesaplanan verileri gösterir.

19 numaralı yer, seçilen (Y) fonksiyonunun değerini ve fonksiyonu belirleyen x değişkeninin program akışı sırasındaki değerlerini gösterir.

20 numaralı yer ise sentezi yapılan fonksiyonu gösterir.

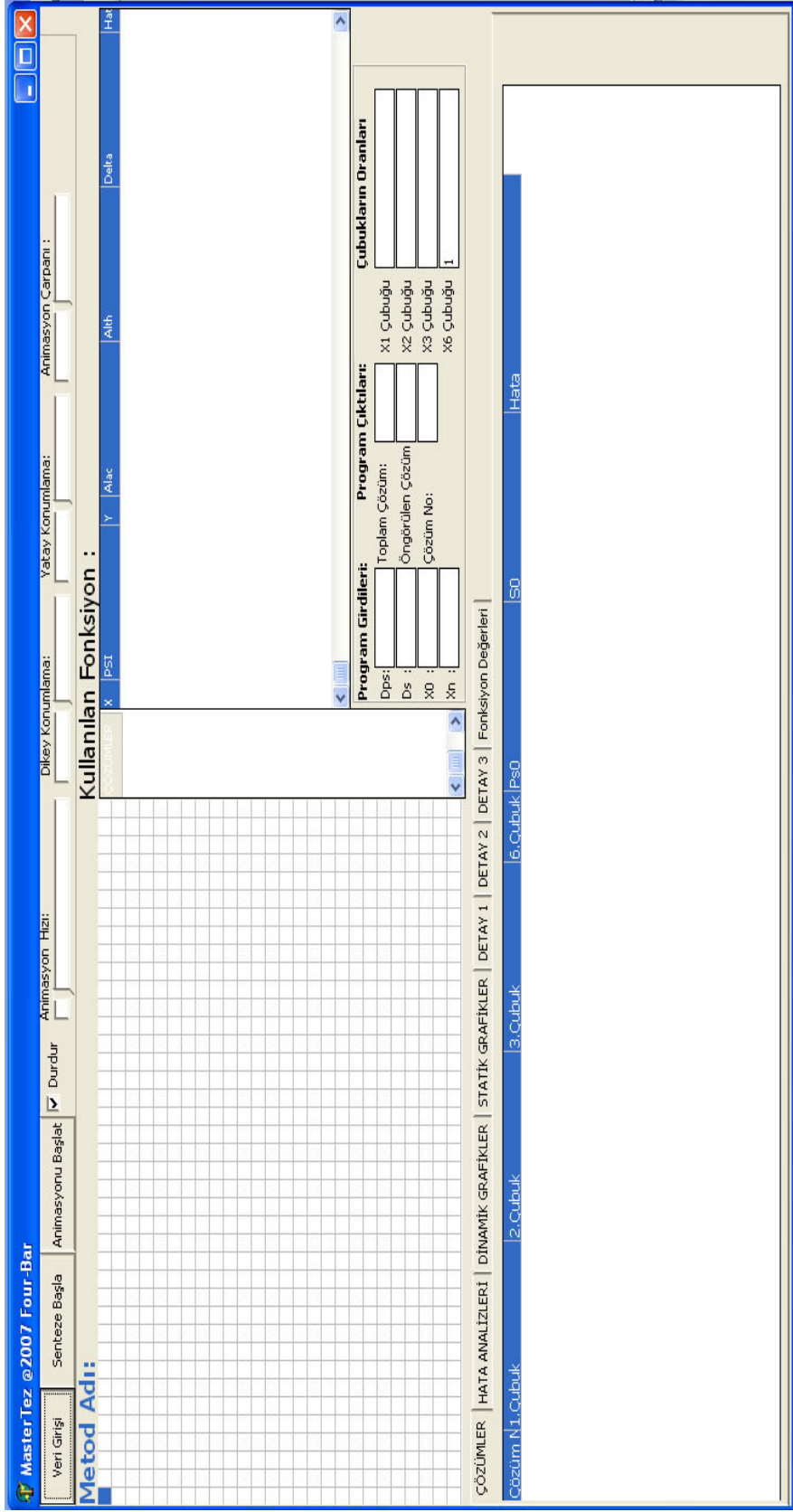
Programın çalışma ekranı, kullanıcının tüm verilere hakim olmasını sağlayan bir düzende dizayn edilmiştir. Girilen veriler, yapılan sentezler, kullanılan yöntemler, bu yöntemlerin programdaki işleyişi, hata analizleri, grafik ve animasyonların tümü, tek bir formda kullanıcıya sunulmuştur.

Sağlıklı bir mekanizma tasarımı yapabilmek için fonksiyon sentezi sırasında bulunan bulguların ve fonksiyon sentezi gerçekleştirilen mekanizmanın animasyonunun aynı anda görülmesi gerekmektedir. Değişen değerlere göre çizilen animasyonların detaylı bir şekilde ve tüm verileriyle görülmesi, ayrıca buna bir de grafiklerin dinamik yapıda olması tasarımcıya büyük kolaylık getirmektedir.

Mek-Gen programında yukarıda bahsekonu tüm detaylar bir arada işlendiğinden, programı kullanmak adeta tasarımcıya zevk vermektedir. Çünkü karmaşık bir yapıda olmayan, gayet basit kullanımlı tek menüsüyle kullanıcı istediği verileri aynı anda görebilmektedir.

Four-Bar Programının Çalışma Yapısı ve Mekanizma Sentezi: Daha önceki bölümlerde, Four-Bar programının işlevi, kullandığı birimler, temel yapısı ve ekran görüntüsü hakkında bilgi verilmiştir. Bu bölümde programın sentez sırasındaki akış diagramı detaylı olarak anlatılacaktır.

Four bar programımızın ana ekranı Şekil 4.18 'deki gibidir:



Şekil 4.18. Four Bar Programının Ekran Görüntüsü

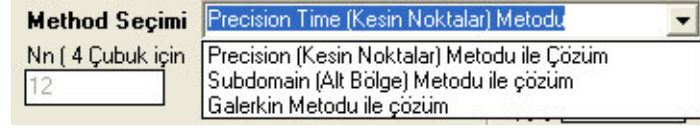
Mekanizma sentezini yapabilmek için öncelikle programa verileri girmek gerekmektedir. Dolayısıyla mekanizma sentezinden bahsetmeden önce programa veri girişi konusuna değinilecektir.

Veri Giriş Menüsü ve İşleyişi: Sol üst köşedeki *veri giriş* butonu, programa veri girmek amacıyla '*Veri Girişi*' menüsünün ekrana getirir. Sentez programına tüm verilerin giriş yapıldığı formdur. Şekilde form detaylı bir şekilde görülmektedir.

Veri Girişi					
Dosyadan Al		Dosyaya Kaydet		Fonksiyon Girişi	
Method Seçimi				Precision Time (Kesin Noktalar) Metodu	
Nn (4 Çubuk için)	Dps(Giriş Kolu Dönüş Mik.)	Precision İçin Kesin Noktalar		Subdomain İçin Kesin Noktalar	
12	90	Xp[1]	0	Xs[1]	0
Rr	Ds (Çıkış Kolu Dönüş Miktarı)	Xp[2]	0.1	Xs[2]	0.01
0	60	Xp[3]	0.6	Xs[3]	0.8
Ss	X0 Kesin Nokta Başlangıcı	Xp[4]	0.8	Xs[4]	0.9
0	0	Xp[5]	1	Xs[5]	0.99
Es	Xn Kesin Nokta Bitiş Açısı			Xs[6]	1
1e-8	1				
Maxit	Aralık Sayısı	Fonksiyon			
2000	10				
X6	Analiz Nokta Sayısı	Varsayılan Değerleri Yükle		Formu Temizle	
1	10				
Çıktı Dosyası	Girdi Dosyası	Senteze Başla			
sonuc.out	sonuc.in				

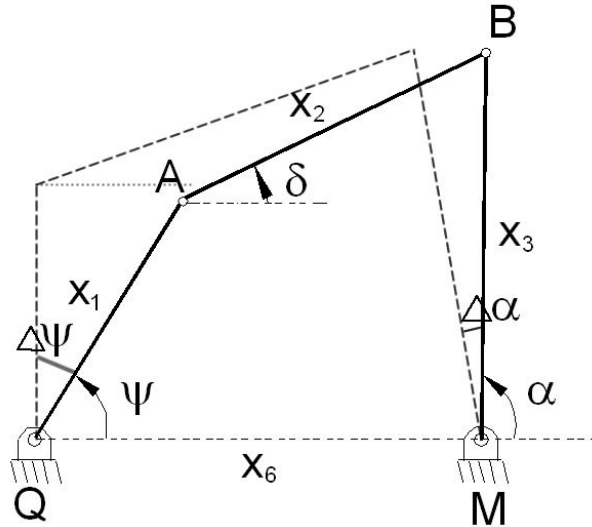
Şekil 4.19. Four Bar Programının Veri Giriş Ekranı

Programda metin kutularında yazılı değerler program çalıştığında önceden tanımlı olarak gelmektedir. Mekanizmanın fonksiyon sentezinin hangi yönteme göre yapılabileceği, *metod seçimi* menüsünden yapılabilmektedir. Precision time metodu, programda önceden tanımlanmış ve seçilidir. Kullanıcı fonksiyon sentezini toplamda 3 metoda göre de yapabilmektedir. Bu metodlar precision time, subdomain ve galerkin metodlarıdır.



Şekil 4.20. Sentez Yapılacak Metod Seçimi

N_n değerinin bulunduğu kutucuk, 4 çubuk mekanizmaları için sabit bir değer olup 12'dir. Bu değer dört çubuk mekanizmalarının sentezinde kullanılan polinomun kökünü belirtir. Değer sabit olduğundan, yani dört çubuk mekanizmalarının sentezini yapabilmek için kullanılacak polinomun 12 kökü olması gerektiğinden kullanıcıya bu değeri değiştirebilme yetkisi verilmemiştir. R_r ve s_s değerleri sentezde kullanılan polinomun reel ve imajiner köklerini temsil eder. Bu değerler polinoma işlenebilecek değerler olarak atandığından, kullanıcı tarafından değiştirilebilmektedir. E_s ve $maxit$; eliminasyon ve maximum iterasyon değerleridir. Bu değerler, fonksiyon sentezi sırasında eliminasyonun ve iterasyonun yapılması gereken aralıkları temsil eder. Sentez sonucu bulunan hata analizleri çok fazla olabilir. E_s ve $maxit$ değerleri ile, hata analiz sonuçları filtrelenir ve en düşük olanları görebilmemiz sağlanır. X_6 değeri, dört çubuk mekanizmasında alt çubuğu temsil etmektedir.



Şekil 4.21. Dört Çubuk mekanizmasının temel yapısı

Sentez sırasında mekanizmanın $X1, X2$ ve $X3$ uzuvları bulunurken, bu değerlerin $X6$ çubuğuna olan oranları bulunur. Yan $X6$ değeri her zaman 1 olarak alınmaktadır. Programda kullanıcının her veriyi görmesi amacıyla $X6$ değerinin de belirtilmesi uygun görülmüştür. Dps değeri $X1$ çubuğunun dönme miktarıdır. Şekilde $\Delta\psi$ olarak belirtilen açısal değer, programa Dps olarak girilmektedir. Ds değeri de yine bir açısal değeri temsil etmekte olup, çıkış kolunun dönüş miktarını belirtmektedir. Şekilde $\Delta\alpha$ olarak belirtilen kinematik büyüklük, programda Ds değerini temsil eder.

Programdaki açısal değerler programa derece olarak girilmektedir. X_0 ve X_n değerleri ise hata analizinin sırasıyla başlangıç ve bitiş aralığını belirtir. X_0 değeri sentezin noktasal olarak başlangıç değeridir. X_n değeri ise yine sentezin noktasal olarak bitiş değeridir. Bu değerler, her metod için kullanılmaktadır. Aralık sayısı olarak nitelendirilen değer, X_0 ve X_n başlangıç değerleri arasında yapılan sentezin kaç noktada olacağını belirtmektedir. Bu değer 10 olarak girildiğinde $x = 0$ dan 1 e kadar tanımlı bir aralıkta, sentez için yapılan hata analizleri 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 ve 1.0 olmak üzere 0 ile 1 aralığını toplamda 10 adet parçaya böler. Hata analizlerinde, her bir hata analizi için biz bu noktadaki değerleri görürüz.

Analiz nokta sayısı ise fonksiyonun yapacağı analizin kaç noktada gerçekleştirilmesi gerektiğini belirtir. 12 adet polinom kökü olan bir dört çubuk mekanizması için maximum 12'dir. 'Precision Point' metodunda, fonksiyon içerisinde kullanılan değerler için program 5 adet nokta belirleyerek bu noktalar için sentez yapar. Bu noktalar kesin çözümün olması istenen noktalar olup, kullanıcı tarafından değiştirilebilmektedir.

Aynı şekilde 'Subdomain' yani 'Alt bölgeler' metodunda da kesin çözümün bulunması istenen noktalar belirtilmektedir. Fonksiyon kutucuğu, sentezi yapılacak olan fonksiyonu içermektedir. Bu değer, fonksiyon girişi butonundan açılan fonksiyon girişi menüsünden yapılır.

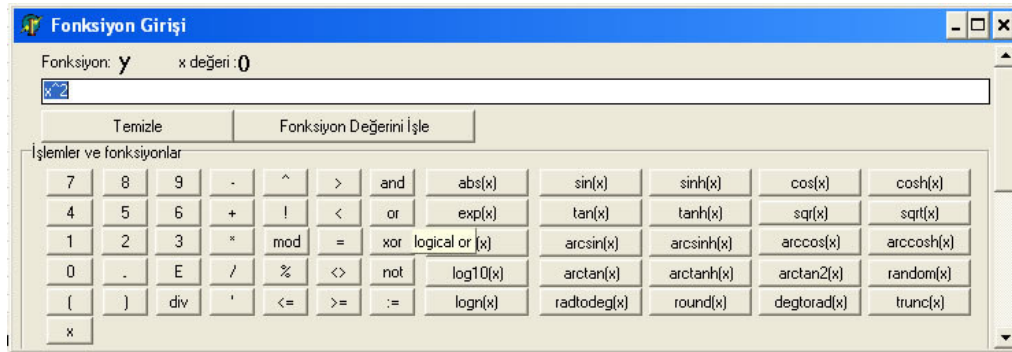
Veri girişi menüsünden programa girilen veriler, global değişkenler ile, ana menümüz olan Unit1'e gönderilmektedir.

Global değişkenler, bir program içerisinde belirli bir kullanılmayan, taşıdığı değerleri tüm birimlere aktarabilen menülerdir. Program içerisinde etkileşimi sağlayan en önemli parçacıklardır.

Mek-Gen programı içerisinde kullanılan global değişkenler, *Unit2* adlı modülde tanımlı bulunmaktadır. Programı oluşturan birimlerin, global değişkenleri tanınması için, barındıkları birimi tanımaları gerekmektedir. Yani unit2'nin diğer birimlerde tanımlı olması gerekir. Mek-Gen'de bu sistem sağlanmıştır ve veriler her modül ve fonksiyonda kullanılabilir hale getirilmiştir.

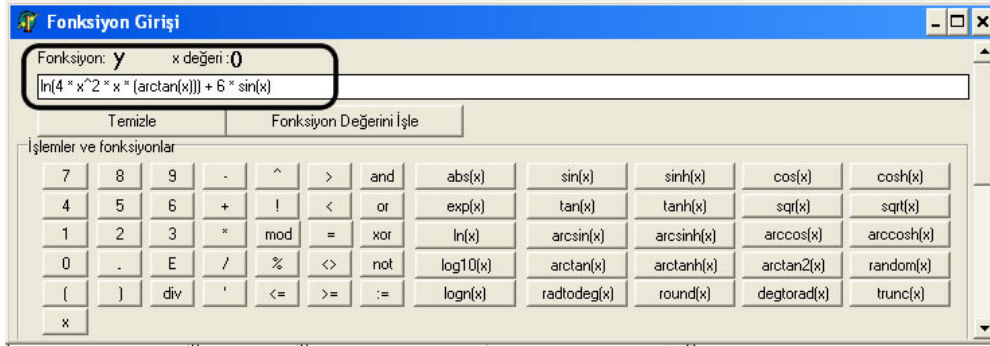
Biraz önce de bahsedildiği gibi, global değişkenler ile veri giriş programından girilen veriler unit1'e, oradan da kullanılacakları diğer fonksiyon ve modüllere gönderilebilmektedirler.

Fonksiyon Giriş Menüsü ve İşleyişi: Mek-gen programını diğer programlardan ayıran en önemli özelliklerden birisi, sentezi yapılacak olan fonksiyonun, kullanıcı tarafından tanımlanabilir olmasıdır. Veri giriş menüsünde bulunan fonksiyon girişi butonu, sentezi yapılacak olan fonksiyonun programa işlenebilmesini sağlamaktadır.



Şekil 4.22. Fonksiyon Giriş Menüsü

Mek-Gen'den önce yazılan sentez programlarında, bir fonksiyon problemi için fonksiyonun dışardan girilmesi çok zordu. Fonksiyon girme yöntemlerinden birisi ise gayet ilkel olan ve kaynak kodlara kullanılacak fonksiyonun, her sentez için işlenmesiydi. Mek-Gen Programı ile fonksiyon girişi, kaynak kodlara hiçbir müdahale edilmeden direkt olarak metin kutusuna yazmakla sağlamaktadır. Yukarıdaki şekilde de görüldüğü gibi istenilen kompleks yapıda ve zorlukta fonksiyon değerleri programa işlenebilir hale getirilmiştir.



Şekil 4.23. İşlenen Fonksiyonun Görünüşü

Şekilde görüldüğü gibi, fonksiyon tanımlanan kutucuğa, istenen uzunlukta ve karmaşıklıkta değer yazılabilmektedir.

Söz konusu fonksiyon giriş menüsü, girilen bu verileri karakter karakter okuyarak *parseexpress* ve *parseclass* adlı birimlere göndermektedir. Bu birimler, girilen karakterin, sayı, string, vektörel veri, trigonometrik veri ya da başka bir formatta olup olmadığını sentezlerler. Girilen metnin uzunluğu ve karmaşıklığına göre bu birimler bazen binlerce kez döngü kurarak, hatasız bir şekilde fonksiyon içindeki karmaşıklığı giderirler. Sonuç değeri birimlerde işlendikten sonra yine fonksiyon girişi birimine, global bir değişkenle gönderilir. Bu işlemler, fonksiyon değerini işle butonuna kullanıcının müdahale etmesi ile olmaktadır. Kullanıcı fonksiyonu ister el ile yazarak, isterse de karakterlere tıklayarak girebilme imkanına sahiptir.

Fonksiyon değerini işle butonuna basılarak, global değişkene atanmış veri aynı zamanda *veri girişi* menüsündeki fonksiyon kutucuğuna da işlenir.

The screenshot shows the 'Veri Girişi' window with the following details:

- Buttons:** Dosyadan Al, Dosyaya Kaydet, **Fonksiyon Girişi** (highlighted), Varsayılan Değerleri Yükle, Formu Temizle, Senteze Başla.
- Method Seçimi:** Precision Time (Kesin Noktalar) Metodu
- Input Fields:**
 - Nn (4 Çubuk için): 12
 - Dps(Giriş Kolu Dönüş Mik.): 315
 - Pr: 1
 - Ds (Çıkış Kolu Dönüş Miktarı): 1.7525
 - Ss: 1
 - X0 Kesin Nokta Başlangıç: 0
 - Es: 1e-6
 - Xn Kesin Nokta Bitiş Açısı: 1
 - Maxit: 200
 - Aralık Sayısı: 10
 - Analiz Nokta Sayısı: 10
 - Çıktı Dosyası: sonuc.out
 - Girdi Dosyası: sonuc.in
- Precision İçin Kesin Noktalar:** Xp[1]: 0, Xp[2]: 0.1, Xp[3]: 0.6, Xp[4]: 0.8, Xp[5]: 1
- Subdomain İçin Kesin Noktalar:** Xs[1]: 0, Xs[2]: 0.01, Xs[3]: 0.8, Xs[4]: 0.9, Xs[5]: 0.99, Xs[6]: 1
- Fonksiyon:** $\ln(4 * x^2 * x * (\arctan(x))) + 6 * \sin(x)$

Şekil 4.24. İşlenen Fonksiyonun Veri Giriş Ekranında Görülmesi

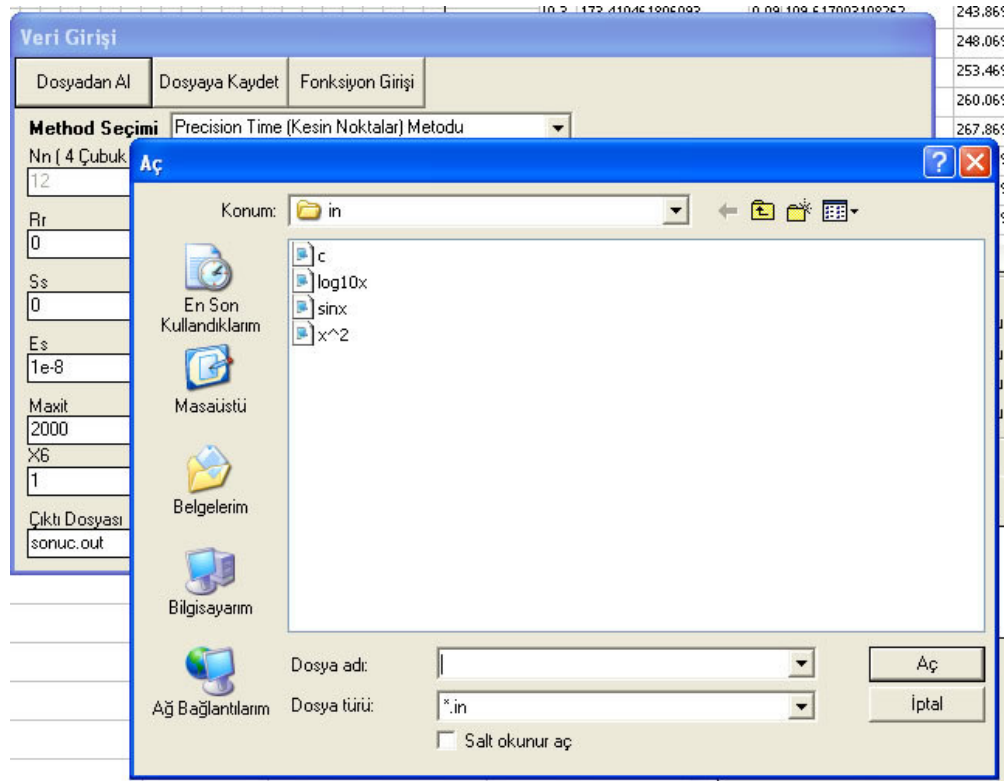
Şekilde, fonksiyon giriş menüsünde yazılmış ve sentez edilmiş fonksiyon değeri veri giriş menüsüne işlenmiştir. Bu işlem ile birlikte, sentez işlemi için gerekli tüm veriler girilmiş bulunmaktadır.

Girilen verileri kullanıcı senteze başla butonu ile direk olarak ana programa gönderebileceği gibi, bu değerleri göndermeden önce programın girdi ve çıktılarını daha sonra kullanmak amacıyla saklayabilir.

Dosyalama Sistemi ve İşleyişi : Çıktı ve girdi dosyaları kutucuğu, sentezi kaydetmek istediğimiz durumda dosya ismi tanımlamak amacıyla kullanılır. Mek-Gen programı, kullanıcıya istediği takdirde yapılan sentezin program girdi ve sentez çıktılarını kaydetme imkanı vermektedir. Ayrıca *dosyadan al* butonu ile, daha önce kaydedilmiş program girdileri tekrar çağırılarak, yeniden sentez yapabilmeye imkanı kullanıcıya sunulmuştur.

Mek-Gen Version1.0. programı, girdi verilerini *.in uzantı şeklinde kayıt eder. Tüm girdi verileri, programın çalıştığı dizin içerisinde bulunun 'in' dosyasının içerisinde barındırılır. Bu varsayılan bir seçimdir. Yani kullanıcı sentez yapacağı verileri, farklı yerlere farklı dosya isimleri ile ve farklı uzantılarda kaydetme imkanına sahiptir. Program ile çalışma yapıldıktan sonra giriş ve çıkış değerlerine toplu olarak rahatça erişilebilmesi için bu sistem varsayılan olarak tanımlanmıştır.

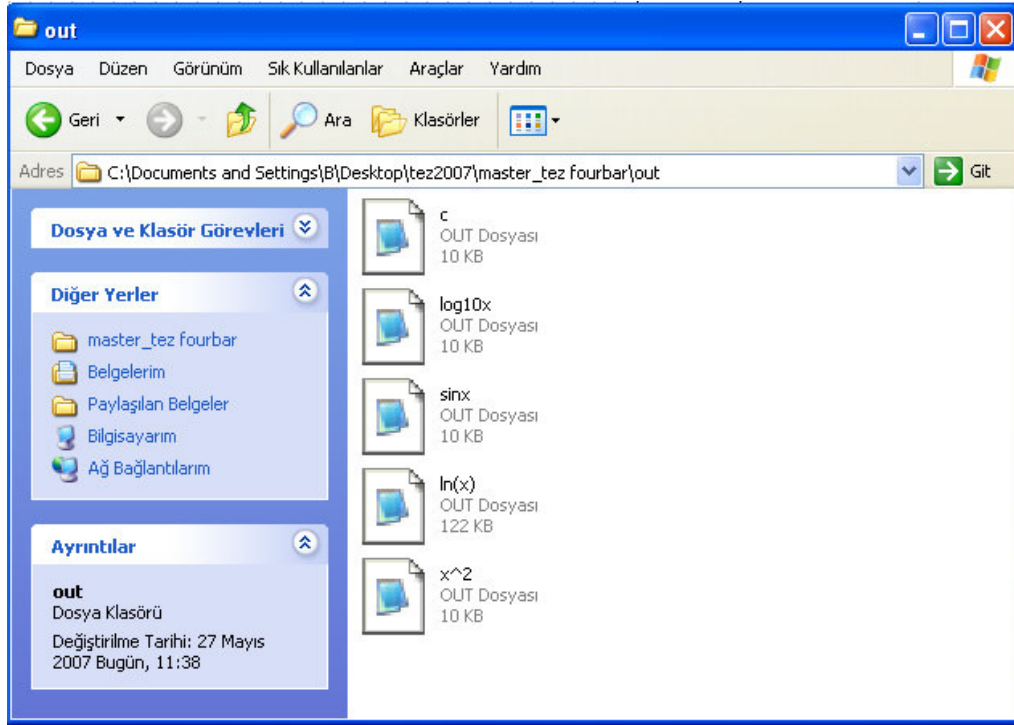
Şekilde *dosyadan al* butonuna basılması sonucu program varsayılan olarak belirlenmiş 'in' dosyasını çağırarak ve içerisindeki girdi dosyalarını gösterecektir.



Şekil 4.25. Kayıt Yapılan Girdi Dosyaları

Veri giriş dosyaları olarak tanımladığımız *.in dosyalarının içeriğini bilgisayarımızdaki herhangi bir metin editöründe açarak işleyebiliriz. Şekilde *notepad* programında açılabilir olarak tanımlanmış '.in' dosyaları görülmektedir.

Mek-Gen programı, veri girişinde olduğu gibi, aynı şekilde veri çıkışında da fonksiyon sentezinin tüm verilerini barındırabilen dosyaları saklama imkanı vermektedir. *.out uzantılı bu dosyalar programda kullanıcının isteğine göre kayıt edilebilir. Program içerisinde, sentezde oluşan ve mekanizma tasarımı için gerekli tüm veriler bulunmaktadır. *.out uzantısı yine varsayılan olarak verilmiştir. Ve yine veri giriş dosyalarında olduğu gibi, program işlev klasörünün içerisindeki 'out' dosyalarının içerisinde varsayılan olarak barındırılırlar. Kullanıcı bu dosyalara istediği takdirde başka uzantılar verebilir ve bunları farklı dizinlerde saklayabilir.



Şekil 4.26. Çıktı Dosyaları

Şekilde görüldüğü gibi, program ana klasörü içerisinde bulunan 'out' dosyasının içerisinde bulunan ve daha önce yapılaş mekanizma sentezlerinin

verilerini bulunduran '.out' uzantılı dosyalar görülmektedir. Bu dosyalar da yine herhangi bir metin editörü yardımı ile açılabilir.

Hem veri giriş, hem de veri çıkış dosyaları, programda kayıt sırasında .xls uzantısında kayıt edilirse, dosyaları Excel veri sayfasına aktarmış oluruz. Aynı şekilde verileri, farklı programlarda kullanılması amacıyla programın tanıyabileceği uzantıda kayıt etmemiz, bu dosya verilerini başka programlarda da kullanabilme imkanı sağlamaktadır.

Veri girişi sonrasında oluşan ekran çıktısı Şekil 4.27 'deki gibidir. Bu aşamadan sonra girilen veriler ile sentez yapmak üzere "senteze başla" butonu tıklanarak, ana menüye geçilir.

Şekil 4.27. Sentezin Başlaması İçin Verilerin Gönderilmesi

Four-Bar Programı İle Mekanizma Sentezi: Girilen tasarım değerleri, global değişkenler ile Four-Bar ana programına gönderildikten sonra, Senteze başla butonu ile girilen veriler işlenmeye başlar.



Şekil 4.28. Sentezin Başlatılması

Girilen veriler ilk olarak sentez işlemine tabi tutulur.

Sentez işleminde şu işlemler gerçekleşir:

- 1) Seçilen metoda bağlı olarak (precision - kesin noktalar – subdomain) girilen ilk değerler fonksiyonda kullanılmak üzere işlenir.
- 2) Kullanılacak olan $X1 - X2$ ve $X3$ çubuk boyları, $X6$ çubuğunun oranına göre tespit edilir. ($X6$ çubuğunun uzunluğu 1 birim olarak alınır. Diğer çubukların bulunan boyları, $X6$ çubuğuna göre oranlarıdır.)
- 3) Kullanılan fonksiyonun girilen X değişkenlerine göre değerleri bulunur. X değerinin, x_0 ve x_n tanım aralığında bulunan ve *veri girişi* menüsünden girilen noktalara ayrılmış kesimlerdeki fonksiyon değerleri bulunur.
- 4) Dört çubuk mekanizması için her bir polinomun toplamda 12 adet kökü vardır. Bu kökler bulunur. Her polinomun kökü reel ve imajiner kısımlardan oluşmaktadır. Hata analizi sırasında polinomların reel kökleri ile imajiner kökleri birbirinden ayrılır ve analize yarayan reel kökleri alınır.
- 5) Çubuk boy oranları bulunan mekanizmanın giriş kolu başlangıç açısı ve çıkış kolu başlangıç açısı hesaplanır.
- 6) Giriş kolunun başlangıç açısı ve çıkış kolunun bağlı açısının, noktaları daha önce belirlenen x ve bağlı $f(x)$ fonksiyonuna tabi tutulması ile hata değerleri saptanır.

7) Mekanizmayı oluşturacak fonksiyonların toplam çözümleri bulunur. Bu çözümlerden hata analizleri yapılır ve hata sapmaları en az olan fonksiyon değerleri mekanizma sentezi için seçilir.

8) Bulunan tüm değerler uygun bölümlere işlenir.

Sentez işleminin bitmesinin ardından girilen fonksiyona bağlı değerler saptanmış olur.

Programda elde edile tüm çıktılar, kullanılan '*Stringgrid*' ve '*editbox*' nesnelere yazdırılmaktadır. *Stringgrid* nesnelere, birbiri ile bağımlı sonuçları bir arada ve düzenli bir şekilde görebilmek için tasarlanmış ideal nesnelere.

Çözümler *stringgrid*inde, programın bulmuş olduğu, hata analizleri en uygun çözümlerin listesi yer almaktadır. Uygun çözümler buradan tek tek seçilebilmekte, çözümün detayları görülebilmektedir. Çözümler *grid*inin yanındaki *grid*e, seçilen çözüme ait olan değerler görülmektedir. Bunlar,

X: Sentezin yapılması istenen aralıktaki noktalar;

Psi: *X* noktasındaki *X1* giriş kolunun açılma açısı,

Y: Sentezin yapılması istenen aralıktaki *x* değerlerine uygun fonksiyon değerleri;

Alac: *X3* Çıkış kolunun bulunan uzunluğu;

Alth: *X3* Çıkış kolunun teorikte olması gereken uzunluğu,

Delta: *Alac* ve *Alth* değerleri arasındaki sapma miktarını;

Hata: bu noktadaki yapısal hatayı göstermektedir.

Program girdileri kısmında, *veri girişi* menüsünden girmiş olduğumuz $\Delta\psi$, $\Delta\alpha$, X_0 ve X_n başlangıç değerleri yer almaktadır. Toplam çözüm sayısı, hata analizleri yapılırken bulunan çözüm sayısını, öngörülen çözüm sayısı ise programda görebildiğimiz, filtrelenmiş çözümleri ifade etmektedir.

Master Tez @2007 Four-Bar

Veri Girişi Sembze Başla Animsasyonu Başlat Durdur Animsasyon Hızı: Yatay Konumlama: Dikey Konumlama: Animsasyon Çarpmanı:

Metod Adı: Precision Time (Kesin Noktalar) Metodu

Kullanılan Fonksiyon : x^2

ÇÖZÜMLER	X	PSI	Y	Alac	Alth	Delta
0	146.410461806093	0	88.0881053834712		238.463385169401	0.205124218843512
7-2	0.1	155.410461806093	0.01	96.4472465221648	239.063385169401	4.333152394443569
8-1	0.2	164.410461806093	0.04	103.631526547737	240.863385169401	8.52250311323813
8-2	0.3	173.410461806093	0.09	109.617003108262	243.863385169401	12.8253052556206
19-1	0.4	182.410461806093	0.16	114.382531124833	246.063385169401	17.2712339305484
19-2	0.5	191.410461806093	0.25	117.923265359194	253.463385169401	21.865810644763
20-1	0.6	200.410461806093	0.36	120.253379752381	260.063385169401	26.5899167421275
20-2	0.7	209.410461806093	0.49	121.397257263938	267.863385169401	31.4011056736021
	0.8	218.410461806093	0.64	121.371385506304	276.863385169401	36.235255758702
	0.9	227.410461806093	0.81	120.159546373907	287.063385169401	41.0062720899624

Program Girdileri: Dps: 90 Ds: 60 X0: 0 Xn: 1

Program Çıktıları: Toplam Çözüm: 24 Öngörülen Çözüm: 4 Çözüm No: 4

Çubukların Oranları: X1 Çubuğu: 0.991533729706845 X2 Çubuğu: 1.84451340384617 X3 Çubuğu: 0.5554683180044 X6 Çubuğu: 1

ÇÖZÜMLER | HATA ANALİZLERİ | DİNAMİK GRAFİKLER | STATİK GRAFİKLER | DETAY 1 | DETAY 2 | DETAY 3 | Fonksiyon Değerleri

Çözüm No	1.Çubuk	2.Çubuk	3.Çubuk	6.Çubuk	PsD	Hata
7	0.991533729706845	1.84451340384617	0.5554683180044	1	146.410461806093	-121.530614830599
8	0.991533729706845	1.84451340384617	-0.5554683180044	1	146.410461806093	58.4693851694014
19	-0.991533729812103	1.84451340344771	0.555468318028603	1	-33.5895381955516	-121.53061488248
20	-0.9915337298121	1.84451340344771	-0.555468318028603	1	-33.5895381955516	58.4693851175198

Şekil 4.29. Sentez Sonrası Ana Ekran

$X1, X2$ ve $X3$ çubuk boyları ise, sentez sonu oluşan çözümlerden, seçili olan çözüme ait çubuk boyunu göstermektedir.

Programın alt kısmında bulunan sayfalardan çözümler kısmında, sentez sonucu bulunmuş çözümlerden öngörülen çözümler ve bu çözümlere bağlı tasarlanabilecek mekanizma uzuvlarının boyutları ile giriş çıkış açıları; mekanizmanın tasarlanması halinde karşılaşılabilecek hata sapması gösterilmektedir

Hata analizleri kısmında, öngörülen çözümlere bağlı değerler ve bu değerlerin hata analizlerinin tümü bir arada yazılmıştır. 3 nolu tabloda bulunan değerler, Şekil 4.30'daki tablonun sadece seçilen çözüm için alınmış kısmıdır. Bu tabloda ise kullanıcı istediği takdirde tüm analiz sonuçlarını bir arada görme imkanına sahiptir.

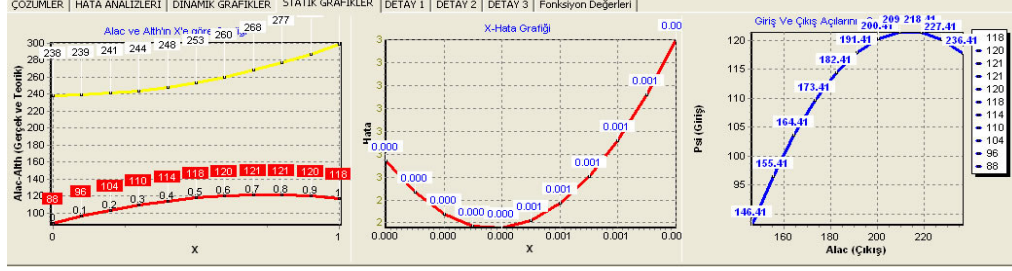
ÇÖZÜMLER		HATA ANALİZLERİ		DİNAMİK GRAFİKLER	STATİK GRAFİKLER	DETAY 1	DETAY 2	DETAY 3	Fonksiyon Değerleri
		HATA	ANALİZ SONUÇLARI						
			7inci Çözüm İçin		Hata Analiz Sonuçları				
		7-1	1inci Dört Çubuk		Analiz Sonuçları				
X	Psi	Y	Alac	Alth	Delta	Hata			
0	146.410461806093	0	88.0881053834712	238.469385169401	0.205124218843512	2.62464846562728			
0.1	155.410461806093	0.01	96.4472445221648	239.069385169401	4.33315294443569	2.48922594053672			
0.2	164.410461806093	0.04	103.631526547797	240.869385169401	8.52250311323813	2.39525249133349			
0.3	173.410461806093	0.09	109.617003108262	243.869385169401	12.8253052656206	2.34314609561226			
0.4	182.410461806093	0.16	114.382531124833	248.069385169401	17.271239305484	2.33327576971081			
0.5	191.410461806093	0.25	117.923265359194	253.469385169401	21.865810644763	2.36572607999083			
0.6	200.410461806093	0.36	120.253379752381	260.069385169401	26.5899167421275	2.44024964151324			
0.7	209.410461806093	0.49	121.397257263938	267.869385169401	31.4011056736021	2.5564208943526			
0.8	218.410461806093	0.64	121.371385506304	276.869385169401	36.2352557758702	2.71395207438609			
0.9	227.410461806093	0.81	120.159546373907	287.069385169401	41.0062720899624	2.91312624095435			

Şekil 4.30. Sentez Sonrası Analiz Sonuçları

Dinamik grafikler kısmında, veriler animasyondan sonra belirdiğinden, programın animasyon kısmında anlatılacaktır.

Statik grafikler kısmında, 2 numaralı tablodan seçilen çözümün grafikleri verilmektedir. İlk grafik, *Alac* ve *Alth* değerlerinin x değişkenine göre değişimini göstermektedir. *Alac* ve *Alth* arasındaki fark, hatayı sapmasını verdiği için, bu grafikteki veriler adeta hatanın görsel olarak analiz edildiği çizimleri oluşturmaktadır. İkinci grafikte, hatanın x değişkenine göre değişimi

yanıtılmaktadır. 3. grafikte ise, giriş açısının, çıkış açısının değişimine göre oluşturduğu değerler görülmektedir.



Şekil 4.31. Sentez Sonrası Statik Grafikler

Detay1- Detay2 ve Detay 3 sayfalarında ise, programın sentezi sırasında oluşan diğer veriler yer almaktadır. Bu veriler, yukarıda anlatılan bulguların elde edilmesi sırasında toplanmış bilgiler olup, detaylar üzerinde çalışmak isteyen tasarımcıların her veriye hakim olabilmesi düşüncesiyle listelenmiştir.

No	aaa değerleri	Kat	Ps0	50	Ps0/kat	50/kat
1	0.000233014339060813	0.0174532925199433	4.16529977022726	-0.0945278311977466	153.000000000855	-10.8000000020971
2	0.00873350120603587	0.0174532925199433	4.16529977022726	10.5788949913392	153.000000000855	169.199999997903
3	0.0828786481052983	0.0174532925199433	4.16529977022726	0	153.000000000855	0
4	0.309540512631783	0.0174532925199433	25.4516995793541	-0.00261799985940063	175.4999999999999	-0.30000000027925
5	0.394622542827431	0.0174532925199433	25.4516995793541	381.970990719951	175.4999999999999	179.699999999972
6	-0.228430835530218	0.0174532925199433	25.4516995793541	0	175.4999999999999	0
7	-0.615356603052689	0.0174532925199433	3.31325298194297	-1.78674803036047	146.410461806093	-121.530614830599
8	0.228430835530218	0.0174532925199433	3.31325298194297	0.559676005238552	146.410461806093	58.4693851694014
9	0.394622542827431	0.0174532925199433	3.31325298194297	0	146.410461806093	0
10	-0.309540512631783	0.0174532925199433	3.54573254236706	-0.0971698782108251	148.499999998373	-11.1000000191494
11	0.0828786481052983	0.0174532925199433	3.54573254236706	10.2912550515947	148.499999998373	168.899999980851
12	-0.00873350120603587	0.0174532925199433	3.54573254236706	0	148.499999998373	0
13	0.000233014339060813	0.0174532925199433	-0.282029168302182	-0.0971698780424543	-31.5000000000031	-11.1000000000036
			0.0174532925199433	10.2912550694269	-31.5000000000031	168.899999999964

Reel[]	İmaj[]	i	Fonk	Çözüm No	Fonksiyon Değerleri	Eps Değeri
4.16529977022726	0	1	-2.18671101775331E-8	1	-4.75422652634894E-17	4.75422652634894E-17
25.4516995793541	0	2	-1.47699127492532	1	-4.75422652634894E-17	6.7234916321566E-17
3.31325298194297	0	3	-1.47324012181713E-9	1	7.54962498766831E-15	7.54992436971381E-15
3.54573254236706	0	4	-2.1477368545672E-9	1	0.0940310947260002	0.0940310947260002
1.51315786319312	-0.567894516246804	5	0.685108792392451	1	0.457527905953167	0.467090603096723
1.51315786319312	0.567894516246804	6	0.685108792392451	1	0.457527905953167	0.467090603096723
-0.579276588840921	-0.217404942467924	7	0.00155091724891493	2	5.87095476400901E-17	5.87095476400901E-17
-0.579276588840921	0.217404942467924	8	0.00155091724891493	2	5.82758667710959E-17	8.27217482408739E-17
-0.282029168302182	0	9	1.03761536038652E-20	2	7.65576838035464E-15	7.65621527788175E-15
-0.240078759080112	0	10	7.87740640946499E-20	2	0.0940308637818778	0.0940308637818778
-0.301818184576251	0	11	-3.54271530189111E-19	2	0.457527110725965	0.467089777658231
-0.0392901070076687	0	12	-1.12364214409203E-20	2	0.457527110725965	0.467089777658231
				3	2.58582218137793E-17	2.58582218137793E-17

ÇÖZÜMLER		HATA ANALİZLERİ	DİNAMİK GRAFİKLER	STATİK GRAFİKLER	DETAY 1	DETAY 2	DETAY 3	Fonksiyon Değerleri		
xth(psi)	kat	psi	psi/kat	y(xth(psi))	sac	sac/kat	sth	sth/kat		
0	0.0174532925199433	2.55534461788172	146.410461806093	0	1.53742747078531	88.0861053834712	4.16207593641259	238.4693		
0.1	0.0174532925199433	2.71242425056121	155.410461806093	0.01	1.68332197138784	96.4472445221648	4.17254791192456	239.0693		
0.2	0.0174532925199433	2.8695038832407	164.410461806093	0.04	1.80871134712696	103.631526547797	4.20396383846045	240.8693		
0.3	0.0174532925199433	3.02658351592019	173.410461806093	0.09	1.91317762040802	109.617003108262	4.25632371602028	243.8693		
0.4	0.0174532925199433	3.18366314859968	182.410461806093	0.16	1.99635177489324	114.382531124833	4.32962754460405	248.0693		
0.5	0.0174532925199433	3.34074278127917	191.410461806093	0.25	2.05814924522091	117.923265359194	4.42387532421174	253.4693		
0.6	0.0174532925199433	3.49782241395866	200.410461806093	0.36	2.09881741333013	120.253379752381	4.53906705484337	260.0693		
0.7	0.0174532925199433	3.65490204663815	209.410461806093	0.49	2.11878184214632	121.397257263938	4.67520273649892	267.8693		
0.8	0.0174532925199433	3.81198167931764	218.410461806093	0.64	2.11833029479232	121.371385506304	4.83228236917841	276.8693		
0.9	0.0174532925199433	3.96906131199713	227.410461806093	0.81	2.09717971192749	120.159546373907	5.01030595288184	287.0693		
1	0.0174532925199433	4.12614094467662	236.410461806093	1	2.0539002464519	117.679815662574	5.20927348760919	298.4693		
0	0.0174532925199433	2.55534461788172	146.410461806093	0	4.16207593641259	238.469385169401	4.16207593641259	238.4693		
0.1	0.0174532925199433	2.71242425056121	155.410461806093	0.01	4.17254791192456	239.069385169401	4.17254791192456	239.0693		

Şekil 4.32. Sentez sonrası detay verilerin ekrana yansımaları

Fonksiyon değerleri sayfasında ise mekanizmanın tasarımında kullanılan $f(x)$ fonksiyonun ve X bağımsız değişkeninin program akışı sırasında aldığı değerler listelenmiştir.

ÇÖZÜMLER		HATA ANALİZLERİ	DİNAMİK GRAFİKLER	STATİK GRAFİKLER	DETAY 1	DETAY 2	DETAY 3	Fonksiyon Değerleri			
Kullanılan Fonksiyon: x^2								x0	y(x0)	xn	y(xn)
NO	xth(p0)	Fonk. Değ. $f(xth(p0))$	ps	Fonk. Değ. $f(y(xth(psi)))$							
1	0	0	0	0	0 0 1 1						
2	0	0	0.1	0.01							
3	0	0	0.2	0.04							
4	0	0	0.3	0.09							
5	0	0	0.4	0.16							
6	0	0	0.5	0.25							
7	0	0	0.6	0.36							
8	0	0	0.7	0.49							

Şekil 4.33. Fonksiyon Değeri ve X Değerinin Sentez Sırasında Aldıkları Değerler

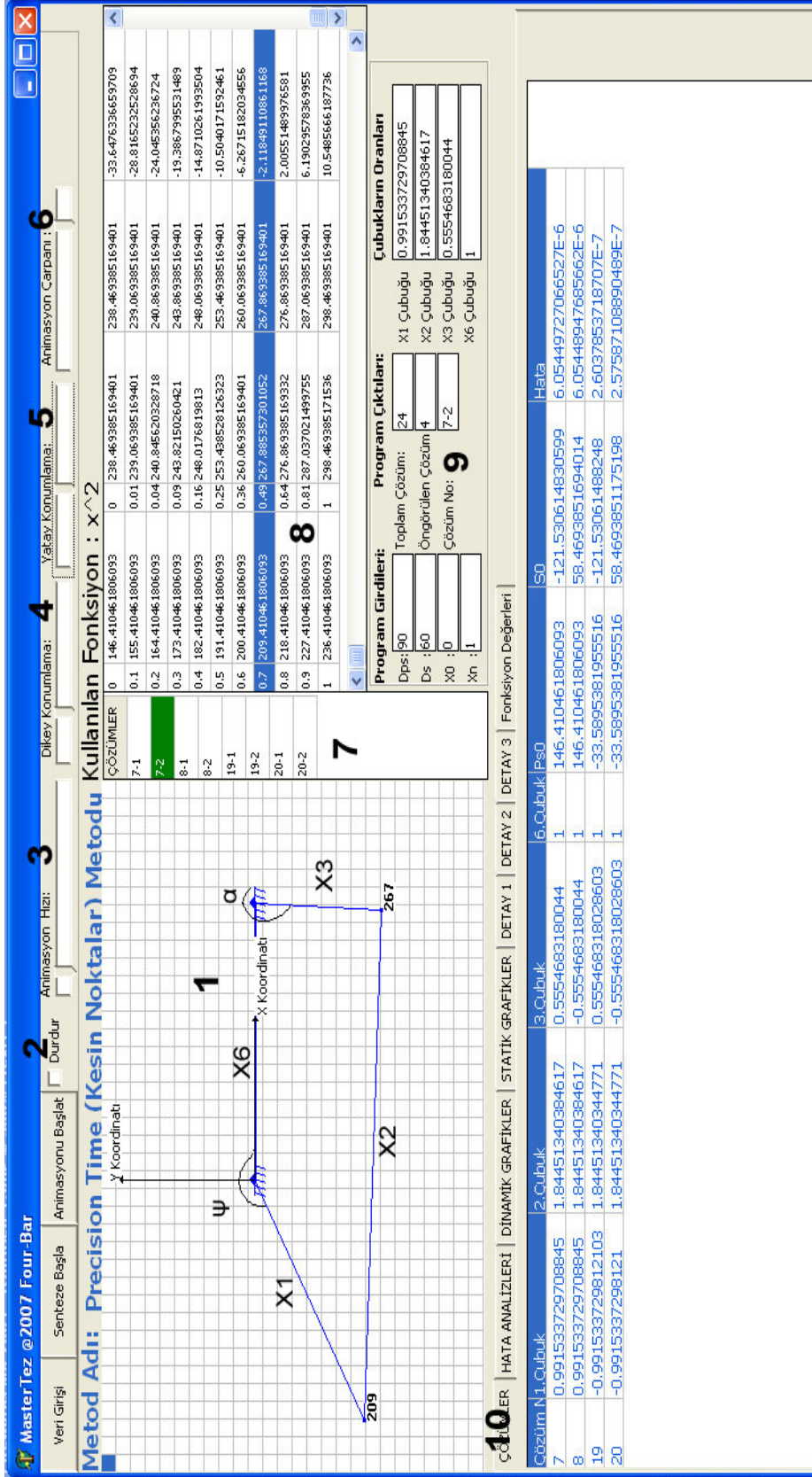
Yukarıda anlatılan çıktılar, ‘Senteze Başla’ butonunun çalıştırılması ile sağlanmaktadır. Bu aşamadan sonra, artık sentezi yapılan fonksiyona göre üretilen mekanizmanın animasyonu oluşturulabilir.

‘Animasyona Başla’ butonu, mekanizmanın animasyon işlevini gerçekleştiren butondur. Programın animasyon oluşturma prosedürlerinin oluşturulma sistemi şu şekildedir;

Senteze Başla butonunun tıklanması ile girilen fonksiyona ilişkin tüm değerler, ilgili listelere yazdırılır. 7 numaralı çözümleri gösteren listeden seçilen çözüme bağlı değerler, 8 numaralı listeye ve çubuk oranları kutucuklarına işlenir. 1 numaralı *stringgridde* de bu verilere ilişkin animasyon çizdirilir.

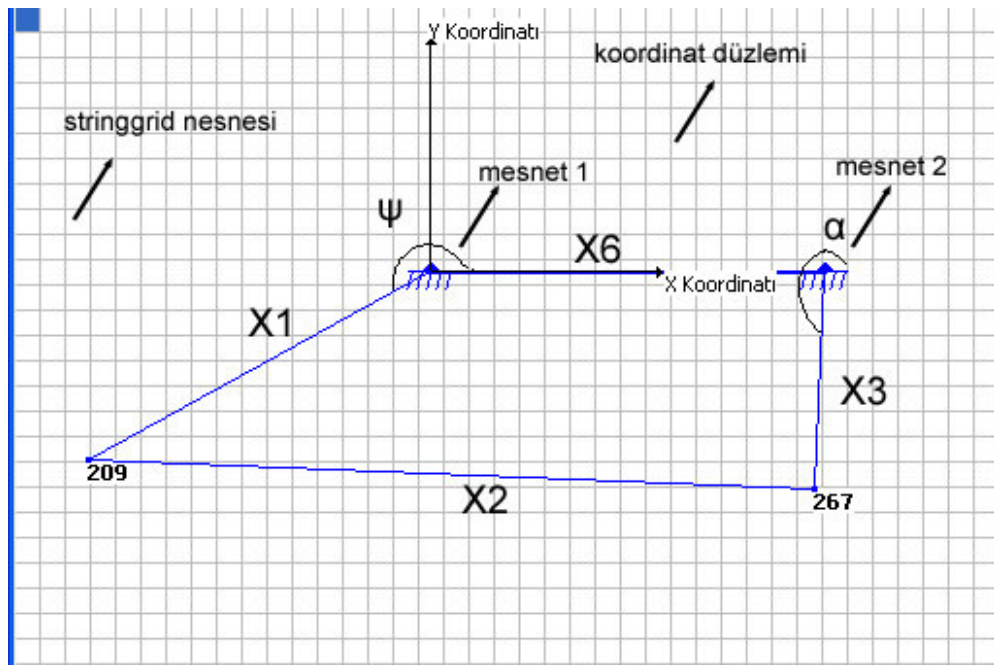
7 numaralı listedeki, 2. sütun $X1$ giriş çubuğunun animasyon sırasındaki açısını (ψ), 4. sütun $X3$ çıkış çubuğunun bulunan açısını (α), 5. sütun $X3$ çubuğunun teorikte olması gereken hareket açısını (***α -teorik***) ifade etmektedir. $X1$, $X2$ ve $X3$ çubuklarının boyları ise, yine sentez sonucu bulunan çubuk oranları menüsündeki değerlerle orantılıdır.

Animasyon için ilk olarak X ve Y koordinatları belirlenir. Animasyonun oluşturulduğu nesne, delphi' nin *stringgrid* nesnesidir. Bu nesne aslında bir tablodur. Yatay ve düşeydeki hücrelerinin aralıkları belirli oranda ayarlanarak bir koordinat düzlemi oluşturulur. Oluşturulan bu koordinat düzleminde, hareketin sağlanacağı X - Y koordinat düzleminin konumu belirlenir. bu konum, programda varsayılan olarak, düzlemsel bölgenin ortası olarak seçilmiştir. X - Y koordinat düzlemi oluşturulduktan sonra, dört çubuk mekanizmasını yere sabitleyecek mesnetler çizilir.



Şekil 4.34. Sentez Sonrası Animasyon Çizimi

Animasyon için ilk olarak X ve Y koordinatları belirlenir. Animasyonun oluşturulduğu nesne, delphi' nin *stringgrid* nesnesidir. Bu nesne aslında bir tablodur. Yatay ve düşeydeki hücrelerinin aralıkları belirli oranda ayarlanarak bir koordinat düzlemi oluşturulur. Oluşturulan bu koordinat düzleminde, hareketin sağlanacağı X - Y koordinat düzleminin konumu belirlenir. bu konum, programda varsayılan olarak, düzlemsel bölgenin ortası olarak seçilmiştir. X - Y koordinat düzlemi oluşturulduktan sonra, dört çubuk mekanizmasını yere sabitleyecek olan mesnetler çizilir.



Şekil 4.35. Animasyonda İşlenen Çıktı Değerleri

Mesnetler çizildikten sonra, sabit değerde olan ve zemin olarak kullanılan $X6$ çubuğu çizilir. $X1$ çubuğunun başlangıç noktası, 1. mesnetin orta noktası olarak kabul edilir. Bu nokta, tüm animasyonun orijini kabul edilir ve uzunluklar bu noktaya göre hesaplanır. $X1$ çubuğu için bilinen uzunluk oranı, o andaki başlangıç çubuğu açısı ψ ile programda belirli fonksiyona gönderilir ve trigonometrik yöntemler kullanılarak $x1$ çubuğunun düzlemdeki bitiş noktaları ($Xx1$, $Yy1$) tespit edilir. Bu nokta, $X1$ çubuğunun uç noktasıdır. Noktanın tespitinden sonra, cismin

canvas özelliğinden faydalanılarak, XI 'in uç noktasına programda belirlenen çizim fonksiyonları doğrultusunda bir çubuk çizilir. 3 çubuğunun da çizim algoritması aynıdır. Yani başlangıç noktası, 2. mesnetin orta noktası olarak seçilir, daha sonra XI 'in konumundaki giriş açısına denk gelen $X3$ çubuğu α çıkış açısı, $X3$ çubuk boyu uzunluğu ile programda belirli fonksiyonlara gönderilerek trigonometrik işlemler sonucu, $X3$ çubuğunun düzlemdeki uç noktası $(Xx3, Yy3)$ bulunur. Bu noktaya, nesnenin *canvas* özelliği kullanılarak belirli grafiksel işlemler dahilinde $X3$ çubuğu çizilir. İki çubuğun çizimi sonucunda, artık $X2$ bağlayıcı çubuğunun başlangıç ve bitiş uçlarının koordinat düzlemindeki konumları belirlenmiş olmaktadır. İki uç nokta $(Xx1, Yy1)$ - $(Xx3, Yy3)$ arasında grafik fonksiyonlar kullanılarak $X2$ çubuğu çizilir.

X	PSI	Y	Alac	Alth	Delta
0	146.410461806093	0	238.469385169401	238.469385169401	-33.6476336659709
0.1	155.410461806093	0.01	239.069385169401	239.069385169401	-28.8165232528694
0.2	164.410461806093	0.04	240.845620328718	240.869385169401	-24.045356236724
0.3	173.410461806093	0.09	243.82150260421	243.869385169401	-19.3867995531489
0.4	182.410461806093	0.16	248.0176819813	248.069385169401	-14.8710261993504
0.5	191.410461806093	0.25	253.438528126323	253.469385169401	-10.5040171592461
0.6	200.410461806093	0.36	260.069385169401	260.069385169401	-6.26715182034556
0.7	209.410461806093	0.49	267.885357301052	267.869385169401	-2.11849110861168
0.8	218.410461806093	0.64	276.869385169332	276.869385169401	2.00551489976581
0.9	227.410461806093	0.81	287.037021499755	287.069385169401	6.19029578369955

Şekil 4.36. Seçilen Çözümün Hata Analizleri

Çubukların Oranları	
X1 Çubuğu	0.991533729708845
X2 Çubuğu	1.84451340384617
X3 Çubuğu	0.5554683180044
X6 Çubuğu	1

Şekil 4.37 Seçilen Çözümün Ürettiği Çubuk Boyları

Yukarıdaki tablolarda, yapılan bir senteze bağlı bir çözümü için eldeki veriler listelenmiştir. Şekil 4.36 'da, bu değerler kullanılmıştır. Mesela, hareketin mavi ile gösterildiği satırında X_1 çubuğunun konumunu bulmak için X_1 çubuk uzunluğu ile psi giriş açısı değeri işlem görmüştür. Aynı şekilde X_2 için de çubuk boyu ve teorikte olması gereken açısı $Alth$ kullanılmıştır.

Şekil 4.36 da, her bir X değeri, x_0 ve x_n başlangıç ve bitiş noktası ve bu aralıktaki nokta sayısını temsil eder. Her x değeri için koordinat düzleminde değerler yukarıda tanımlandığı gibi sırasıyla çizilir. Bu işlem belirli zaman aralıklarında tüm X noktalarına uygulanmaktadır. İşte bu belirlenen zaman aralıklarında çizilen değerler, animasyonu oluşturur.

Örnekte X_0 ve X_n sırasıyla 0 ve 1 olarak verilmiş ve bu noktalar arası 10 eşit parçaya bölünmüştür. Dolayısıyla programın bulduğu çözümlerden herhangi birini seçtiğimizde, animasyonu çizebilmek için elimizde birbiri ile bağımlı 10 farklı değerler dizisi bulunmaktadır. (bu sayı, aralık sayısına göre değişir sabit değildir). Her değerler dizisi için düzlemde çubukları ve konumları çizmekle bahsekonu çözüm için gerekli animasyonu elde etmiş oluruz.

Mek-Gen programının animasyon bölümünde, şekil 4.36 da *alac'* ta gösterilen değerlerden mavi ile seçili satır, animasyonun o anda kullanmış olduğu değerleri göstermektedir. Bu sistematik yapı, animasyon işleyişinde yukarı aşağı doğru gezinmekte ve kullanıcıya rahat bir izleme ortamı sunmaktadır.

Animasyon çizimi yapılan koordinat düzleminde, her çubuğun uç noktasında, çubuğa bağlı açısal değerler de gösterilmektedir. Bu açısal değerler, yine kullanım kolaylığı ve görsellik açısından rahatlık sağlamaktadır.

Oluşturulan animasyon, şekilde gösterilen 2 numaralı '*durdur*' işaret kutucuğu işaretlenene kadar bir döngü içerisinde işlemektedir. Bu kutucuğun işaretlenmesi ile, mekanizma işlem yaptığı satırda durdurulabilmektedir. Bu sayede, animasyonun durdurulduğu aşamada kullandığı verilerin gözden geçirilme imkanı sağlanmaktadır.

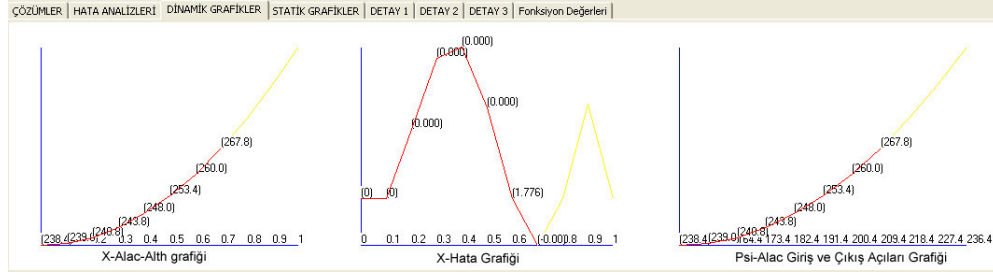
Ayrıca *durdur* butonu işaretlenmeden, programdan çıkış yapılamamaktadır. Bu özelliği kodlara eklemekteki amaç, sentez sırasında kapat düğmesine yanlışlıkla tıklansa bile çıkışın kontrol edilebilirliğini sağlamaktadır.

Oluşturulan animasyonun hızı, 3 nolu nesne sayesinde kontrol edilebilmektedir. Bu kontrol kursörü sağa sola çekilerek animasyonun hızlılık ve yavaşlık derecesi ayarlanabilmektedir. 4 nolu özellik sayesinde animasyonun dikey (Y) düzlemdeki konumu değiştirilebilmektedir. 5 nolu özellik ise animasyonun yatay (X) düzlemdeki konumu değiştirilebilmektedir. 6 nolu özellik ile, oluşturulan mekanizmanın büyüklük ve küçüklüğü ayarlanmaktadır.

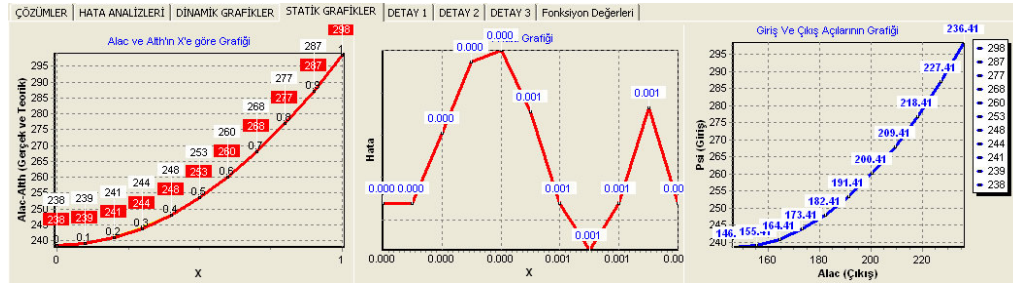
Sentez sonucu oluşan çözümlerde, mekanizmanın uzuvları ($X1, X2, X3$ çubukları) arasındaki boyutsal oranlar çok büyük olabilir. Bu gibi durumlarda, animasyonun çizildiği koordinat düzleminin default başlangıcı belirlenen orta noktaya göre yapılan çizim belirlenen boyutlara göre tam ve net olarak görülmeyebilir. Animasyonu büyültüp küçültme, yatay ve dikey hareket sağlama kursörelere sayesinde kullanıcıya bu konuda kolaylık sağlanmaktadır. Kullanıcı bu nesnelere yardımıyla oluşturduğu mekanizmayı istediği konum ve büyüklüğe getirerek animasyonu rahat görme imkanına sahiptir.

Mek-Gen programında, koordinat düzleminde yapılan animasyon ile birlikte, aynı algoritmik düzende işleyen dinamik grafikler de bulunmaktadır. Aşağıda bu grafikler hakkında bilgi verilmiştir.

Dinamik Grafikler ve İşleyişleri: Dinamik grafikler, animasyonun işleyişi sırasında, çizilen animasyona eşdeğer olarak işlemektedir. Şekil 4.38' de grafiklerin bir arada görünümü verilmektedir.



Şekil 4.38. Dinamik grafikler



Şekil 4.39. Statik grafikler

Statik grafikler, fonksiyon sentezi yapılırken, mekanizma verilerinin oluşturulması ile her bir çözüm için bir defaya mahsus olarak oluşturulmaktadır. Dinamik grafikler ise, her çözüm için çizilen animasyon sırasında, animasyonla birlikte bir döngü içerisinde sürekli çizilmektedir. Mek-Gen programında her çözümün bütün ayrıntılarının saptanması en önemli hedeflerdendir. Normal grafiklerden bağımsız ayrıca statik grafiklerin de olması, kullanıcının veri akışını rahat görebilmesini sağlamaktadır.

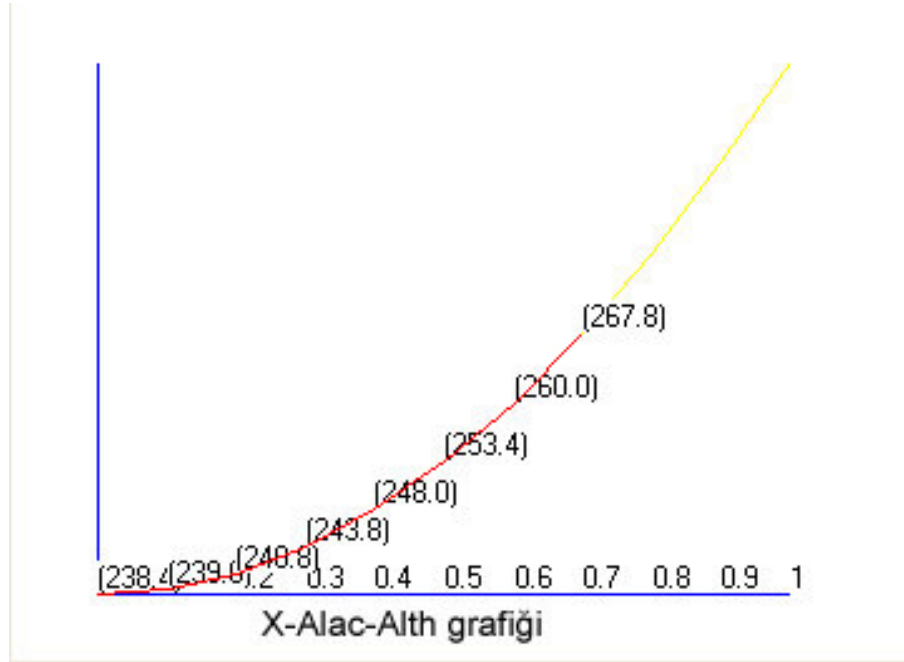
Statik grafikler, koordinat düzleminde olduğu gibi, çözümler listesinden seçilen çözüm ve buna bağlı olarak listelenen dizisel veriler yardımı ile çizilir.

ÇÖZÜMLER	X	PSI	Y	Alac	Alth	Delta
7-1	0	146.410461806093	0	238.469385169401	238.469385169401	-33.6476336659709
7-2	0.1	155.410461806093	0.01	239.069385169401	239.069385169401	-28.8165232528694
8-1	0.2	164.410461806093	0.04	240.845620328718	240.869385169401	-24.045356236724
8-2	0.3	173.410461806093	0.09	243.82150260421	243.869385169401	-19.3867995531489
19-1	0.4	182.410461806093	0.16	248.0176819813	248.069385169401	-14.8710261993504
19-2	0.5	191.410461806093	0.25	253.438528126323	253.469385169401	-10.5040171592461
20-1	0.6	200.410461806093	0.36	260.069385169401	260.069385169401	-6.26715182034556
20-2	0.7	209.410461806093	0.49	267.885357301052	267.869385169401	-2.11849110861168
	0.8	218.410461806093	0.64	276.869385169332	276.869385169401	2.00551489976581
	0.9	227.410461806093	0.81	287.037021499755	287.069385169401	6.19029578369955

Şekil 4.40. Seçilen Çözüme Göre Hata Analizlerinin Değişimi

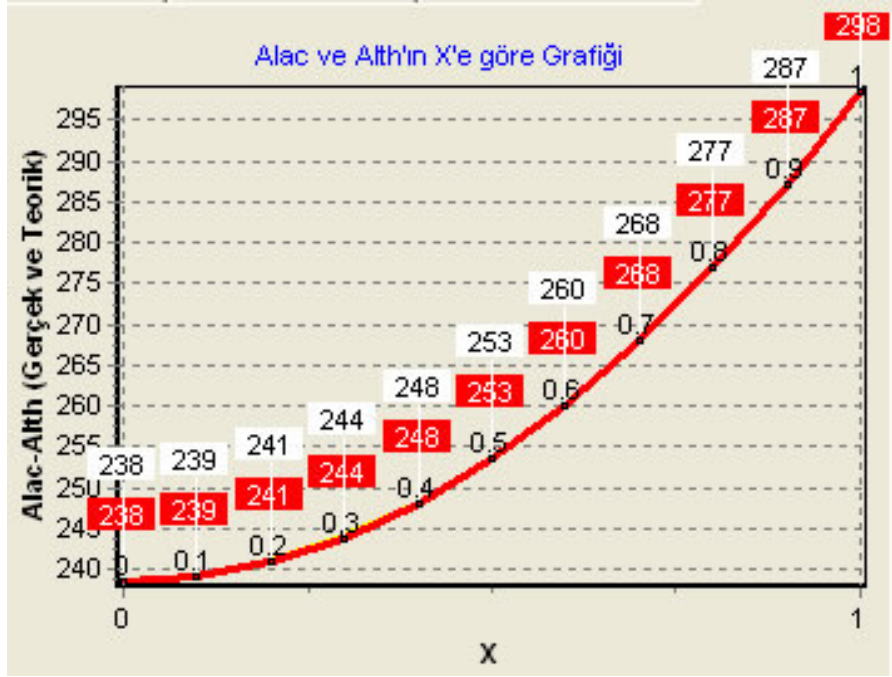
Seçilen çözüme bağlı olarak değişen dizi değerleri belirli aralıklar ile grafiklere aktarılır. Döngü içerisinde bu sistemin çalışması sonucu dinamik grafikler ortaya çıkar.

Programda toplam 3 adet dinamik grafik bulunmaktadır.



Şekil 4.41. X'e Göre Alac-Alth Hata Sapma Grafiği (Dinamik)

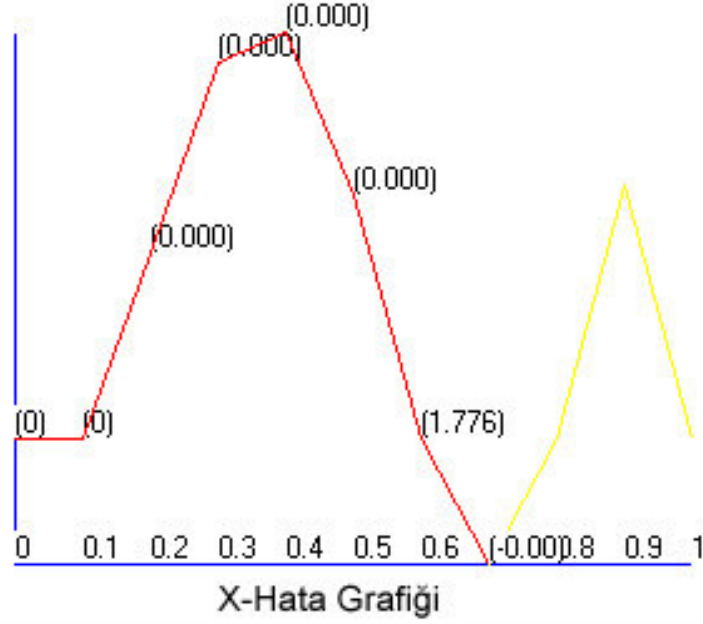
Gösterilen grafik, X_3 çubuğunun bulunan ve olması gereken çıkış açılarının, X aralıklarına göre değişimini gösterir. $Alac$ ve $Alth$ arasındaki fark, analiz edilen çözümün hata sapması olduğundan, bu grafik hata sapmasını göstermiş bulunmaktadır. Sarı ile çizilen eğri, teorikteki olması gereken eğriyi ifade eder. Bu bir defaya mahsus çizilir. Animasyon ile, çıkış açısının $Alth$ ve $Alac$ değerleri kırmızı ve mavi renk ile aynı anda çizilir. Şekilde, seçilen çözümün hata sapması çok ok az olduğundan, kırmızı eğrinin altındaki mavi çizgi çok zor görünmektedir. Eğer ki hata sapması yüksek olsaydı, programda kırmızı, mavi ve sarı eğriler arasında uzaklıklar belirecekti. Aynı grafiğin dinamik çizimi ise Şekil 4.42' deki gibidir.



Şekil 4.42. X 'e Göre $Alac$ - $Alth$ Hata Sapma Grafiği (Statik)

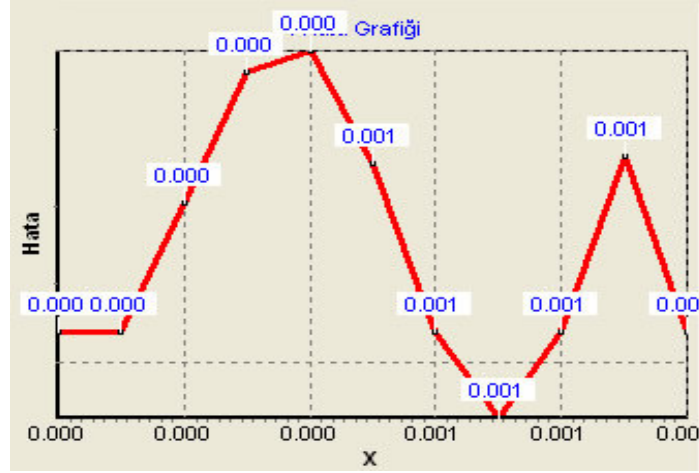
Grafiklerin tamamlanmış çizimleri görüldüğü gibi aynıdır.

2. grafiğin animasyon halinde görünüşü Şekil 4.43' deki gibidir.



Şekil 4.43. X 'e Göre Hata Sapma Grafiği (Dinamik)

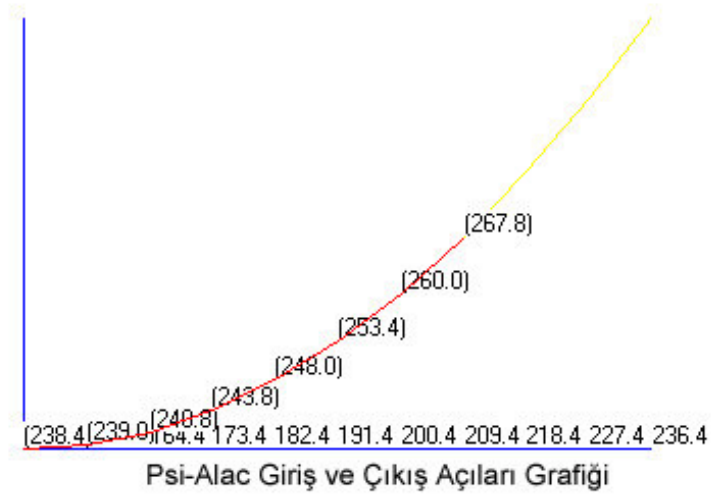
Şekilde gösterilen grafik, seçilen çözümün, X 'e göre hata sapmasını göstermektedir. Grafiğe göre, hata sapması en az olan noktaya bakılarak mekanizma tasarımı yapılabilir. Sarı ile gösterilen eğri, grafiğin izlediği yolu gösterir. Kırmızı ile gösterilen çizim ise animasyon sırasındaki konuma göre konulan çizgiler dizisinden oluşur. Çalışma prensibi 1. grafikte aynıdır. Aşağıda belirtilen hata değerleri ve aynı satırda bulunan X değeri için, zaman aralıklarıyla grafiğe çizim eklenir. Bu şekilde grafiğin dinamiği sağlanır. Aynı grafiğin sentez sırasında oluşturulan statifi ise şekil 4.44' deki gibidir.



Şekil 4.44. X'e Göre Hata Sapma Grafiği (Statik)

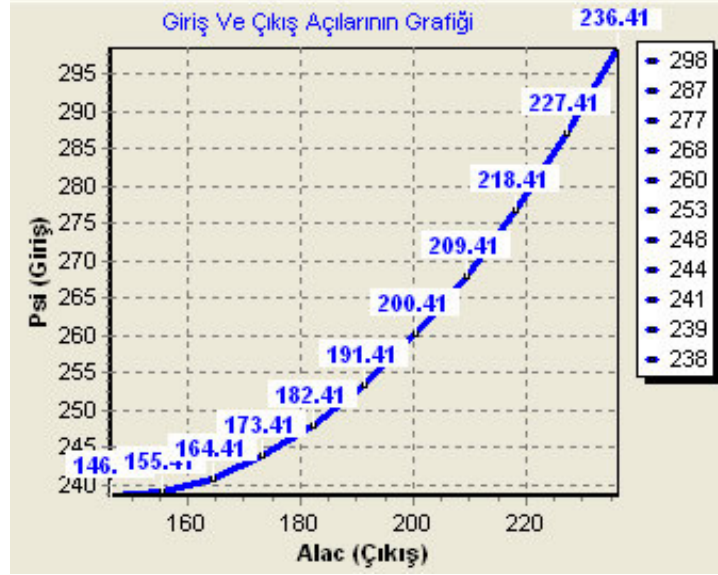
Görüldüğü gibi, statik ve dinamik grafiklerin sonuçta oluşturdukları çizimler aynıdır. Hata grafiklerinde hatanın çok küçük boyutlarda (virgülden sonraki sondan 3 rakamın 0 olması) olması bazen grafikte verilerin noktalarını sadece 0.000 olarak görebilmemizi etkin kılar.

3. dinamik grafiğin görüntüsü ise şu şekildedir.



Şekil 4.45. Psi-Alac Değişim Grafiği (Dinamik)

Bu grafik ile, $X1$ giriş kolu açısı ve $X3$ çıkış kolu açısının oluşturduğu eğim görülmektedir. Çalışma prensibi diğer iki grafikte aynıdır. Statik grafik görüntüsü Şekil 4.46' daki gibidir.



Şekil 4.46. Psi-Alac Değişim Grafiği (Statik)

Dinamik grafikler ile statik grafiklerin çalışma algoritmaları farklılık göstermektedir. Dinamik grafikler delphi programının kullanıcıya sağladığı hazır nesnelere sahiptir. Bu grafiklere program içerisinde girdi değerleri atanarak grafiğin çizimi sağlanmış olur. Object-oriented özelliğinin en canlı örneğidir.

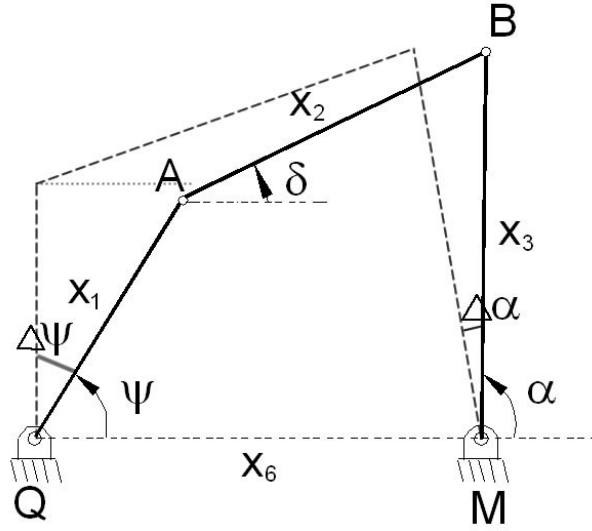
Statik grafikler, program içerisinde vektörel, grafik ve matematiksel fonksiyonlar kullanılarak gerçekleştirilir. 'Complex Math' ve 'graphics primitives' birimleri, bu grafiklerin çizimi için kullanılan birimlerdir. İçerik olarak dinamik vektörel çizim işlemlerini gerçekleştiren fonksiyonlar barındırmaktadır. Dinamik grafiklerin çalışma prensibi şu şekildedir:

Her üç grafik için ayrı ayrı resim kutuları oluşturulmuştur. bu nesnelerin *canvas* ve vektör çizim fonksiyonları için bahsedilen birimlerde parametrik fonksiyonlar tanımlanmıştır.

Grafik çizimi için var olan değerlerden en büyük ve en küçük veriler saptanır. Bu değerlere uygun nesne içerisinde analitik düzlem oluşturulur. Bu analitik düzlemde *maximum* ve *minimum* hareket noktaları, eldeki verilerin *min* ve *max* değerleri olarak atanır. Grafiği çizilecek ikili verilerin *X* ve *Y* düzlemleri için kaçar noktadan oluştukları belirlenir. Daha sonra bu noktalar yatay ve dikey olarak grafiğin omurgasını oluşturmak üzere nesne içerisine *canvas* ve *vektor* çizim yöntemleriyle çizilir. Daha sonra, her bir veri dizisinde yer alan *X* ve *Y* düzlemine denk gelen veriler düzlemde belirlenir. Grafiğin başlangıcı olarak belirlenen noktadan itibaren bu noktalar arasında yine nesnenin *canvas* özelliği kullanılarak çizgiler çizilir. Bu işlemlerin belirli aralıklarla tekrarlanması sonucu dinamik grafiklerimiz elde edilir.

4.3.4.2. Four-Bar Delta Programı

Mek-Gen programından, 4 nolu seçeneği işaretleyip tamam butonuna basmak, kullanıcıyı dört çubuk mekanizma sentezi yapma üzere Four - Bar delta adlı alt programa götürmektedir. Bu programın işleyiş yapısı, Four - Bar programı ile aynı yapıdadır.



Şekil 4.47. Dört -Çubuk Mekanizması

Four - Bar programı ile Four - Bar Delta programı arasındaki tek fark, Giriş değerlerindeki α ve δ açılarıdır. Four - Bar programında fonksiyon sentezi X_1 giriş kolu dönüş miktarı $\Delta\psi$ ve X_3 çıkış kolu dönüş miktarına $\Delta\alpha$ göre yapılırken, Four-Bar delta programında Fonksiyon sentezi X_1 Giriş kolu dönüş miktarı $\Delta\psi$ ve X_2 bağlantı kolu dönüş $\Delta\delta$ miktarına göre yapılmaktadır.

İki programın işleyiş kısımları aynı olduğundan, burada sadece Four-Bar Delta programının farklı olan bölümleri üzerinde durulacaktır.

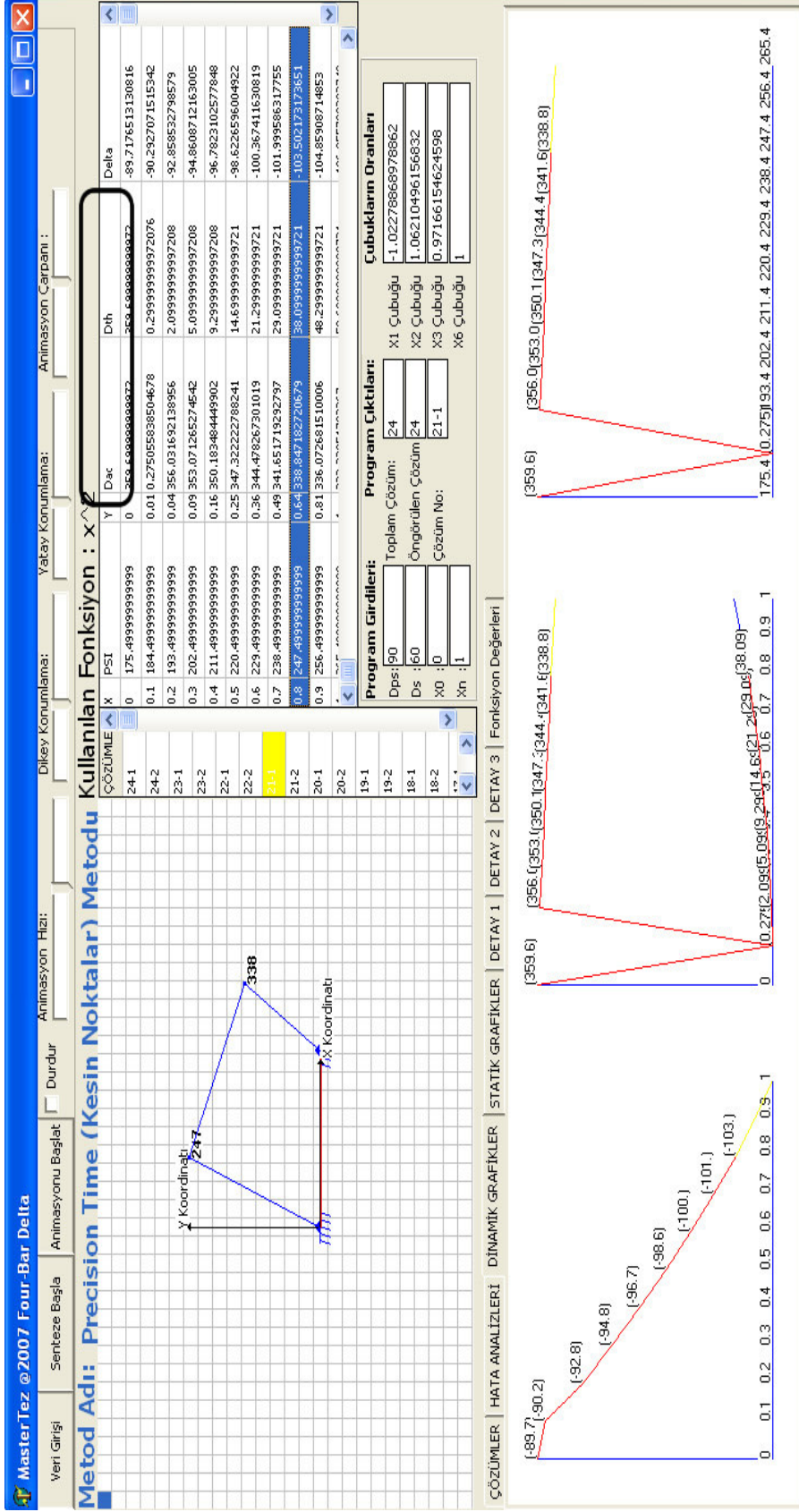
Four - Bar Delta Programı Veri Girişi: Four bar programının veri giriş menüsü Şekil 4.48’de verilmiştir. Şekilde de görüldüğü gibi programa diğer programdan farklı olarak çıkış kolu dönüş miktarı yerine bağlantı kolu dönüş miktarının girilmesi istenmektedir. Program aynı şekilde global değişkenler kullanarak girilen verileri ana program birimine ve ilgili birimlere aktarır.

Şekil 4.48. Veri Girişindeki Değişiklik

Şekilde de görüldüğü gibi, Four-bar delta programının veri giriş kısmı da Four-bar programı ile yanıdır. Tek fark X_1 açısı yerine X_2 açısının dönüş miktarı kullanılmasıdır.

Four-Bar Programı ile Mekanizma Sentezi: Four-Bar Delta Programı, Sentezde girilen X_1 çubuğu dönüş miktarını $\Delta\psi$ ve X_2 çubuğu Dönüş miktarını $\Delta\delta$ kullanmaktadır. İşleyiş, kullanılan birim ve fonksiyonlar, hata analizleri ve animasyon çizimleri Four-Bar programı ile aynıdır. Sonuç değerlerinde Four-Bar programından farklı olarak sonuçları $\Delta\delta$ için üretir. Animasyon ve grafiklerdeki değerlerde $\Delta\delta$ için üretilen değerler kullanılır.

Four-Bar Delta programının sentez sırasındaki görüntüsü Şekil 4.49' daki gibidir. $\Delta\delta$ değerine bağlı olarak bir önceki programda $\Delta\alpha$ değerine bağlı olan $Alac$ ve $Alth$ çıkış kolu açıları burada bağlantı kolu dönüş miktarı $\Delta\delta$ için $Dlac$ ve Dth değerlerini alırlar. Sentezde ana programın içerisindeki birkaç değişik fonksiyon tanımından başka hiçbir fark yoktur. Animasyon ve grafik çizimlerinde ise, $\Delta\delta$ değeri vektörel ve trigonometrik işlemlere tabi tutularak, X_2 ye bağlı değerler üretilmiştir.



Şekil 4.49. Four Bar Delta Programı Ekran Çıktısı

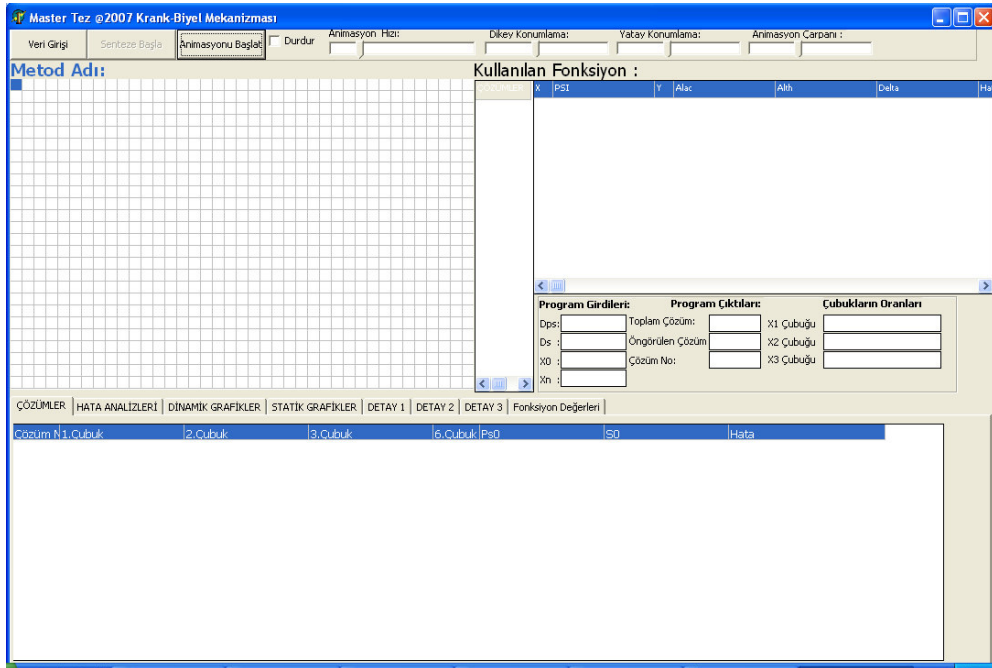
4.3.4.3. Krank-Biyel Programı

Bu bölümde, Krank-Biyel mekanizmalarının sentezi ile ilgili Krank-Biyel programının işleyiş yapısı, yapılan mekanizma sentezleri, animasyon tasarımı ve veri girişleri üzerinde durulacaktır.

Mek-Gen programından, 1 nolu seçeneği işaretleyip tamam butonuna basmak, kullanıcıyı Krank-Biyel mekanizma sentezi yapmak üzere Krank-Biyel adlı alt programa götürmektedir.

Krank-Biyel mekanizmaları basit düzenekli çubuk mekanizmaları olup, zorluk dereceleri ve işleyiş yapıları bakımından dört-çubuk mekanizmaları ile aynı yapıdadır.

Programın ana menüsü, diğer iki alt program ile aynıdır.



Şekil 4.50. Krank-Biyel Mekanizması Ana Görüntüsü

Krank-Biyel Programının Çalışma Yapısı ve Mekanizma Sentezi: Krank biyel mekanizmasının çalışma yapısı, Four-Bar programının çalışma yapısı ile aynıdır. Krank-biyel programında ilk olarak yapılış gereken diğer programlarda olduğu gibi; veri girişinin yapılmasıdır.

Veri Giriş Menüsü ve İşleyişi: Sol üst köşedeki *veri giriş* butonu, programa veri girmek amacıyla ‘*Veri Girişi*’ menüsünü ekrana getirir. Sentez programına tüm verilerin giriş yapıldığı formdur. Form yapısı diğer programlardaki gibidir.

Method Seçimi	
Precision Time (Kesin Noktalar) Metodu	
Nn (4 Çubuk için)	Dps[Giriş Kolu Dönüş Mik.]
12	315
Rr	Delta S çıkış kolu har. mik.
1	1.7525
Ss	X0 Kesin Nokta Başlangıç
1	0
Es	Xn Kesin Nokta Bitiş Açısı
1e-6	1
Maxit	Aralık Sayısı
200	10
Çıktı Dosyası	Girdi Dosyası
sonuc.out	sonuc.in

Precision İçin Kesin Noktalar	Subdomain İçin Kesin Noktalar
Xp[1] 0	Xs[1] 0
Xp[2] 0.1	Xs[2] 0.01
Xp[3] 0.6	Xs[3] 0.8
Xp[4] 0.8	Xs[4] 0.9
Xp[5] 1	Xs[5] 0.99
	Xs[6] 1

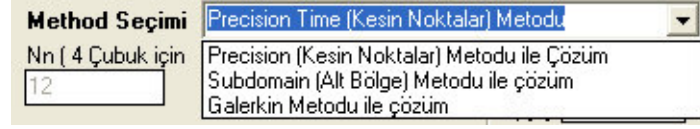
Fonksiyon:

Varsayılan Değerleri Yükle Formu Temizle

Senteze Başla

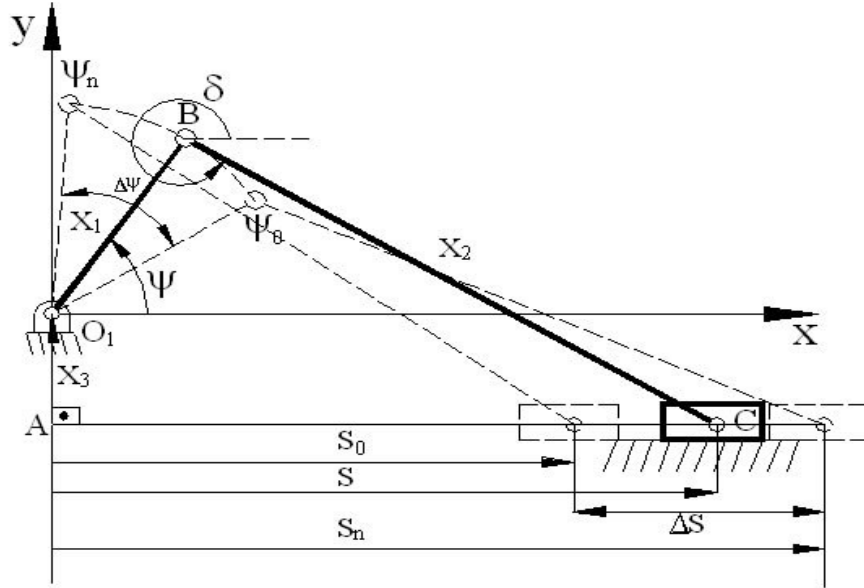
Şekil 4.51. Krank-Biyel Mekanizması Veri Girişi

Programda metin kutularında yazılı değerler program çalıştığında varsayılan olarak gelmektedir. Mekanizmanın fonksiyon sentezinin hangi yöneme göre yapılabileceği, metod seçimi menüsünden yapılabilmektedir. Precision time metodu, programda varsayılan olarak seçilidir. Kullanıcı fonksiyon sentezini toplamda 3 metoda göre de yapabilmektedir. Bu metodlar precision time, subdomain ve galerkin metodlarıdır.



Şekil 4.52. Krank-Biyel Mekanizması Metod Seçimi

N_n değerinin bulunduğu kutucuk, 4 çubuk mekanizmaları için sabit bir değer olup 12'dir. Bu değer krank-biyel mekanizmalarının sentezinde kullanılan polinomun kökünü belirtir. Değer sabit olduğundan, yani Krank-Biyel mekanizmalarının sentezini yapabilmek için kullanılacak polinomun 12 kökü olması gerektiğinden kullanıcıya bu değeri değiştirebilme yetkisi verilmemiştir. R_r ve s_s değerleri sentezde kullanılan polinomun reel ve imajiner köklerini temsil eder. Bu değerler polinoma işlenebilecek değerler olarak atandığından, kullanıcı tarafından değiştirilebilmektedir. E_s ve $maxit$; eliminasyon ve maximum iterasyon değerleridir. Bu değerler, fonksiyon sentezi sırasında eliminasyonun ve iterasyonun yapılması gereken aralıkları temsil eder. Sentez sonucu bulunan hata analizleri çok fazla olabilir. E_s ve $maxit$ değerleri ile, hata analiz sonuçları filtrelenir ve en düşük olanları görebilmemiz sağlanır.



Şekil 4.53. Krank-Biyel Mekanizması

Şekilde;

X_1 : Giriş kolunu,

X_2 : Aktarım kolunu,

X_3 : Giriş kolu ile pistonun dikey pozisyonundaki hareket uzaklığını,

Ψ : Giriş kolu başlangıç açısını;

$\Delta \Psi$: Giriş kolu hareket miktarını (açı)

S : Pistonun orijine başlangıç uzaklığını,

ΔS : Pistonun düzlemdeki hareket miktarını,

δ : X_2 kolu Başlangıç açısını

temsil eder.

Dps değeri X_1 çubuğunun dönme miktarıdır. Şekilde $\Delta \Psi$ olarak belirtilen açısal değer, programa Dps olarak girilmektedir. ΔS değeri pistonun düzlemdeki hareket miktarını belirtir. Şekilde ΔS olarak belirtilen kinematik büyüklük, programda Ds değerini temsil eder.

Programdaki açısal değerler programa derece olarak girilmektedir. X_0 ve X_n değerleri ise hata analizinin sırasıyla başlangıç ve bitiş aralığını belirtir. X_0 değeri sentezin noktasal olarak başlangıç değeridir. X_n değeri ise yine sentezin noktasal olarak bitiş değeridir. Bu değerler, her metod için kullanılmaktadır. Aralık sayısı olarak nitelendirilen değer, X_0 ve X_n başlangıç değerleri arasında yapılan sentezin kaç noktada olacağını belirtmektedir. Bu değer 10 olarak girildiğinde $X=0$ dan 1 e kadar tanımlı bir aralıkta, sentez için yapılan hata analizleri 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 ve 1.0 olmak üzere 0 ile 1 aralığını toplamda 10 adet parçaya böler. Hata analizlerinde, her bir hata analizi için biz bu noktalardaki değerleri görürüz.

Analiz nokta sayısı ise fonksiyonun yapacağı analizin kaç noktada gerçekleştirilmesi gerektiğini belirtir. 12 adet polinom kökü olan bir dört çubuk mekanizması için max. 12'dir.

'Precision Point' metodunda, fonksiyon içerisinde kullanılan değerler için program 5 adet nokta belirleyerek bu noktalar için sentez yapar. Bu noktalar kesin çözümün olması istenen noktalar olup, kullanıcı tarafından değiştirilebilmektedir.

Aynı şekilde 'Subdomain' yani 'alt bölgeler' metodunda da kesin çözümün bulunması istenen noktalar belirtilmektedir. Fonksiyon kutucuğu, sentezi yapılacak olan fonksiyonu içermektedir. Bu değer, fonksiyon girişi butonundan açılan fonksiyon girişi menüsünden yapılır.

Veri girişi menüsünden programa girilen veriler, global değişkenler ile, ana menümüz olan Unit1'e gönderilmektedir.

Global değişkenler, bir program içerisinde belirli bir kullanılmayan, taşıdığı değerleri tüm birimlere aktarabilen menülerdir. Program içerisinde etkileşimi sağlayan en önemli parçacıklardır.

Mek-Gen programı içerisinde kullanılan global değişkenler, Unit2 adlı modülde tanımlı bulunmaktadır. Programı oluşturan birimlerin, global değişkenleri tanınması için, barındıkları birimi tanımaları gerekmektedir. Yani unit2'nin diğer birimlerde tanımlı olması gerekir. Mek-Gen'de bu sistem sağlanmıştır ve veriler her modül ve fonksiyonda kullanılır hale getirilmiştir.

Biraz önce de bahsedildiği gibi, global değişkenler ile veri giriş programından girilen veriler unit1'e, oradan da kullanılacakları diğer fonksiyon ve modüllere gönderilebilmektedirler.

Fonksiyon Girişi Menüsü ve İşleyişi: Mek-Gen programında fonksiyon girişi menüsü tüm programlar için aynı olduğundan ve daha önce Four-Bar programında

detaylı olarak fonksiyon girişinden bahsedildiğinden, bu bölümde bahsedilmeyecektir.

Krank-Biyel Programı İle Mekanizma Sentezi: Girilen tasarım değerleri, global değişkenler ile Krank-Biyel ana programına gönderildikten sonra, Senteze başla butonu ile girilen veriler işlenmeye başlar.



Şekil 4.54. Krank-Biyel Mekanizması Senteze Başlama

Girilen veriler ilk olarak sentez işlemine tabi tutulur.

Sentez işleminde şu işlemler gerçekleşir:

- 1) Seçilen metoda bağlı olarak (precision - kesin noktalar - subdomain) girilen ilk değerler fonksiyonda kullanılmak üzere işlenir.
- 2) Kullanılacak olan X_1 - X_2 ve X_3 çubuk boyları tespit edilir.
- 3) Kullanılan fonksiyonun girilen x değişkenlerine göre değerleri bulunur. X değerinin, x_0 ve x_3 tanım aralığında bulunan ve *veri girişi* menüsünden girilen noktalara ayrılmış kesimlerdeki fonksiyon değerleri bulunur.
- 4) Krank-Biyel mekanizması için her bir polinomun toplamda 12 adet kökü vardır. Bu kökler bulunur. Her polinomun kökü reel ve imajiner kısımlardan oluşmaktadır. Hata analizi sırasında polinomların reel kökleri ile kompleks kökleri birbirinden ayrılır ve analize yarayan reel kökleri alınır.
- 5) Çubuk boy oranları bulunan mekanizmanın giriş kolu başlangıç açısı ve çıkış kolu başlangıç açısı hesaplanır.

- 6) Giriş kolunun başlangıç açısı ve çıkış kolunun düzlemde bağlı salınım miktarı, noktaları daha önce belirlenen x ve bağlı $f(x)$ fonksiyonuna tabi tutulması ile hata değerleri saptanır.
- 7) Mekanizmayı oluşturacak fonksiyonların toplam çözümleri bulunur. Bu çözümlerden hata analizleri yapılır ve hata sapmaları en az olan fonksiyon değerleri mekanizma sentezi için uygun kabul edilir.
- 8) Bulunan tüm değerler uygun bölümlere işlenir.

Sentez işleminin bitmesinin ardından girilen fonksiyona bağlı değerler saptanmış olur.

Programda elde edile tüm çıktılar, kullanılan '*Stringgrid*' ve '*Editbox*' nesnelere yazdırılmaktadır. *Stringgrid* nesnelere, birbiri ile bağımlı sonuçları bir arada ve düzenli bir şekilde görebilmek için tasarlanmış ideal nesnelere.

Çözümler stringgridinde, programın bulmuş olduğu, hata analizleri en uygun çözümlerin listesi yer almaktadır. Uygun çözümler buradan tek tek seçilebilmekte, çözümün detayları görülebilmektedir. Çözümler gridinin yanındaki gridde, seçilen çözüme ait olan değerler görülmektedir. Bunlar,

X : Sentezin yapılması istenen aralıktaki noktalar;

Psi : X noktasındaki X_1 giriş kolunun açılma değeri,

Y : Sentezin yapılması istenen aralıktaki X değerlerine uygun fonksiyon değerleri;

$Alac$: X_1 ye bağlı pistonun olması gereken hareket başlangıç S_0 uzaklığını,

$Alth$: X_2 ye bağlı pistonun hareket başlangıç S_0 uzaklığının teorik değerini,

$Delta$: $Alac$ ve $Alth$ değerleri arasındaki sapma miktarını;

Hata: bu noktadaki yapısal hatayı göstermektedir.

Program girdileri kısmında, *veri giriři* menüsünden girmiş olduđumuz $\Delta\psi$, ΔS , X_0 ve X_n başlangıç deđerleri yer almaktadır. Toplam çözüm sayısı, hata analizleri yapılırken bulunan çözüm sayısını, öngörülen çözüm sayısı ise programda görebildiđimiz, filtrelenmiş çözümleri ifade etmektedir.

1 nolu kısım animasyon penceresidir.

2 nolu kısımda çözümler sıralanmaktadır.

3 nolu kısım, seçilen çözüme bađlı hata analizlerini ekrana getirir.

4 nolu kısım toplam çözümleri gösterir.

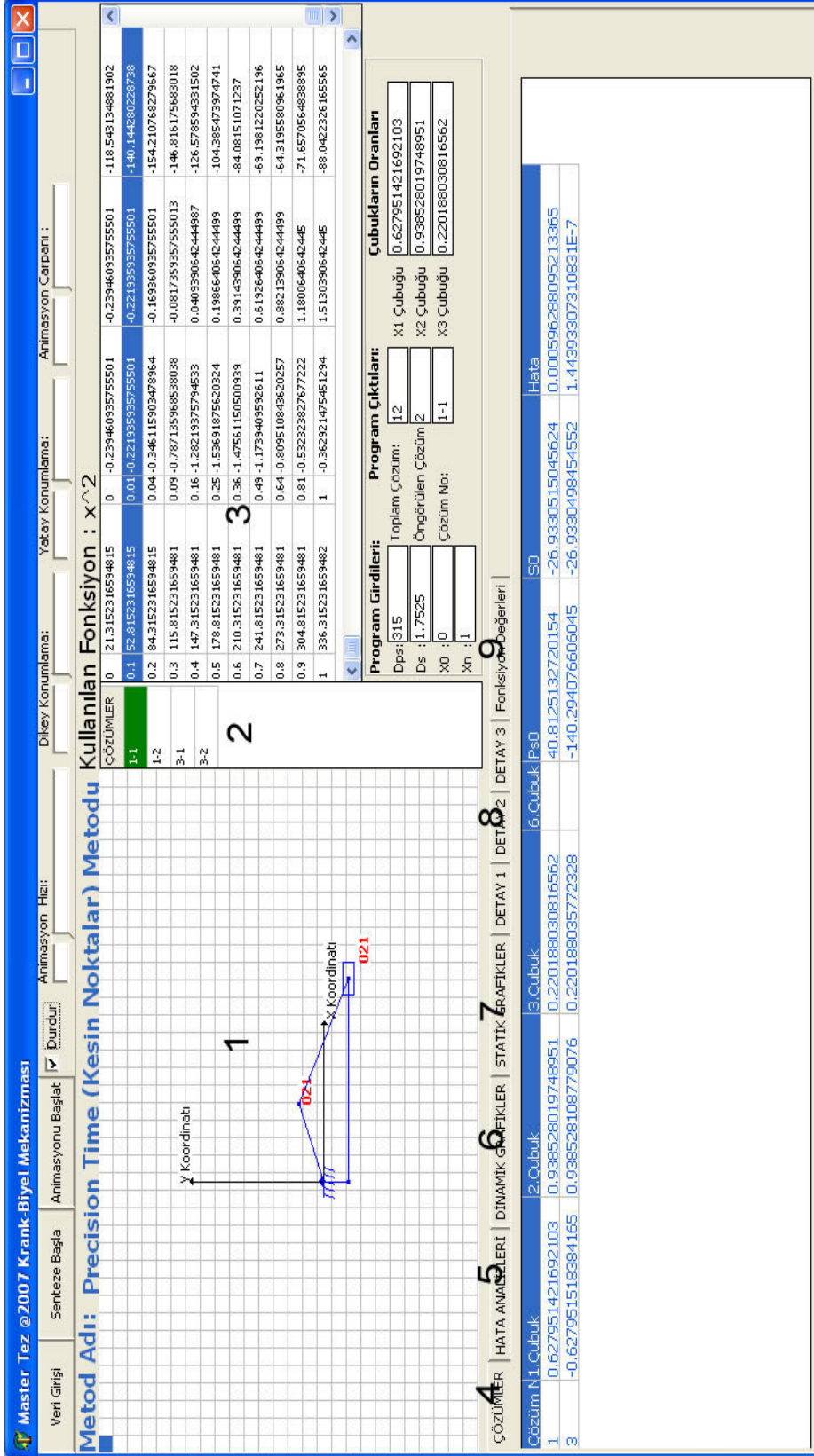
5 nolu kısım hata analizlerini gösterir.

6 nolu kısım dinamik grafikleri gösterir.

7 nolu kısım statik grafikleri gösterir.

8 nolu kısım çözümler ile ilgili detayları gösterir

9 nolu kısım, çözümlerde kullanılan fonksiyonların x deđişkenine bađlı deđerlerini gösterir.



Şekil 4.55. Krank-Biyel Mekanizması Sentez Çıktı Ekranı

X_1, X_2 ve X_3 çubuk boyları ise, sentez sonu oluşan çözümlerden, seçili olan çözüme ait çubuk boyunu göstermektedir.

Programın alt kısmında bulunan sayfalardan çözümler kısmında, sentez sonucu bulunmuş çözümlerden öngörülen çözümler ve bu çözümlere bağlı tasarlanabilecek mekanizma uzuvlarının boyutları ile giriş çıkış açıları; mekanizmanın tasarlanması halinde karşılaşılabilecek hata sapması gösterilmektedir.

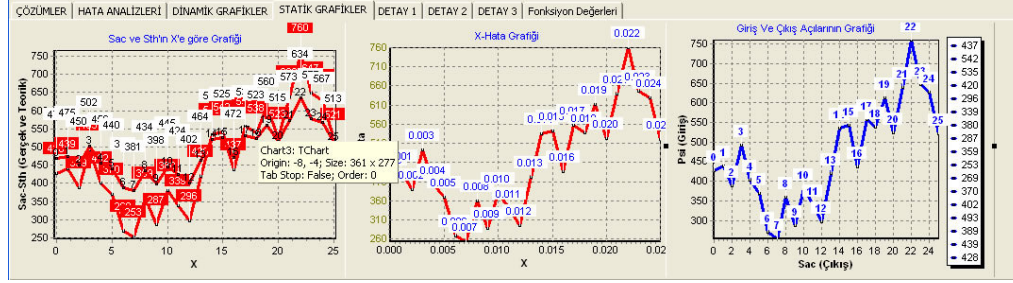
Hata analizleri kısmında, öngörülen çözümlere bağlı değerler ve bu değerlerin hata analizlerinin tümü bir arada yazılmıştır. 3 nolu kısımda, seçilen çözüme göre üretilen değerler görünürken, programın 5 nolu hata analizleri kısmında tüm çözümler ve bu çözümlere bağlı hata analizleri görülmektedir.

ÇÖZÜMLER		HATA ANALİZLERİ	DİNAMİK GRAFİKLER	STATİK GRAFİKLER	DETAY 1	DETAY 2	DETAY 3	Fonksiyon Değerleri
		HATA	ANALİZ SONUÇLARI					
			1inci Çözüm İçin	Hata Analiz Sonuçları				
		1-1	1inci Krank-Biyel	Analiz Sonuçları				
X	Psi	Y	Alac	Alth	Delta	Hata		
0	21.3152316594815	0	-0.239460935755501	-0.239460935755501	-118.543134881902	2.77555756156289E-17		
0.1	52.8152316594815	0.01	-0.221935935755501	-0.221935935755501	-140.144280228738	2.77555756156289E-17		
0.2	84.3152316594815	0.04	-0.346115903478964	-0.169360935755501	-154.210768279667	0.176754967723462		
0.3	115.815231659481	0.09	-0.787135968538038	-0.0817359357555013	-146.816175683018	0.705400032782536		
0.4	147.315231659481	0.16	-1.28219375794533	0.0409390642444987	-126.578594331502	1.32313282218983		
0.5	178.815231659481	0.25	-1.53691875620324	0.198664064244499	-104.385473974741	1.73558282044774		
0.6	210.315231659481	0.36	-1.47561150500939	0.391439064244499	-84.08151071237	1.86705056925389		
0.7	241.815231659481	0.49	-1.1739409592611	0.619264064244499	-69.1981220252196	1.7932050235056		
0.8	273.315231659481	0.64	-0.809510843620257	0.882139064244499	-64.3195580961965	1.69164990786476		
0.9	304.815231659481	0.81	-0.532323827672222	1.18006406424445	-71.6570564838895	1.71238789192172		

Şekil 4.56. Krank-Biyel Mekanizması Hata Analizleri

Dinamik grafikler kısmında, veriler animasyondan sonra belirdiğinden, programın animasyon kısmında anlatılacaktır.

Statik grafikler kısmında, 2 numaralı tablodan seçilen çözümün grafikleri verilmektedir. İlk grafik, S_{ac} ve S_{th} değerlerinin x değişkenine göre değişimini göstermektedir. S_{ac} ve S_{th} arasındaki fark, hata sapmasını vermektedir. İkinci grafikte, hatanın x değişkenine göre değişimi yansıtılmaktadır. 3. grafikte ise, giriş açısının, çıkış kolu değişimine göre oluşturduğu değerler görülmektedir.



Şekil 4.57. Krank-Biyel Mekanizması Statik Grafikler

Detay1- Detay2 ve Detay 3 sayfalarında ise, programın sentezi sırasında oluşan diğer veriler yer almaktadır. Bu veriler, yukarıda anlatılan bulguların elde edilmesi sırasında toplanmış bilgiler olup, detaylar üzerinde çalışmak isteyen tasarımcıların her veriye hakim olabilmesi düşüncesiyle listelenmiştir.

ÇÖZÜMLER HATA ANALİZLERİ DİNAMİK GRAFİKLER STATİK GRAFİKLER DETAY 1 DETAY 2 DETAY 3 Fonksiyon Değerleri						
No	aaa değerleri	Kat	Ps0	S0	Ps0/kat	S0/kat
1	-0.000330507729083687	0.0174532925199433	0.372020973283287	-0.239460935755501	40.8125132720154	-26.9330515045624
2	-0.007873648806212	0.0174532925199433	2.78816348006094	-0.3242125	140.538339579396	-35.9266845623231
3	0.0233476761681282	0.0174532925199433	-2.76957162731771	-0.239460920446843	-140.294076606045	-26.9330498454552
4	0.225941852808245	0.0174532925199433	3.06305283725005	-0.31545	143.839212010861	-35.0157600114452
5	-0.486849672133252	0.0174532925199433	-1.62752080865622	-0.0881315307145998	-116.864250770929	-10.0731034783148
6	0.304944001887542	0.0174532925199433	-1.62752080865622	-0.0881315307145998	-116.864250770929	-10.0731034783148
7	0.4864476338669825	0.0174532925199433	0.757389704542859	-0.0850367327745533	74.2798263272479	-9.72110489721954
8	-0.304944001887542	0.0174532925199433	0.757389704542859	-0.0850367327745533	74.2798263272479	-9.72110489721954
9	-0.486849672133252	0.0174532925199433	-0.353429207415726	-0.324212499836883	-38.9297915807577	-35.9266845454092
10	0.225941852808245	0.0174532925199433	-0.0785397369877943	-0.315449994243358	-8.98155364486325	-35.015759411484
11	0.0233476761681282	0.0174532925199433	-1.845685683994	-0.0087625	-123.101970678706	-1.00408283819549
12	0.007873648806212	0.0174532925199433	1.29590696960579	-0.00876250000000003	104.688111122868	-1.00408283819549
13	-0.000330507729083687					

ÇÖZÜMLER HATA ANALİZLERİ DİNAMİK GRAFİKLER STATİK GRAFİKLER DETAY 1 DETAY 2 DETAY 3 Fonksiyon Değerleri						
Result[]	Imaj[]	i	Fonk	Çözüm No	Fonksiyon Değerleri	Eps Değeri
0.188185914258394	0	1	-3.95321426296857E-10	1	-1.38777878078145E-17	1.38777878078145E-17
5.59981456068062	0	2	-1.96848205448603E-8	1	0	1.38777878078145E-17
-5.31389322185206	0	3	1.08390299592978E-8	1	-4.85722573273506E-17	5.05159101305033E-17
25.4516995793572	0	4	-0.00207216265711913	1	1.3394743220152E-7	1.3394743220152E-7
-1.05839639363173	-1.24086737084685	5	-0.476364268394743	1	3.29363990303477E-7	3.5559492493183E-7
-1.05839639363173	1.24086737084685	6	-0.476364268394743	2	3.64291929955129E-17	3.64291929955129E-17
0.397900150459385	-0.466499426079501	7	0.00639399056840343	2	3.72965547335014E-17	5.21355837916235E-17
0.397900150459385	0.466499426079501	8	0.00639399056840343	2	3.81639164714898E-17	6.46111725457466E-17
-0.17857735949461	0	9	-3.05356291420067E-10	2	0.909795216613711	0.909795216613711
-0.0392900672704462	0	10	-3.39374920770548E-10	2	0.0414532660616262	0.910739100643189
-1.32103852907481	0	11	-2.60786197866548E-16	3	5.55111512312578E-17	5.55111512312578E-17
0.756980192470503	0	12	-8.35894463997906E-18	3	5.55111512312578E-17	5.55111512312578E-17

ÇÖZÜMLER		HATA ANALİZLERİ	DİNAMİK GRAFİKLER	STATİK GRAFİKLER	DETAY 1	DETAY 2	DETAY 3	Fonksiyon Değerleri		
xth(psi)/kat	psi	psi/kat	y(xth(psi))	sac	sac/kat	sth	sth/kat			
0	0.0174532925199433	0.372020973283287	21.3152316594815	0	-0.239460935755501	-13.7201009770436	-0.239460935755501	-13.7201		
0.1	0.0174532925199433	0.9217996876615	52.8152316594815	0.01	-0.221935935755501	-12.7159924410768	-0.221935935755501	-12.7159		
0.2	0.0174532925199433	1.47157840203971	84.3152316594815	0.04	-0.346115903478964	-19.830980491702	-0.169360935755501	-9.70366		
0.3	0.0174532925199433	2.02135711641793	115.815231659481	0.09	-0.787135968538038	-45.0995689001719	-0.0817359357555013	-4.68312		
0.4	0.0174532925199433	2.57113583079614	147.315231659481	0.16	-1.28219375794533	-73.4642908482858	0.0409390642444987	2.34563E		
0.5	0.0174532925199433	3.12091454517436	178.815231659481	0.25	-1.53691875620324	-88.0589581849417	0.198664064244499	11.38261		
0.6	0.0174532925199433	3.67069325955257	210.315231659481	0.36	-1.47561150500939	-84.5463114379856	0.391439064244499	22.4278C		
0.7	0.0174532925199433	4.22047197393078	241.815231659481	0.49	-1.1739409592611	-67.2618623632004	0.619264064244499	35.48121		
0.8	0.0174532925199433	4.770250688309	273.315231659481	0.64	-0.809510843620257	-46.3815548095155	0.882139064244499	50.5428A		
0.9	0.0174532925199433	5.32002940268721	304.815231659481	0.81	-0.532323827677222	-30.4999086601542	1.1800640642445	67.61265		
1	0.0174532925199433	5.86980811706543	336.315231659482	1	-0.362921475451294	-20.7938688380199	1.5130390642445	86.6907E		
0	0.0174532925199433	0.372020973283287	21.3152316594815	0	0	0	-0.239460935755501	-13.7201		
0.1	0.0174532925199433	0.9217996876615	52.8152316594815	0.01	0	0	-0.221935935755501	-12.7159		

Şekil 4.58. Krank-Biyel Mekanizması Detaylar Menüsü

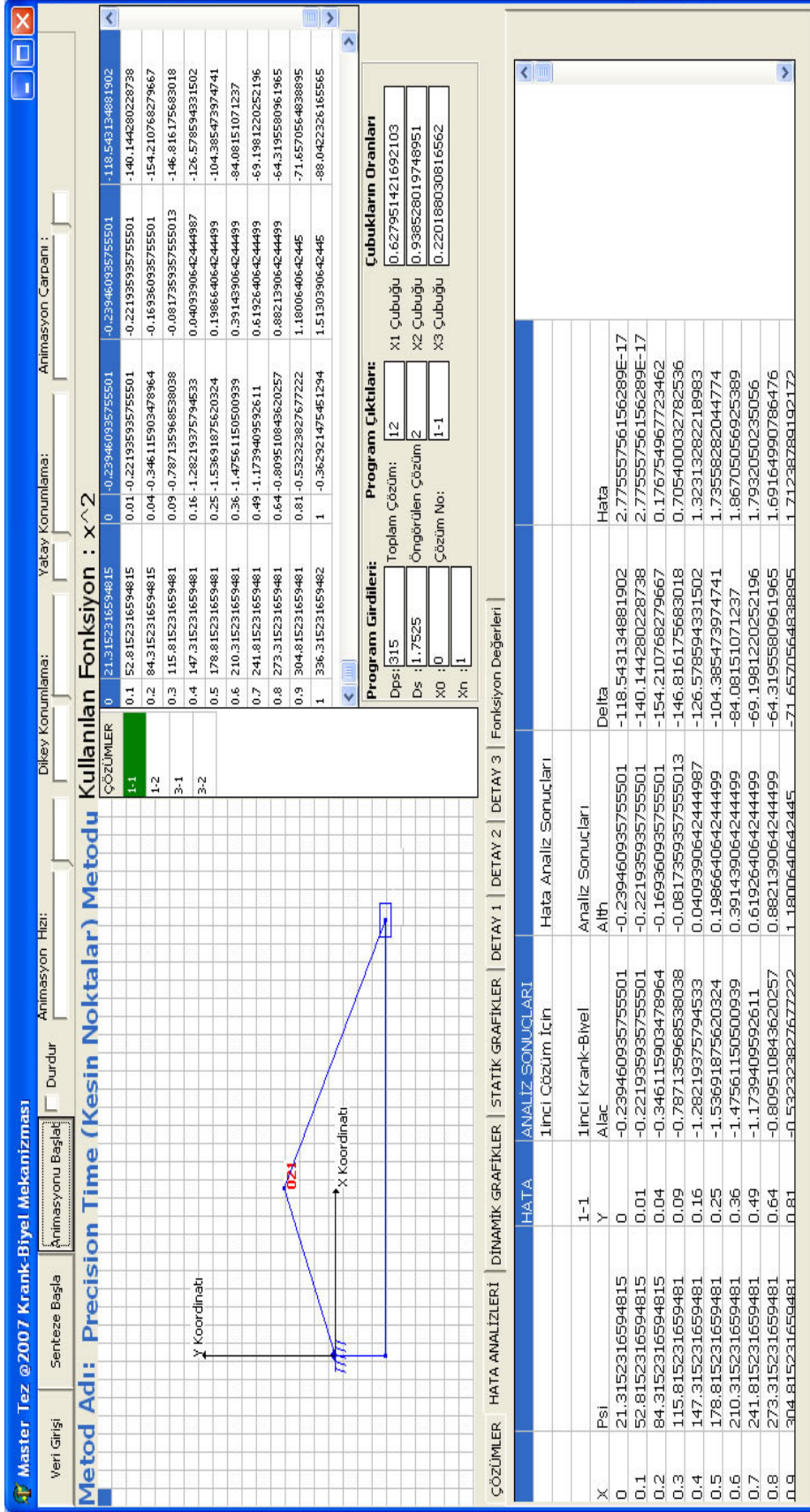
Fonksiyon değerleri sayfasında ise mekanizmanın tasarımında kullanılan $f(x)$ fonksiyonun ve X bağımsız değişkeninin program akışı sırasında aldığı değerler listelenmiştir.

ÇÖZÜMLER		HATA ANALİZLERİ	DİNAMİK GRAFİKLER	STATİK GRAFİKLER	DETAY 1	DETAY 2	DETAY 3	Fonksiyon Değerleri			
Kullanılan Fonksiyon: x^2								x_0	$y(x_0)$	x_n	$y(x_n)$
NO	xth(p0)	Fonk. Değ. (Y(xth(p0)))	ps	Fonk. Değ. (Y(xth(ps)))							
1	0	0	0	0				0	0	1	
2	0	0	0.1	0.01							
3	0	0	0.2	0.04							
4	0	0	0.3	0.09							
5	0	0	0.4	0.16							
6	0	0	0.5	0.25							
7	0	0	0.6	0.36							
8	0	0	0.7	0.49							

Şekil 4.59. Krank-Biyel Mekanizması Fonksiyonunun Sentez Sırasındaki Değişimi

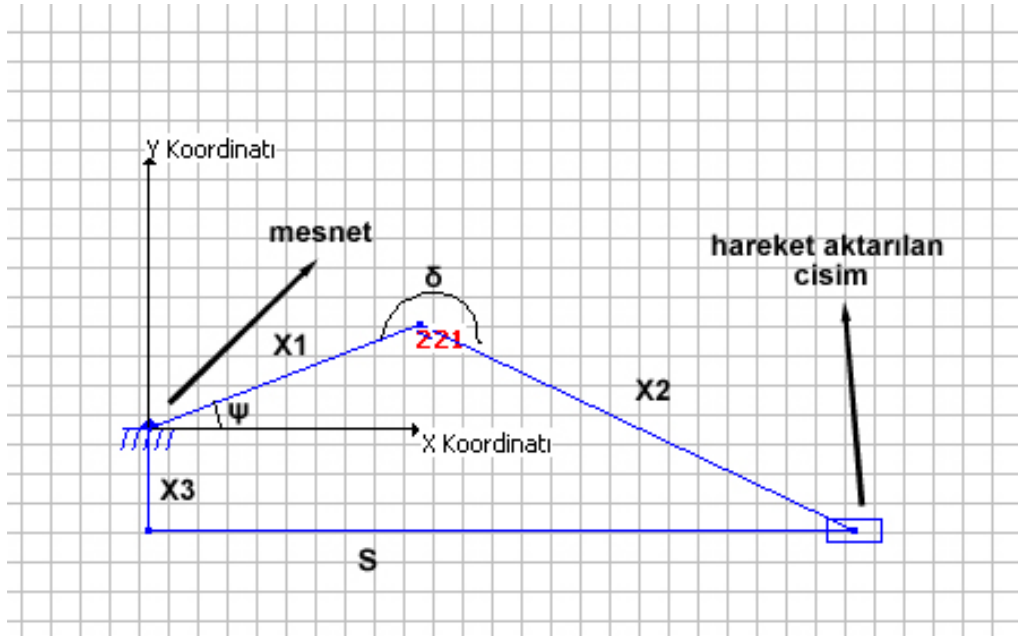
Yukarıda anlatılan çıktılar, Four-Bar programının çalışma yapısı ile aynı özelliktedir. Sadece, girilen değerlere bağlı değişkenlerde farklılıklar bulunmaktadır. Bu aşamadan sonra, artık sentezi yapılan fonksiyona göre üretilen mekanizmanın animasyonu oluşturulabilir.

'Animasyona Başla' butonu, Krank- Biyel mekanizmasının animasyonunu gerçekleştiren butondur ve çalışma prensibi Four-Bar programı ile aynıdır. Bu konuya daha önce değinildiğinden, Şekil 4.60' da sadece animasyon sonucu oluşan ekran çıktısı verilmiştir.



Şekil 4.60. Krank-Biyel Mekanizması Animasyon Sonrası Çıktı Ekranı

Animasyon için ilk olarak X ve Y koordinatları belirlenir. Animasyonun oluşturulduğu nesne, delphi 'nin *stringgrid* nesnesidir. Bu nesne aslında bir tablodur. Yatay ve düşeydeki hücrelerinin aralıkları belirli oranda ayarlanarak bir koordinat düzlemi oluşturulur. Oluşturulan bu koordinat düzleminde, hareketin sağlanacağı X - Y koordinat düzleminin konumu belirlenir. Bu konum, programda varsayılan olarak, düzlemsel bölgenin ortası olarak seçilmiştir. X - Y koordinat düzlemi oluşturulduktan sonra, dört çubuk mekanizmasını yere sabitleyecek olan mesnet çizilir.



Şekil 4.61. Krank-Biyel Mekanizması Animasyonda Çubukların Yerleşimi

Mesnet çizildikten sonra X_1 çubuğunun başlangıç noktası, 1. mesnetin orta noktası olarak kabul edilir. Bu nokta, tüm animasyonun orijini kabul edilir ve uzunluklar bu noktaya göre hesaplanır. X_1 çubuğu için bilinen uzunluk oranı, o andaki başlangıç çubuğu açısı ψ ile programda belirli fonksiyona gönderilir ve trigonometrik yöntemler kullanılarak X_1 çubuğunun düzlemdeki bitiş noktaları (X_{x1} , Y_{y1}) tespit edilir. Bu nokta, X_1 çubuğunun uç noktasıdır. Noktanın tespitinden sonra, cismin *canvas* özelliğinden faydalanılarak, X_1 'in uç noktasına programda belirlenen çizim fonksiyonları doğrultusunda bir çubuk çizilir.

X_3 çubuğunun da çizim algoritması aynıdır. Yani başlangıç noktası, 1. mesnetin orta noktası olarak seçilir, X_3 çubuğu Y koordinatına bağlı bir uzunluktur. Sentez sonucu bulunan X_3 uzunluğu sayesinde Y koordinat düzlemindeki uç noktası olan (X_{x3}, Y_{y3}) bulunur. Bu noktaya, nesnenin *canvas* özelliği kullanılarak belirli grafiksel işlemler dahilinde X_3 çubuğu çizilir. S çubuğu çizilirken, başlangıç orijini olan X_3 'ün bitiş noktası alınır. Yatay düzlemde hareket eden S çubuğu için bilinen X uzunluk değeri ile (S_{x2}, S_{y2}) uç nokta bulunur. Nesnenin *canvas* özelliği kullanılarak bu noktaya çizim yapılır.

ÇÖZÜMLER	X	PSI	Y	Sac	Sth	Delta
1-1	0	-158.684765304484	0	-0.239460920446843	-0.239460920446843	-118.543136866758
1-2	0.1	-127.184765304484	0.01	-0.221935920446843	-0.221935920446843	-140.14428344428
3-1	0.2	-95.6847653044839	0.04	-0.346115915442967	-0.169360920446843	-154.210771690255
3-2	0.3	-64.1847653044839	0.09	-0.787136084743778	-0.0817359204468434	-146.816176017329
	0.4	-32.6847653044839	0.16	-1.28219394651835	0.0409390795531566	-126.578592515362
	0.5	-1.18476530448389	0.25	-1.53691895222328	0.198664079553157	-104.385470922378
	0.6	30.3152346955161	0.36	-1.47561165366812	0.391439079553157	-84.0815068214857
	0.7	61.8152346955161	0.49	-1.17394103437384	0.619264079553157	-69.198117821532
	0.8	93.3152346955161	0.64	-0.809510860489735	0.882139079553157	-64.3195546368135
	0.9	124.815234695516	0.81	-0.532323820572286	1.18006407955316	-71.6570547208061

Şekil 4.62. Çözüme Göre Hata Analizlerinin Görülmesi

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps:	315	Toplam Çözüm:	12	X1 Çubuğu	0.627951421692103
Ds :	1.7525	Öngörülen Çözüm	2	X2 Çubuğu	0.938528019748951
X0 :	0	Çözüm No:	1-1	X3 Çubuğu	0.220188030816562
Xn :	1				

Şekil 4.63. Çözüme Göre Hata Çubuk Boylarının Görülmesi

Yukarıdaki tablolarda, yapılan bir senteze bağlı bir çözümü için eldeki veriler listelenmiştir. Şekil 4.63 'de, bu değerler kullanılmıştır. Mesela, hareketin mavi ile gösterildiği satırında X_1 çubuğunun konumu bulmak için X_1 çubuk uzunluğu ile *psi*

giriş açısı değeri işlem görmüştür. Aynı şekilde S için de çubuk boyu ve teorikte olması gereken açısı $St\theta$ kullanılmıştır.

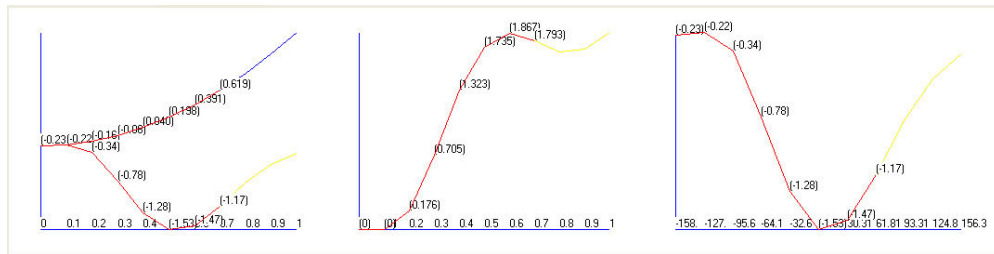
Şekil 4.63'de, her bir X değeri, x_0 ve x_n başlangıç ve bitiş noktası ve bu aralıktaki nokta sayısını temsil eder. Her x değeri için koordinat düzleminde değerler yukarıda tanımlandığı gibi sırasıyla çizilir. Bu işlem belirli zaman aralıklarında tüm X noktalarına uygulanmaktadır. İşte bu belirlenen zaman aralıklarında çizilen değerler, animasyonu oluşturur.

Örnekte X_0 ve X_n sırasıyla 0 ve 1 olarak verilmiş ve bu noktalar arası 10 eşit parçaya bölünmüştür. Dolayısıyla programın bulduğu çözümlerden herhangi birini seçtiğimizde, animasyonu çizebilmek için elimizde birbiri ile bağımlı 10 farklı değerler dizisi bulunmaktadır. (bu sayı, aralık sayısına göre değişir sabit değildir). Her değerler dizisi için düzlemde çubukları ve konumları çizmekle bahsekonu çözüm için gerekli animasyonu elde etmiş oluruz.

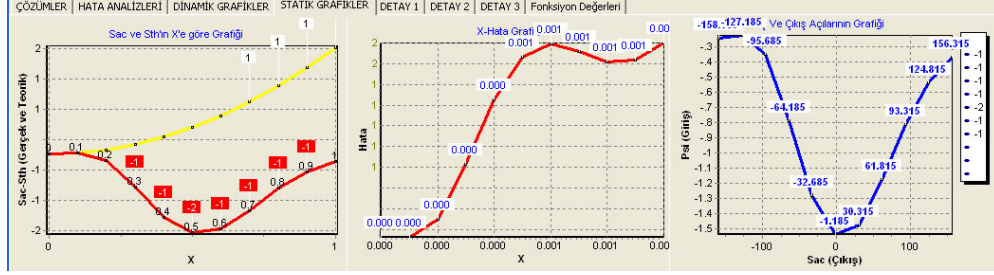
Krank-Biyel mekanizmasının animasyon oluşturan fonksiyonel yapısı, four bar ile aynı yapıdadır.

Four-Bar programında, koordinat düzleminde yapılan animasyon ile birlikte, aynı algoritmik düzende işleyen dinamik grafikler de bulunmaktadır. Aşağıda bu grafikler hakkında bilgi verilmiştir.

Dinamik Grafikler ve İşleyişleri: Dinamik grafikler, animasyonun işleyişi sırasında, çizilen animasyona eşdeğer olarak işlemektedir. Şekil 4.64' de, grafiklerin bir arada görünümü verilmektedir.



Şekil 4.64. Dinamik Grafikler

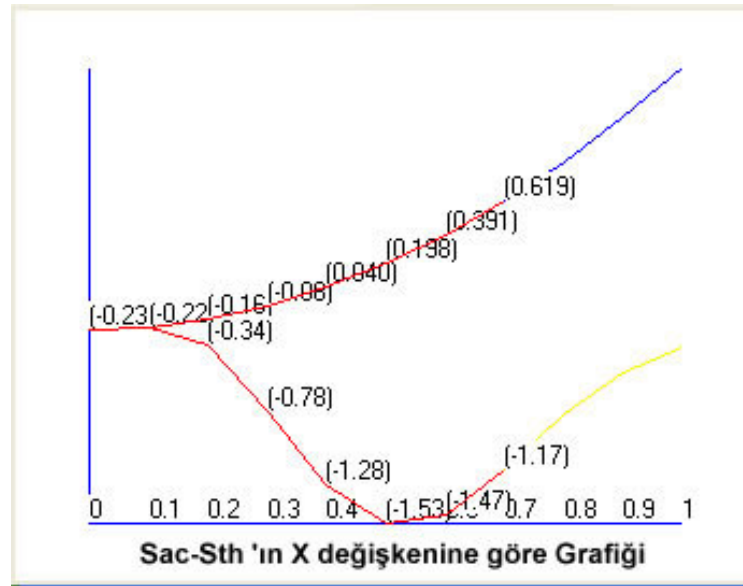


Şekil 4.65 Statik Grafikler

Grafiklerin çalışma prensibi Four-Bar programı ile aynıdır.

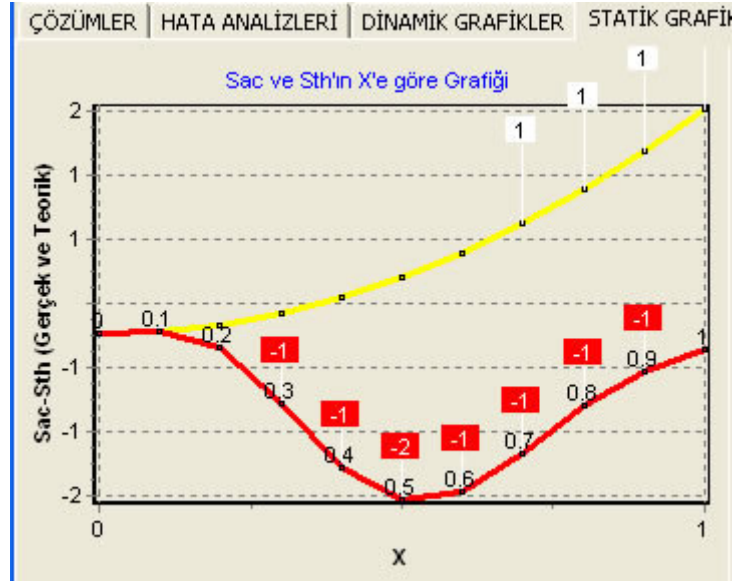
Statik grafikler, koordinat düzleminde olduğu gibi, çözümler listesinden seçilen çözüm ve buna bağlı olarak listelenen dizisel veriler yardımı ile çizilir. Seçilen çözüme bağlı olarak değişen dizi değerleri belirli aralıklar ile grafiklere aktarılır. Döngü içerisinde bu sistemin çalışması sonucu dinamik grafikler ortaya çıkar.

Programda toplam 3 adet dinamik grafik bulunmaktadır.



Şekil 4.66 Sac-Sth-X Grafiği (Dinamik)

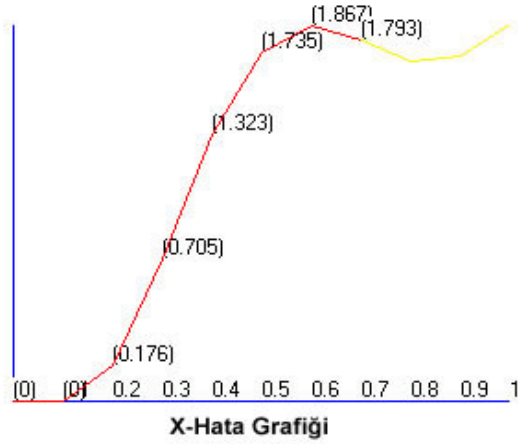
Gösterilen grafik, S çubuğunun bulunan ve olması gereken çıkış açılarının, X aralıklarına göre değişimini gösterir. Sac ve Sth arasındaki fark, analiz edilen çözümün hata sapması olduğundan, bu grafik hata sapmasını göstermiş bulunmaktadır. Sarı ve mavi ile çizilen eğri, tamamlanacak eğriyi ifade eder. Bu bir defaya mahsus çizilir. Animasyon ile çıkış açısının Sth ve Sac değerleri kırmızı renktedir ve aynı anda çizilir. Şekil 4.67’ de seçilen çözümün hata sapması büyük olduğundan, grafikler arasındaki mesafe fazladır. Hata sapmasının 0 ‘ a yaklaştığı kesimlerde grafikler tek çizgide toplanmaya yaklaşır. Aynı grafiğin dinamik çizimi ise Şekil 4.67’ deki gibidir.



Şekil 4.67 Sac-Sth-X Grafiği (Statik)

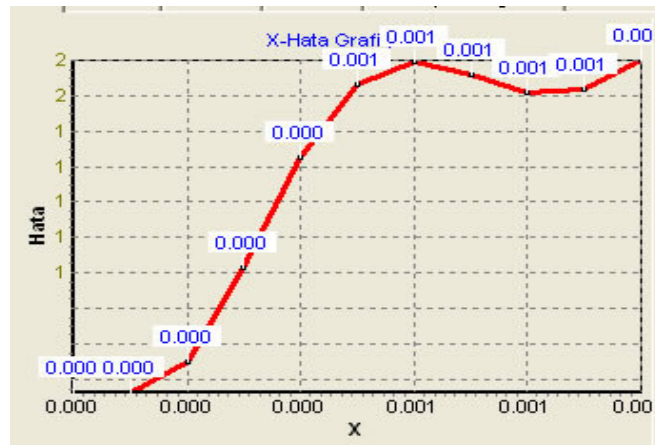
Grafiklerin tamamlanmış çizimleri görüldüğü gibi aynıdır.

2. grafiğin animasyon halinde görünüşü Şekil 4.68 'deki gibidir.



Şekil 4.68 X-Hata Grafiği (Dinamik)

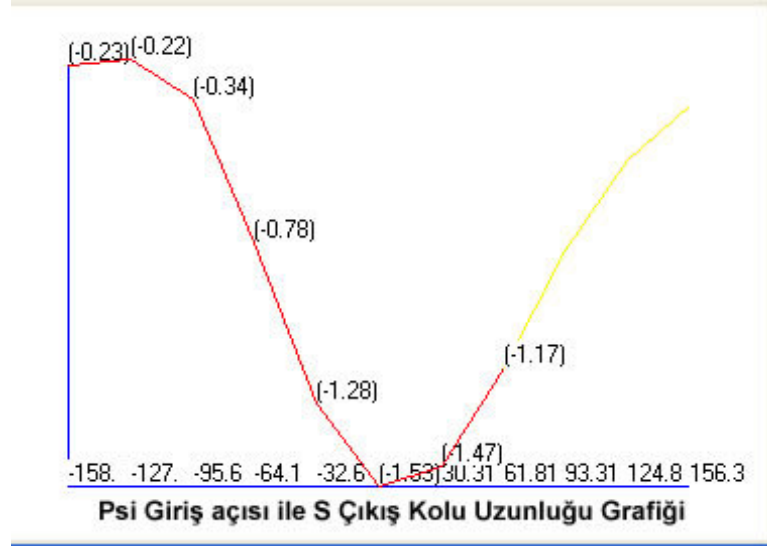
Şekil 4.68 'de gösterilen grafik, seçilen çözümün, X 'e göre hata sapmasını göstermektedir. Grafiğe göre, hata sapması en az olan noktaya bakılarak mekanizma tasarımı yapılabilir. Sarı ile gösterilen eğri, grafiğin izlediği yolu gösterir. Kırmızı ile gösterilen çizim ise animasyon sırasındaki konuma göre konulan çizgiler dizisinden oluşur. Çalışma prensibi Şekil 4.68'de grafikte aynıdır. Aşağıda belirtilen hata değerleri ve aynı satırda bulunan X değeri için, zaman aralıklarıyla grafiğe çizim eklenir. Bu şekilde grafiğin dinamiği sağlanır. Aynı grafiğin sentez sırasında oluşturulan statifi ise Şekil 4.69'daki gibidir.



Şekil 4.69 X-Hata Grafiği (Statik)

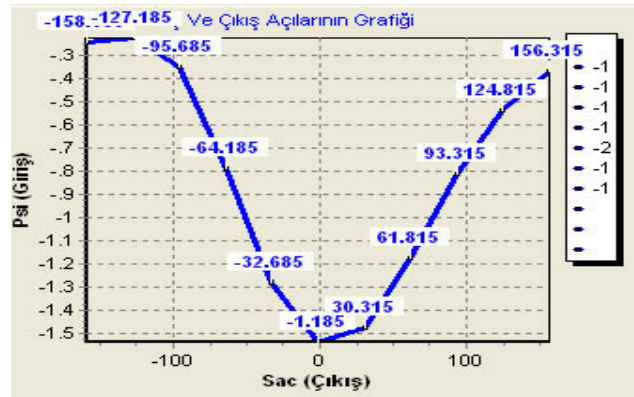
Görüldüğü gibi, statik ve dinamik grafiklerin sonuçta oluşturdukları çizimler aynıdır. Hata grafiklerinde hatanın çok küçük boyutlarda (virgülden sonraki sondan 3 rakamın 0 olması) olması bazen grafikte verilerin noktalarını sadece 0.000 olarak görebilmemizi mümkün kılar.

3. dinamik grafiğin görüntüsü ise şu şekildedir.



Şekil 4.70 Psi-S Grafiği (Dinamik)

Bu grafik ile, X_I giriş kolu açısı ve S çıkış kolu uzunluğunun oluşturduğu eğim görülmektedir. Çalışma prensibi diğer iki grafikte aynıdır. Statik grafik görüntüsü Şekil 4.71'deki gibidir.



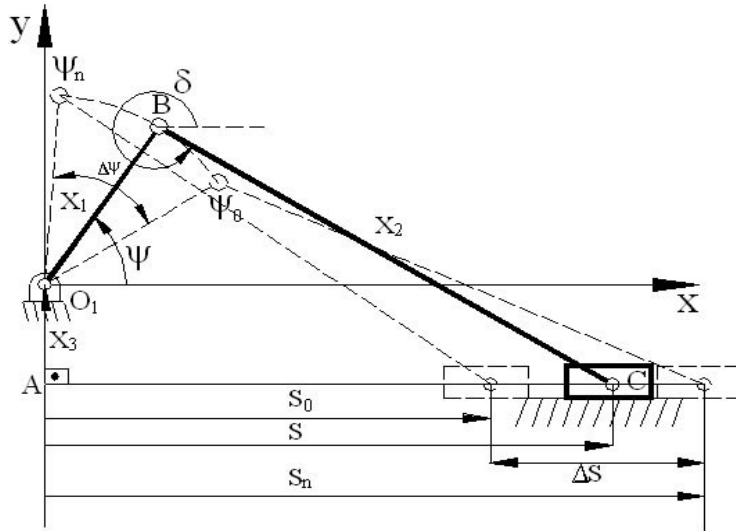
Şekil 4.71 Psi-Sac Grafiği (Statik)

Dinamik grafikler ile statik grafiklerin çalışma algoritmaları farklılık göstermektedir. Dinamik grafikler delphi programının kullanıcıya sağladığı hazır nesnelere sahiptir. Bu grafiklere program içerisinde girdi değerleri atanarak grafiğin çizimi sağlanmış olur. Object-Oriented özelliğinin en canlı örneğidir.

Statik grafikler, program içerisinde vektörel, grafik ve matematiksel fonksiyonlar kullanılarak gerçekleştirilir. *Complex math* ve *graphics primitives* birimleri, bu grafiklerin çizimi için kullanılan birimlerdir. İçerik olarak dinamik vektörel çizim işlemlerini gerçekleştiren fonksiyonlar barındırmaktadır. Dinamik grafiklerin çalışma prensibi için dört çubuk mekanizmalarının dinamik grafik çalışma prensibine bakılabilir.

4.3.4.4. Krank-Biyel Delta Programı

Mek-Gen programından, 2 nolu seçeneği işaretleyip tamam butonuna basmak, kullanıcıyı Krank-Biyel mekanizması için sentez yapmak üzere Krank-Biyel delta adlı alt programa götürmektedir. Bu programın işleyiş yapısı, Krank-Biyel programı ile aynı yapıdadır.



Şekil 4.72 Krank-Biyel Delta İçin Mekanizma Tipi

Krank-Biyel programı ile Krank-Biyel Delta programı arasındaki tek fark, Giriş değerlerindeki S uzunluğu ve δ açılarıdır. Krank-Biyel programında fonksiyon sentezi X_1 giriş kolu dönüş miktarı $\Delta\psi$ ve S çıkış kolu dönüş miktarına ΔS 'e göre yapılırken, Krank-Biyel Delta programında Fonksiyon sentezi X_1 Giriş kolu dönüş miktarı $\Delta\psi$ ve X_2 bağlantı kolu dönüş $\Delta\delta$ miktarına göre yapılmaktadır.

İki programların işleyiş kısımları aynı olduğundan, burada sadece Krank-Biyel Delta programının farklı olan bölümleri üzerinde durulacaktır.

Krank-Biyel Delta Programı Veri Girişi: Krank-Biyel Delta programının veri giriş menüsü Şekil 4.73' de verilmiştir. Şekilde de görüldüğü gibi programa diğer programdan farklı olarak çıkış kolu dönüş miktarı yerine bağlantı kolu dönüş miktarının girilmesi istenmektedir. Program aynı şekilde global değişkenler kullanarak girilen verileri ana program birimine ve ilgili birimlere aktarır.

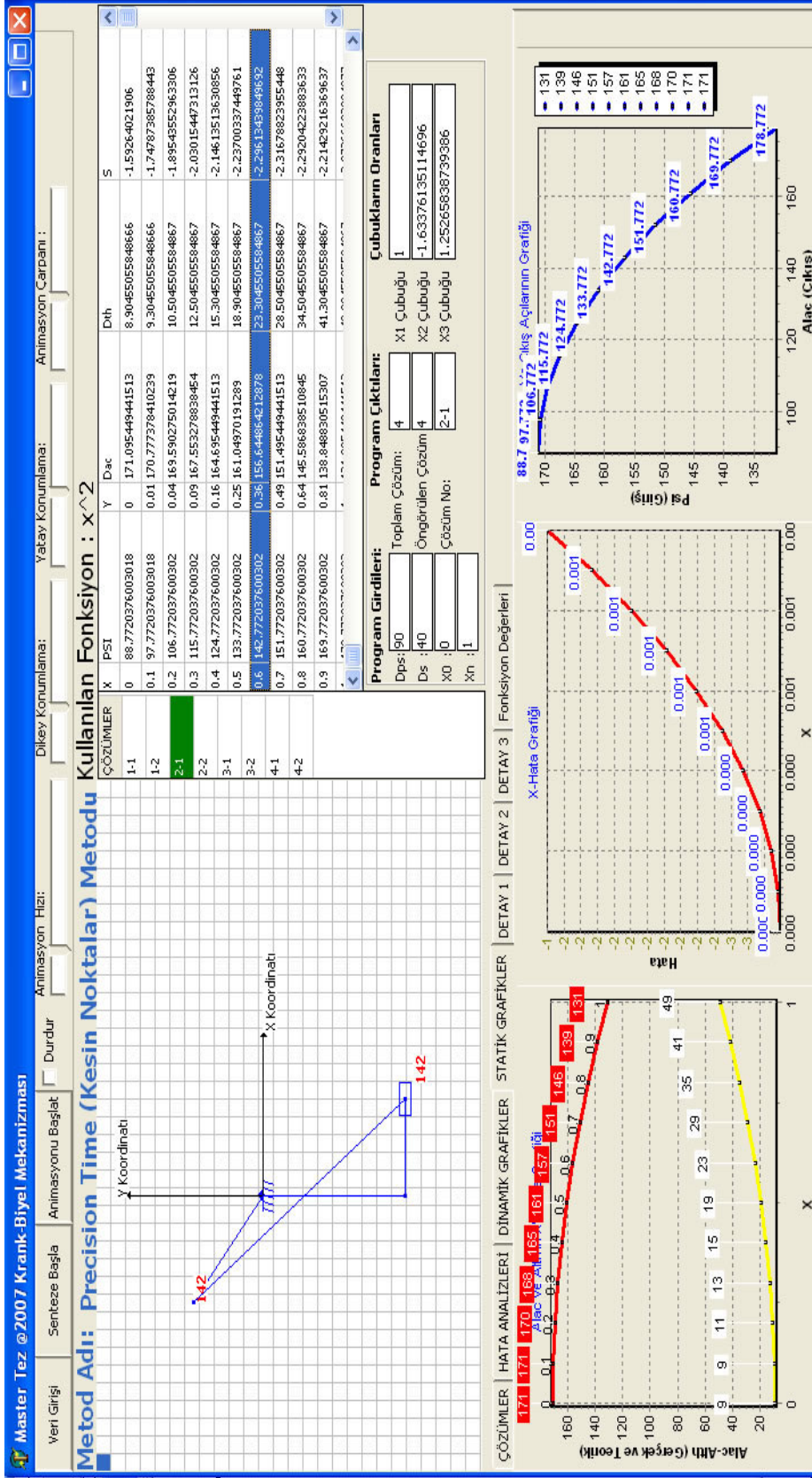
Veri Girişi					
Dosyadan Al		Dosyaya Kaydet		Fonksiyon Girişi	
Method Seçimi				Precision Time (Kesin Noktalar) Metodu	
Nn (Krank sider için)	Dps[Giriş Kolu Dönüş Mik.	Precision İçin Kesin Noktalar		Subdomain İçin Kesin Noktalar	
2	90	Xp[1]	0	Xs[1]	0
Rr	X2 kolunun Dönüş Miktarı	Xp[2]	0.4	Xs[2]	0.01
-1	-45	Xp[3]	0.7	Xs[3]	0.8
Ss	X0 Kesin Nokta Başlangıcı	Xp[4]	1	Xs[4]	0.9
-1	0			Xs[5]	1
Es	Xn Kesin Nokta Bitiş Açısı	Fonksiyon			
1e-10	1				
Maxit	Aralık Sayısı	Varsayılan Değerleri Yükle		Formu Temizle	
20000	10				
	Analiz Nokta Sayısı				
	10				
Çıktı Dosyası	Girdi Dosyası	Senteze Başla			
sonuc.out	sonuc.in				

Şekil 4.73 Veri Girişindeki Değişiklik

Şekil 4.73 'de görüldüğü gibi, Krank-Biyel delta programının veri giriş kısmı da Krank-Biyel programı ile yanıdır. Tek fark S hareket kolu uzunluğu yerine X_2 açısının dönüş miktarı kullanılmasıdır.

Krank-Biyel Programının Çalışma Yapısı ve Mekanizma Sentezi: Krank-Biyel Delta Programı, Sentezde girilen X_1 çubuğu dönüş miktarını $\Delta\psi$ ve X_2 çubuğu Dönüş miktarını $\Delta\delta$ kullanmaktadır. İşleyiş, kullanılan birim ve fonksiyonlar, hata analizleri ve animasyon çizimleri Krank-Biyel programı ile aynıdır. Sonuç değerlerinde Krank-Biyel programından farklı olarak sonuçları $\Delta\delta$ için üretir. Animasyon ve grafiklerdeki değerlerde $\Delta\delta$ için üretilen değerler kullanılır.

Krank-Biyel Delta programının sentez sırasındaki görüntüsü Şekil 4.74 'deki gibidir. $\Delta\delta$ değerine bağlı olarak bir önceki programda ΔS değerine bağlı olan Sac ve Sth çıkış kolu açıları burada bağlantı kolu dönüş miktarı $\Delta\delta$ için $Dlac$ ve Dth değerlerini alırlar. Sentezde ana programın içerisindeki birkaç değişik fonksiyon tanımından başka hiçbir fark yoktur. Animasyon ve grafik çizimlerinde ise, $\Delta\delta$ değeri vektörel ve trigonometrik işlemlere tabi tutularak, X_2 ye bağlı değerler üretilmiştir.



Şekil 4.74 Program Çıktı Ekranı

5. ARAŞTIRMA BULGULARI VE TARTIŞMA

5.1. GİRİŞ

Bu kısımda, Mek-Gen bilgisayar program paketinin alt programları sayesinde elde edilebilecek bulgulara yer verilmiştir. Programa girilen veriler ve bu verilere ilişkin çıktılar detaylı olarak gösterilmiştir.

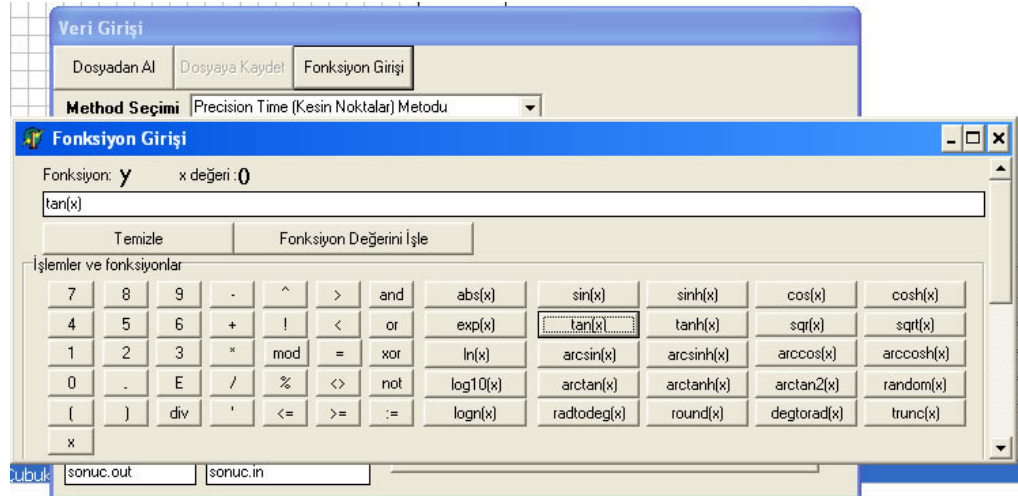
Verilen örnekler, sentezi yapılan birçok mekanizma tasarımlarından en iyi yapısal hatayı üreten çözümlerdir.

5.2. DÖRT- ÇUBUK TASARIMLARI

5.2.1. Galerkin Metodu İle Dört Çubuk Tasarım Sonuçları

Mek-Gen programının ana menüsünden, Four-Bar programı işletilerek, veri giriş kısmına Şekil 5.1' deki veriler işlenir.

5.2.1.1. $Y=Tan(x)$ fonksiyonu için dört çubuk tasarımı



Şekil 5.1. $Y=Tan(x)$ Fonksiyon Girişi

$Y = \tan(x)$ için girilen tasarım değerleri Şekil 5.2' deki gibidir.

Veri Girişi			
Dosyadan Al	Dosyaya Kaydet	Fonksiyon Girişi	
Method Seçimi: Galerkin Metodu ile çözüm			
Nn (4 Çubuk için)	Dps(Giriş Kolu Dönüş Mik.)	Precision İçin Kesin Noktalar	Subdomain İçin Kesin Noktalar
12	90	Xp[1] 0	Xs[1] 0
Rr	Ds (Çıkış Kolu Dönüş Miktarı)	Xp[2] 0.1	Xs[2] 0.01
0	75	Xp[3] 0.6	Xs[3] 0.8
Ss	X0 Kesin Nokta Başlangıcı	Xp[4] 0.8	Xs[4] 0.9
0	0	Xp[5] 1	Xs[5] 0.99
Es	Xn Kesin Nokta Bitiş Açısı		Xs[6] 1
1e-8	0.785398163397448	Fonksiyon: tan(x)	
Maxit	Aralık Sayısı	Varsayılan Değerleri Yükle	Formu Temizle
2000	10	Senteze Başla	
X6	Analiz Nokta Sayısı		
1	10		
Çıktı Dosyası	Girdi Dosyası		
radgalerkintanx1.out	radgalerkintanx1.in		

Şekil 5.2. $Y = \tan(x)$ Veri Girişi

Burada girilen değerler $Y = \tan(x)$ fonksiyonu içindir. $0 < x < 45$ derecelik bir açısal aralıktadır.

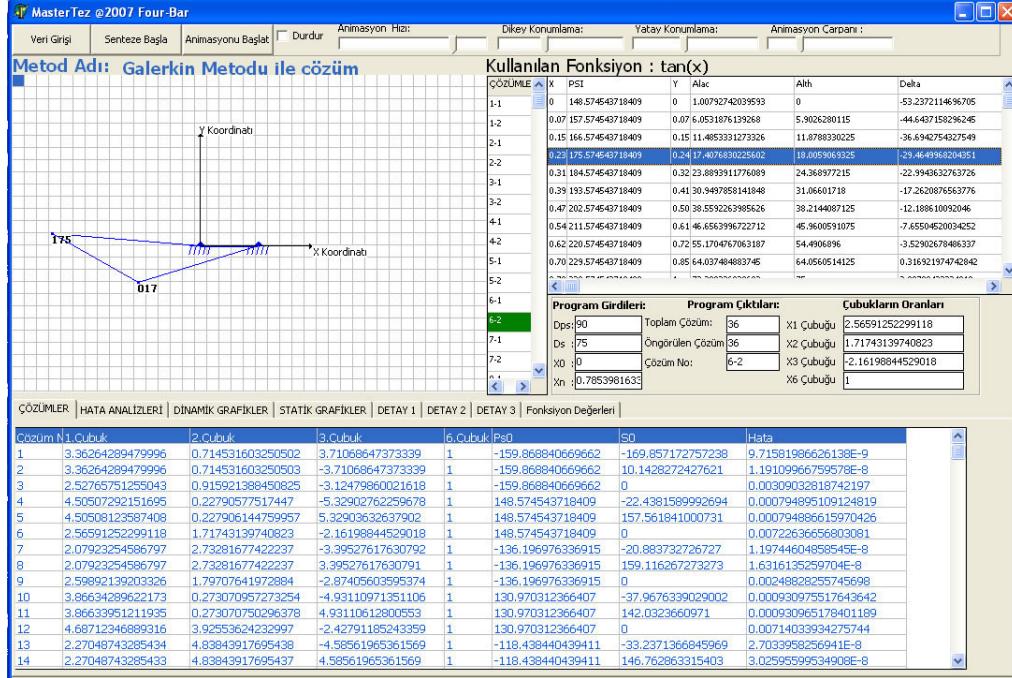
Fonksiyonda kullanılan

90 = giriş kolu dönme açısını,

75 = X3 çıkış kolu dönme açısını temsil eder.

X değişkeninin 0 ile 45 derece aralığındaki değeri derece olarak verilmektedir. Biz bu değerleri X_0 ve X_n in programa girerken radyana çevirmektediriz. Dolayısıyla X_n değerine 45 derecelik açıya denk gelen 0.785398163397448 radyan değeri girilmektedir. Metod otomatik olarak Galerkin seçilidir ve çıktılar *radgalerkintanx1.in* ve *.out* isimli giriş ve çıkış dosyalarına kaydedilir.

Program çalıştığında öngörülen çözümler Şekil 5.3' deki gibidir:



Şekil 5.3. $Y=Tan(x)$ Çıktı Ekranı

Four-bar programı $y=tan(x)$ için Şekil 5.4' deki gibi bir ekran çıktısı sağlar.

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $0 < x < 45$ aralığında tanımlı $Y = tan(x)$ fonksiyonu için toplamda 36 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 36 çözümü önermiştir.

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps:	90	Toplam Çözüm:	36	X1 Çubuğu	2.56591252299118
Ds :	75	Öngörülen Çözüm	36	X2 Çubuğu	1.71743139740823
X0 :	0	Çözüm No:	6-2	X3 Çubuğu	-2.16198844529018
Xn :	0.7853981633			X6 Çubuğu	1

Şekil 5.4. $Y=Tan(x)$ Program Girdileri Çubuk Boyları ve Çözümler

Programın verilerinin kaydedildiği *radgalerkintanx1.out* dosyasının görünüşü Şekil 5.5’ deki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu gibi görünmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.

```

radgalerkintanx1 - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
** Galarkin method ile fonksiyon ureten dört çubuk tasarlar ****

Girilen tasarım degerleri
dps,ds,x0,xn = 90, 75.0000, 0.0, 0.8

coz.no  x1          x2          x3          ps0          s0          hata
-----  --          --          --          --          --          --
 1  3.362643    0.714532    3.710686   -159.868841  -169.857173  9.72E-0009
 2  3.362643    0.714532   -3.710686   -159.868841   10.142827  1.19E-0008
 3  2.527658    0.915921   -3.124799   -159.868841   0.000000  3.09E-0003
 4  4.505073    0.227906   -5.329028   148.574544   -22.438159  7.95E-0004
 5  4.505081    0.227906    5.329036   148.574544   157.561841  7.95E-0004
 6  2.565913    1.717431   -2.161988   148.574544   0.000000  7.23E-0003
 7  2.079233    2.732817   -3.395276   -136.196976   -20.883733  1.19E-0008
 8  2.079233    2.732817    3.395276   -136.196976   159.116267  1.63E-0008
 9  2.598921    1.797076   -2.874056   -136.196976   0.000000  2.49E-0003
10  3.866343    0.273071   -4.931110   130.970312   -37.967634  9.31E-0004
11  3.866340    0.273071    4.931106   130.970312   142.032366  9.31E-0004
12  4.687123    3.925536   -2.427912   130.970312   0.000000  7.14E-0003
13  2.270487    4.838439   -4.585620   -118.438440   -33.237137  2.70E-0008
14  2.270487    4.838439    4.585620   -118.438440   146.762863  3.03E-0008
15  3.154117    2.780350   -2.778410   -118.438440   0.000000  4.40E-0003
16  3.742213    0.217219   -4.669082   123.552152   -43.430782  1.03E-0003
17  3.742218    0.217219    4.669087   123.552152   136.569218  1.03E-0003
18  6.799490    6.124688   -2.488337   123.552152   0.000000  7.07E-0003
19  -3.736838    0.217168   -4.663866   -56.447848   -43.430782  1.01E-0003
20  -3.736849    0.217168    4.663877   -56.447848   136.569218  1.01E-0003
21  -6.799490    6.124688   -2.488337   -56.447848   0.000000  7.07E-0003
22  -2.270487    4.838439   -4.585620    61.561560   -33.237137  3.08E-0008
23  -2.270487    4.838439    4.585620    61.561560   146.762863  2.48E-0008
24  -3.154117    2.780350   -2.778410    61.561560   0.000000  4.40E-0003
25  -3.861086    0.272749   -4.925538   -49.029688   -37.967634  9.17E-0004
26  -3.861084    0.272749    4.925536   -49.029688   142.032366  9.16E-0004
27  -4.687123    3.925536   -2.427912   -49.029688   0.000000  7.14E-0003
28  -2.079233    2.732817   -3.395276   43.803024   -20.883733  1.10E-0008
29  -2.079233    2.732817    3.395276   43.803024   159.116267  3.25E-0008
30  -2.598921    1.797076   -2.874056   43.803024   0.000000  2.49E-0003
31  -4.506983    0.227991   -5.331027   -31.425456   -22.438159  7.93E-0004
32  -4.507015    0.227992    5.331061   -31.425456   157.561841  7.93E-0004
33  -2.565913    1.717431   -2.161988   -31.425456   0.000000  7.23E-0003
34  -3.362643    0.714532    3.710686   20.131159   -169.857173  1.26E-0008
35  -3.362643    0.714532   -3.710686   20.131159   10.142827  2.01E-0008
36  -2.527658    0.915921   -3.124799   20.131159   0.000000  3.09E-0003

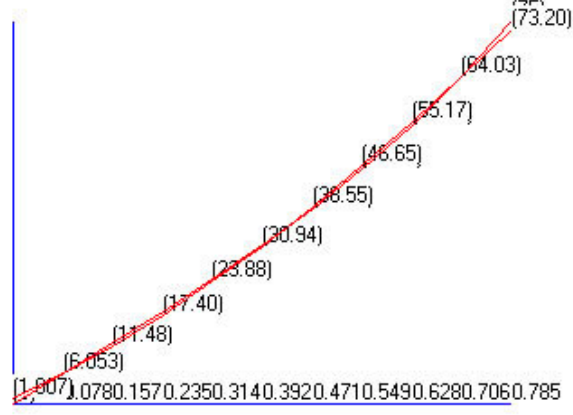
****1 inci çözüm hata analiz sonuçları ****

***** 1 inci Dört çubuk analiz sonuçları*****
  x      psi      y      alac      alth      delta      hata
  -      ---      -      ---      ---      -----      --

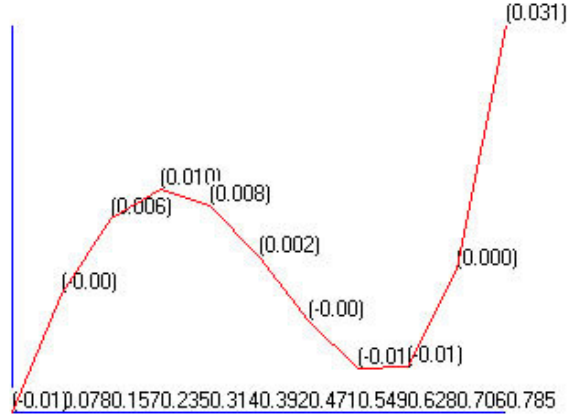
```

Şekil 5.5. $Y=Tan(x)$ Text Dosyası

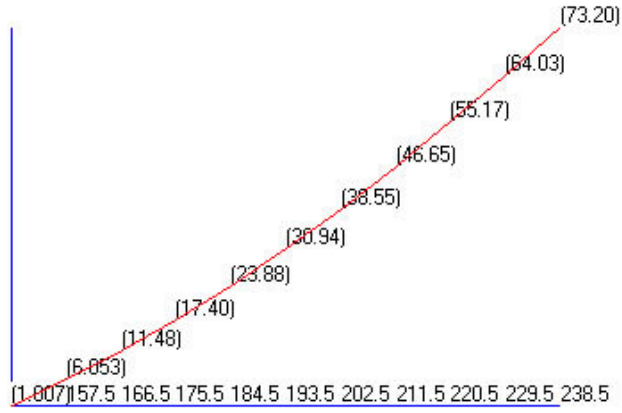
$Y=\tan(x)$ fonksiyonunun grafikleri ise Şekil 5.6 ve 5.7'deki gibidir.



Şekil 5.6. $Y=\tan(x)$ Psi-Alac Hata Grafiği

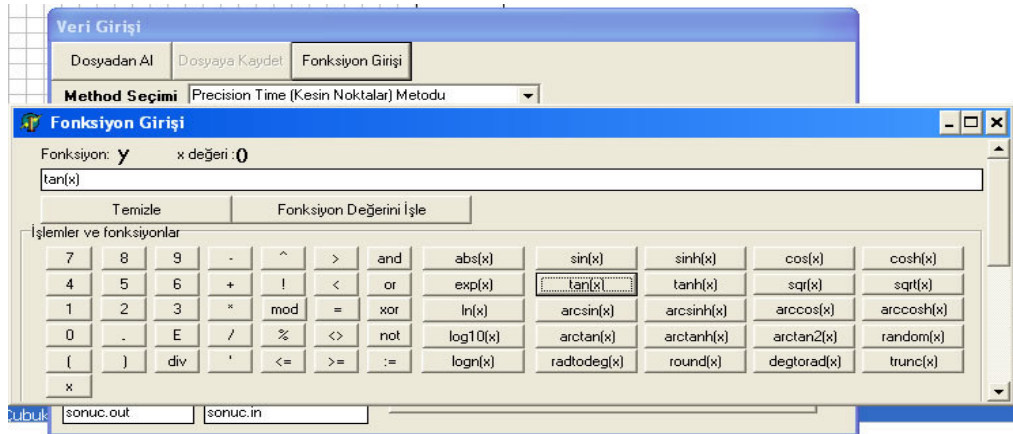


Şekil 5.7. $Y=\tan(x)$ X-Hata Grafiği



Şekil 5.8. $Y = \tan(x)$ Psi-Alac Grafiği

5.2.1.2. $Y = x^2$ Fonksiyonu İçin Dört Çubuk Tasarımı



Şekil 5.9. $Y = x^2$ Fonksiyon Girişi

$Y = x^2$ için girilen tasarım değerleri Şekil 5.10' daki gibidir.

Veri Girişi

Dosyadan Al Dosyaya Kaydet Fonksiyon Girişi

Method Seçimi Galerkin Metodu ile çözüm

Nn (4 Çubuk için)	Dps(Giriş Kolu Dönüş Mik.)	Precision İçin Kesin Noktalar	Subdomain İçin Kesin Noktalar
12	90	Xp[1] 0	Xs[1] 0
Rr	Ds (Çıkış Kolu Dönüş Miktan)	Xp[2] 0.1	Xs[2] 0.01
0	60	Xp[3] 0.6	Xs[3] 0.8
Ss	X0 Kesin Nokta Başlangı	Xp[4] 0.8	Xs[4] 0.9
0	0	Xp[5] 1	Xs[5] 0.99
Es	Xn Kesin Nokta Bitiş Açısı		Xs[6] 1
1e-8	1		
Maxit	Aralık Sayısı	Fonksiyon	
2000	10	x ²	
X6	Analiz Nokta Sayısı	Varsayılan Değerleri Yükle	Formu Temizle
1	10	Senteze Başla	
Çıktı Dosyası	Girdi Dosyası		
sonuc.out	sonuc.in		

Şekil 5.10. $Y = x^2$ Veri Girişi

Burada girilen değerler $Y = x^2$ fonksiyonu içindir. $0 < x < 1$ birimlik bir açısıl aralıktadır.

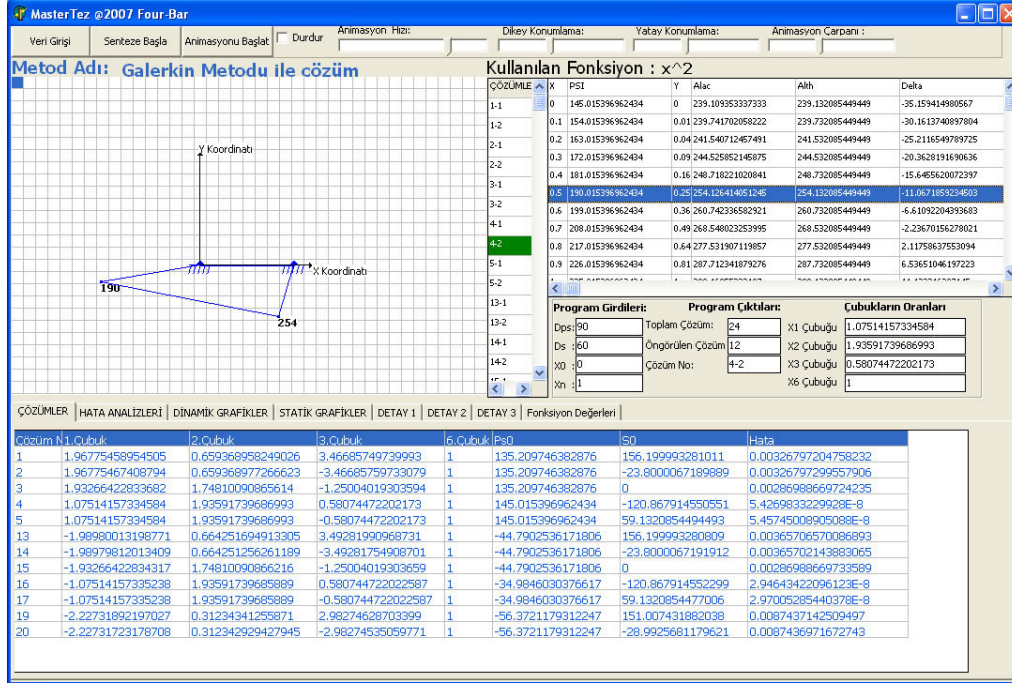
Fonksiyonda kullanılan

90 = giriş kolu dönme açısını,

60 = X3 çıkış kolu dönme açısını temsil eder.

X değişkeninin 0 ile 1 birim aralığındaki değerleri için sentez yapılır. Metod otomatik olarak galerkin seçilidir ve çıktılar *sonuc.in* ve *sonuc.out* isimli giriş ve çıkış dosyalarına kaydedilir.

Program çalıştığında öngörülen çözümler Şekil 5.11 'deki gibidir:



Şekil 5.11. $Y = x^2$ Program Çıktısı

Four-bar programı $Y = x^2$ için Şekil 5.12' deki gibi bir ekran çıktısı sağlar.

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $0 < x < 1$ aralığında tanımlı $Y = x^2$ fonksiyonu için toplamda 24 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 12 çözümü önermiştir.

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps:	90	Toplam Çözüm:	24	X1 Çubuğu	1.07514157334584
Ds :	60	Öngörülen Çözüm	12	X2 Çubuğu	1.93591739686993
X0 :	0	Çözüm No:	4-2	X3 Çubuğu	0.58074472202173
Xn :	1			X6 Çubuğu	1

Şekil 5.12. $Y = x^2$ Program Girdileri Çubuk Boyları ve Çözümler

Programın verilerinin kaydedildiği *sonuc.out* dosyasının görünüşü ise Şekil 5.13'deki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu gibi görünmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.

```

sonuc - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
*** Galarkin method ile fonksiyon ureten dört çubuk tasarımlar ***
Girilen tasarım degerleri
dps,ds,x0,xn = 90, 60.0000, 0.0, 1.0

coz.no  x1          x2          x3          ps0          s0          hata
-----  ---          ---          ---          ---          ---          ---
 1  1.967755    0.659369    3.466857    135.209746    156.199993    3.27E-0003
 2  1.967755    0.659369    -3.466858    135.209746    -23.800007    3.27E-0003
 3  1.932664    1.748101    -1.250040    135.209746    0.000000    2.87E-0003
 4  1.075142    1.935917    0.580745    145.015397    -120.867915    5.43E-0008
 5  1.075142    1.935917    -0.580745    145.015397    59.132085    5.46E-0008
13  -1.989800    0.664252    3.492820    -44.790254    156.199993    3.66E-0003
14  -1.989798    0.664251    -3.492818    -44.790254    -23.800007    3.66E-0003
15  -1.932664    1.748101    -1.250040    -44.790254    0.000000    2.87E-0003
16  -1.075142    1.935917    0.580745    -34.984603    -120.867915    2.95E-0008
17  -1.075142    1.935917    -0.580745    -34.984603    59.132085    2.97E-0008
19  -2.227319    0.312343    2.982746    -56.372118    151.007432    8.74E-0003
20  -2.227317    0.312343    -2.982745    -56.372118    -28.992568    8.74E-0003
****1 inci çözüm hata analiz sonuçları ****

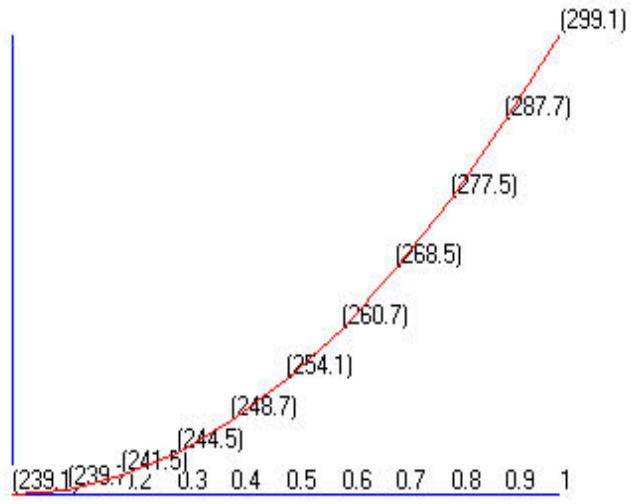
***** 1 inci Dört Çubuk analiz sonuçları*****
      x      psi      y      alac      alth      delta      hata
      ---      ---      ---      ---      ---      ---      ---
*****Hareket Yok*****
 0.1000  144.2097  0.0100  152.3712  156.8000  136.1320  0.07729653
 0.2000  153.2097  0.0400  156.2036  158.6000  129.0702  0.04182490
 0.3000  162.2097  0.0900  161.1581  161.6000  128.1625  0.00771308
 0.4000  171.2097  0.1600  166.6214  165.8000  130.4895  -0.01433705
 0.5000  180.2097  0.2500  172.4367  171.2000  135.3340  -0.02158401
 0.6000  189.2097  0.3600  178.5771  177.8000  142.5411  -0.01356369
 0.7000  198.2097  0.4900  185.1066  185.6000  152.3165  0.00861106
 0.8000  207.2097  0.6400  192.2256  194.6000  165.4539  0.04144061
 0.9000  216.2097  0.8100  200.6376  204.8000  -174.8245  0.07264668
*****Hareket Yok*****

***** 2 inci Dört Çubuk analiz sonuçları*****
      x      psi      y      alac      alth      delta      hata
      ---      ---      ---      ---      ---      ---      ---
*****Hareket Yok*****

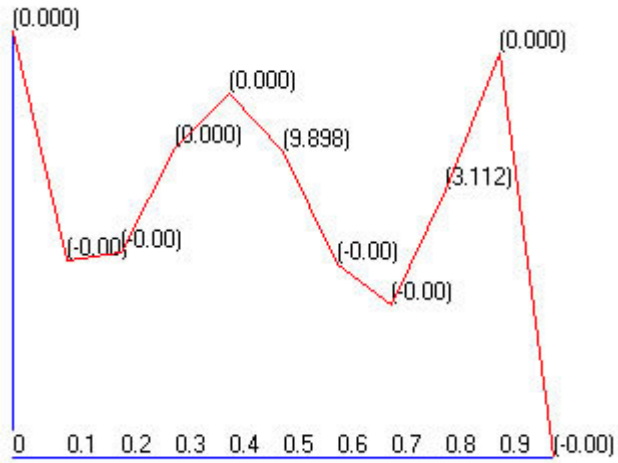
```

Şekil 5.13. $Y=x^2$ Metin Dosyası

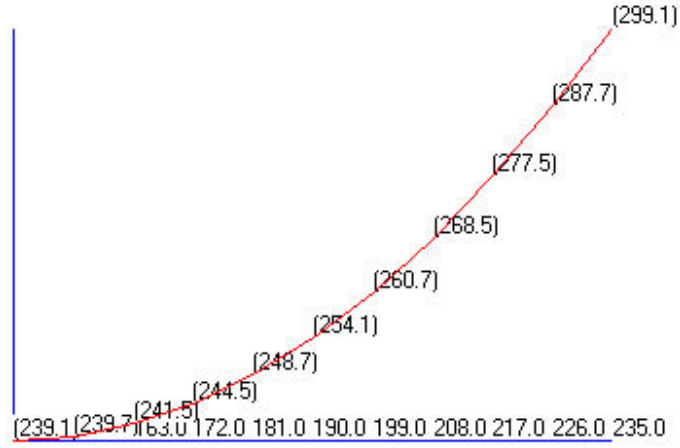
$Y=x^2$ fonksiyonunun grafikleri ise Şekil 5.14' deki gibidir.



Şekil 5.14. $Y=x^2$ Psi-Alac Hata Grafiği

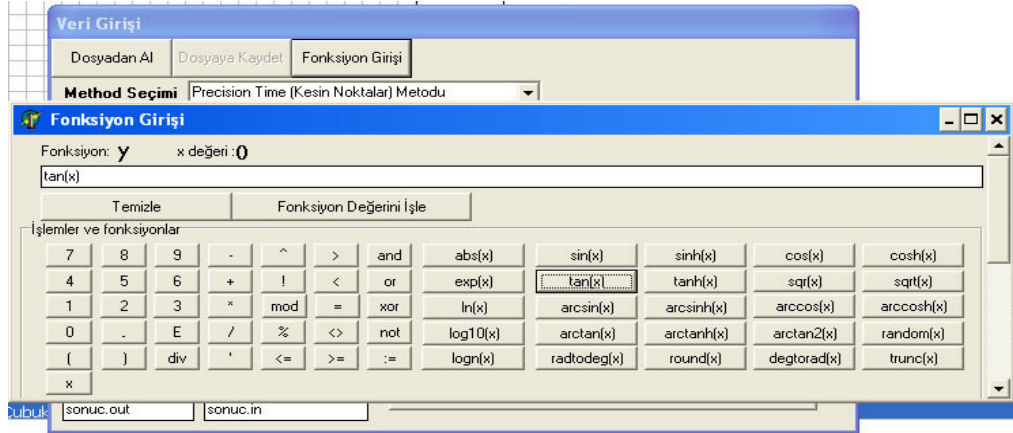


Şekil 5.15. $Y=x^2$ X-Hata Grafiği



Şekil 5.16. $Y=x^2$ Psi-Alac Grafığı

5.2.1.3. $Y = \sin(x)$ Fonksiyonu İçin 4 Çubuk Tasarımı



Şekil 5.17. $Y = \sin(x)$ Fonksiyon Girişi

$Y = \sin(x)$ için girilen tasarım değerleri şekil 5.18' deki gibidir.

Şekil 5.18. $Y=\sin(x)$ Veri Girişi

Burada girilen değerler $Y = \sin(x)$ fonksiyonu içindir. $0 < x < 90$ derecelik bir açısal aralıktadır.

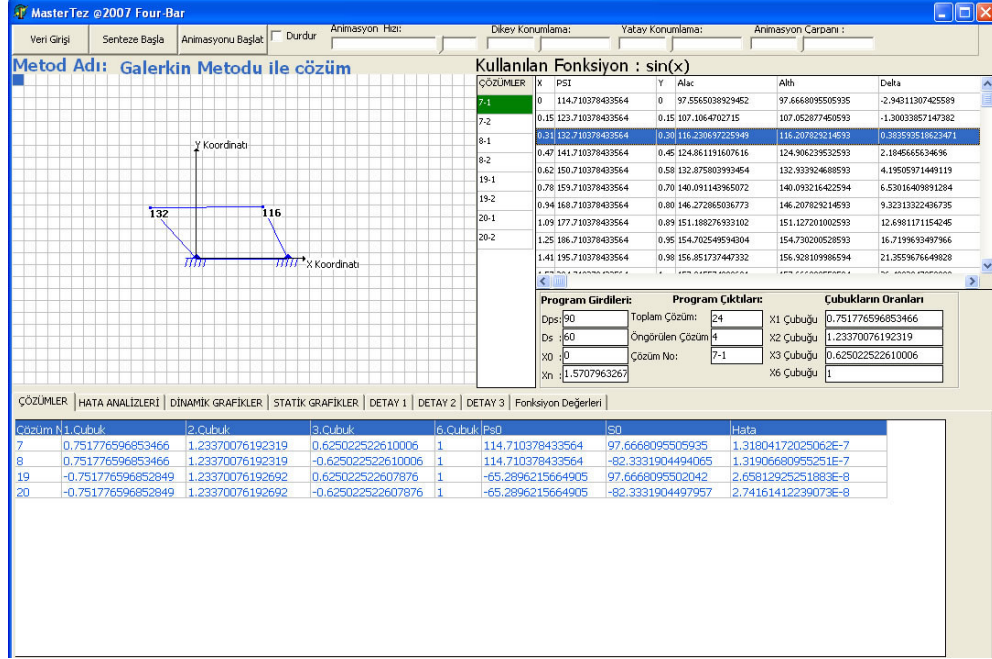
Fonksiyonda kullanılan

90 = giriş kolu dönme açısını,

60 = X_3 çıkış kolu dönme açısını temsil eder.

X değişkeninin 0 ile 90 derece aralığındaki değerleri için sentez yapılır. Programda X_0 ve X_n değerleri birim girişleri için tasarlanmıştır. Bir önceki örnekte X^2 fonksiyonu için X değerinin 0 ile 1 değerleri arasındaki değişimi için x_0 ve x_n 0 ve 1 olarak girilmiştir. Burada, girilecek değer derece olarak verildiğinden programa girerken radyan cinsinden girilmelidir. Yani $0 < x < 90$ derece aralığındaki x değerinin x_0 ve x_n değerleri programa $x_0 = 0$, $x_n = 1.5707963267949$ olarak girilmektedir. Metod otomatik olarak galerkin seçilidir ve çıktılar $\sinx.in$ ve $\sinx.out$ isimli giriş ve çıkış dosyalarına kaydedilir.

Program çalıştığında öngörülen çözümler Şekil 5.19 'daki gibidir:



Şekil 5.19. $Y = \sin(x)$ Program Çıktısı

Four-bar programı $Y = \sin(x)$ için Şekil 5.20' deki gibi bir ekran çıktısı sağlar.

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $0 < x < 90$ aralığında tanımlı $Y = \sin(x)$ fonksiyonu için toplamda 24 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 4 çözümü önermiştir.

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps: 90	Toplam Çözüm: 24	X1 Çubuğu	0.751776596853466		
Ds : 60	Öngörülen Çözüm 4	X2 Çubuğu	1.23370076192319		
X0 : 0	Çözüm No: 7-1	X3 Çubuğu	0.625022522610006		
Xn : 1.5707963267		X6 Çubuğu	1		

Şekil 5.20. $Y = \sin(x)$ Program Girdileri Çubuk Boyları ve Çözümler

Programın verilerinin kaydedildiği *sinx.out* dosyasının görünüşü ise Şekil 5.21' deki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu gibi

görülmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.

```

sinx - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
** Galarkin method ile fonksiyon ureten dört çubuk tasarılar ****
Girilen tasarım degerleri
dps,ds,x0,xn = 90, 60.0000, 0.0, 1.6

coz.no  x1          x2          x3          ps0          s0          hata
-----  --          --          --          --          --          --
7       0.751777    1.233701    0.625023    114.710378    97.666810    1.32E-0007
8       0.751777    1.233701    -0.625023   114.710378    -82.333190    1.32E-0007
19      -0.751777    1.233701    0.625023    -65.289622    97.666810    2.66E-0008
20      -0.751777    1.233701    -0.625023   -65.289622    -82.333190    2.74E-0008
****7 inci çözüm hata analiz sonuçları ****

***** 1 içi Dört Çubuk analiz sonuçları*****
x      psi      y      alac      alth      delta      hata
-----  --      --      --      --      --      --
0.0000  114.7104    0.0000  97.5565    97.6668    -2.9431    0.00192520
0.1571  123.7104    0.1564  107.1065   107.0529   -1.3003    -0.00093537
0.3142  132.7104    0.3090  116.2307   116.2078    0.3836    -0.00039912
0.4712  141.7104    0.4540  124.8612   124.9062    2.1846    0.00078623
0.6283  150.7104    0.5878  132.8758   132.9339    4.1951    0.00101440
0.7854  159.7104    0.7071  140.0911   140.0932    6.5302    0.00003617
0.9425  168.7104    0.8090  146.2729   146.2078    9.3231    -0.00113509
1.0996  177.7104    0.8910  151.1883   151.1272   12.6981    -0.00106598
1.2566  186.7104    0.9511  154.7025   154.7302   16.7200    0.00048260
1.4137  195.7104    0.9877  156.8517   156.9281   21.3560    0.00133295
1.5708  204.7104    1.0000  157.8156   157.6668   26.4893    -0.00259643

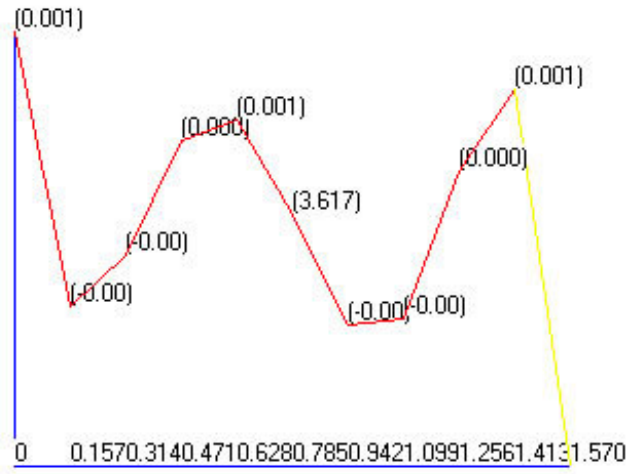
***** 2 içi Dört Çubuk analiz sonuçları*****
x      psi      y      alac      alth      delta      hata
-----  --      --      --      --      --      --
0.0000  114.7104    0.0000  207.5276    97.6668   -51.9728   -1.91743249
0.1571  123.7104    0.1564  205.2736   107.0529   -46.3196   -1.71427538
0.3142  132.7104    0.3090  203.5797   116.2078   -40.5732   -1.52492609
0.4712  141.7104    0.4540  202.4816   124.9062   -34.8418   -1.35394508
0.6283  150.7104    0.5878  202.0757   132.9339   -29.2435   -1.20675211
0.7854  159.7104    0.7071  202.5240   140.0932   -23.9150   -1.08962340
0.9425  168.7104    0.8090  204.0423   146.2078   -19.0079   -1.00940233
1.0996  177.7104    0.8910  206.8466   151.1272   -14.6633   -0.97248660
1.2566  186.7104    0.9511  211.0559   154.7302   -10.9615   -0.98306916
1.4137  195.7104    0.9877  216.6187   156.9281    -7.8855   -1.04179807
1.5708  204.7104    1.0000  223.3392   157.6668    -5.3345   -1.14619999
****8 inci çözüm hata analiz sonuçları ****

***** 1 içi Dört Çubuk analiz sonuçları*****

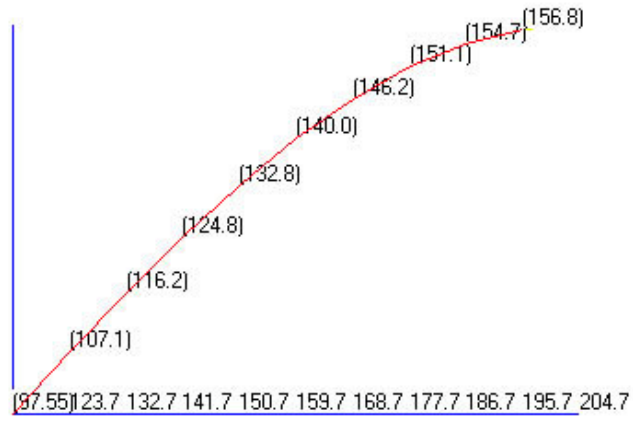
```

Şekil 5.21. $Y = \sin(x)$ metin dosyası

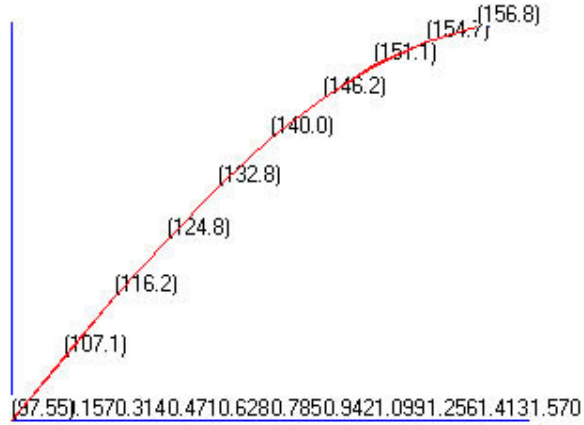
$Y = \sin(x)$ fonksiyonunun grafikleri ise Şekil 5.22'deki gibidir.



Şekil 5.22. $Y=\sin(x)$ X- Hata Grafiği



Şekil 5.23. $Y = \sin(x)$ Psi-Alac Grafiği

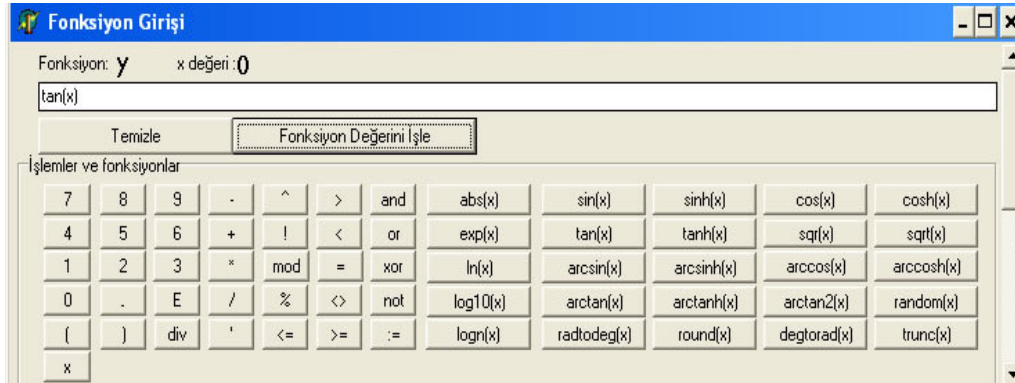


Şekil 5.24. $Y = \sin(x)$ Psi-Alac Hata Grafiği

5.2.2. Alt Bölgeler Metodu İle Dört Çubuk Tasarım Sonuçları

Four-Bar programının mekanizma sentezini 3 ayrı metodu yapabildiğine daha önceki konularda değinilmiştir. Alt bölge metodu ile yapılan sentezlerde, fonksiyon değışeni X 'in 6 ayrı noktadaki değeri için sentez yapılabilir.

5.2.2.1. $Y = \tan(x)$ fonksiyonu için Dört Çubuk Tasarımı



Şekil 5.25. $Y = \tan(x)$ Fonksiyon Giriş'i

$Y = \tan(x)$ için girilen tasarım değerleri Şekil 5.26'daki gibidir.

Veri Girişi			
Dosyadan Al	Dosyaya Kaydet	Fonksiyon Girişi	
Method Seçimi Subdomain (Alt Bölge) Metodu ile çözüm			
Nn (4 Çubuk için)	Dps(Giriş Kolu Dönüş Mik.)	Precision İçin Kesin Noktalar	Subdomain İçin Kesin Noktalar
12	90	Xp[1] 0	Xs[1] 0
Rr	Ds (Çıkış Kolu Dönüş Miktarı)	Xp[2] 0.1	Xs[2] 0.01
0	60	Xp[3] 0.6	Xs[3] 0.3
Ss	X0 Kesin Nokta Başlangıcı	Xp[4] 0.8	Xs[4] 0.4
0	0	Xp[5] 1	Xs[5] 0.6
Es	Xn Kesin Nokta Bitiş Açısı		Xs[6] 8163397448
1e-8	0.785398163397448		
Maxit	Aralık Sayısı	Fonksiyon	
2000	10	tan(x)	
X6	Analiz Nokta Sayısı	Varsayılan Değerleri Yükle	Formu Temizle
1	10	Senteze Başla	
Çıktı Dosyası	Girdi Dosyası		
altbolgetanx.out	altbolgetanx.in		

Şekil 5.26. $Y = \tan(x)$ Veri Girişi

Burada girilen değerler $Y = \tan(x)$ fonksiyonu içindir. $0 < x < 45$ derecelik bir açısal aralıktadır.

Fonksiyonda kullanılan

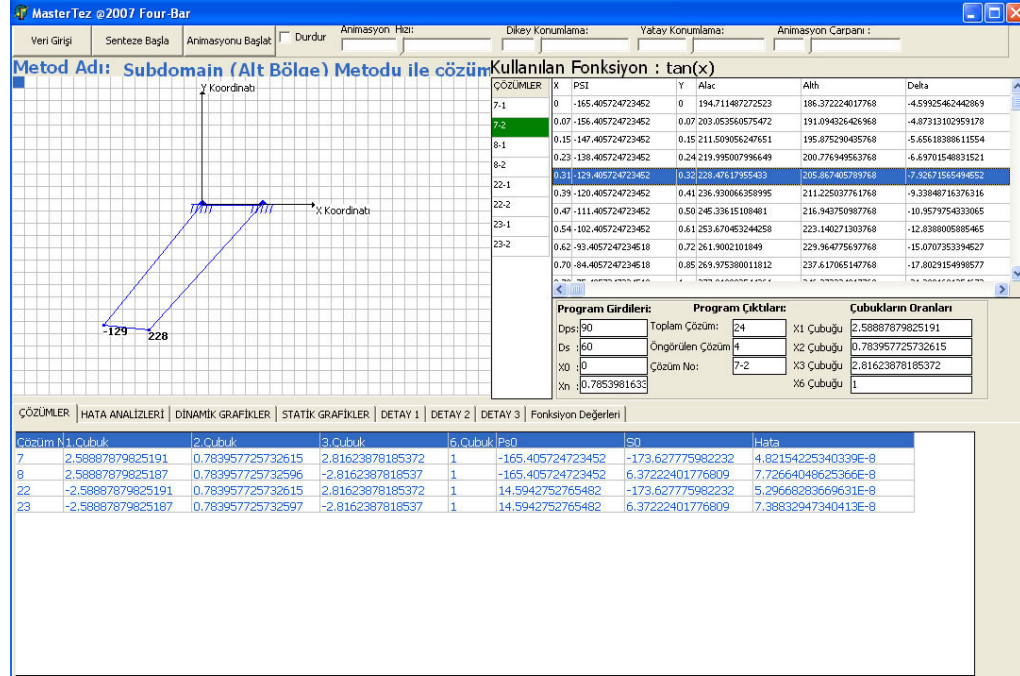
90 = Giriş kolu dönme açısını,

60 = X_3 çıkış kolu dönme açısını temsil eder.

X değişkeninin 0 ile 45 derece aralığındaki değeri derece olarak verilmektedir. Biz bu değerleri X_0 ve X_n in programa girerken radyana çevirmekteyiz. Dolayısıyla X_n değerine 45 derecelik açıya denk gelen 0.785398163397448 değeri girilmektedir. Subdomain metodu ile çözüm yapılması için, programa 6 adet $X_s[i]$ değerinin girilmesi istenir. Bu değerler, fonksiyon değerleri istenen alt bölgelerin değişkenleridir. $X_s[1]$ olarak tanımlanan değer, X_0 değeri, $x_s[6]$ olarak tanımlanan değer X_n değerini temsil eder. Metod olarak

Subdomain (Alt Bölgeler metodu seçilir) seçilir ve çıktılar *altbolgetanx.in* ve *altbolgetanx.out* isimli giriş ve çıkış dosyalarına kaydedilir.

Four - Bar programı $Y = \tan(x)$ için Şekil 5.27'deki gibi bir ekran çıktısı sağlar.



Şekil 5.27. $Y = \tan(x)$ Çıktı Ekranı

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $0 < x < 45$ aralığında tanımlı $Y = \tan(x)$ fonksiyonu için toplamda 24 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 4 çözümü önermiştir.

Program Girdileri:	Program Çıktıları:	Çubukların Oranları
Dps: 90	Toplam Çözüm: 24	X1 Çubuğu: 2.58887879825191
Ds : 60	Öngörülen Çözüm: 4	X2 Çubuğu: 0.783957725732615
X0 : 0	Çözüm No: 7-2	X3 Çubuğu: 2.81623878185372
Xn : 0.7853981633		X6 Çubuğu: 1

Şekil 5.28. $Y = \tan(x)$ Program Girdileri Çubuk Boyları ve Çözümler

Programın verilerinin kaydedildiği *altbolgetanx.out* dosyasının görünüşü ise Şekil 5.29 'daki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu gibi görünmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.

```

altbolgetanx - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
** subdomain method ile fonksiyon ureten dört çubuk tasarlar ****
Girilen tasarim degerleri
dps,ds,x0,xn = 90, 60.0000, 0.0, 0.8
Alt bolgeler xs[i] i:=1,6 = 0.000, 0.010, 0.300, 0.400, 0.600, 0.785,

coz.no  x1          x2          x3          ps0          s0          hata
-----
1  2.514878  0.569619  2.945687  -175.680003  -177.845118  1.53E-0007
2  2.514878  0.569619  -2.945687  -175.680003  2.154882  1.65E-0007
22 -2.514878  0.569619  2.945687  4.319997  -177.845118  1.53E-0007
23 -2.514878  0.569619  -2.945687  4.319997  2.154882  1.47E-0007
****1 inci çözüm hata analiz sonuçları ****

***** 1 içi Dört Çubuk analiz sonuçları*****
x      psi      y      alac      alth      delta      hata
-----
0.0000 -175.6800  0.0000  182.1490  182.1549  7.9701  0.00010336
0.0785 -166.6800  0.0787  186.8871  186.8770  23.3948  -0.00017603
0.1571 -157.6800  0.1584  191.6576  191.6579  39.1829  0.00000598
0.2356 -148.6800  0.2401  196.5555  196.5596  55.2315  0.00007177
0.3142 -139.6800  0.3249  201.6481  201.6501  71.6286  0.00003419
0.3927 -130.6800  0.4142  207.0098  207.0077  88.5027  -0.00003661
0.4712 -121.6800  0.5095  212.7297  212.7264  106.0255  -0.00005747
0.5498 -112.6800  0.6128  218.9205  218.9229  124.4348  0.00004191
0.6283 -103.6800  0.7265  225.7332  225.7474  144.0843  0.00024793
0.7069 -94.6800  0.8541  233.3873  233.3997  165.5610  0.00021739
0.7854 -85.6800  1.0000  242.2517  242.1549  -169.9701  -0.00169027

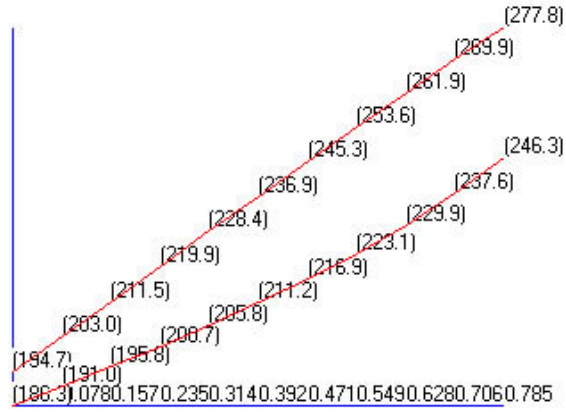
***** 2 içi Dört Çubuk analiz sonuçları*****
x      psi      y      alac      alth      delta      hata
-----
0.0000 -175.6800  0.0000  184.0336  182.1549  -1.7875  -0.03279035
0.0785 -166.6800  0.0787  192.1948  186.8770  -4.3129  -0.09281423
0.1571 -157.6800  0.1584  200.3822  191.6579  -7.1431  -0.15226760
0.2356 -148.6800  0.2401  208.5428  196.5596  -10.1332  -0.20914636
0.3142 -139.6800  0.3249  216.6548  201.6501  -13.3256  -0.26188283
0.3927 -130.6800  0.4142  224.6952  207.0077  -16.7977  -0.30870442
0.4712 -121.6800  0.5095  232.6335  212.7264  -20.6623  -0.34744420
0.5498 -112.6800  0.6128  240.4264  218.9229  -25.0878  -0.37530713
0.6283 -103.6800  0.7265  248.0060  225.7474  -30.3451  -0.38848606
0.7069 -94.6800  0.8541  255.2537  233.3997  -36.9201  -0.38142354
0.7854 -85.6800  1.0000  261.9240  242.1549  -45.8541  -0.34503601
****2 inci çözüm hata analiz sonuçları ****

***** 1 içi Dört Çubuk analiz sonuçları*****

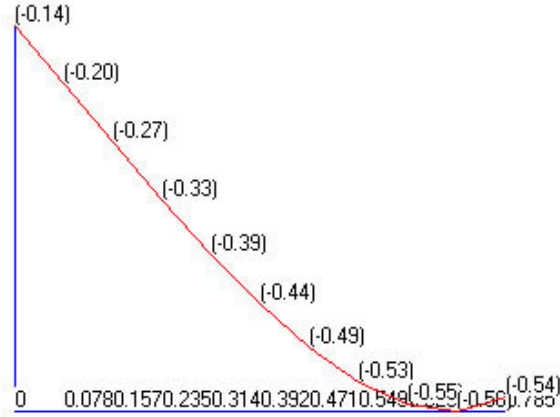
```

Şekil 5.29. $Y=Tan(x)$ Text Dosyası

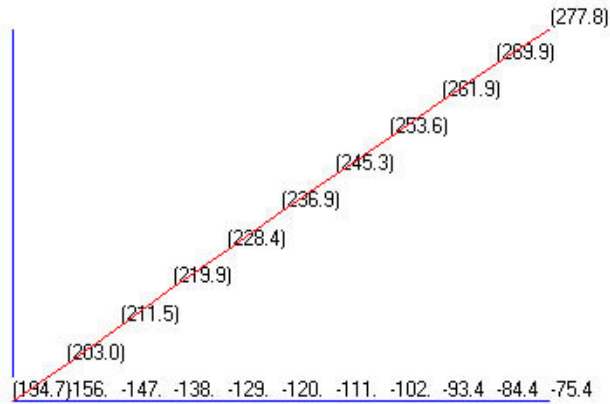
$Y = Tan(x)$ fonksiyonunun grafikleri ise Şekil 5.30, Şekil 5.31, Şekil 5.32 'deki gibidir.



Şekil 5.30. $Y=Tan(x)$ Psi-Alac Hata Grafiği

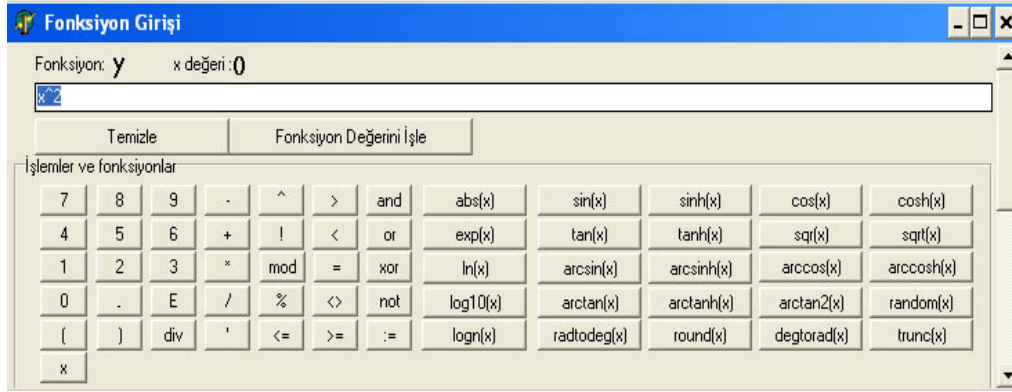


Şekil 5.31. $Y=Tan(x)$ X-Hata Grafiği



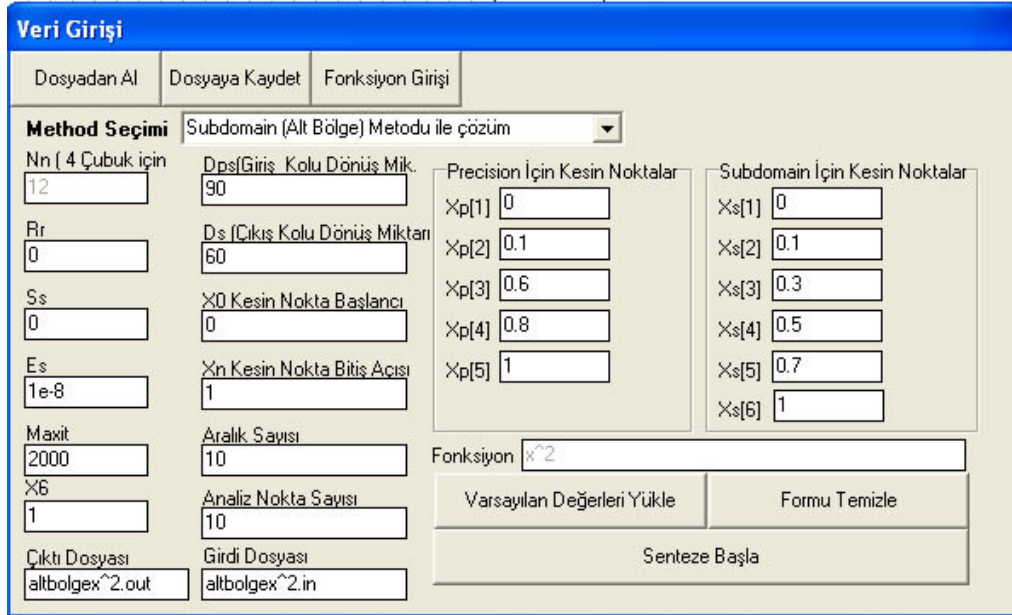
Şekil 5.32. $Y=Tan(x)$ Psi-Alac Grafiği

5.2.2.2. $Y=x^2$ Fonksiyonu İçin Dört Çubuk Tasarımı



Şekil 5.33. $Y=x^2$ Fonksiyon Girişi

$Y=x^2$ için girilen tasarım değerleri Şekil 5.34' deki gibidir.



Şekil 5.34. $Y=x^2$ Veri Girişi

Burada girilen değerler $Y = x^2$ fonksiyonu içindir. $0 < x < 1$ birimlik bir açılal aralıktadır.

Fonksiyonda kullanılan

90 = giriş kolu dönme açısını,

60 = X_3 çıkış kolu dönme açısını temsil eder.

X değişkeninin 0 ile 1 birim aralığındaki değer alır .

$$X_s[1] = 0$$

$$X_s[2] = 0.1$$

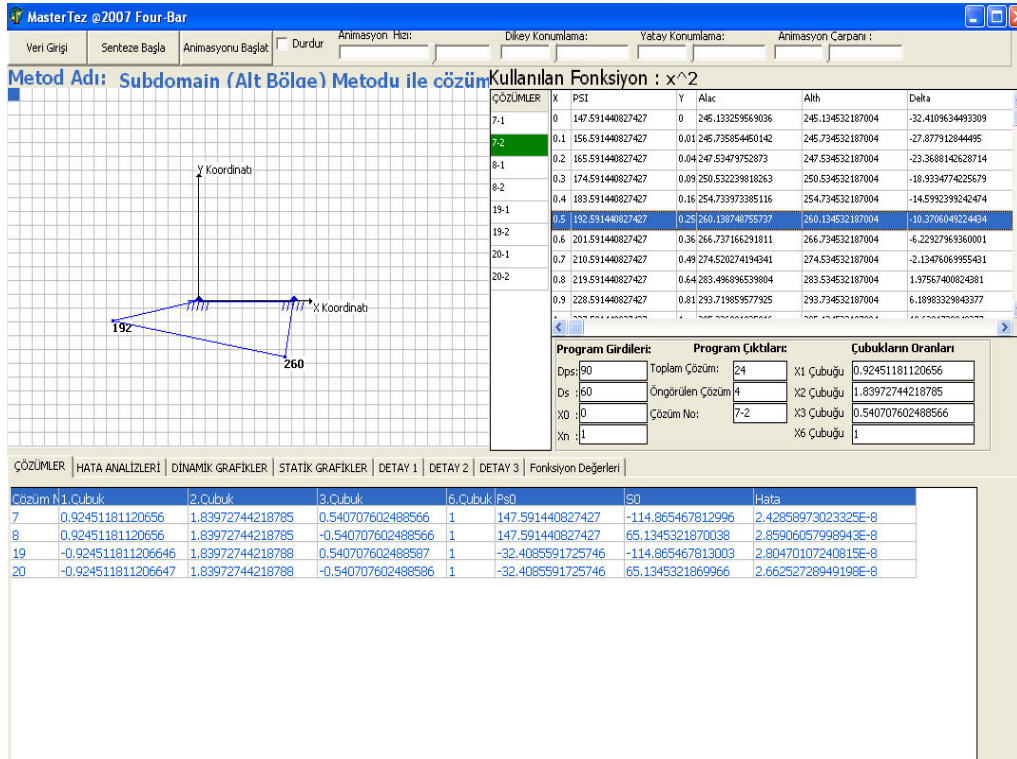
$$X_s[3] = 0.3$$

$$X_s[4] = 0.5$$

$$X_s[5] = 0.7$$

$$X_s[6] = 1$$

Four-bar programı $Y = x^2$ için Şekil 5.35'deki gibi bir ekran çıktısı sağlar.



Şekil 5.35. $Y = x^2$ Program Çıktısı

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $0 < x < 1$ aralığında tanımlı $y = x^2$ fonksiyonu için toplamda 24 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 4 çözümü önermiştir.

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps:	90	Toplam Çözüm:	24	X1 Çubuğu	0.92451181120656
Ds :	60	Öngörülen Çözüm:	4	X2 Çubuğu	1.83972744218785
X0 :	0	Çözüm No:	7-2	X3 Çubuğu	0.540707602488566
Xn :	1			X6 Çubuğu	1

Şekil 5.36. $Y=x^2$ Program Girdileri Çubuk Boyları ve Çözümler

Programın verilerinin kaydedildiği *altbolgex^2.out* dosyasının görünüşü ise Şekil 5.37' deki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu gibi görünmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.


```

altbolgex^2 - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
** Subdomain method ile fonksiyon ureten dört çubuk tasarımlar ****

Girilen tasarım degerleri
dps,ds,x0,xn = 90, 60.0000, 0.0, 1.0
Alt bolgeler xs[i] i:=1,6 = 0.000, 0.100, 0.300, 0.500, 0.700, 1.000,

coz.no  x1          x2          x3          ps0          s0          hata
-----  --          --          --          --          --          ---
 7  0.924512    1.839727    0.540708    147.591441   -114.865468  2.43E-0008
 8  0.924512    1.839727    -0.540708   147.591441    65.134532  2.86E-0008
19 -0.924512    1.839727    0.540708    -32.408559   -114.865468  2.80E-0008
20 -0.924512    1.839727    -0.540708   -32.408559    65.134532  2.66E-0008
****7 inci çözüm hata analiz sonuçları ****

***** 1 inci Dört çubuk analiz sonuçları*****
  x      psi      y      alac      alth      delta      hata
  --      --      -      --      --      --      ---
0.0000  147.5914    0.0000  83.7644  245.1345    1.3086    2.81644098
0.1000  156.5914    0.0100  91.7867  245.7345    5.4005    2.68689575
0.2000  165.5914    0.0400  98.6248  247.5345    9.5284    2.59896501
0.3000  174.5914    0.0900  104.2715  250.5345   13.7372    2.55277136
0.4000  183.5914    0.1600  108.7165  254.7345   18.0498    2.54849458
0.5000  192.5914    0.2500  111.9568  260.1345   22.4662    2.58618924
0.6000  201.5914    0.3600  113.9972  266.7345   26.9637    2.66576919
0.7000  210.5914    0.4900  114.8419  274.5345   31.4970    2.78716162
0.8000  219.5914    0.6400  114.4768  283.5345   35.9980    2.95061404
0.9000  228.5914    0.8100  112.8427  293.7345   40.3727    3.15715781
1.0000  237.5914    1.0000  109.7942  305.1345   44.4910    3.40933148

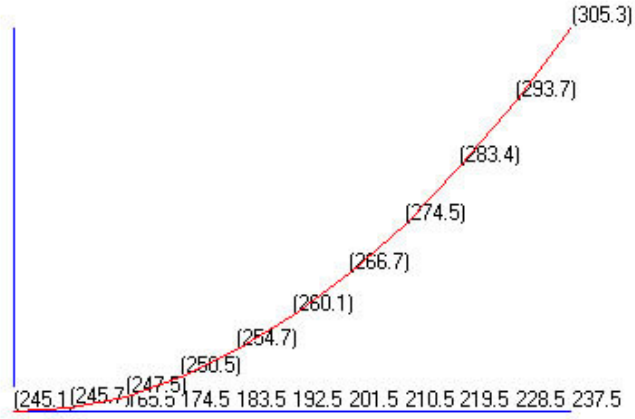
***** 2 inci Dört çubuk analiz sonuçları*****
  x      psi      y      alac      alth      delta      hata
  --      --      -      --      --      --      ---
0.0000  147.5914    0.0000  245.1333  245.1345   -32.4110    0.00002221
0.1000  156.5914    0.0100  245.7359  245.7345   -27.8779   -0.00002308
0.2000  165.5914    0.0400  247.5348  247.5345   -23.3688   -0.00000463
0.3000  174.5914    0.0900  250.5322  250.5345   -18.9335    0.00004001
0.4000  183.5914    0.1600  254.7340  254.7345   -14.5992    0.00000975
0.5000  192.5914    0.2500  260.1387  260.1345   -10.3706   -0.00007359
0.6000  201.5914    0.3600  266.7372  266.7345    -6.2293   -0.00004597
0.7000  210.5914    0.4900  274.5203  274.5345    -2.1348    0.00024885
0.8000  219.5914    0.6400  283.4969  283.5345    1.9757    0.00065687
0.9000  228.5914    0.8100  293.7199  293.7345    6.1898    0.00025609
1.0000  237.5914    1.0000  305.3270  305.1345   10.6302   -0.00335906
****8 inci çözüm hata analiz sonuçları ****

***** 1 inci Dört çubuk analiz sonuçları*****

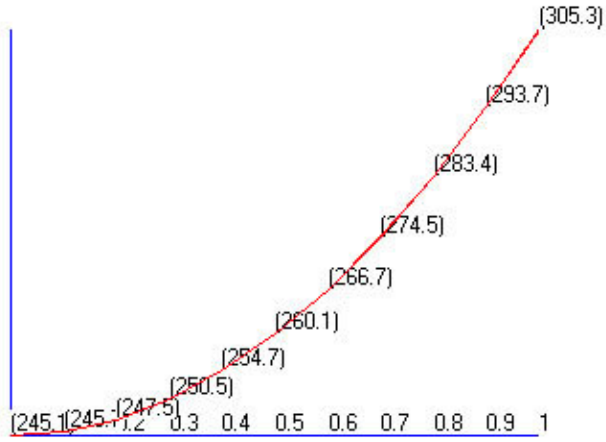
```

Şekil 5.37. $Y = x^2$ Metin Dosyası

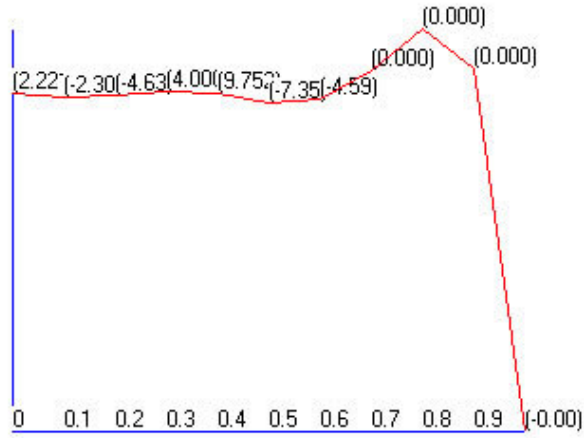
$Y=x^2$ fonksiyonunun grafikleri ise Şekil 5.38, Şekil 5.39 ve Şekil 5.40 'daki gibidir.



Şekil 5.38. $Y = x^2$ Psi-Alac Grafiği



Şekil 5.39. $Y = x^2$ Psi-Alac Hata Grafiği



Şekil 5.40. $Y=x^2$ X-Hata Grafiği

5.2.2.3. $Y = \text{Log}_{10}(x)$ fonksiyonu için dört çubuk tasarımı



Şekil 5.41. $Y = \text{Log}(x)$ Fonksiyon Girişi

$Y = \text{Log}10(x)$ için girilen tasarım değerleri Şekil 5.42 'deki gibidir.

Method Seçimi	Subdomain (Alt Bölge) Metodu ile çözüm
Nn (4 Çubuk için)	Dps(Giriş Kolu Dönüş Mik.)
12	90
Pr	Ds (Çıkış Kolu Dönüş Miktarı)
0	60
Ss	X0 Kesin Nokta Başlangıç
0	1
Es	Xn Kesin Nokta Bitiş Açısı
1e-8	10
Maxit	Aralık Sayısı
2000	10
X6	Analiz Nokta Sayısı
1	10
Çıktı Dosyası	Girdi Dosyası
log10x.out	log10xin
Precision İçin Kesin Noktalar	
Xp[1]	0
Xp[2]	0.1
Xp[3]	0.6
Xp[4]	0.8
Xp[5]	1
Subdomain İçin Kesin Noktalar	
Xs[1]	1
Xs[2]	3
Xs[3]	5
Xs[4]	7
Xs[5]	9
Xs[6]	10
Fonksiyon	
log10(x)	
Varsayılan Değerleri Yükle	
Formu Temizle	
Senteze Başla	

Şekil 5.42. $Y = \text{Log}(x)$ Veri Girişi

Burada girilen değerler $Y = \text{Log}10(x)$ fonksiyonu içindir. $1 < x < 10$ birimlik bir aralıktadır.

Fonksiyonda kullanılan

90 = giriş kolu dönme açısını,

60 = X_3 çıkış kolu dönme açısını temsil eder.

X değişkeninin 1 ile 10 birim aralığındaki değer alır.

$$X_s [1] = 1$$

$$X_s [2] = 3$$

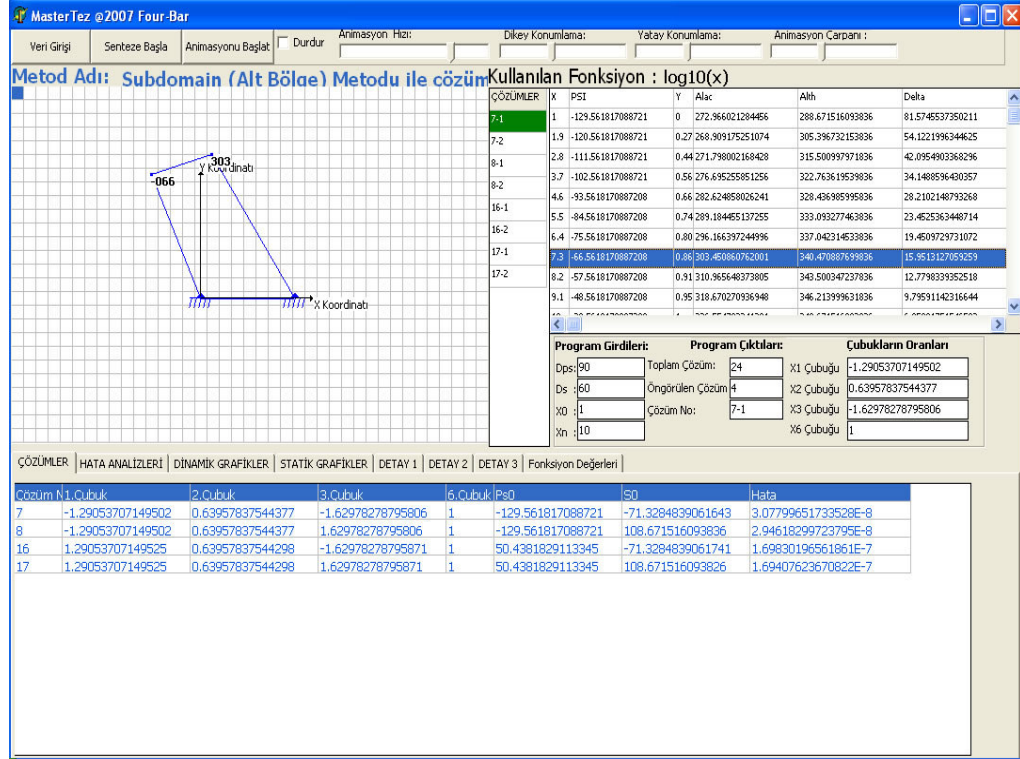
$$X_s [3] = 5$$

$$X_s [4] = 7$$

$$X_s [5] = 9$$

$$X_s [6] = 10$$

Four - Bar programı $Y = \text{Log}10(x)$ için Şekil 5.42'deki gibi bir ekran çıktısı sağlar.



Şekil 5.43. $Y = \text{Log}(x)$ Program Çıktısı

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $1 < x < 10$ aralığında tanımlı $Y = \text{Log}10(x)$ fonksiyonu için toplamda 24 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 4 çözümü önermiştir.

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps:	90	Toplam Çözüm:	24	X1 Çubuğu	-1.29053707149502
Ds:	60	Öngörülen Çözüm:	4	X2 Çubuğu	0.63957837544377
X0:	1	Çözüm No:	7-1	X3 Çubuğu	-1.62978278795806
Xn:	10			X6 Çubuğu	1

Şekil 5.44. $Y = \text{Log}(x)$ Program Girdileri Çubuk Boyları ve Çözümler

Programın verilerinin kaydedildiği *log10x.out* dosyasının görünüşü ise Şekil 5.45’ deki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu gibi görünmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.

```

log10x - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
** subdomain method ile fonksiyon ureten dört çubuk tasarlar ****

Girilen tasarım degerleri
dps,ds,x0,xn = 90, 60.0000, 1.0, 10.0
Alt bölgeler xs[i] i:=1,6 = 1.000, 3.000, 5.000, 7.000, 9.000, 10.000,

coz.no  x1          x2          x3          ps0          s0          hata
-----  --          --          --          --          --          --
 7 -1.290537      0.639578     -1.629783    -129.561817   -71.328484   3.08E-0008
 8 -1.290537      0.639578     1.629783     -129.561817   108.671516   2.95E-0008
16 1.290537       0.639578     -1.629783     50.438183     -71.328484   1.70E-0007
17 1.290537       0.639578     1.629783     50.438183     108.671516   1.69E-0007
*****7 inci çözüm hata analiz sonuçları*****

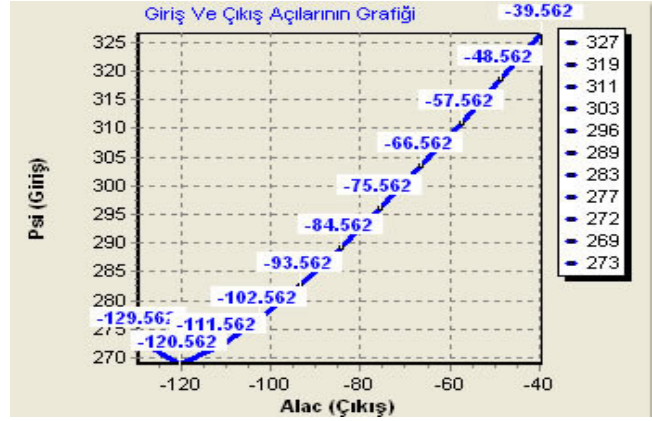
***** 1 inci Dört çubuk analiz sonuçları*****
  x      psi      y      alac      alth      delta      hata
  --      --      --      --      --      --      --
 1.0000 -129.5618    0.0000  272.9660  288.6715  81.5746  0.27411260
 1.9000 -120.5618    0.2788  268.9092  305.3967  54.1222  0.63682800
 2.8000 -111.5618    0.4472  271.7980  315.5010  42.0955  0.76276117
 3.7000 -102.5618    0.5682  276.6953  322.7636  34.1489  0.80404463
 4.6000 -93.5618     0.6628  282.6249  328.4370  28.2102  0.79957247
 5.5000 -84.5618     0.7404  289.1845  333.0933  23.4525  0.76635352
 6.4000 -75.5618     0.8062  296.1664  337.0423  19.4510  0.71341934
 7.3000 -66.5618     0.8633  303.4509  340.4709  15.9513  0.64612136
 8.2000 -57.5618     0.9138  310.9656  343.5003  12.7798  0.56783762
 9.1000 -48.5618     0.9590  318.6703  346.2140  9.7959   0.48072875
10.0000 -39.5618     1.0000  326.5547  348.6715  6.8500   0.38601120

***** 2 inci Dört çubuk analiz sonuçları*****
  x      psi      y      alac      alth      delta      hata
  --      --      --      --      --      --      --
 1.0000 -129.5618    0.0000  287.3256  288.6715  118.7170  0.02349110
 1.9000 -120.5618    0.2788  305.4731  305.3967  160.2601 -0.00133357
 2.8000 -111.5618    0.4472  315.5106  315.5010 -174.7869 -0.00016724
 3.7000 -102.5618    0.5682  322.7616  322.7636 -154.6920  0.00003455

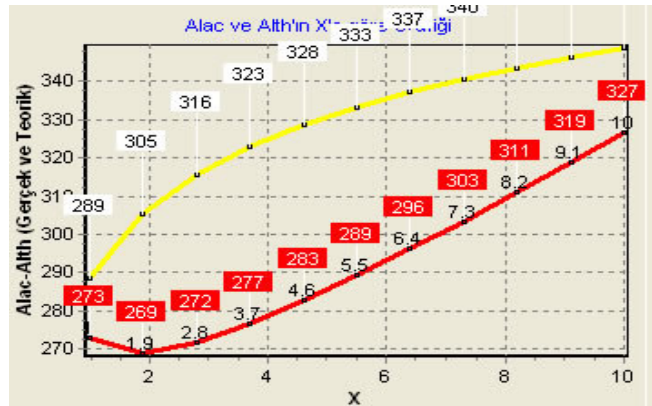
```

Şekil 5.45. $Y = \text{Log}(x)$ Metin Dosyası

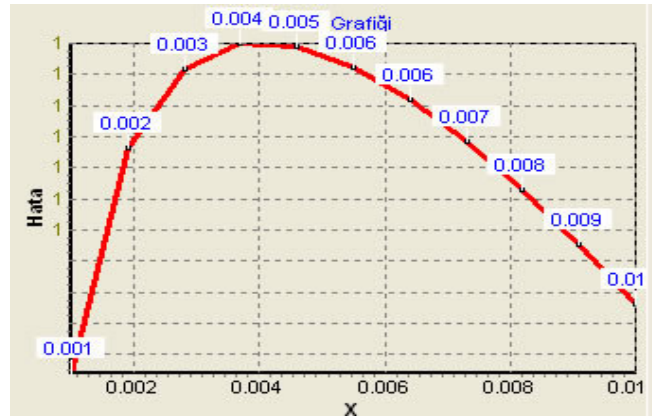
$Y = \text{Log}_{10}(x)$ fonksiyonunun grafikleri ise Şekil 5.46, Şekil 5.47, Şekil 5.48'deki gibidir.



Şekil 5.46. $Y = \text{Log}(x)$ Psi-Alac Grafiği



Şekil 5.47. $Y = \text{Log}(x)$ Psi-Alac Hata Grafiği

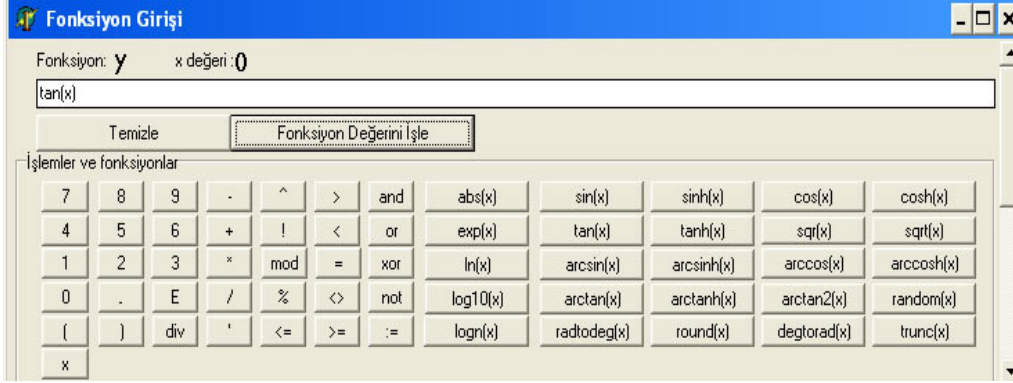


Şekil 5.48. $Y = \log(x)$ X- Hata Grafiği

5.2.3. Kesin Noktalar Metodu ile dört çubuk tasarım sonuçları

Kesin Noktalar metodu, Alt bölge metodunda olduğu gibi fonksiyon değişkeni X 'in 5 ayrı noktadaki değerleri için sentez yapılabilmektedir.

5.2.3.1. $Y=Tan(x)$ Fonksiyonu İçin Dört Çubuk Tasarımı



Şekil 5.49. $Y=Tan(x)$ Fonksiyon Girişi

$Y = Tan(x)$ için girilen tasarım değerleri Şekil 5.50' deki gibidir.



Şekil 5.50. $Y=Tan(x)$ Veri Girişi

Burada girilen deęerler $Y = \text{Tan}(x)$ fonksiyonu iindir. $0 < x < 1$ aralıęındadır.

Fonksiyonda kullanılan

90 = giriş kolu dönme açısını,

60 = X_3 çıkış kolu dönme açısını temsil eder.

X_0 deęeri 0 ve X_n deęeri 1 olarak tanımlanmıştır.

Kesin noktalar yönteminde mekanizmanın 5 noktada sentezi yapılır. Bu noktalar x_0 ve x_n aralıęında belirlenmelidir. $Y = \text{Tan}(x)$ fonksiyonu için ařaęıdaki gibi belirlenmiştir.

$$X_p [1] = 0$$

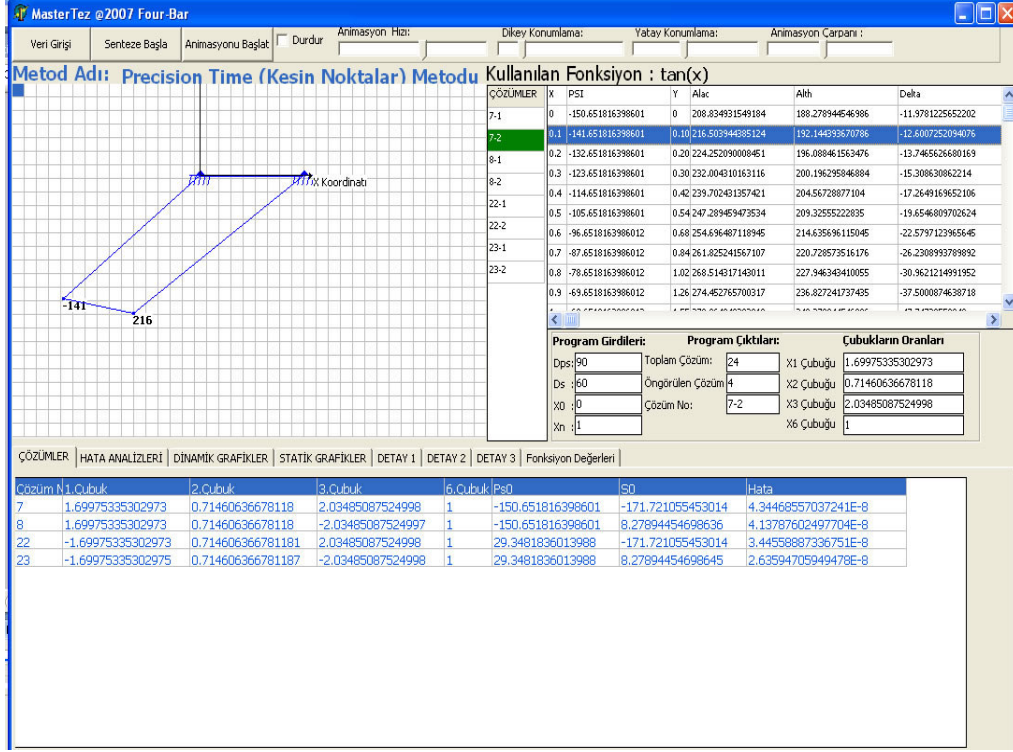
$$X_p [2] = 0.1$$

$$X_p [3] = 0.6$$

$$X_p [4] = 0.8$$

$$X_p [5] = 1$$

Four - Bar programı $Y = \text{Tan}(x)$ için Őekil 5.51'deki gibi bir ekran ıktısı saęlar.



Şekil 5.51. $Y = \tan(x)$ Ekran Çıktısı

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $0 < x < 1$ aralığında tanımlı $Y = \tan(x)$ fonksiyonu için toplamda 24 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 4 çözümü önermiştir.

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps:	90	Toplam Çözüm:	24	X1 Çubuğu	1.69975335302973
Ds :	60	Öngörülen Çözüm	4	X2 Çubuğu	0.71460636678118
X0 :	0	Çözüm No:	7-2	X3 Çubuğu	2.03485087524998
Xn :	1			X6 Çubuğu	1

Şekil 5.52. $Y = \tan(x)$ Program Girdileri Çubuk Boyları ve Çözümler

Programın verilerinin kaydedildiği *kesintanx.out* dosyasının görünüşü ise Şekil 5.53 'deki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu

gibi görünmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.

```

kesintanx - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
** Precision point method ile fonksiyon ureten dört çubuk tasarlar ****
Girilen tasarım degerleri
dps,ds,x0,xn = 90, 60.0000, 0.0, 1.0
kesin noktalar xp [i] i:=1,5 = 0.000, 0.100, 0.600, 0.800, 1.000,

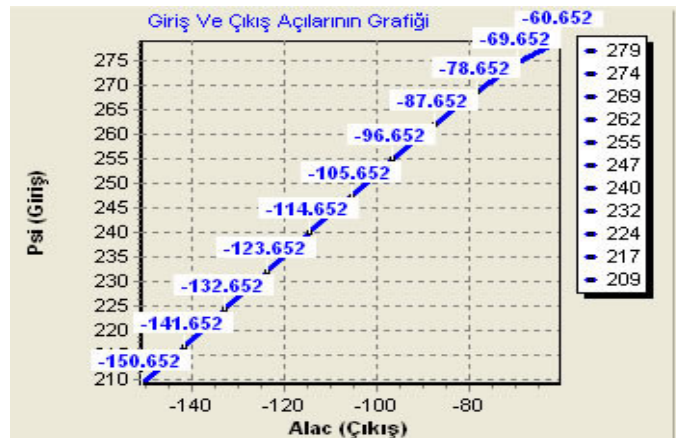
coz.no  x1          x2          x3          ps0          s0          hata
-----  --          --          --          --          --          --
7  1.699753      0.714606      2.034851     -150.651816  -171.721055  4.34E-0008
8  1.699753      0.714606      -2.034851     -150.651816   8.278945  4.14E-0008
22 -1.699753      0.714606      2.034851      29.348184   -171.721055  3.45E-0008
23 -1.699753      0.714606      -2.034851      29.348184    8.278945  2.64E-0008
****7 inci çözüm hata analiz sonuçları ****

***** 1 inci dört çubuk analiz sonuçları*****
      x      psi      y      alac      alth      delta      hata
      -      -      -      -      -      -      -
0.0000 -150.6518  0.0000  188.2789  188.2789  49.0920  0.00000000
0.1000 -141.6518  0.1003  192.1444  192.1444  61.2491  0.00000000
0.2000 -132.6518  0.2027  196.0624  196.0885  74.0610  0.00045560
0.3000 -123.6518  0.3093  200.1512  200.1963  87.4641  0.00078744
0.4000 -114.6518  0.4228  204.5220  204.5673  101.4893  0.00079052
0.5000 -105.6518  0.5463  209.2985  209.3256  116.2426  0.00047261
0.6000 -96.6518  0.6841  214.6357  214.6357  131.9119  0.00000000
0.7000 -87.6518  0.8423  220.7461  220.7286  148.8023  -0.00030669
0.8000 -78.6518  1.0296  227.9463  227.9463  167.4228  0.00000000
0.9000 -69.6518  1.2602  236.7630  236.8272  -171.2841  0.00112110
1.0000 -60.6518  1.5574  248.2789  248.2789  -145.1090  -0.00000000

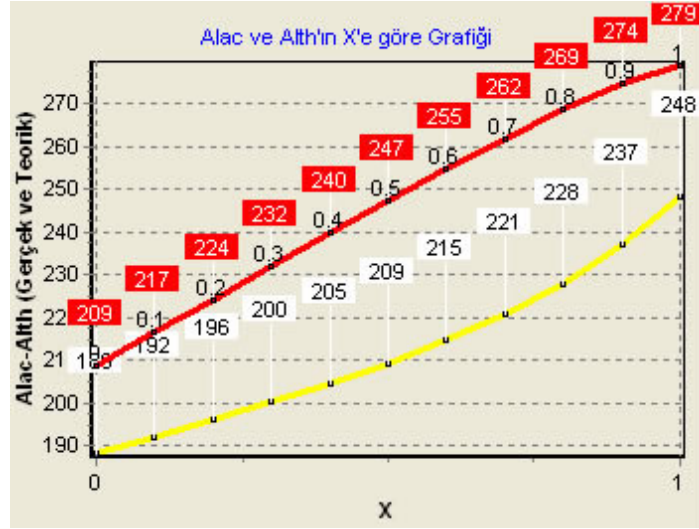
```

Şekil 5.53. $Y = \tan(x)$ Program Text Dosyası

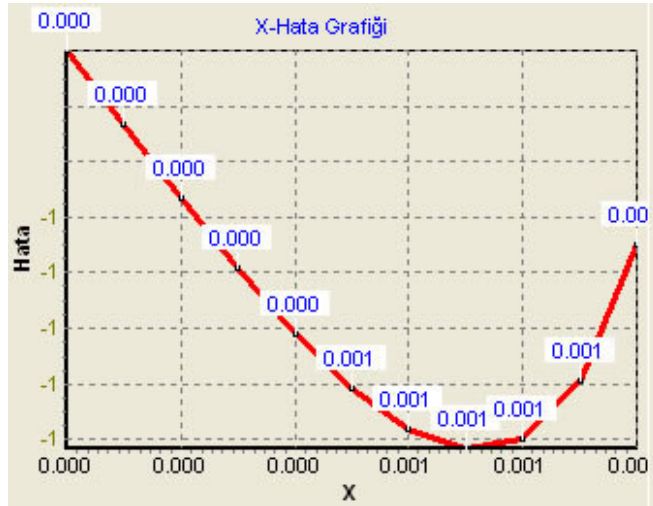
$Y = \tan(x)$ fonksiyonunun grafikleri ise Şekil 5.54, Şekil 5.55, Şekil 5.56'daki gibidir.



Şekil 5.54. $Y = \tan(x)$ Psi-Alac Grafiği

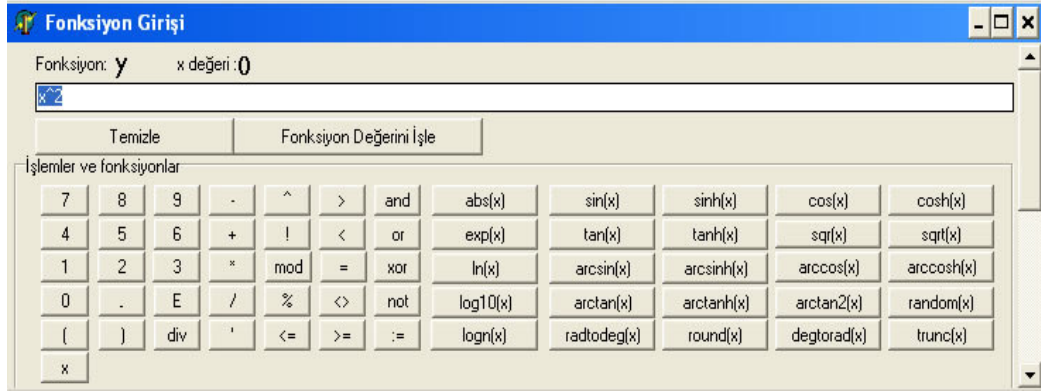


Şekil 5.55. $Y = \tan(x)$ Psi-Alac Hata Grafiği



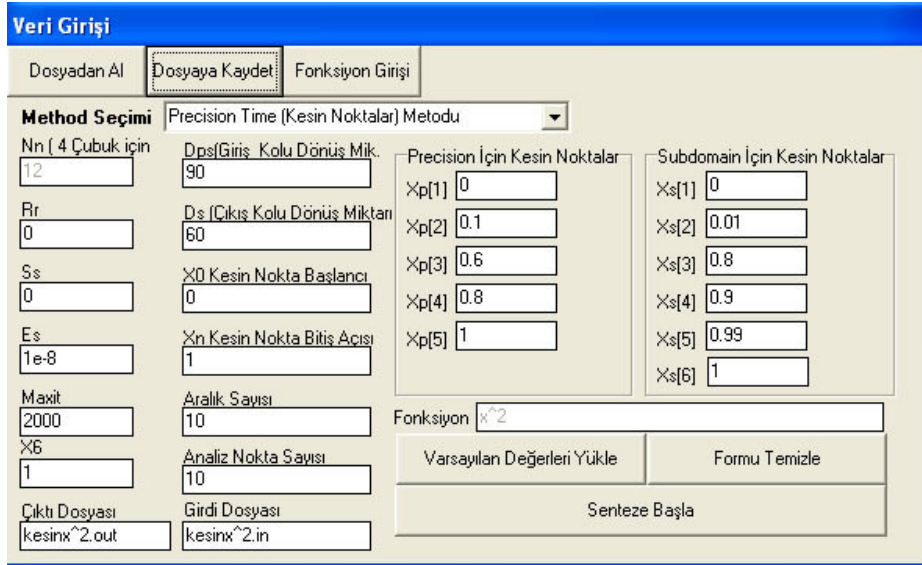
Şekil 5.56. $Y = \tan(x)$ X-Hata Grafiği

5.2.3.2. $Y = x^2$ Fonksiyonu İçin dört çubuk tasarımı



Şekil 5.57. $Y = x^2$ Fonksiyon Girişi

$Y = x^2$ için girilen tasarım değerleri Şekil 5.58 'deki gibidir.



Şekil 5.58. $Y = x^2$ Veri Girişi

Burada girilen değerler $Y = x^2$ fonksiyonu içindir. $0 < x < 1$ birimlik bir açısız aralıktadır.

Fonksiyonda kullanılan

90 = giriş kolu dönme açısını,

60 = X3 çıkış kolu dönme açısını temsil eder.

X değişkeninin 0 ile 1 birim aralığındaki değer alır.

$$X_p [1] = 0$$

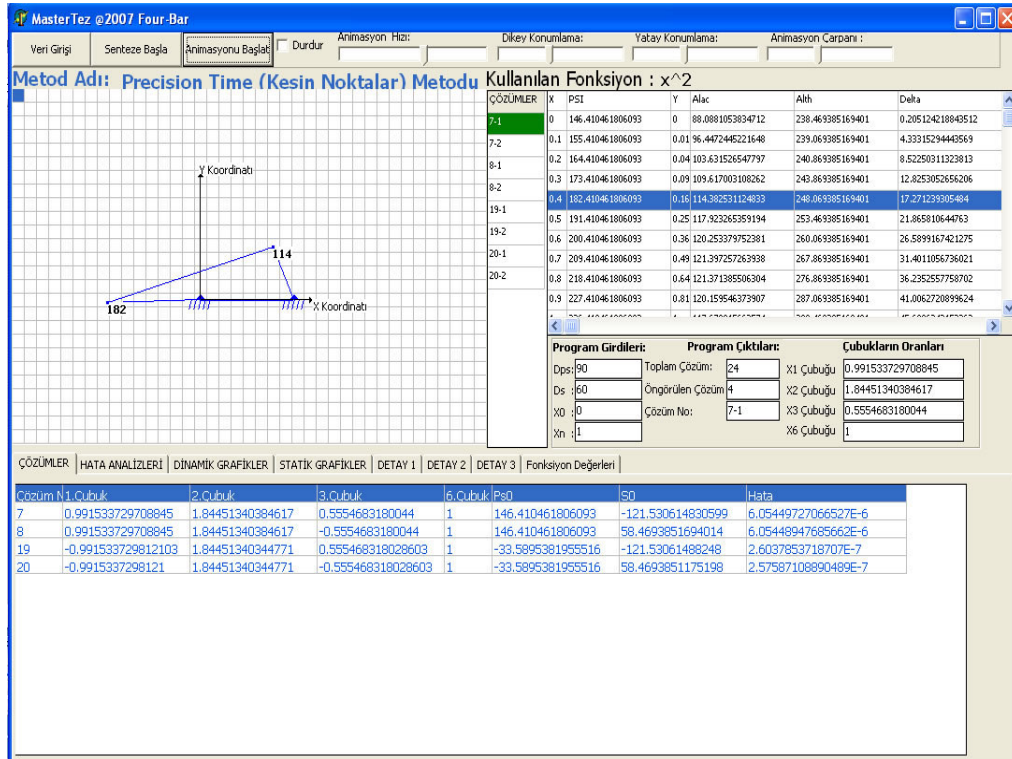
$$X_p [2] = 0.1$$

$$X_p [3] = 0.6$$

$$X_p [4] = 0.8$$

$$X_p [5] = 1$$

Four - Bar programı $Y = x^2$ için Şekil 5.59 'daki gibi bir ekran çıktısı sağlar.



Şekil 5.59. $Y = x^2$ Program Çıktısı

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $0 < x < 1$ aralığında tanımlı $Y = x^2$ fonksiyonu için toplamda 24 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 4 çözümü önermiştir.

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps: 90	Toplam Çözüm: 24	X1 Çubuğu	0.991533729708845		
Ds : 60	Öngörülen Çözüm 4	X2 Çubuğu	1.84451340384617		
X0 : 0	Çözüm No: 7-1	X3 Çubuğu	0.5554683180044		
Xn : 1		X6 Çubuğu	1		

Şekil 5.60. $Y = x^2$ Program Girdileri Çubuk Boyları ve Çözümler

Programın verilerinin kaydedildiği *kesinx^2.out* dosyasının görünüşü ise şekil 5.61'deki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu gibi görünmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.

```

kesinx^2 - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
*** Precision point method ile fonksiyon ureten dört çubuk tasarlar ****
Girilen tasarım degerleri
dps,ds,x0,xn = 90, 60.0000, 0.0, 1.0
kesin noktalar xp [i] i:=1,5 = 0.000, 0.100, 0.600, 0.800, 1.000,

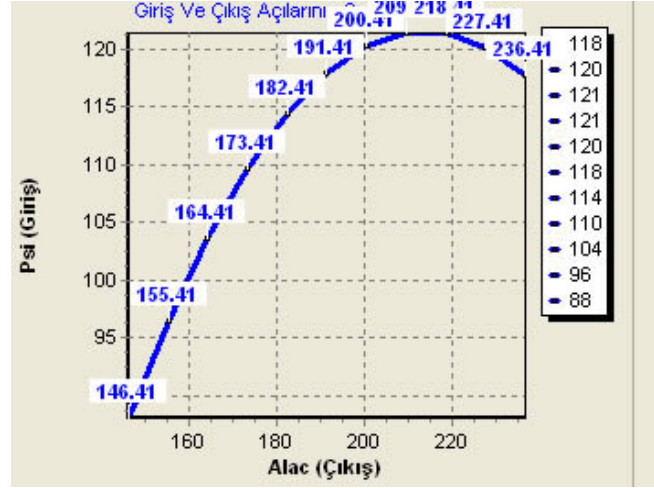
coz.no  x1          x2          x3          ps0          s0          hata
-----  --          --          --          --          --          --
7       0.991534     1.844513     0.555468     146.410462   -121.530615  6.05E-0006
8       0.991534     1.844513     -0.555468     146.410462   58.469385    6.05E-0006
19      -0.991534     1.844513     0.555468     -33.589538   -121.530615  2.60E-0007
20      -0.991534     1.844513     -0.555468     -33.589538   58.469385    2.58E-0007
*****7 inci çözüm hata analiz sonuçları *****

***** 1 inci Dört çubuk analiz sonuçları*****
x      psi      y      alac      alth      delta      hata
-      -      -      -      -      -      -
0.0000 146.4105  0.0000  88.0881  238.4694  0.2051  2.62464847
0.1000 155.4105  0.0100  96.4472  239.0694  4.3332  2.48922594
0.2000 164.4105  0.0400  103.6315  240.8694  8.5225  2.39525249
0.3000 173.4105  0.0900  109.6170  243.8694  12.8253  2.34314610
0.4000 182.4105  0.1600  114.3825  248.0694  17.2712  2.33327577
0.5000 191.4105  0.2500  117.9233  253.4694  21.8658  2.36572608
0.6000 200.4105  0.3600  120.2534  260.0694  26.5899  2.44024964
0.7000 209.4105  0.4900  121.3973  267.8694  31.4011  2.55642089
0.8000 218.4105  0.6400  121.3714  276.8694  36.2353  2.71395207
0.9000 227.4105  0.8100  120.1595  287.0694  41.0063  2.91312624

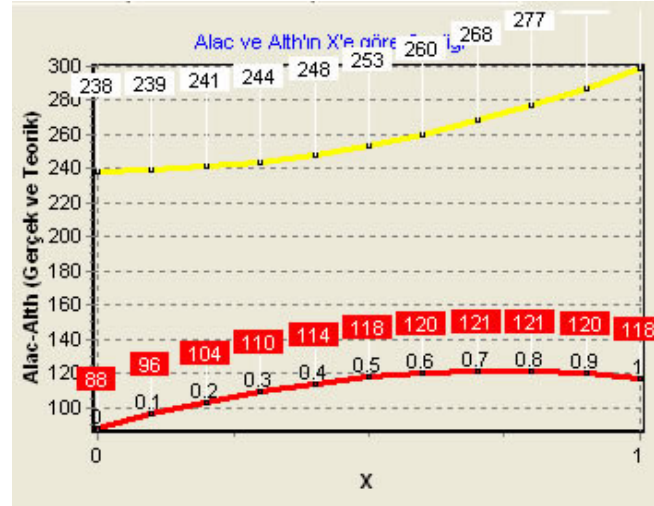
```

Şekil 5.61. $Y = x^2$ Text Dosyası

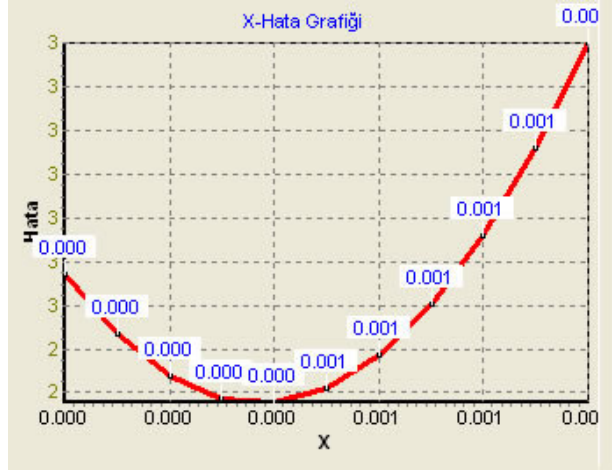
$Y = x^2$ fonksiyonunun grafikleri ise Şekil 5.62, Şekil 5.63, Şekil 5.64 'daki gibidir.



Şekil 5.62. $Y = x^2$ Psi-Alac Grafiği



Şekil 5.63. $Y = x^2$ Psi-Alac Hata Grafiği



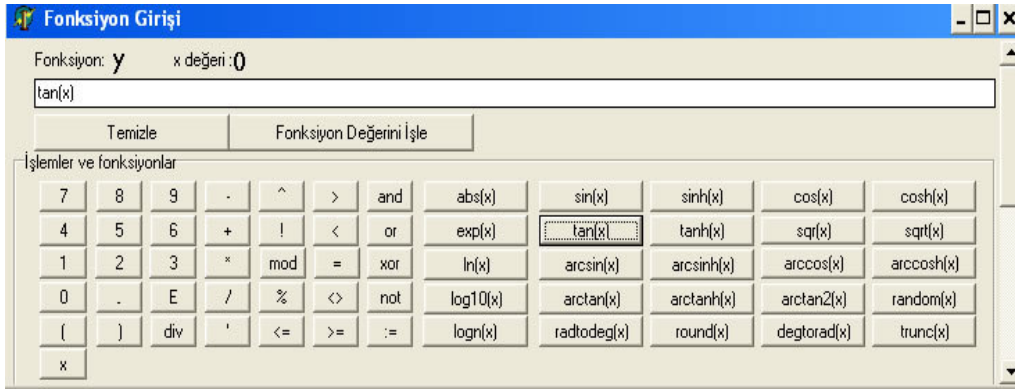
Şekil 5.64. $Y = x^2$ X-Hata Grafiği

5.3. KRANK – BİYEL TASARIMLARI

5.3.1. Galerkin Metodu İle Krank-Biyel Tasarım Sonuçları

Mek-Gen programının ana menüsünden, Krank-Biyel yada Krank Biyel Delta programı işletilerek, veri giriş kısmına Şekil 5.65 'deki veriler işlenir.

5.3.1.1. $Y = \tan(x)$ Fonksiyonu İçin Krank – Biyel Tasarımı



Şekil 5.65. $Y = \tan(x)$ Fonksiyon Girişi

$Y = \tan(x)$ için girilen tasarım değerleri Şekil 5.66'daki gibidir.

Veri Girişi

Dosyadan Al Dosyaya Kaydet Fonksiyon Girişi

Method Seçimi Galerkin Metodu ile çözüm

Nn [Krank sider için] 2	Dps[Giriş Kolu Dönüş Mik. 90	Precision İçin Kesin Noktalar Xp[1] 0	Subdomain İçin Kesin Noktalar Xs[1] 0
Rr -1	X2 kolunun Dönüş Miktarı 60	Xp[2] 0.4	Xs[2] 0.01
Ss -1	X0 Kesin Nokta Başlangıcı 0	Xp[3] 0.7	Xs[3] 0.8
Es 1e-10	Xn Kesin Nokta Bitiş Açısı 1	Xp[4] 1	Xs[4] 0.9
Maxit 20000	Aralık Sayısı 10	Fonksiyon tan(x)	Xs[5] 1
Çıktı Dosyası galtanx.out	Analiz Nokta Sayısı 10	Varsayılan Değerleri Yükle	Formu Temizle
Girdi Dosyası galtanx.in		Senteze Başla	

Şekil 5.66. $Y = \tan(x)$ Veri Girişi

Burada girilen değerler $Y = \tan(x)$ fonksiyonu içindir. $0 < x < 57$ derecelik bir açısal aralıktadır.

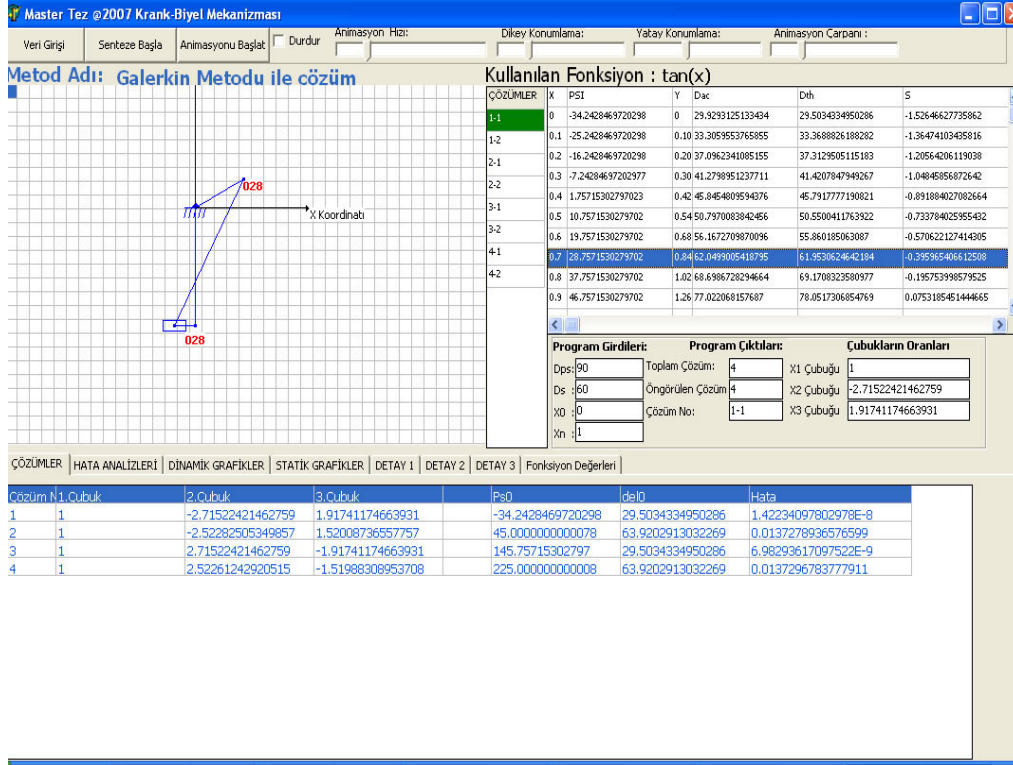
Fonksiyonda kullanılan

90 = Giriş kolu dönme açısını,

60 = X_2 çıkış kolu dönme açısını temsil eder.

X değişkeninin 0 ile 57 derece aralığındaki değeri derece olarak verilmektedir. Biz bu değerleri X_0 ve X_n in programa girerken radyana çevirmekteyiz. Dolayısıyla X_n değerine 57derecelik açığa denk gelen 1 değeri girilmektedir. Metod galerkin seçilidir ve çıktılar *galtanx.in* ve *galtanx.out* isimli giriş ve çıkış dosyalarına kaydedilir.

Program çalıştığında öngörülen çözümler Şekil 5.67 'deki gibidir:



Şekil 5.67. $Y = \tan(x)$ Çıktı Ekranı

Four-bar programı $Y = \tan(x)$ için Şekil 5.68' deki gibi bir ekran çıktısı sağlar.

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $0 < x < 57$ aralığında tanımlı $Y = \tan(x)$ fonksiyonu için toplamda 4 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 4 çözümü önermiştir.

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps:	90	Toplam Çözüm:	4	X1 Çubuğu	1
Ds :	60	Öngörülen Çözüm	4	X2 Çubuğu	-2.71522421462759
X0 :	0	Çözüm No:	1-1	X3 Çubuğu	1.91741174663931
Xn :	1				

Şekil 5.68. $Y=Tan(x)$ Program Girdileri Çubuk Boyları ve Çözümler

Programın verilerinin kaydedildiği *galtanx.out* dosyasının görünüşü ise Şekil 5.69 'daki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu gibi görünmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.

```

galtanx - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
** Galarkin method ile fonksiyon ureten krank-biyel tasarlar ****
Girilen tasariim degerleri
dps,ds,x0,xn = 90, 1.0472, 0.0, 1.0

coz.no  x1          x2          x3          ps0          del0          hata
-----  --          --          --          --          --          --
  1  1.000000    -2.715224    1.917412    -34.242847    29.503433    1.42E-0008
  2  1.000000    -2.522825    1.520087    45.000000    63.920291    1.37E-0002
  3  1.000000    2.715224    -1.917412    145.757153    29.503433    6.98E-0009
  4  1.000000    2.522612    -1.519883    225.000000    63.920291    1.37E-0002
****1 inci cozum hata analiz sonuclari ****

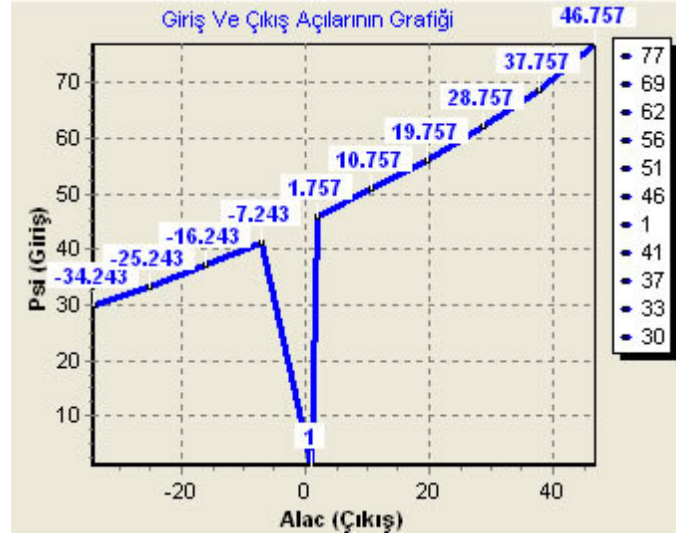
***** 1 inci krank-Biyel analiz sonuclari*****
      x          psi          y          dac          dth          s          hata
      --          --          --          --          --          --          --
0.0000    -34.2428    0.0000    29.9293    29.5034    -1.5265    -0.00743299
0.1000    -25.2428    0.1003    33.3060    33.3689    -1.3647    0.00109829
0.2000    -16.2428    0.2027    37.0962    37.3130    -1.2056    0.00378241
0.3000     -7.2428    0.3093    41.2799    41.4208    -1.0485    0.00245899
0.4000     1.7572    0.4228    45.8455    45.7918    -0.8919    -0.00093730
0.5000    10.7572    0.5463    50.7970    50.5500    -0.7338    -0.00431039
0.6000    19.7572    0.6841    56.1673    55.8602    -0.5706    -0.00535966
0.7000    28.7572    0.8423    62.0499    61.9531    -0.3960    -0.00169014
0.8000    37.7572    1.0296    68.6987    69.1708    -0.1958    0.00824074
0.9000    46.7572    1.2602    77.0221    78.0517     0.0753    0.01797100
*****Hareket yok*****

***** 2 inci krank-Biyel analiz sonuclari*****
      x          psi          y          dac          dth          s          hata

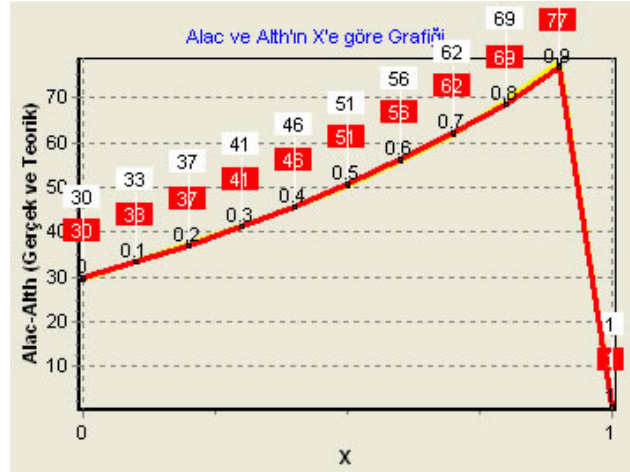
```

Şekil 5.69. $Y=Tan(x)$ Metin Dosyası

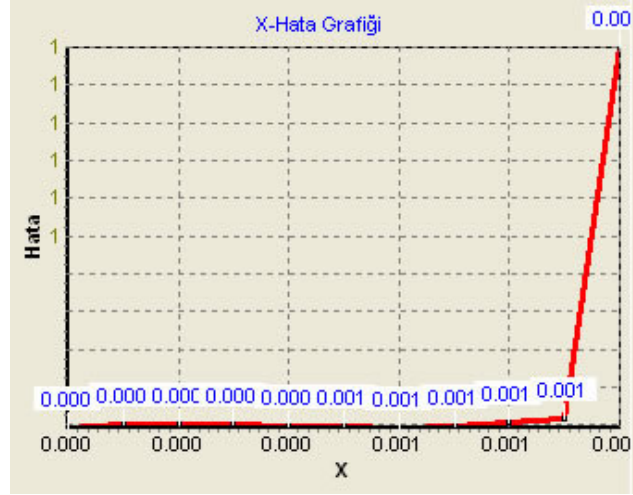
$Y = \tan(x)$ fonksiyonunun grafikleri ise Şekil 5.70, Şekil 5.71, Şekil 5.72'deki gibidir.



Şekil 5.70. $Y = \tan(x)$ Psi-Alac Grafiği

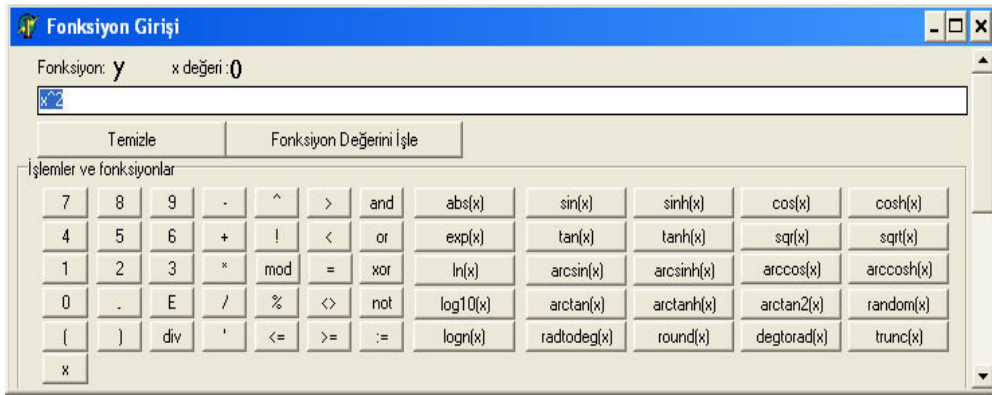


Şekil 5.71. $Y = \tan(x)$ Psi-Alac Hata Grafiği



Şekil 5.72. $Y=Tan(x)$ X-Hata Grafiği

5.3.1.2. $Y = x^2$ Fonksiyonu İçin Krank-Biyel Tasarımı



Şekil 5.73. $Y=x^2$ Fonksiyon Girişi

$Y=x^2$ için girilen tasarım değerleri Şekil 5.74 'deki gibidir.

Method Seçimi	Galerkin Metodu ile çözüm		
Nn (Krank sider için)	Dps(Giriş Kolu Dönüş Mik.	Precision İçin Kesin Noktalar	Subdomain İçin Kesin Noktalar
2	90	Xp(1) 0	Xs(1) 0
Fr	X2 kolunun Dönüş Miktarı	Xp(2) 0.4	Xs(2) 0.01
-1	60	Xp(3) 0.7	Xs(3) 0.8
Ss	X0 Kesin Nokta Başlangıç	Xp(4) 1	Xs(4) 0.9
-1	0		Xs(5) 1
Es	Xn Kesin Nokta Bitiş Açısı		
1e-10	1		
Maxit	Aralık Sayısı	Fonksiyon	
20000	10	x^2	
	Analiz Nokta Sayısı	Varsayılan Değerleri Yükle	Formu Temizle
	10	Senteze Başla	
Çıktı Dosyası	Girdi Dosyası		
galx^2.out	galx^2.in		

Şekil 5.74. $Y=x^2$ Veri Girişi

Burada girilen değerler $Y = x^2$ fonksiyonu içindir. $0 < x < 1$ birimlik bir açısal aralıktadır.

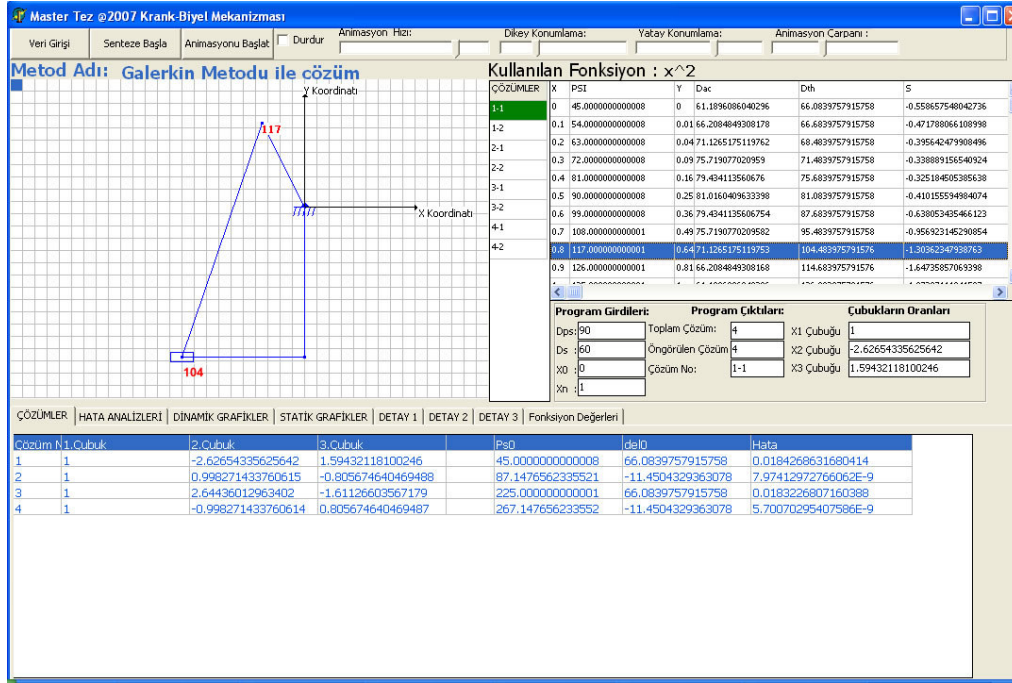
Fonksiyonda kullanılan

90 = Giriş kolu dönme açısını,

60 = X_2 çıkış kolu dönme açısını temsil eder.

X değişkeninin 0 ile 1 birim aralığındaki değerleri için sentez yapılır. Metod olarak galerkin seçilidir ve çıktılar *galx^2.in* ve *galx^2.out* isimli giriş ve çıkış dosyalarına kaydedilir.

Program çalıştığında öngörülen çözümler Şekil 5.75 'deki gibidir:



Şekil 5.75. $Y = x^2$ Program Çıktısı

Four - Bar programı $Y = x^2$ için Şekil 5.76' daki gibi bir ekran çıktısı sağlar.

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $0 < x < 1$ aralığında tanımlı $Y = x^2$ fonksiyonu için toplamda 4 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 4 çözümü önermiştir.

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps:	90	Toplam Çözüm:	4	X1 Çubuğu	1
Ds :	60	Öngörülen Çözüm	4	X2 Çubuğu	-2.62654335625642
X0 :	0	Çözüm No:	1-1	X3 Çubuğu	1.59432118100246
Xn :	1				

Şekil 5.76. $Y = x^2$ program girdileri çubuk boyları ve çözümler

Programın verilerinin kaydedildiği *galx^2.out* dosyasının görünüşü ise Şekil 5.77'deki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu gibi görünmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.

```

galx^2 - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
** Galarkin method ile fonksiyon ureten krank-biyel tasarlar ****

Girilen tasarim degerleri
dps,ds,x0,xn = 90, 1.0472, 0.0, 1.0

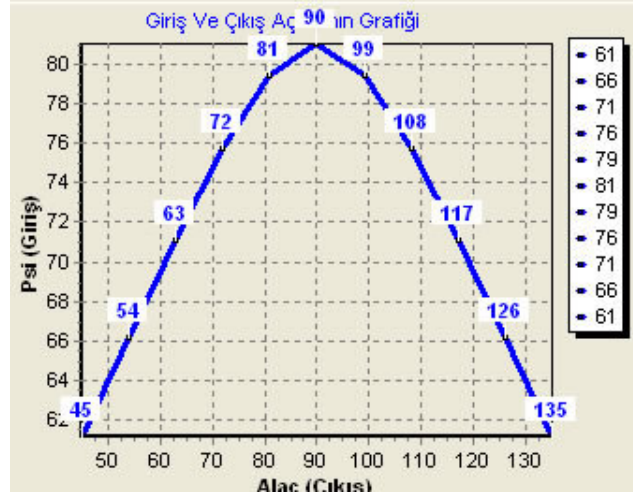
coz.no  x1          x2          x3          ps0          del0          hata
-----  --          --          --          --          --          ---
1  1.000000    -2.626543    1.594321    45.000000    66.083976    1.84E-0002
2  1.000000     0.998271   -0.805675    87.147656   -11.450433    7.97E-0009
3  1.000000     2.644360   -1.611266    225.000000    66.083976    1.83E-0002
4  1.000000    -0.998271     0.805675    267.147656   -11.450433    5.70E-0009
****1 inci cozum hata analiz sonuclari ****

***** 1 inci krank-Biyel analiz sonuclari*****
x      psi      y      dac      dth      s      hata
-----  --      --      --      --      --      ---
0.0000  45.0000  0.0000  61.1896  66.0840  -0.5587  0.08542282
0.1000  54.0000  0.0100  66.2085  66.6840  -0.4718  0.00829888
0.2000  63.0000  0.0400  71.1265  68.4840  -0.3956  -0.04612105
0.3000  72.0000  0.0900  75.7191  71.4840  -0.3389  -0.07391646
0.4000  81.0000  0.1600  79.4341  75.6840  -0.3252  -0.06545225
0.5000  90.0000  0.2500  81.0160  81.0840  -0.4102  0.00118569
0.6000  99.0000  0.3600  79.4341  87.6840  -0.6381  0.14398726
0.7000  108.0000  0.4900  75.7191  95.4840  -0.9569  0.34496256
0.8000  117.0000  0.6400  71.1265  104.4840  -1.3036  0.58219748
0.9000  126.0000  0.8100  66.2085  114.6840  -1.6474  0.84605692
1.0000  135.0000  1.0000  61.1896  126.0840  -1.9729  1.13262037

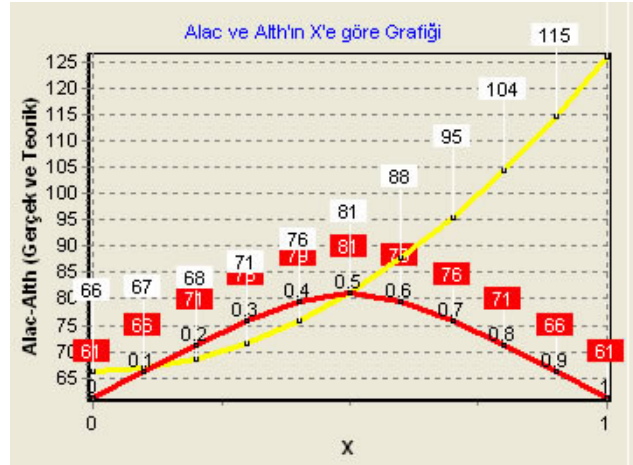
```

Şekil 5.77. $Y = x^2$ Metin Dosyası

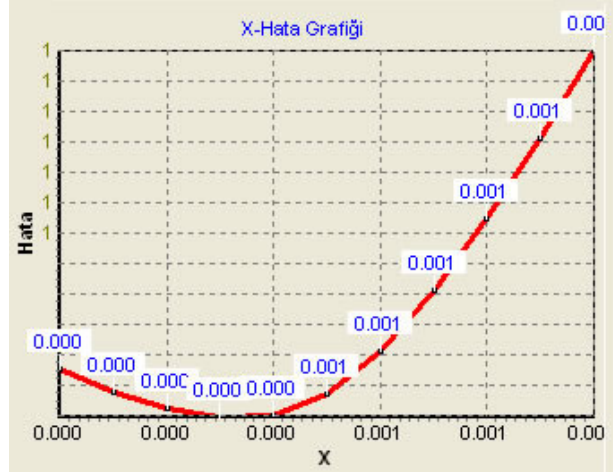
$Y = x^2$ fonksiyonunun grafikleri ise Şekil 5.78 , Şekil 5.79, Şekil 5.80'deki gibidir.



Şekil 5.78. $Y = x^2$ Psi-Alac Grafiği

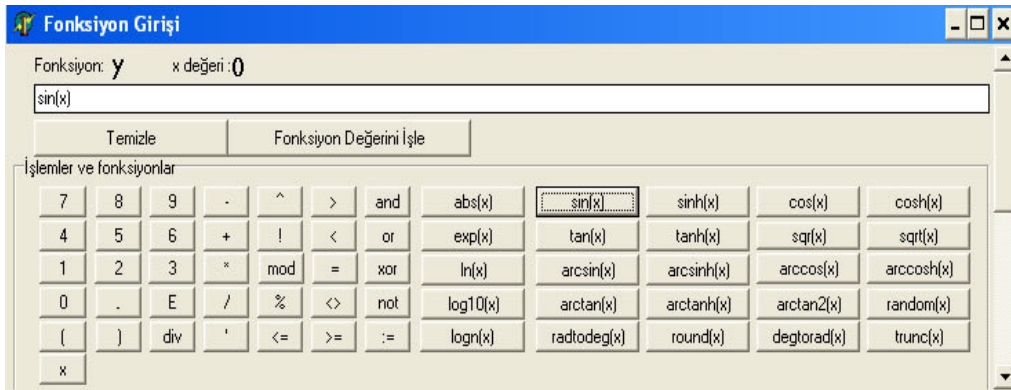


Şekil 5.79. $Y = x^2$ Psi-Alac Hata Grafiği



Şekil 5.80. $Y = x^2$ X-Hata Grafiği

5.3.1.3. $Y = \sin(x)$ Fonksiyonu için 4 çubuk tasarımı



Şekil 5.81. $Y = \sin(x)$ Fonksiyon Girişi

$Y = \sin(x)$ için girilen tasarım değerleri Şekil 5.82 'deki gibidir.

Field	Value
Method Seçimi	Galerkin Metodu ile çözüm
Nn (Krank sider için)	2
Dps(Giriş Kolu Dönüş Mik.	90
Rr	-1
X2 kolunun Dönüş Miktarı	60
Ss	-1
X0 Kesin Nokta Başlangıç	0
Es	1e-10
Xn Kesin Nokta Bitiş Açısı	1
Maxit	20000
Aralık Sayısı	10
Analiz Nokta Sayısı	10
Çıktı Dosyası	galsinx.out
Girdi Dosyası	galsinx.in
Xp[1]	0
Xp[2]	0.4
Xp[3]	0.7
Xp[4]	1
Xs[1]	0
Xs[2]	0.01
Xs[3]	0.8
Xs[4]	0.9
Xs[5]	1
Fonksiyon	sin(x)

Şekil 5.82. $Y = \sin(x)$ Veri Girişi

Burada girilen değerler $Y = \sin(x)$ fonksiyonu içindir. $0 < x < 1$ radyanlık bir açısal aralıktadır.

Fonksiyonda kullanılan

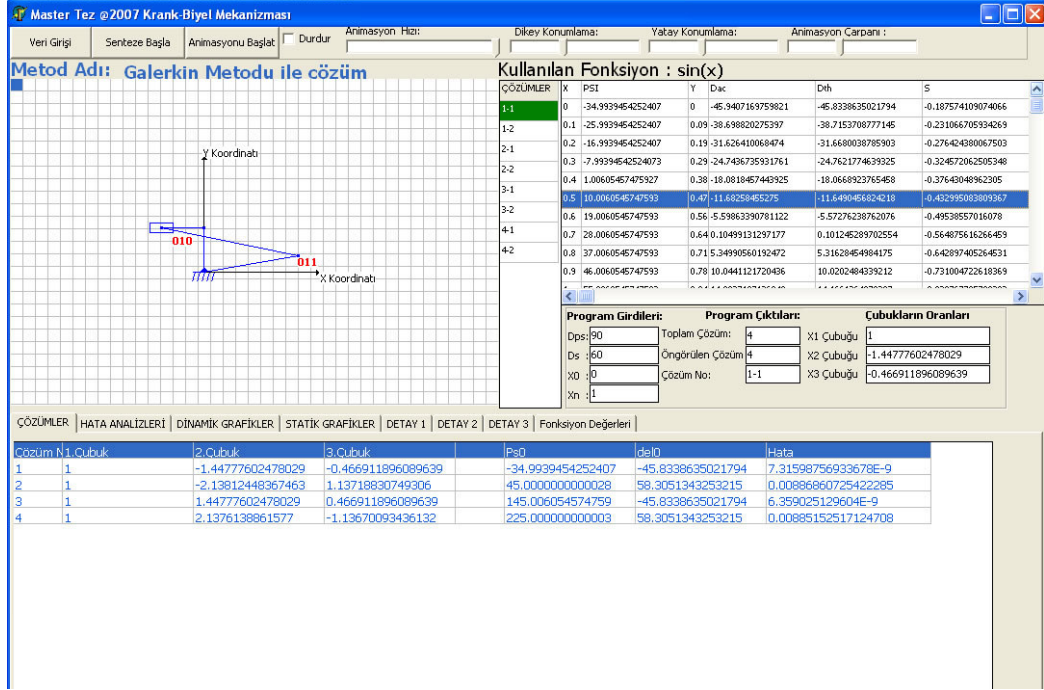
90 = Giriş kolu dönme açısını,

60 = X^2 çıkış kolu dönme açısını temsil eder.

X değişkeninin 0 ile 1 radyan aralığındaki değerleri için sentez yapılır.

Programda X_0 ve X_n değerleri birim girişleri için tasarlanmıştır. Bir önceki örnekte X^2 fonksiyonu için X değerinin 0 ile 1 değerleri arasındaki değişimi için x_0 ve x_n 0 ve 1 olarak girilmiştir. Metod olarak galerkin seçilidir ve çıktılar *galsinx.in* ve *galsinx.out* isimli giriş ve çıkış dosyalarına kaydedilir.

Program çalıştığında öngörülen çözümler Şekil 5.83 'deki gibidir:



Şekil 5.83. $Y = \sin(x)$ Program Çıktısı

Four-Bar programı $Y = \sin(x)$ için Şekil 5.84' deki gibi bir ekran çıktısı sağlar.

Belirlenen her çözüm için çubuk boyları yazdırılır. Toplam çözüm ve öngörülen çözüm sayıları sabittir. Program $0 < x < 1$ aralığında tanımlı $Y = \sin(x)$ fonksiyonu için toplamda 4 çözüm yapmış ve bu çözümlerden hata analizleri en az olan 4 çözümü önermiştir.

Program Girdileri:		Program Çıktıları:		Çubukların Oranları	
Dps:	90	Toplam Çözüm:	4	X1 Çubuğu	1
Ds :	60	Öngörülen Çözüm	4	X2 Çubuğu	-1.44777602478029
X0 :	0	Çözüm No:	1-1	X3 Çubuğu	-0.466911896089639
Xn :	1				

Şekil 5.84. $Y = \sin(x)$ Program Girdileri Çubuk Boyları ve Çözümler

Programın verilerinin kaydedildiği *galsinx.out* dosyasının görünüşü ise Şekil 5.85'deki gibidir. Bu dosyada, tüm analiz sonuçları yine programda olduğu gibi görünmektedir. Buradan hata değeri en az olan çözüm seçilerek, mekanizma tasarımında kullanılır.

```

galsinx - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
** Galarkin method ile fonksiyon ureten krank-biyel tasarlar ****
Girilen tasarim degerleri
dps,ds,x0,xn = 90, 1.0472, 0.0, 1.0

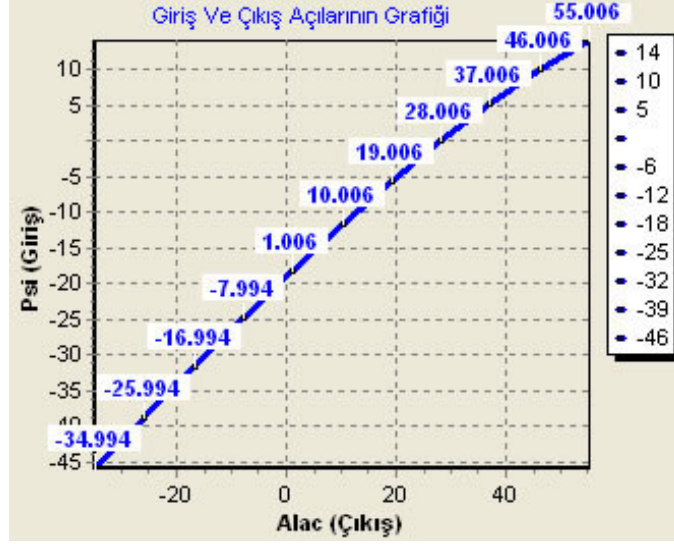
coz.no  x1          x2          x3          ps0          de10        hata
-----  --          --          --          --          --          --
 1  1.000000  -1.447776  -0.466912  -34.993945  -45.833864  7.32E-0009
 2  1.000000  -2.138124  1.137188   45.000000   58.305134  8.87E-0003
 3  1.000000  1.447776   0.466912  145.006055  -45.833864  6.36E-0009
 4  1.000000  2.137614  -1.136701  225.000000   58.305134  8.85E-0003
****1 inci çözüm hata analiz sonuçları ****

***** 1 inci Krank-Biyel analiz sonuçları*****
      x      psi      y      dac      dth      s      hata
      -      -      -      -      -      -      -
0.0000  -34.9939  0.0000  -45.9407  -45.8339  -0.1876  0.00186494
0.1000  -25.9939  0.0998  -38.6988  -38.7154  -0.2311  -0.00028886
0.2000  -16.9939  0.1987  -31.6264  -31.6680  -0.2764  -0.00072595
0.3000  -7.9939  0.2955  -24.7437  -24.7622  -0.3246  -0.00032295
0.4000  1.0061  0.3894  -18.0818  -18.0669  -0.3764  0.00026099
0.5000  10.0061  0.4794  -11.6826  -11.6490  -0.4330  0.00058536
0.6000  19.0061  0.5646  -5.5986  -5.5728  -0.4954  0.00045154
0.7000  28.0061  0.6442  0.1050  0.1012  -0.5649  -0.00006538
0.8000  37.0061  0.7174  5.3499  5.3163  -0.6429  -0.00058680
0.9000  46.0061  0.7822  10.0441  10.0202  -0.7210  -0.00041650

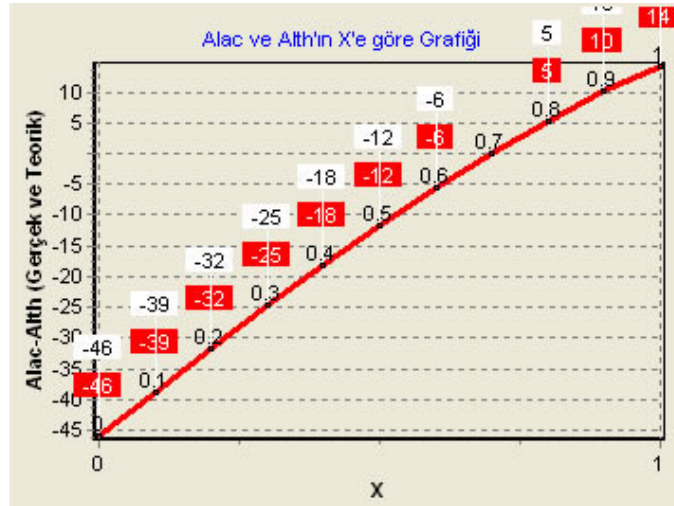
```

Şekil 5.85. $Y = \sin(x)$ Metin Dosyası

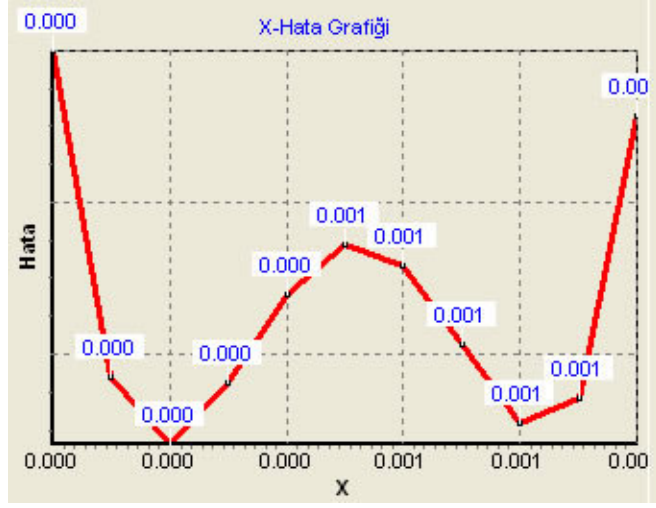
$Y = \sin(x)$ fonksiyonunun grafikleri ise Şekil 5.86, Şekil 5.87 ve Şekil 5.88'deki gibidir.



Şekil 5.86. $Y = \sin(x)$ X- Hata Grafiği



Şekil 5.87. $Y = \sin(x)$ Psi-Alac Grafiği



Şekil 5.88. $Y = \sin(x)$ Psi-Alac Hata Grafiği

6. SONUÇ VE ÖNERİLER

Yapılan çalışmalar sonucunda, girilen mekanizma değerlerine uygun mekanizma tasarımlarının üretilmesini sağlayan bilgisayar programı geliştirilmiştir. Bu yazılıma '*Mechanism-Generator*' yani mekanizma üreticisi kelimelerinin kısaltılmışı olan 'Mek-Gen' adı verilmiştir. Program daha ileriki çalışmalarda geliştirileceğinden, bu aşamadaki adı Mek-Gen Version 1.0 olarak tanımlanmıştır.

Mek-Gen kullanıcıya, mekanizma sentezini 3 ayrı metod ile yapabilme imkanı sağlamıştır. Daha önce geliştirilen birçok yazılımda, sentezde aynı anda kullanılacak metod sayısı fazla tutulamamıştır. Bunun nedeni, bu metodların tek program içerisinde kullanılmasının karmaşıklığa neden olmasıdır. Mek-Gen bu problemi güçlü omurga yapısı ile çözmüştür. Programa modül olarak istenildiği kadar metod eklenebilir.

Mek- Gen programında yine diğer yazılımlardan farklı olarak kullanıcıya, dışardan fonksiyon girebilme imkanı tanınmıştır. Daha önce yazılan mekanizma sentez programları kullanıcıya fonksiyonu dışarıdan girme imkanı sunmamaktaydı. Ancak programı oluşturan kişiler kodlara müdahale ederek bu fonksiyonları tanımlayabilmekteydiler. Mek-Gen ile, karmaşık fonksiyonların sentezi bile yapılabilmektedir. Programın bu özelliği birçok karmaşık denklem için denenmiş ve herhangi bir sorun üretmediği gözlenmiştir.

Mek-gen programında sentez , animasyon, nümerik ve grafik işlemleri için yapılacak işlemler birbirinden ayrı prosedürlerde barındırılmıştır. Programda 8 ayrı birim yazılmıştır. Tanımlı ve birbiri ile bağdaşık prosedürler aynı birimlerde kümelendirilmiştir. Böylece programda düzenli bir yapı sağlamıştır. Yapılan tüm işlemler için kurulan algoritmalar programın omurgasını güçlendiren en önemli etkenlerdir. Bu sayede programın kod yapısının karmaşıklığı giderilmiştir. Ayrıca programın testleri yapılmış sorunları giderilmiş ve bug (program işleyişi sırasında verilen hatalar) 'lar giderilmiştir.

Mek-Gen programı tüm Windows işletim sistemlerinde çalışabilir niteliktedir. Programı ekranda sorunsuz görebilmek için en az 1024 X 768 çözünürlük kullanılması gerekmektedir.

Mek-Gen programı mekanizma sentezinde 3 ayrı metod kullanabilmektedir. Bu metodlar Galerkin metodu, Kesin Noktalar metodu ve Alt Bölgeler metodudur. 3 metod fonksiyon sentezi sırasında en çok kullanılan ve sonucu en iyi verebilen metodlardır. 3 ayrı metod ile kullanıcıya mekanizma sentezi yapma imkanının verilmesiyle kullanıcı aynı fonksiyon ve değerler için 3 ayrı çözüm yapıp bu çözümler içerisinde hata analizleri en düşük olanlara göre daha rahat bir şekilde mekanizma tasarımı yapabilir. Mek-Gen programında, fonksiyon sentez problemlerinin çözümü için bir ayrıcalık daha getirilmiştir. Daha önceki metodlarda ve bilgisayar programlarında belirlenen noktalara göre sentez yapılırken kullanılabilen nokta sayısı maximum 3 yada 4 nokta oluyordu. Mek-gen programı ile belirtilen hareket aralığındaki hata analizleri istenen nokta sayıları 6 tane olarak belirlenebilmektedir. Bu özellik programın sentez sırasında kesin çözüme yakın değerler üretmesinde yardımcı olmaktadır.

Sentezi yapılan fonksiyonun çözüm ve hata analizlerinin belirlenmesinin ardından, animasyon çizme işlemine başlanır. Seçilen çözüme bağlı değerlere göre animasyon çizilebilmektedir. Animasyon çizimi sırasında kullanıcı yatay ve düşeyde animasyonu hareket ettirebilir, boyutunu büyütüp küçültebilir, animasyonun hızını değiştirebilir ve animasyonu durdurup yeniden başlatabilir. Bu özellikleri ile Mek-Gen kullanıcı dostu bir programdır.

Mekanizma sentezi için, programın işleyişini göstermesi açısından günlük yaşamda ve mühendislikte kullanılan bazı fonksiyonlara bağlı örnekler verilmiştir. Bu örneklerin tüm metodlar ile çözümleri ayrı ayrı işlenmiştir. Bu sayede kullanıcının belirlenen fonksiyon probleminde ne kadar rahat hata analizi yapabileceği gösterilmiştir.

Mek-gen programı, ileride geliştirilmesi amaçlanarak yazılmıştır. Bundan dolayı güçlü bir omurga yapısı üzerine oluşturulmuştur. 1.0 versiyonu oluşturulan program ileriki çalışmalarda değişik metodlar kullanarak çubuk mekanizmalarından farklı mekanizmaların sentezini yapabilecek seviyeye getirilmesi hedeflenmektedir.

Bu çalışma kapsamında tasarımı yapılan mekanizmadan istenilen kriterler olarak yapısal hata ve boyutların birbirine yakın olması istenmiştir. Bunların dışında hız ve ivmede ani değişikliklerin olmaması, hareket aktarımını zorlaştırıcı yada mekanizmanın çalışmasını engelleyici tekil konumların olmaması gibi kriterler de tasarımda mekanizmalardan talep edilebilir.

KAYNAKLAR

- [1] Mutlu,H. “Fonksiyon Sentezine Modüler Yaklaşım 1990 Master Tezi” , (1982).
- [2] Söylemez,E. “Mechanisms.”, Ankara Üniversitesi Basımevi., ANKARA, 402.s., (1985).
- [3] Akçalı.,İ.D. “Design of Slider-Crank Mechanism for Function Generation., The Theory of Machines and Mechanisms, Proceedings of the 7th World congress.”, Pergamon Press., **Vol.1**,:119-124, (1987).
- [4] Hartenberg.,R.S., Denavit., J. “Kinematic Synthesis of Linkages.”, McGraw-Hill, Inc., U.S.A., 435.s., (1964).
- [5] Akçalı.,İ.D., Lindholm., I.C. “A Novel Method For The 4-bar Function Generation., V. World Congren On Theory Of Machines and Mechanisms.”, MONTREAL, 263.s.,(1979).
- [6] Akçalı.,I.D., Dittrich., G. ” Function Generation by Galerkin Method., Mech. Mach. Theory.”, **Vol.24, No.1** ,:39-43, (1989).
- [7] Akçalı.,İ.D. “Subdomain-Least Squares Method of Six-Bar for Mechanizing Functions.”, Ph.D. Dissertation., KSU. MANHATTAN,(1977).
- [8] Erdman,A.G. “Modern Kinematics – Develpoment in the Last Forty Years”, Wiley Series in Design Engineering, John Wiley&Sons, Inc,112.s., (1993).
- [9] Deaux,R. “Introduction to the Geometry of Complex Numbers”, F. Ungar Publications Co., New York, 185.s.,(1956).

- [10] Polat,M. “Computer Aided Synthesis of Planar Mechanisms”, M.Sc. Thesis, METU,240.s.,(1985).
- [11] Erdman,A.G., Sandor G.N., “Mechanism Design, Analysis and Synthesis Volume 2”, Prentice-Hall, USA,245.s., (1984).
- [12] Tesar,D. “The Generalized Concept of Three Multiply Separated Positions in Coplanar Motion”, Journal of Mechanisms, **Vol. 2**,: 462-474, (1967).
- [13] Freudenstein,F. “An Analytical Approach to the Design of Four-Link Mechanisms”, Transactions of the ASME, **Vol. 76**,:483-92,(1954).
- [14] Akçalı,İ.D. “Mekanizma Sentezi Ders Notları.”, 113.s, (1979).
- [15] Harrisberger.,L “Mechanization of motion”, John Wiley & Sons Inc., New York,185.s.,(1961).
- [16] Hartenberg,R.S., Denavit,J. “Kinematic Synthesis of Linkages”, Mc-Graw Hill, New – York,168.s., (1964).
- [17] Beyer, “Kinematic Synthesis of Mechanisms”, Choptmoll and Hall Ltd., London,176.s., (1963).
- [18] Hall, “Kinematics and Linkage Design”, Prentice-Hall Inc., N.J.,244.s., (1961).
- [19] Loerch,R.J., Erdman,A.G., Sandor,G.N., “On the Existence of Circle Point an Center Point Circles for ,Three Precision-Point Dyad Synthesis”, Journal of Mechanical Design, **Vol. 2** :554-62, (1979).
- [20] Akçalı.,I.D., Dittrich.,G., “Function Generation by Galerkin Method.”, Mech. Mach. Theory.,**Vol.24, No.1**:39-43,(1989).

- [21] Akçalı.,İ.D., Lindholm.,I.C. “A Novel Method For The 4-bar Function Generation.”, V. World Congren On Theory Of Machines and Mechanisms,MONREAL,(1979).
- [22] Kaufman,R. “Mechanism Design by Computer”, Machine Design, **Vol.24**:94-100, (1978).
- [23] Erdman,A.G., Gustafson,J. “LINKAGES: Linkage Interactive Computer Analysis and Graphically Enchanced Synthesis Paskages”, Technical Report 77-Det-5. American Society of Mechanical Engineers, (1977)
- [24] Ruth,A.D., McCarthy,J.M. “The Design of Spherical 4R Linkages for Four Specified Orientations”, Mechanism and Machine Theory, Vol 34, s.667-692, (1999).
- [25] Sancak, O. (11 Ocak 2006). Nesne yönelimli programlama (NYP), Erişim: http://tr.wikipedia.org/wiki/Nesne_Y%C3%B6nelimli_Programlama [15 Nisan 2006].
- [26] Cantu,M. “Mastering Delphi 7”, Sybex, Italy,263.s.,(2003)
- [27] Erdman,A.G., Midha,A. “Man-Made Mechanism Models Multiply Mental Motivation Proceeding 3rd Applied Mechanism Conference. 3.1 – 3.21” ,(1973).
- [28] Moppert,C.F. “A Self- Balancing Crane. Mechanism And Machine Theory”, **Vol. 9** ,:359-366,(1974).
- [29] Meriam,J.L. “Statik”, Birsen Kitapevi, İSTANBUL,345.s,(1973).

ÖZGEÇMİŞ

1983 yılında Adana'da Doğdum. İlköğretimimi Ceyhan / BOTAŞ İlköğretim okulunda tamamladım. 1993 yılında Anadolu Lisesi sınavlarımı kazanarak Ceyhan Anadolu Lisesi'nde Orta ve Lise Öğretimimi tamamladım. 2000 yılında Mersin Üniversitesi Bilgisayar Mühendisliği Bölümünü Kazandım. 2004 yılında mezun olup, aynı yıl Mersin Üniversitesi Fen Bilimleri Enstitüsü Makina Mühendisliği Ana bilim dalında Yüksek Lisans eğitimi almaya hak kazandım. T.C. Sanayi ve Ticaret Bakanlığı KOSGEB Mersin İşletme Geliştirme Merkez Müdürlüğünde Dış Uzman/Danışman olarak çalışmaktayım.

EKLER

EK 1

1. Uygulama Örnekleri

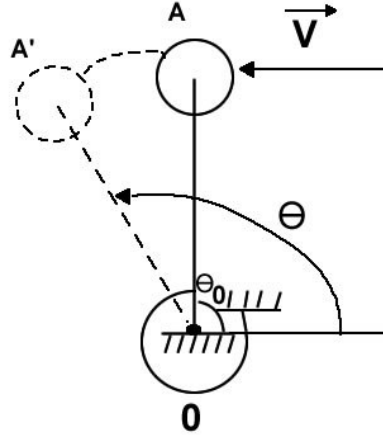
Bu kısımda bazı ölçme işlemleri (hız, ağırlık, ivme debi vs.) bazı kontrol olayları ile ilgili, fonksiyon sentez uygulamalarına yer verilmiştir. Uygulamalarda ölçülecek büyüklüklerle, açısız yer deęiřtirmeler arasındaki fonksiyonel iliřki belirlenerek, fonksiyon sentezine uygulanabilir duruma getirilecektir. Bundan sonra istenirse, elde edilen fonksiyonun uygun 4-çubuk mekanizmaları, bu çalıřma kapsamında yer alan program paketleri yardımı ile tasarlanarak imal edilebilir.

1.1. Akıřkan Hızı Ölçme [27]

řekil 8.1. de görölen düzenekte, bir çubuk ucuna baęlanmış küresel cisme etkiyen akıřkan hızı, çubuk boyu 1 ile orantılı olarak 0 noktası etrafında bir moment 0 noktasındaki bir spiral yay vasıtası ile dengelenirse, bu denge řartından akıřkan hızı V ile Θ arasındaki fonksiyonel iliřki

$$V=Z^{1/2} \left| \frac{\Theta - \Theta_0}{\cos(\Theta - \Theta_0)} \right|^{1/2} \text{ olur.}$$

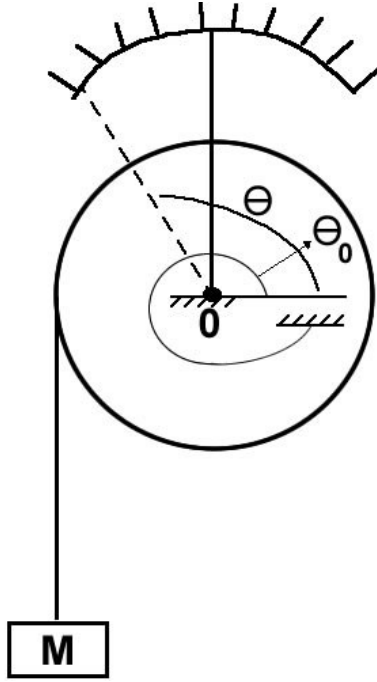
Burada $Z = 2k/lA\delta c_d$ dir. Z de yer alan büyüklükler ise, k : spiral yayın yay sabiti, δ : akıřkan yoęunluęu, A : küresel cismin kesit alanı, c : küresel cisim ile akıřkan arasındaki sürüklenme katsayısı, l : çubuk boyu olarak simgelenmiřtir.



Şekil 8.1. Akışkan hızı ölçen düzenek.

1.2. Ağırlık Ölçme [28]

Bir m kütesinin ağırlığını ölçmek için Şekil 8.2 de görülen r yarıçapında bir dönel diskin 0 noktası etrafında oluştuğu momenti belirli bir θ açısından dengelemek amacıyla spiral yaydan faydalanılmıştır.



Şekil 8.2. Ağırlık ölçmek için bir düzenek

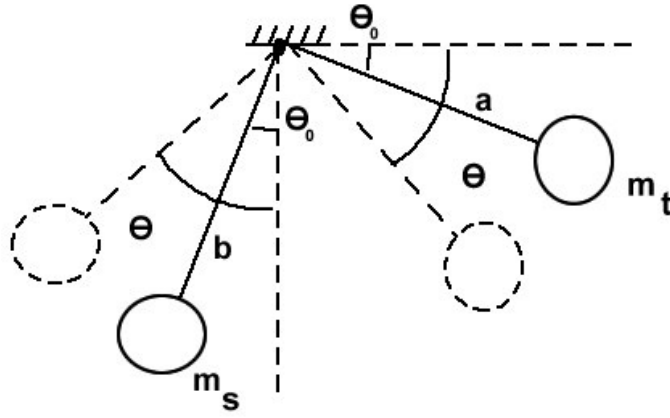
0 noktasına göre m kütesinin yaratacağı mgr momenti, buna karşı koyan $k \theta$ yay momentine eşit olacaktır. Burada, m ölçülecek kütle, g , yerçekimi ivmesi, r , dönen diskin yarıçapı, k , spiral yayın yay sabiti, θ belirli bir başlangıç açısıdır. Buna göre m kütesi ile θ açısı arasında aşağıdaki doğrusal eşitlik bulunur.

$$m = \frac{k}{g r} (\theta - \theta_0)$$

1.3. Ağırlık Ölçme [29]

Ağırlık ölçmede bir başka yöntem ise, şekil 8.3 de görülen, bir ucunda ölçülecek kütle m_t , bulunan a boyundaki kol ile, b boyunda ucunda dengeleme kütlesi m_s bulunan bir başka kol birbirine rijit bağlanarak, O noktasında mesnetlenmiştir. Buna göre oluşan momentlerin dengesinden aşağıdaki şekilde bulunur.

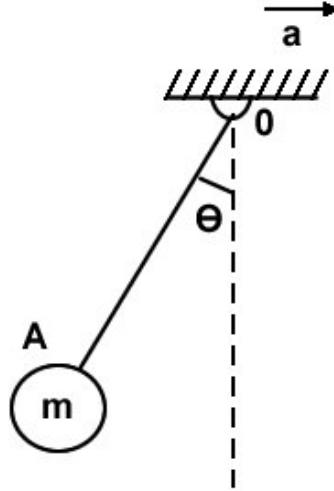
$$m_t = \frac{b}{a} m_s \tan(\Theta - \Theta_0)$$



Şekil 8.3. Ağırlık ölçen düzenek

1.4. İvme Ölçer [28]

Doğrusal hareket eden bir cismin ivmesinin ölçülebilmesi için, a ivmesi açısal yer değiştirmeye (θ) dönüştürecek bir düzenek şekil 8.4 de görülen, bir ucunda m kütlesi bulunan sarkaç kol olarak düşünülmüştür. İvmelenmeden dolayı oluşan yatay atalet kuvvetinin m kütlesi ile orantılı O noktası etrafında yarattığı moment, yer çekiminden ötürü düşey yönde oluşan ağırlığın etkisi belirli bir θ açılarından dengeleyecektir. Bu denge konumunda atalet kuvveti ve ağırlık kuvvetinin O etrafındaki moment eşitliğinden, ivme (a) ile θ açısal yer değiştirme arasındaki aşağıdaki fonksiyonel ilişki elde edilir.



Şekil 8.4. ivme ölçen bir düzenek

$$a = g \tan \theta$$

burada g yerçekimi ivmesidir.